Table of Contents

Agilian User's Guide 4
01 Part I: Getting started 4
01. Introduction to Agilian 5
02. Installing Agilian 13
03. Uninstalling Agilian 23
04. User interface 25
05. Working with projects 82
02 Part II: UML modeling 90
01. Use case diagram 91
02. Class diagram 109
03. Sequence diagram 124
04. Communication diagram 134
05. State machine diagram 138
06. Activity diagram 142
07. Component diagram 148
08. Deployment diagram 153
09. Package diagram 158
10. Object diagram 162
11. Composite structure diagram 166
12. Timing diagram 170
13. Interaction overview diagram 177
03 Part III: Enterprise architecture 182
01. Zachman Framework 183
02. Business Motivation Model diagram 191
03. ArchiMate diagram 196
04 Part IV: SoaML modeling 205
01. Service interface diagram 206
02. Service participant diagram 209
03. Service contract diagram 213
04. Services architecture diagram 216
05. Service categorization diagram 219
05 Part V: Business modeling 220
01. Business Process Diagram 221
02. Conversation Diagram 246
03. Data Flow Diagram 248
04. Event-driven Process Chain Diagram 251
05. Process Map Diagram 256
06. Organization Chart 258
07. RACI Chart 261
06 Part VI: Business rule 263
01. Manage business rule 264
02. Fact diagram 272
03. Decision table 275
07 Part VII: Simulacian 279
01. Visual simulation 280
02. Simulacian charts 291
08 Part VIII: Database modeling 303
01. Entity relationship diagram 304
02. Object relational mapping diagram 312
09 Part IX: Requirements gathering 314
01. UeXceler 315
02. Actor List 320
03. Requirement List 326
04. Use Case List 329
05. Use Case Notes 339
06. Use Case Flow of Events 343
07. Wireframe 362
08. Use Case Note List 397
09. Scenario List 401
10. Requirement diagram 404
11. Textual analysis 413

12. CRC card diagram ... 426
10 Part X: Glossary management ... 428
01. Working with glossary ... 429
02. Grid diagram ... 436
11 Part XI: Mind mapping ... 442
01. Mind Mapping Diagram ... 443
02. Brainstorm ... 455
12 Part XII: Diagramming toolset ... 458
01. Creating diagrams ... 459
02. Project management properties ... 483
03. Documenting model elements ... 487
04. Style and formatting ... 494
05. General modeling techniques ... 514
06. Advanced modeling techniques ... 559
07. Annotations and freehand shapes ... 598
08. Resource referencing ... 614
09. Using shape editor ... 624
10. Customizing user interface ... 630
13 Part XIII: Modeling toolset ... 633
01. Organizing works with model ... 634
02. Animacian ... 638
03. Maintaining project reference ... 657
04. Model element nicknaming ... 670
05. Visual Diff ... 679
06. Using design pattern ... 692
07. Model transitor ... 698
08. Using stereotypes ... 703
09. Customizing elements with profile ... 711
14 Part XIV: Impact analysis ... 716
01. Introduction of impact analysis ... 717
02. Matrix diagram ... 719
03. Analysis diagram ... 726
04. Chart Diagram ... 733
15 Part XV: Documentation generation ... 743
01. Generating HTML/PDF/Word report ... 744
02. Customizing report ... 754
03. Publishing project to web site ... 772
04. Report composer ... 780
05. Report composer templates ... 803
16 Part XVI: Interoperability ... 833
01. Export and import XML ... 834
02. Export and import VP project ... 838
03. Export and import Microsoft Excel ... 842
04. Export and import XMI ... 848
05. Importing Visio drawing ... 853
06. Importing Rational Rose model ... 856
07. Importing Rational Software Architect file ... 859
08. Importing Erwin Data Modeler project file ... 862
09. Importing Telelogic Rhapsody and System Architect project file ... 865
10. Importing NetBeans 6.x UML diagrams ... 869
11. Importing BizAgi ... 872
12. Export diagram to various graphic formats ... 875
13. Extend functionalities with Open API ... 883
14. Command line operations ... 900
15. Printing diagrams ... 911
17 Part XVII: Team collaboration ... 923
01. Getting started ... 924
02. Basic features ... 931
03. Advanced features ... 943
04. PostMania ... 958
18 Part XVIII: Appendix A - Application Options ... 964
01. General ... 965
02. Diagramming ... 974

03. View 987
04. Office Exchange 989
05. User Path 991
06. File Types 993
07. Spell Checking 995
08. Keys 997
09. Import/Export 999
19 Part XIX: Appendix B - Project Options 1000
01. Diagramming 1001
02. ORM 1026
03. Data Type 1029
04. Model Quality 1031
20 Part XX: Appendix C 1032
01. Automatic update 1033
02. Connection rules 1038

Introduction to Agilian

This chapter gives you an introduction about Agilian. The following topics will be covered:

About Agilian

A brief description of Agilian which outlines some of the key features that Agilian supports.

Editions

A summary of editions and their supported features.

Licensing

Agilian need to run with a key. This page shows you how to import different kinds of key into Agilian.

Software maintenance

Describes what software maintenance provides and tell you why you need it.

System requirements

A description of hardward requirements.

Agilian User's Guide Trademark Disclaimer

Product names, logos, brands and other trademarks referred to within Visual Paradigm's products and services and within visual-paradigm.com are the property of their respective trademark holders. These trademark holders are not affiliated with Visual Paradigm International, our products, or our website. They do not sponsor or endorse our materials. Below is a partial listing of these trademarks and their owners. This list is subject to change without notice.

- ActionScript is a trademark of Adobe Systems Inc.
- Android is a trademark of Google Inc.
- ArchiMate is a trademark of The Open Group.
- BPMN, CORBA and UML are trademarks of the Object Management Group
- C#, C++, MS SQL Server, Microsoft Windows, Microsoft Visio, VB.NET and Visual Studio are trademarks of Microsoft Corporation.
- Cache is a trademark of InterSystems Corporation.
- Cloudscape, DB2, Informix and Rhapsody are trademarks of International Business Machines Corporation.
- Derby is a trademark of the Apache Software Foundation.
- Eclipse is a trademark of Eclipse Foundation, Inc.
- Erwin is a trademark of CA Inc.
- Firebird is a trademark of Firebird Foundation Inc.
- FrontBase is a trademark of FrontBase Inc.
- H2 is a trademark of H2, Inc.
- Hibernate is a trademark of Red Hat, Inc.
- HSQL is a trademark of the hsql Development Group.
- IntelliJ IDEA is a trademark of JetBrains.
- Ingres is a trademark of Actian Corporation.
- iPad and iPhone are trademarks of Apple Inc.
- Java, MySQL and Oracle are trademarks of Oracle Corporation.
- NetBeans is a trademark of Sun Microsystems, Inc.
- Objective-C is a trademark of Stepstone Corporation
- OpenEdge is a trademark of Progress Software Corporation.
- Perl is a trademark of The Perl Foundation.
- PHP is a trademark of The PHP Group.
- PostgreSQL is a trademark of the PostgreSQL Global Development Group.
- Python is a trademark of the Python Software Foundation
- SQLite is a trademark of Hipp, Wyrick & Company, Inc.
- Sybase ASE are Sybase SQL Anywhere are trademarks of Sybase Inc.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Agilian official website
- Contact us if you need any help or have any suggestion

Agilian product overview

Agilian is an easy-to-use software engineering diagramming tool that supports all contemporary modeling notations. Agilian provides flexible modeling environment for agile software development practitioners to communicate effectively with UML, BPMN, ERD, DFD and mind map.

UML Modeling

You can draw all kinds of UML 2.x diagrams in Agilian, which include:

- Class diagram
- Use case diagram
- Sequence diagram
- Communication diagram
- State machine diagram
- Activity diagram
- Component diagram
- Deployment diagram
- Package diagram
- Object diagram
- Composite structure diagram
- Timing diagram
- Interaction overview diagram

Business process modeling

With Agilian, you can design and model your system with the following diagrams:

- Business process diagram
- Conversation diagram
- Process map diagram
- Event-driven process chain (EPC) diagram
- Data flow diagram
- Organization chart

Data modeling

Business analysts model processes with diagrams like business process diagram (BPD) and process map diagram. They model that data that is required or produced by processes using entity relationship diagram. ERD, short for entity relationship diagram, presents data in entities form, with relationships show in between, presenting the inter-relationship among pieces of data.

Team collaboration

For users that work as a team, team collaboration support lets you perform modeling collaboratively and concurrently with any one of the following tools or technologies:

- VP Teamwork Server (Need to buy Visual Paradigm Teamwork Server additionally)
- CVS
- Subversion
- Perforce
- ClearCase

Documentation generation

Share your design with your customers in popular document formats, including:

- HTML (report generation)
- HTML (project publisher)
- PDF
- Microsoft Word

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Agilian official website
- Agilian's enterprise architecture supports

Editions

Belows are the kinds of features that can be found in each edition of Agilian. For details, please visit: http://www.visual-paradigm.com/product/ag/editions/

	Enterprise	Standard
UML Modeling	●	●
Use Case Diagram	●	●
Flow of Events Editor	●	
Requirement Diagram	●	●
SoaML Modeling	●	
Glossary Grid	●	●
Entity Relationship Diagram	●	●
Business Process Modeling	●	
Mind Mapping	●	●
UML Profile	●	●
Design Pattern	●	●
Visual Diff	●	●
Animacian	●	
Simulacian	●	
Team Collaboration	●	●
Impact Analysis	●	
PDF, HTML, Word Report Generation	●	●
Report Template Designer	●	●

The filled circle indicates the support of certain feature in certain edition

A summary of features supported by Agilian

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Full comparison of Agilian editions, from Agilian Standard to Agilian Enterprise
- Overview of Agilian Enterprise, with features and strength listed
- Overview of Agilian standard edition, with features and strength listed

Licensing

Agilian needs to run with a valid license key. The various licensing options listed in this page vary in price and functionality.

Various licensing options

Single seat license

Visual Paradigm's single-seat (team-member-based) license allows a licensee to install the software on a computer that belongs, and provides sole access, to the named user only. Since the license is team-member-based, the software must be used by the licensee only, without running more than one instance concurrently. The single-seat license only allows installation on a maximum of three computers.

Edition	Unit price	Unit price (with 1 year maintenance)
Enterprise	$999.00	$1195.50
Standard	$299.00	$358.50
Viewer	Free	Not applicable

Price of single seat licenses (Prices are provided in US dollars)

Floating license

The floating license supports sharing of the pool of licenses among your team. Instead of purchasing a single-seat license for each team member, optimize your budget by purchasing floating licenses for the maximum number of simultaneous software users or access points. This approach allows greater flexibility in using our software. Users can then export the license files to a laptop to use the software offsite (to deliver a presentation, for example), and then import the license back to the server at a later time.

Edition	Unit price	Unit price (with 1 year maintenance)
Enterprise	$1298.50	$1558.00
Standard	$388.50	$466.00
Viewer	Not applicable	Not applicable

Price of floating licenses (Prices are provided in US dollars)

In order to work with the floating license, installation of a VP Server that stores the license key file(s) and also automatically manages access requests from clients is required. The client must enable the connection to the license server when requesting access to the software.

For more information about floating licenses, please visit
http://www.visual-paradigm.com/shop/floatinglicense.jsp

Subscription license

Edition	Unit price (per month)
Enterprise	$49.00
Standard	$15.00
Viewer	Not applicable

Price of subscription licenses (Prices are provided in US dollars)

Academic license

Academic licenses are available for higher education, with the aim of providing free site licenses for the teaching of software engineering. Educational institutions that join the Academic Partners Program are entitled to free licenses for the Standard Edition of Visual Paradigm's software, which can then be used solely for educational purposes. The academic license is not limited to use on campus, but can also be used at home by students and teachers.

For more information about academic licenses, please visit
http://www.visual-paradigm.com/partner/academic/ .

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Buy Agilian, the enterprise architecture software online in Visual Paradigm online store
- Use Agilian for educational purposes. Join Visual Paradigm Academic Partner Program (VPAPP)
- Wanna know more about how floating license can help you share license between users? Visit this page
- See the most complete, updated Agilian price details
- Can you switch to another edition after purchase? Yes, you can. See how to do
- Keep Agilian updated. Know more about Visual Paradigm's product maintenance policy

Software maintenance

The Visual Paradigm Software Maintenance package includes both version upgrades and technical support services for our customers. The following benefits are all included in the Visual Paradigm Software Maintenance package.

Version upgrades

Typically, Visual Paradigm produces two to three versions per year, with approximately five to ten major new features and enhancements per version. You are entitled to install any new versions of Visual Paradigm product that are released within your software maintenance period.

Technical support

You and your team members can submit technical support tickets to our Technical Support Team at http://www.visual-paradigm.com/support/technicalsupport.jsp

Our Technical Support Team will respond to your message within one working day. Normally, you will receive our response by email within a few hours.

Visual Paradigm is committed to delivering extraordinary technical support to our customers. Our Technical Support Team employs the following technologies to back up our products.

Email with text and screen shot attachments

In most cases we can provide assistance by guiding you with the aid of screen shots.

Flash demo

Sometimes, a short movie is more descriptive than a thousand words. If the answer to your question is complex, we can prepare a short Flash demonstration to guide you in resolving your difficulty.

Secure online sessions

We can schedule an online meeting with you to take an interactive look at your issue. Online meetings are held using a secure Internet connection. During the meeting, our team can remotely access and operate your PC while speaking with you by telephone or while chatting with you using the built-in chat program.

Telephone

You can leave a callback request at the following URL. Our Technical Support Team will return your call as soon as possible. To make a call, visit: http://www.visual-paradigm.com/support/callme.jsp

Price

Software maintenance is purchased on an annual basis (e.g., June 20, 2012 to June 19, 2013).

If you decide to purchase the software maintenance package with your product, or if you decide to extend a current maintenance contract, the yearly cost is 20% of the product list price. To take advantage of this 20% offer, you must extend your maintenance contract at least one week prior to its expiration date.

If you decide to purchase a software maintenance package separately, the yearly cost is 30% of the product list price.

You can purchase software maintenance to cover up to three years from the date of purchase.

Detailed software maintenance package pricing is listed below.

Single seat license

Prices are provided in US dollar

Edition	1 Year Maintenance (extend current maintenance)	1 Year Maintenance (buy maintenance separately)
Enterprise	$199.50	$299.50
Standard	$59.50	$89.50
Viewer	not applicable	not applicable

The above software maintenance contract prices are for 1 year only.

Price for single-seat license

Floating license

Prices are provided in US dollar

Edition	1 Year Maintenance (extend current maintenance)	1 Year Maintenance (buy maintenance separately)
Enterprise	$259.50	$389.50
Standard	$77.50	$116.50
Viewer	not applicable	not applicable

The above software maintenance contract prices are for 1 year only.

Price for floating license

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Buy Agilian, the enterprise architecture software online in Visual Paradigm online store
- Use Agilian for educational purposes. Join Visual Paradigm Academic Partner Program (VPAPP) for FREE
- See the most complete, updated Agilian price details
- Can you switch to another edition after purchase? Yes, you can. See how to do

System requirements

Hardware requirements

- Intel Pentium 4 at 2.0 GHz or higher
- Minimum 512MB RAM, but 1.0 GB is recommended
- Minimum 800MB disk space
- Microsoft Windows (98/2000/XP/2003/Vista/7/8), Linux, Mac OS X, Solaris or all other Java-enabled platforms
- JDK 1.6 for Mac OS X

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Download Agilian and try it FREE

Getting started

This chapter covers mainly the installation of Agilian on various platforms, as well as the steps of switching between product editions and how to remove Agilian.

Installing Agilian on Windows 2000/NT/2003/XP/Vista/7/8

List the steps of installing Agilian on Microsoft Windows as well as the use of no-install (zip) version.

Installing Agilian on Mac OS X

List the steps of installing Agilian on Mac OS X as well as the use of no-install (zip) version.

Installing Agilian on Linux and Unix

List the steps of installing Agilian on Linux and Unix as well as the use of no-install (zip) version.

Starting Agilian

List the steps of starting Agilian, with a brief description on 'workspace'.

Changing edition

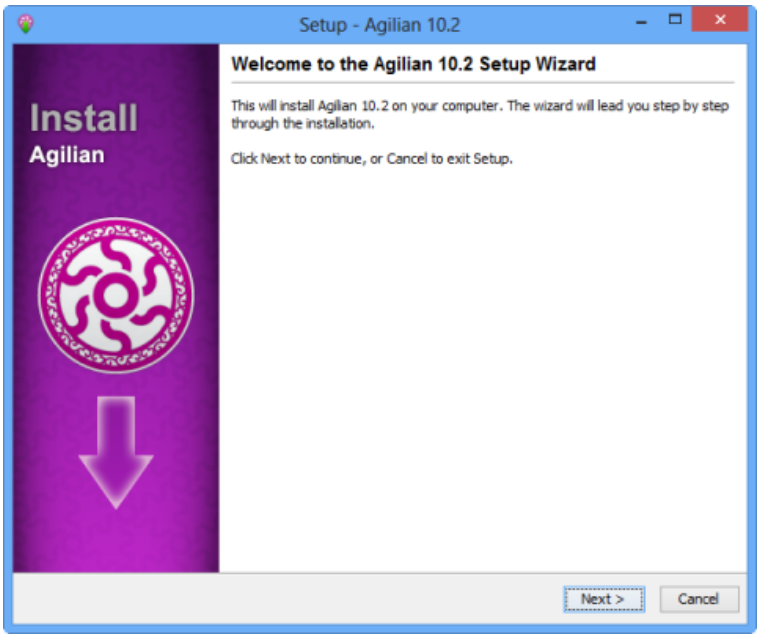
You can switch between editions without re-installation. This page shows you how to do.

Installing Agilian on Windows 2000/ NT/ 2003/XP/ Vista/7/8

Having downloaded the installer of Agilian , execute it, run through the installation to install Agilian. If you are using the "no-install" zip version, you just need to unzip it and run Agilian directly. In this chapter, we will go through the installation of Agilian both with installer (.exe) and "no-install" (.zip).

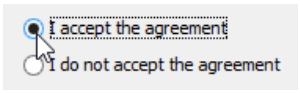
Using installer (.exe)

1. Execute the downloaded Agilian installer file. The setup wizard appears as below.



Agilian welcome screen

2. Click **Next** to proceed to the **License Agreement** page.
3. Read through the license agreement carefully. Make sure you accept the terms before continuing with the installation. If you accept the agreement, select **I accept the agreement** and click **Next** to proceed to the **Select Destination Directory** page.



The License Agreement

4. Specify the directory for installing Agilian. Click **Next** to proceed to the next page.
5. Specify the name of the Start Menu folder that will be used to store the shortcuts. Keep **Create shortcuts for all users** checked if you want the shortcut to be available in all the user accounts in the machine. Click **Next** to proceed.
6. In the **File Association** page, keep **Visual Paradigm Project (*.vpp)** checked if you want your system able to open the project file upon direct execution (i.e. double click). Click **Next** to start the file copying process.
7. Upon finishing, you can select whether to start Agilian or not. Keep **Agilian** selected and click **Finish** will run Agilian right away.

Using "no install" version (.zip)

Decompress the downloaded zip file into a directory. This creates a subdirectory named "Agilian 10.2" where 10.2 is the version number. That's it. To start Agilian, execute **bin\Agilian.exe**.

Installation FAQ

Question: What is the difference between Installer and "No Install" Version?
Answer: Installer version creates shortcut and registers the menus that make the system more easy to use. We suggest user to use installer version for a long term usage. The "No Install" version is good for evaluation and testing the release candidate.
Question: I cannot complete the installation due to a file is missing when copying files. What can I do?
Answer: This can be caused by a corrupted installer file. Please download the installer file again with a different mirror site and run it again to solve the problem.
Question: I cannot start the application after installing the software. What can I do?
Answer: There are several possible causes of the problem. If you are sure that your installation was performed correctly, contact Visual Paradigm's support team (support-team@visual-paradigm.com) for assistance. It is recommended to include the **ag.log** file in **%HOME-DIR%\visualparadigm** (e.g. *C:\Users\Peter\visualparadigm\ag.log*).
Question: I don't have administrator right, can I install the software?
Answer: Yes, you can.
Question: The installer file is detected to contain a virus. What can I do?
Answer: Our installer files are all packed by ourselves in a secure environment, and are scanned for virus before releasing to public. If a virus is detected, please update to the latest virus profile first. After that, we recommend you perform a full system scan, download the installer file from our official site, and run the installation again. If the problem remain, please contact us (support-team@visual-paradigm.com) or the virus scanner vendor for assistance.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Download Agilian and try it FREE
- New to Agilian? We have a lot of tutorials written to help you get started with enterprise architecture with Agilian
- Have difficulties when installing Agilian? Contact us. We will help you out

Installing Agilian on Mac OS X

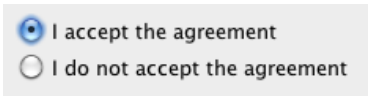
Having downloaded the installer of Agilian , execute it, run through the installation to install Agilian. If you are using the "no-install" .tgz version, you just need to unzip it and run Agilian directly. In this chapter, we will go through the installation of Agilian both with installer (.dmg) and "no-install" (.tgz).

Using installer (.dmg)

1. Execute the downloaded Agilian installer file. The setup wizard appears as below.

Agilian welcome screen

2. Click **Next** to proceed to the **License Agreement** page.
3. Read through the license agreement carefully. Make sure you accept the terms before continuing with the installation. If you accept the agreement, select **I accept the agreement** and click **Next** to proceed to the **Select Destination Directory** page.



The License Agreement

4. Specify the directory for installing Agilian. Click **Next** to proceed to the next page.
5. In the **File Association** page, keep **Visual Paradigm Project (*.vpp)** checked if you want your system able to open the project file upon direct execution (i.e. double click). Click **Next** to start the file copying process.
6. Upon finishing, you can select whether to start Agilian or not. Keep **Agilian** selected and click **Finish** will run Agilian right away.

Using "no install" version (.tgz)

Decompress the downloaded .tgz file into a directory. This creates a subdirectory named "Agilian 10.2" where 10.2 is the version number. That's it. To start Agilian, execute **bin\ Agilian .app**.

Installation FAQ

Question: What is the difference between Installer and "No Install" Version?
Answer: Installer version creates shortcut and registers the menus that make the system more easy to use. We suggest user to use installer version for a long term usage. The "No Install" version is good for evaluation and testing the release candidate.
Question: I cannot complete the installation due to a file is missing when copying files. What can I do?
Answer: This can be caused by a corrupted installer file. Please download the installer file again with a different mirror site and run it again to solve the problem.
Question: I cannot start the application after installing the software. What can I do?
Answer: There are several possible causes of the problem. If you are sure that your installation was performed correctly, contact Visual Paradigm's support team (support-team@visual-paradigm.com) for assistance. It is recommended to include the **ag.log** file in **%HOME-DIR%\visualparadigm** (e.g. *C:\Users\Peter\visualparadigm\ag.log*).
Question: The installer file is detected to contain a virus. What can I do?
Answer: Our installer files are all packed by ourselves in a secure environment, and are scanned for virus before releasing to public. If a virus is detected, please update to the latest virus profile first. After that, we recommend you perform a full system scan, download the installer file from our official site, and run the installation again. If the problem remain, please contact us (support-team@visual-paradigm.com) or the virus scanner vendor for assistance.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Download Agilian and try it FREE
- New to Agilian? We have a lot of tutorials written to help you get started with enterprise architecture with Agilian

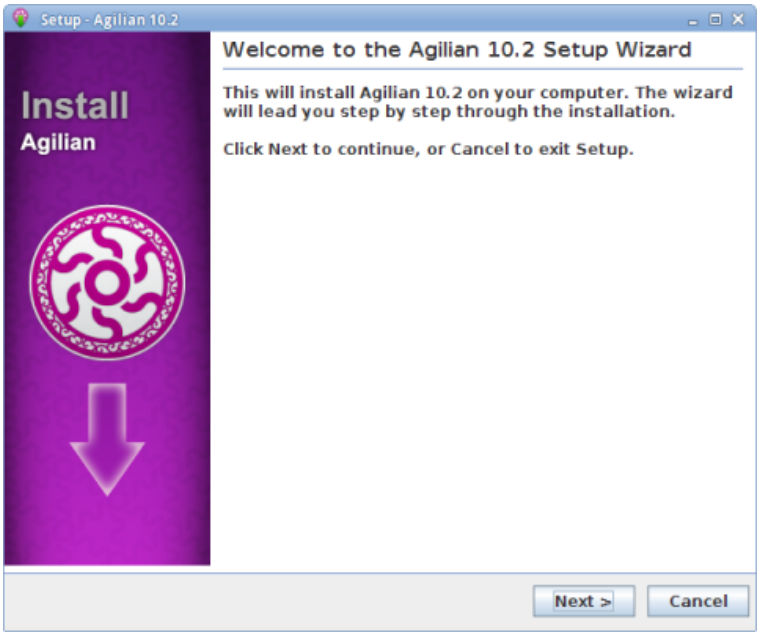
- Have difficulties when installing Agilian? Contact us. We will help you out

Installing Agilian on Linux and Unix

Having downloaded the installer of Agilian, execute it, run through the installation to install Agilian. If you are using the "no-install" zip version, you just need to unzip it and run Agilian directly. In this chapter, we will go through the installation of Agilian both with installer (.sh) and "no-install" (.tar.gz).

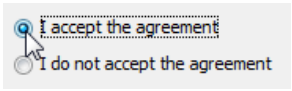
Using installer (.sh)

1. Execute the downloaded Agilian installer file.
   bash ./%AGILIAN-INSTALLER-FILENAME%
   The setup wizard appears as below. If you are prompted an error like "bin/unpack200: /lib/ld-Linux.so.2: bad ELF interpreter: No such file or directory. Error unpacking jar files. The architecture or bitness (32/64)", make sure you are executing the right installer - 64 bit / 32 bit. You can download any of them from our official website.



Agilian welcome screen

2. Click **Next** to proceed to the **License Agreement** page.
3. Read through the license agreement carefully. Make sure you accept the terms before continuing with the installation. If you accept the agreement, select **I accept the agreement** and click **Next** to proceed to the **Select Destination Directory** page.



The License Agreement

4. Specify the directory for installing Agilian. Click **Next** to proceed to the next page.
5. Select a folder for creating symlinks. You may uncheck **Create symlinks** if you do not want to. Click **Next** to the start file copying process.
6. Upon finishing, you can select whether to start Agilian or not. Keep **Agilian** selected and click **Finish** will run Agilian right away.

Using "no install" version (.tar.gz)

Decompress the downloaded .tar.gz file into a directory: *tar -zxf %NO-INSTALL-FILE.tar.gz% -C %DESTINATION-FOLDER%*

This creates a subdirectory named "Agilian 10.2" where 10.2 is the version number. That's it. To start Agilian, execute **bin\Agilian**.

Installation FAQ

Question: What is the difference between Installer and "No Install" Version?
Answer: Installer version creates shortcut and registers the menus that make the system more easy to use. We suggest user to use installer version for a long term usage. The "No Install" version is good for evaluation and testing the release candidate.
Question: I cannot complete the installation due to a file is missing when copying files. What can I do?
Answer: This can be caused by a corrupted installer file. Please download the installer file again with a different mirror site and run it again to solve the problem.
Question: I cannot start the application after installing the software. What can I do?
Answer: There are several possible causes of the problem. If you are sure that your installation was performed correctly, contact Visual Paradigm's support team (support-team@visual-paradigm.com) for assistance. It is recommended to include the **ag.log** file in **%HOME-DIR%\visualparadigm** (e.g. *C:\Users\Peter\visualparadigm\ag.log*).
Question: The installer file is detected to contain a virus. What can I do?
Answer: Our installer files are all packed by ourselves in a secure environment, and are scanned for virus before releasing to public. If a virus is detected, please update to the latest virus profile first. After that, we recommend you perform a full system scan, download the installer file from our official site, and run the installation again. If the problem remain, please contact us (support-team@visual-paradigm.com) or the virus scanner vendor for assistance.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Download Agilian and try it FREE
- New to Agilian? We have a lot of tutorials written to help you get started with enterprise architecture with Agilian
- Have difficulties when installing Agilian? Contact us. We will help you out

Starting Agilian

Ways of starting Agilian

Agilian can be started through accessing the **Start** menu for Windows 7 or earlier, or **Start** screen for Windows 8. For Linux users, Agilian can be started through the shortcuts in desktop, created by the installer.

Starting Agilian (for floating license client whose host is IP-4-enabled)

If you are a floating license client, and if your host is IP-4 enabled, you need to start Agilian with a startup script in order to connect to the server. Here are the steps:

1. Copy **Agilian.bat** under the scripts folder of VP Suite installation directory to become Startup.bat
2. Edit Startup.bat
3. Add -Djava.net.preferIPv4Stack=true to the script

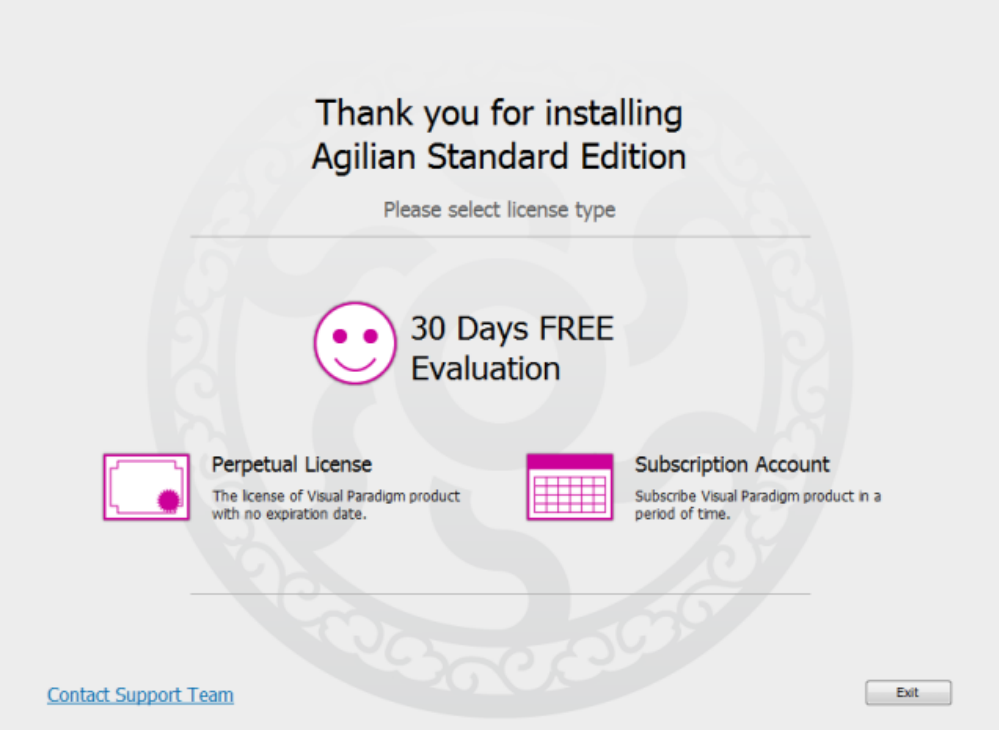
```
pushd ..\bin
start ..\jre\bin\javaw -Xms256m -Xmx768m -XX:MaxPermSize=256m -Dvpproduct=AG -cp ".;..\
lib\vpplatform.jar;..\lib\jniwrap.jar;..\lib\winpack.jar;..\ormlib\orm.jar;..\ormlib\orm-
core.jar;..\lib\xalan.jar;..\lib\lib01.jar;..\lib\lib02.jar;..\lib\lib03.jar;..\lib\
lib04.jar;..\lib\lib05.jar;..\lib\lib06.jar;..\lib\lib07.jar;..\lib\lib08.jar;..\lib\
lib09.jar;..\lib\lib10.jar" -Djava.net.preferIPv4Stack=true RV %*
popd
```

Editing the start up script

4. Save
5. From now on, execute **Startup.bat** to run Agilian.

Starting Agilian the first time

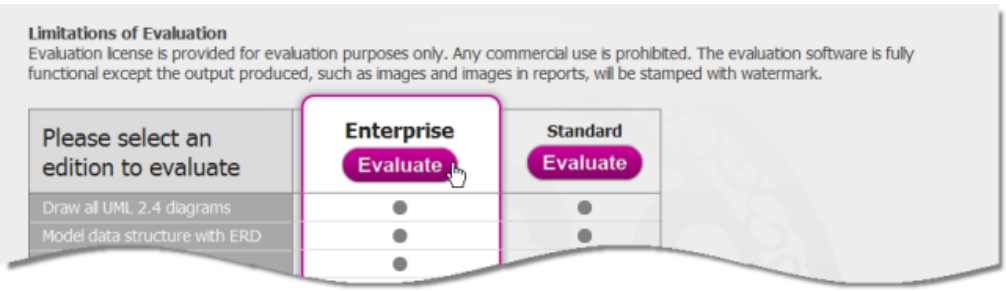
When you start Agilian the first time, you are asked to select a way to "unlock" Agilian.



Select a way to unlock Agilian

30 Days FREE Evaluation

If you want to evaluate Agilian, click this. You will then be asked to select the edition of product to evaluate. Agilian features vary by product edition. For more details on the features supported by different editions, check the Edition Comparison page. Click on the **Evaluate** button to confirm your edition selection. Then, you can start your 30 days evaluation.



Perpetual License
If you have purchased a license and you want to unlock Agilian with it, click this.

There are several ways you can take to import your license key into Agilian. The first way is to enter the activation code and click **Activate**. You can obtain the code by visiting your customer account at our Customer Service Center. Alternatively, the licensee should have received our Email notification with activation code included.

The second way is to import the .zvpl license key file by expanding the **License Key** section and clicking on the button **...** to select the key file. Again, you can find the key file in your customer account.

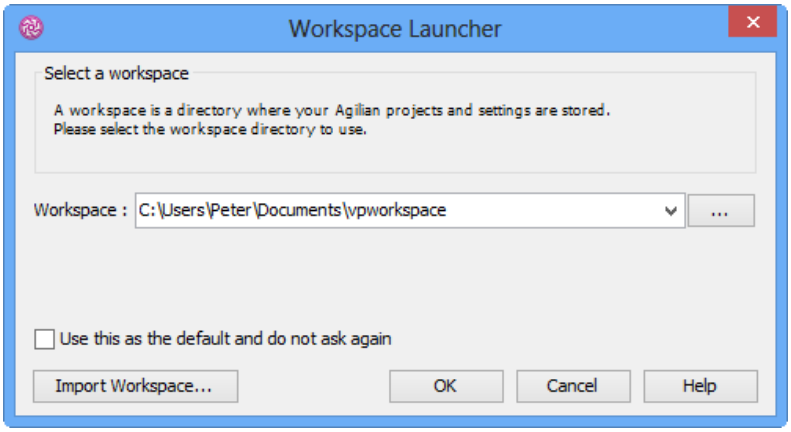
If you are using a floating license, expand the **Floating License** section, enter the connection settings of the host machine where the license is installed and click **Apply**.

Subscription Account
If you have subscribed to run Agilian for certain period of time, click this. Enter the login details of your customer account and click **Sign in** to continue.

Select workspace

A workspace is a directory used to store all settings, user interface perspectives and other preferences defined for the working environment (settings can be configured via **Tools > Application Options...** in Agilian). A workspace also stores all the teamwork login information and local copies of teamwork projects. In the case of switching computers, you simply need to copy the whole workspace directory to the new computer and specify the new workspace when starting Agilian. All your teamwork information and settings will then be transferred to the new computer.

The **Workspace Launcher** appear when running Agilian.



Workspace Launcher

Specify the workspace folder.

If you do not want this dialog box to appear again, check **Use this as the default and do not ask again**. This will cause Agilian to open the specified workspace folder automatically the next time.

If you already have an existing workspace, you can import the settings from there by clicking **Import Workspace...**.

Click **OK** to continue.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Wanna use Agilian for educational purposes? Join our Academic Partner Program now
- Learn more about floating license - how to import, export, return a floating license
- Full Agilian edition comparison

Changing edition

You may want to run the product in a different to use the functions supported by that edition. This is particularly common during evaluation. You want to try out different editions to find out the one that suit you best. Product edition can be changed by updating the license applied, without the need of re-installation.

1. Select **Tools > License Manager...** from the main menu.
2. Click **Change License...** at bottom left. Answer **OK** when you are prompted to delete the existing license.
3. If you want to evaluate another edition, click on the smiley face and click **Evaluate** at the column of edition you want to switch to. If you own a perpetual license of the target edition, click on **Perpetual License** and enter the activation code of the license, or update the connection of floating license access if your team owns a floating license. If you have subscribed antoher edition of product, click on **Subscription Account** and sign in with your licensee ID and password.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- Full comparison of Agilian editions, from Agilian Standard to Agilian Enterprise
- Overview of Agilian Enterprise, with features and strength listed
- Overview of Agilian Standard, with features and strength listed
- Can you switch to another edition after purchase? Yes, you can. See how to do

Uninstalling Agilian

This chapter teaches you how to uninstall Agilian from your system.

Uninstalling Agilian

List the steps of uninstalling Agilian.

Uninstalling Agilian

If you want to remove Agilian from your system, you can perform an uninstallation.

Uninstalling Agilian will remove the files in your Agilian installation from system. If you have installed Agilian through installer, you can uninstall it by running the **uninstall** file right under the installation directory. If you are using a no-install version of Agilian, which means that the installation was produced by decompressing the zip file that contains the installation, you just need to delete the whole installation folder to have Agilian removed.

Note that uninstallation does not clear the setting files that are stored under the Home directory.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- VP-UML product home page
- Logizian product home page

User interface

This chapter walks through the various panes and components in user interface.

Interface overview

A summary of the user interface you can see when Agilian is started.

Main menu

The main menu enables you to access most of the core functions in Agilian.

Toolbar

The toolbar is a by default horizontal bar below the main menu bar which covers most of the core functions in Agilian.

Dockable user interface

Dockable user interface refers to the ability to drag out a pane and dock it to another part of the application screen.

Diagram navigator

Diagram navigator is a pane that lets you access and create diagrams.

Model explorer

Model explorer is a pane that lets you access and create models, and browse for their specifications.

Class repository

Class repository is a pane that lets you access and create classes, and browse for their specifications.

Logical view

Logical view is a pane that lets you organize diagrams with user-named views.

Property pane

Property pane is a pane that lets you read and edit chosen element(s) 's properties.

Diagram overview

Diagram overview is a thumbnail of diagram which enables you to navigate and zoom into a diagram.

Documentation pane

Documentation pane is a pane that lets you read and edit documentation of chosen element.

Stencil pane

Stencil pane is a pane that list stencil, and lets you drag out a shape to diagram.

Diagram specification

Diagram specification enables you to adjust some of the diagram settings

Perspective

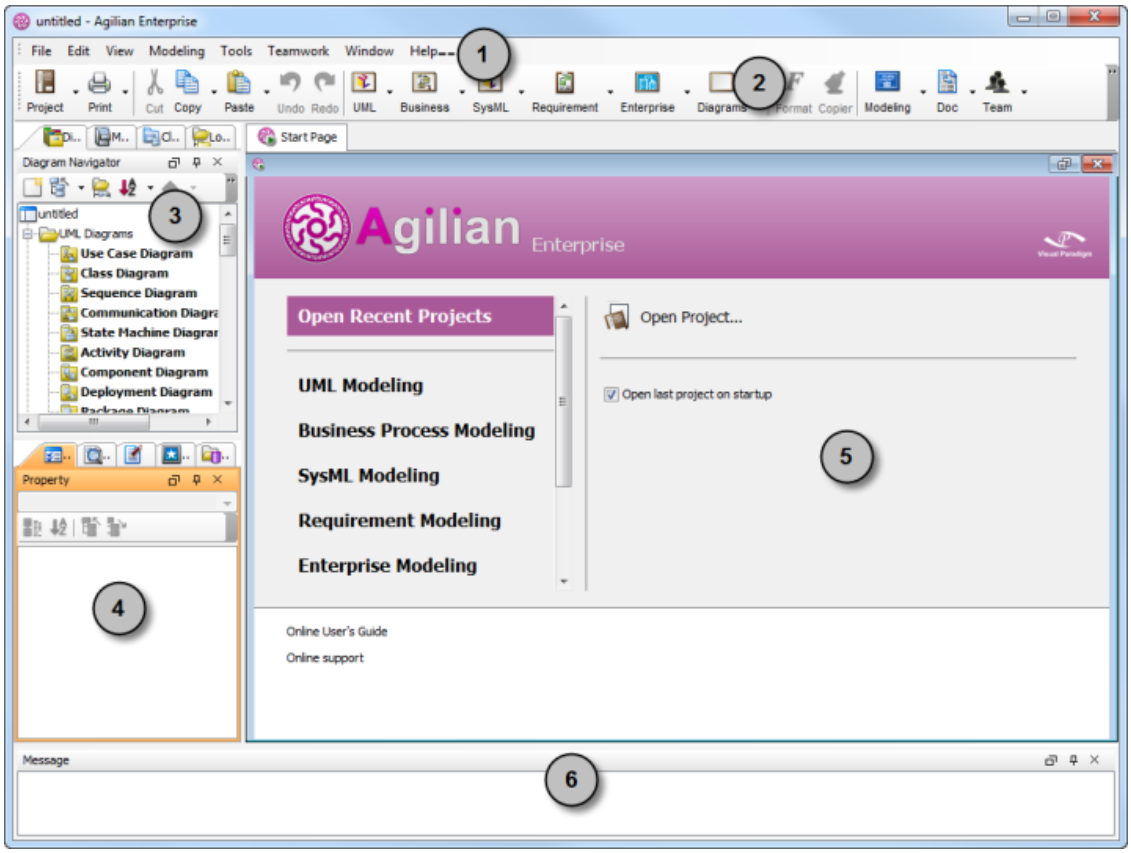
Perspective define way to position panes.

Model element specification

Properties of model elements can be set and read through the specification dialog box.

Interface overview

Agilian 's user interface comprises a menu bar, a toolbar, several panes for model navigation, a message pane and a diagram pane that occupy most spaces.



*User Interface of **Agilian***

No.	Name	Description
1	Menu bar	The menu bar at the top of the window allows you to select and perform various operations in Agilian.
2	Toolbar	Toolbar, which is below the menu bar, is the extension of menu. All buttons are presented as groups of icons that handily placed for users.
3	Diagram navigator	A place where diagrams are listed, and where you can create and access diagrams base on their types.
4	Property pane	The properties of chosen model/ shapes will be shown on properties pane upon selection.
5	Diagram pane	The diagram will be displayed in diagram pane.
6	Message pane	Possible information, notification or warnings will be shown here.

Description of user interface

Related Resources

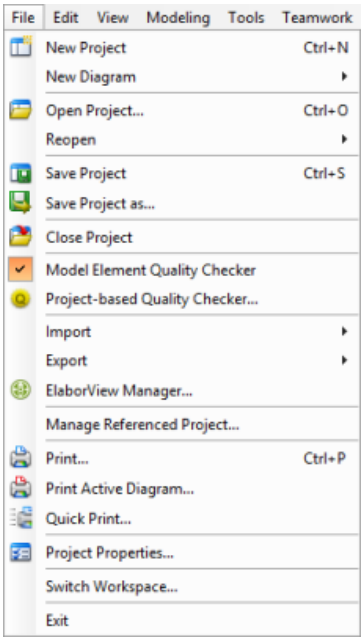
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Menu bar

The main menu, which is on the top of the window, allows you to select and perform various operations in Agilian .

File menu

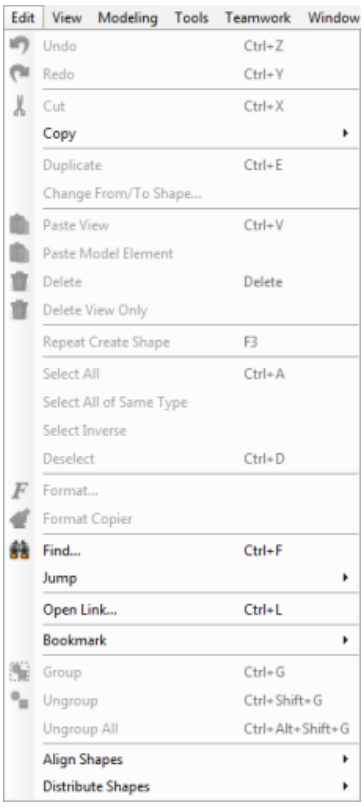


*The **File** menu*

The **File** menu enables you to:

- Create a project
- Create a diagram
- Open project
- Save project
- Import project data from the following media: Agilian Project, Rose Project, XMI, XML, ERwin Project, Telelogic Rhapsody Project, Telelogic System Architect, Rational Model, Rational DNX, PowerDesigner DataArchitect, Excel (Exported from VP), Visio, NetBeans
- Export project into the following formats: Agilian Project, XMI, XML, Excel
- Export images (JPG, PNG, SVG, EMF, PDF)
- Printing
- Set project properties
- Exit

Edit menu

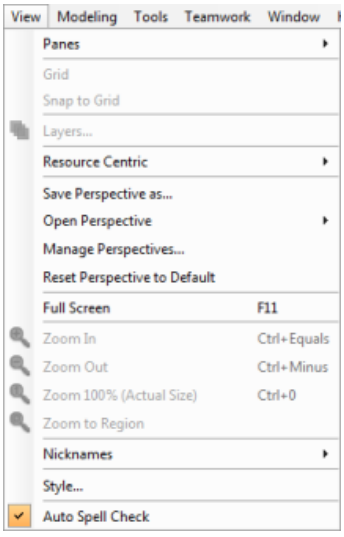


*The **Edit** menu*

The **Edit** menu enables you to:

- Undo and Redo
- Cut
- Copy
- Duplicate
- Delete
- Change the end model element of connector
- Repeat an action
- Select everything in diagram
- Find a model element/diagram
- Jump to a diagram or an element
- Add or manage bookmarks
- Grouping
- Shapes alignment and distribution

View menu

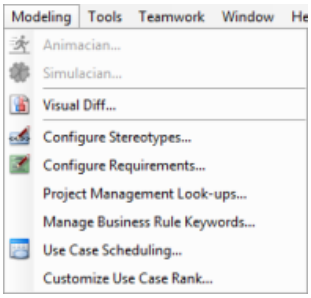


*The **View** menu*

The **View** menu enables you to:

- Show/Hide a pane
- Show and manage grid
- Manage Layers
- Change Resource Centric behavior
- Save, open and manage perspective
- Change Agilian to show in full screen
- Zoom diagram in and out
- Manage nickname
- Manage style
- Auto spell check

Modeling menu

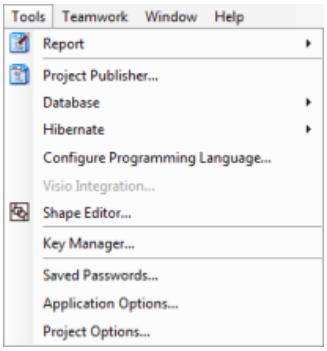


*The **Modeling** menu*

The **Modeling** menu enables you to:

- Launch Animacian
- Launch Simulacian
- Compare diagrams with Visual Diff
- Perform use case scheduling
- Configure stereotypes
- Configure requirements
- Customize use case ranks

Tools menu



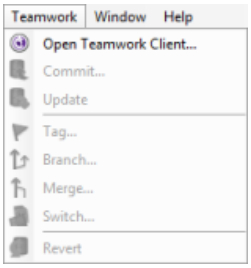
*The **Tools** menu*

The **Tools** menu enables you to:

- Generate report
- Publish project
- Configure programming language
- Perform ORM
- Perform State Machine Code Generation
- Perform Visio integration
- Launch Shape Editor
- Launch License Key Manager

- Configure application options through the Options dialog box

Teamwork menu

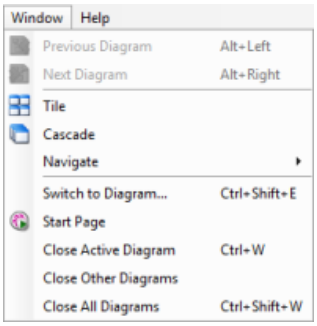


*The **Teamwork** menu*

The **Tools** menu enables you to:

- Commit changes
- Update changes
- Tagging
- Branching
- Merge
- Revert changes

Window menu

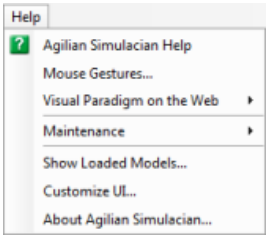


*The **Window** menu*

The **Window** menu enables you to:

- Navigate between diagrams
- Rearrange diagram windows
- Switch to another diagram
- Show the Start Page
- Close other diagrams, except the active diagram
- Close Diagrams

Help menu



*The **Help** menu*

The **Help** menu enables you to:

- Browse the help contents
- Check the instruction of Mouse Gesture
- Visit Visual Paradigm online support
- Repair project
- Customize the user interface
- Check the environment using the About dialog box

Related Resources

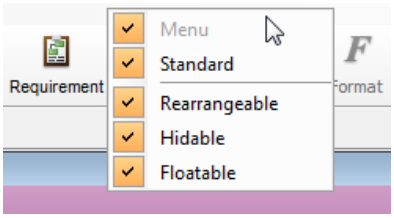
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Toolbar

Showing/Hiding toolbar

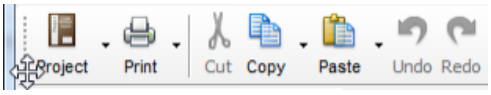
A toolbar can be shown or hidden. To show a toolbar, right click on the standard toolbar, and check **Standard** to show. Similarly, you can uncheck it to hide it.



Show or hide toolbar

Repositioning toolbar

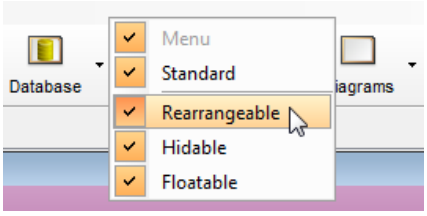
A toolbar can be repositioned by pressing on the handle of toolbar, which appears on the left edge of a toolbar, and dragging it to your target position.



Drag the left edge of toolbar

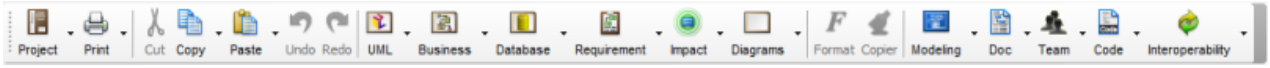
Locking toolbar

To freeze toolbars' position and make the toolbars immovable, right click on any toolbar and uncheck **Rearrangable** from the pop-up menu.



Uncheck Rearrangable option

Standard toolbar



The Standard toolbar

Icon	Name	Description
	New Project	Create a project.
	Open Project	Open a project.
	Save Project	Save the changes made in the working project.
	Print...	Select and print diagram(s) after configure the advanced printing set up.
	Print Active Diagram...	Print the active diagram after configure the advanced printing set up.
	Quick Print...	Select and print diagram(s) after configure the simple printing set up in **Quick Print** dialog box.
	Cut	Cut selected diagram elements.
	Copy within AG SN	Copy selected diagram elements ready to be used within Agilian.
	Copy to Clipboard as Image (JPG)	Copy selected diagram elements as JPG image.

Copy to Clipboard as Image (EMF)	Copy selected diagram elements as EMF image.
Copy to Clipboard as XML	Copy selected diagram elements as XML data.
Paste View	Paste copied diagram elements as view of original model element.
Paste Model Element	Paste copied diagram elements as a new model element.
Undo	Roll back undesired changes.
Redo	Rerun an undone task.
UML Modeling	Select and create a diagram type under UML Modeling, including: Use Case Diagram, Use Case Grid, Actor Grid, Class Diagram, Composite Structure Diagram, Object Diagram, Sequence Diagram, Communication Diagram, Activity Diagram, State Machine Diagram, Timing Diagram, Interaction Overview Diagram, Component Diagram, Deployment Diagram and Package Diagram.
Business Process Modeling	Select and create a diagram type under Business Process Modeling, including: Business Process Diagram, Conversation Diagram, Data Flow Diagram, Event-driven Process Chain Diagram, Process Map Diagram, Organization Diagram and ArchiMate Diagram.
Database Modeling	Select and create a diagram type under Database Modeling, including: Entity Relationship Diagram and ORM Diagram.
Requirement Modeling	Select and create a diagram type under Requirement Modeling, including: Textual Analysis, Requirement Diagram, Basic Diagram, Open Requirements Grid, Open Glossary Grid and CRC Card Diagram.
Impact Analysis	Select and create Matrix Diagram.
Diagrams	Select and create other diagrams, including: EJB Diagram, Overview Diagram, User Interface and Mind Mapping Diagram.
Format	Click it to open **Formats** dialog box. You can format font name, font style, font size, font color, line style, shape's foreground and background color and style, etc.
Copier	Copy the selected diagram element's format and apply on another shape.
Modeling	Click it to perform the following: Visual Diff, Animacian, Simulacian, Nicknames, Apply Design Pattern and Transit to New Diagram. **Visual Diff**: Launch Visual Diff for comparing diagrams. **Animacian**: Launch Animacian for animating the active diagram. **Simulacian**: Launch Simulacian for simulating the execution of business process. **Nicknames**: Manage and select the nickname to be applied on the working project. **Apply Design Pattern**: Apply the defined design pattern to the target diagram. **Transit to New Diagram**: Transit the current diagram to a new diagram.
Generate HTML Report	Open the **Generate HTML** dialog box to generate HTML report.
Generate PDF Report	Open the **Generate PDF** dialog box to generate PDF report.
Generate Word Report	Open the **Generate Word** dialog box to generate Word report.
Report Writer	Open **Report Writer** where you can create and edit your report(s).
Project Publisher	Publish project to Web pages through Project Publisher.
Open Teamwork Client	Open the **Teamwork Client** dialog box.
Commit	Commit changes to server.
Update	Update changes from server.

	Tag	Create a tag.
	Branch	Create a branch.
	Merge	Merge changes between trunk and branches.
	Switch	Switch between trunk, branches and tags.
	Revert	Click it to undo all non-committed changes in working project.
	Generate Code	Generate state machine code from class diagram and state machine diagram.
	Reverse Code	Reverse state machine code back into the opening project.
	Interoperability	**Import**: Agilian Project: Import an Agilian project into the opening project. Rose Project: Import diagrams and model elements from a Rose model. XMI: Import diagrams and model elements from XMI. XML: Import diagrams and model elements from XML. ERwin Project (XML): Import diagrams and model elements from ERwin Project. Telelogic Rhapsody Project: Import diagram and model elements from Telelogic Rhapsody Project. Telelogic System Architect: Import diagram and model elements from Telelogic System Architect. Rational Model: Import diagram and model elements from Rational Model. Rational DNX: Import diagram and model elements from Rational DNX. MS Word to Use Case Model: Import Use Case Report (MS Word) back into the opening project. PowerDesigner DataArchitect: Import diagram and model elements from Power Designer Data Model. Visual UML: Import diagram and model elements from Visual UML XML file. Excel: Import Excel file back into the working project. Visio: Import Visio diagrams into Agilian. NetBeans UML Project: Import NetBeans UML diagrams into Agilian. Linked Project: Import dependent projects into the opening project. **Export**: Agilian Project: Export the open project into an Agilian project file. XMI: Export XMI from the opening project. XML: Export XML from the opening project. Active Diagram as Image: Export active diagram as image file. Diagrams as Image: Export any diagram(s) as image file(s). Selection as Image: Export selection on active diagram as image file. Use Case Model to MS Word: Export Use Case Report (MS Word) from Use Case diagrams and use case models. Active Diagram to Excel: Export active diagram into Excel report. Excel: Export any diagram(s) into Excel report(s).

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

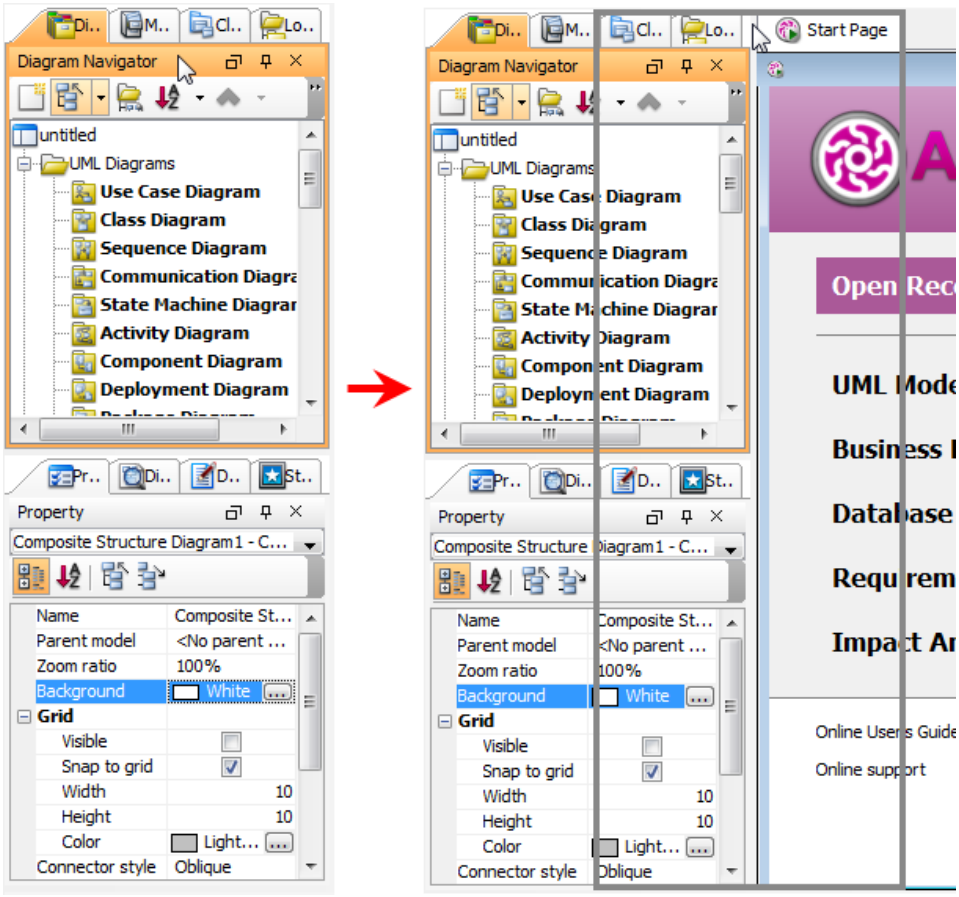
Dockable user interface

Agilian adapts a Dockable User Interface which allows you to drag UI components around to customize your favorite working environment. You can save the environment as a perspective which you can reopen later. It allows you to use different perspectives for different purposes.

Using the dockable user interface

The Dockable User Interface is composed of a number of windows called dockable frames. A dockable frame may be standalone (floating) or docked into another container (split pane/tab pane).

You can press on the title bar of a dockable frame or press on a tab to drag it to anywhere you like.



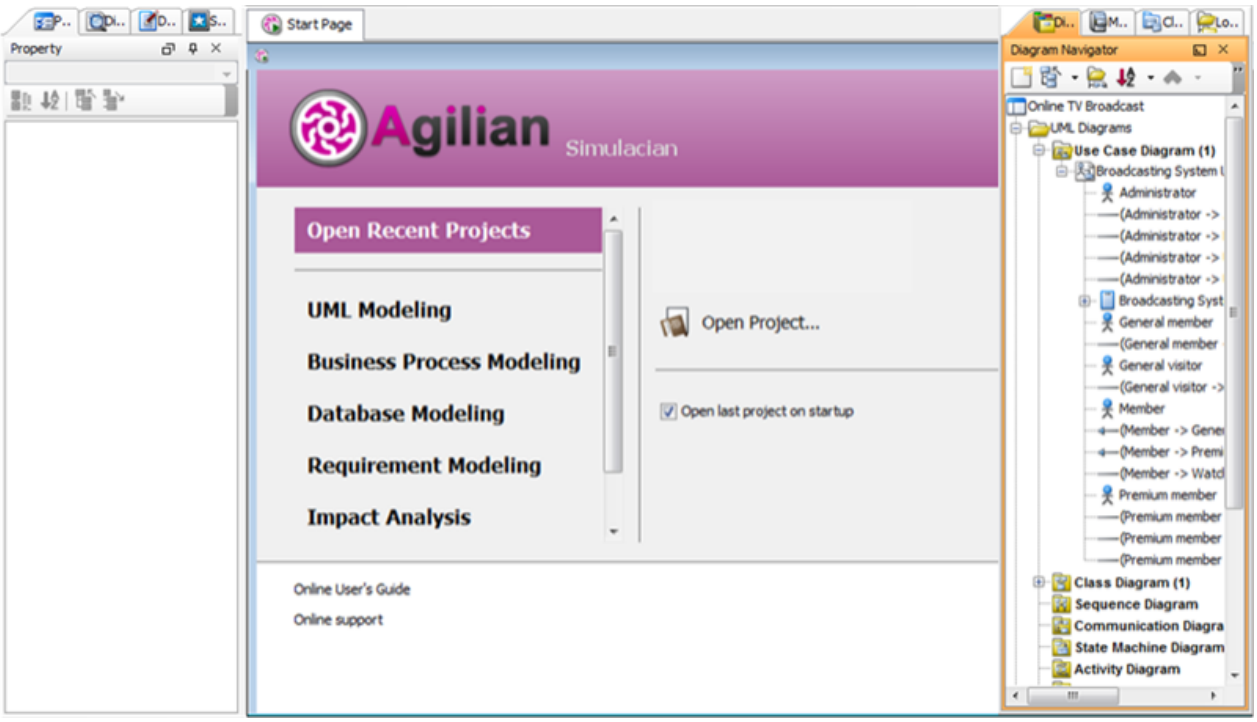
Frame can be dragged out and dock to elsewhere

A gray outline will appear while you are dragging a frame/tab. This outline tells you where the dockable frame/tab will be docked to.

Docking a dockable frame to elsewhere

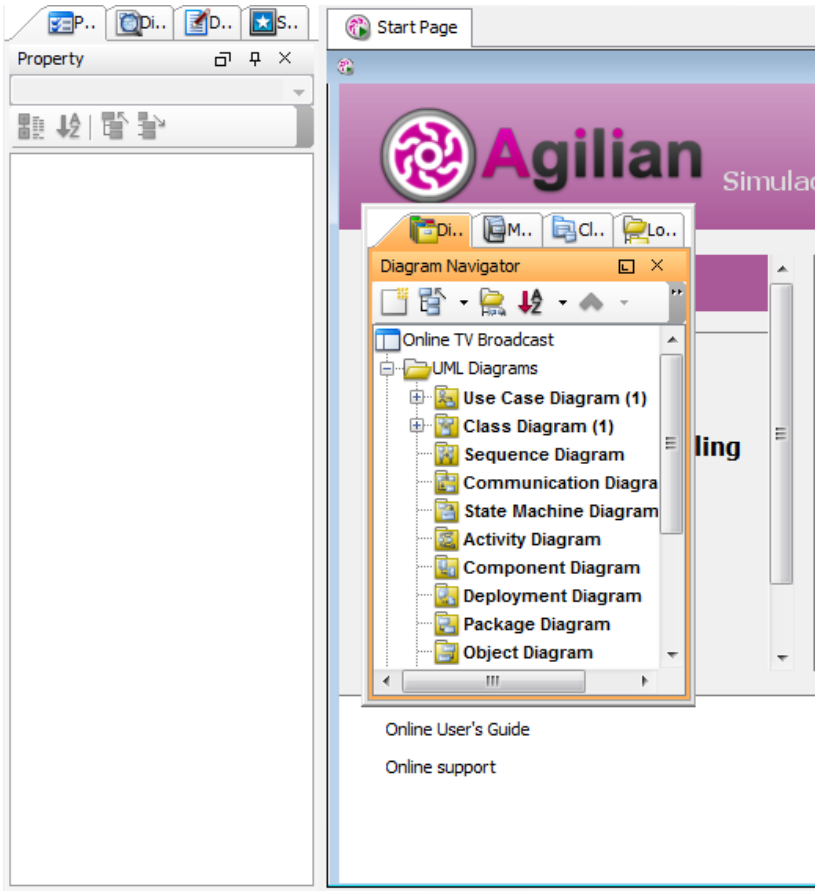
By dragging out a dockable frame/tab and moving the cursor to a specific position, the frame/tab will be repositioned accordingly.

If you drag the dockable frame/tab and release it over another container, the gray outline will change its shape to fit the dockable area of the container. If you release the frame/tab, it will be docked into the underlying container and also removed from its original container.



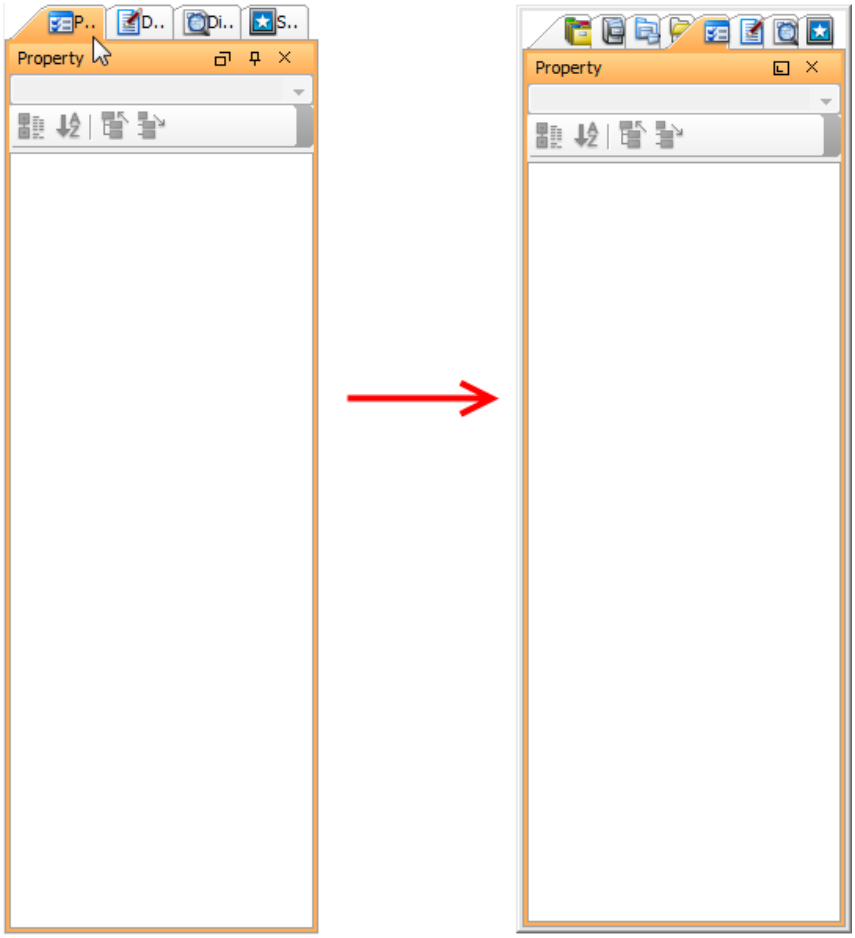
A frame docked to the right of application screen

You also can drag a frame/tab out to make it a floating window.



A frame is floating on the application screen

You can also drag a frame/tab and dock it into another tabbed pane. The outline will be changed to a tab shape if you drag over a tab pane.



A frame is docked to another frame

Auto-hiding a dockable frame

A dockable frame can be set to "auto hide", meaning it will automatically disappear when not active. To set a dockable frame to "auto hide", click on the **Toggle auto-hide** button on the top right corner of the frame (the button with a pin as the icon, see figure below)

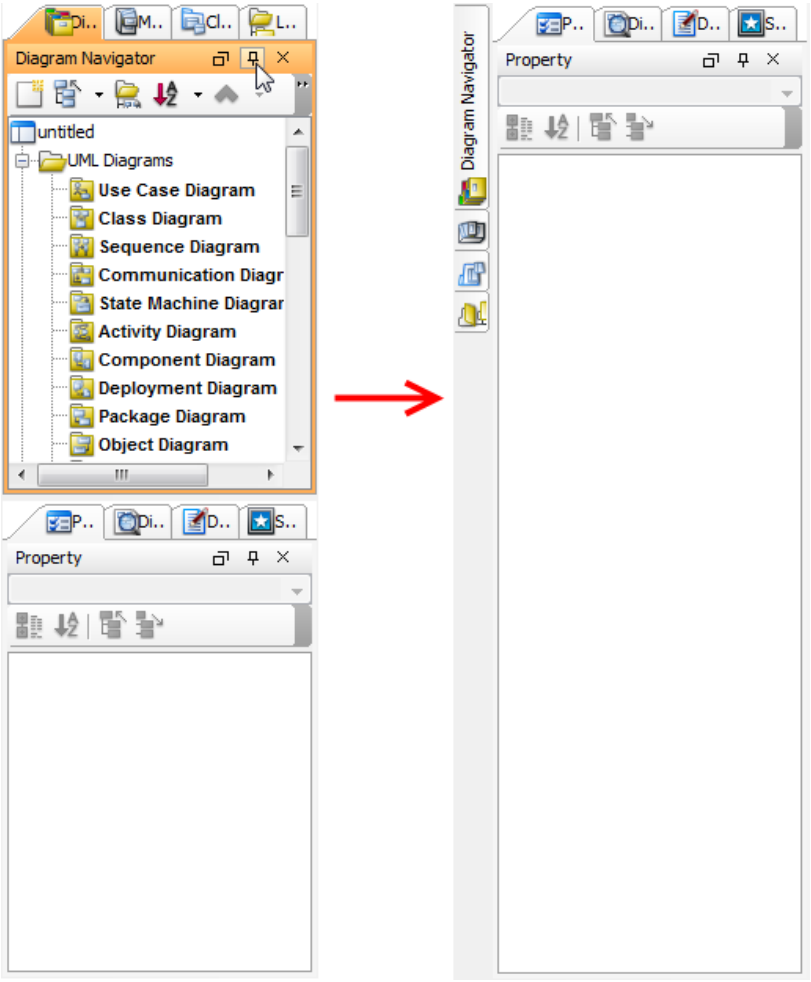


Diagram Navigator is hidden automatically

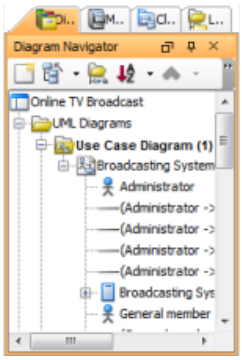
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagram navigator

Diagram navigator is the location where diagrams are listed. You can create your own diagrams easily as they are listed by categories, such as UML Diagrams, Requirements Capturing, Database Modeling, Business Process Modeling and Others. With diagram navigator, you can create, open and delete diagrams.



The diagram navigator

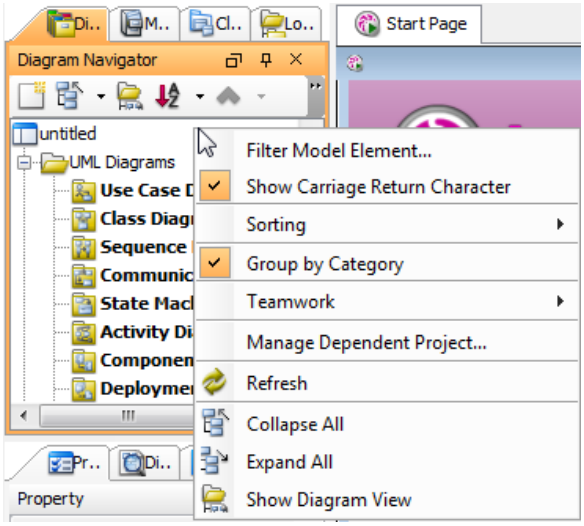
The toolbar

Name	Icon	Description
New diagram		Create a new diagram through the **New Diagram** dialog box.
Collapse		Collapse the selected diagram.
Expand		Expand the selected diagram.
Show Diagram View		Overview the diagram elements in the selected diagram.
Sort Diagram Element By Name		Sort diagram elements by their name in alphabetical order.
Sort Diagram Element By Type		Sort diagram elements by their type, disregard their name.
Move Selected Diagram		Move the selected diagram down or up. This option only works within the same diagram type when there are multiple diagrams.
Refresh		Update the content of diagram tree.

The description of icons on diagram navigator

Pop-up menu

Pop-up menu of diagram navigator



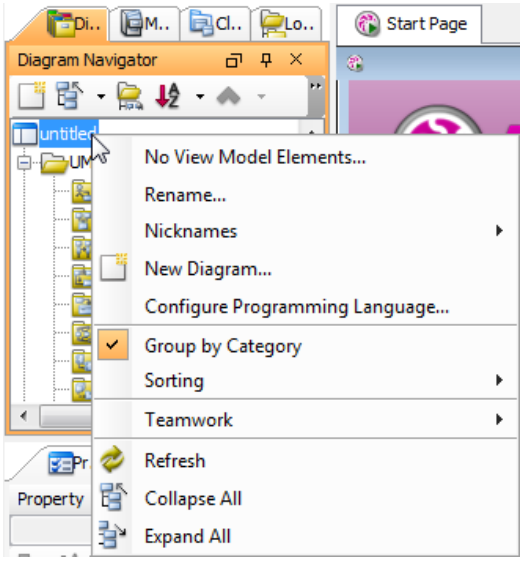
The pop-up menu of diagram navigator

Menu Title	Description

Filter Model Element...	Open the Model Element Filter dialog box to filter the diagram elements to appear in the diagram navigator.
Show Carriage Return Character	Display line breaks of multi-lined diagram name as carriage return character.
Sorting	Select the way to sort diagram elements in diagram navigator.
Group by Category	Categorize diagrams based on their types.
Teamwork	Perform teamwork activities.
Manage Dependent Project...	Add or remove dependent project.
Refresh	Refresh diagram navigator content.
Collapse All	Collapse all tree nodes.
Expand All	Expand all tree nodes.
Show Diagram View	Make the Diagram Navigator to expand all diagram types' nodes to show the diagrams nodes, but do not display any diagram element nodes.

The description of pop-up menu of diagram navigator

Pop-up menu of project node

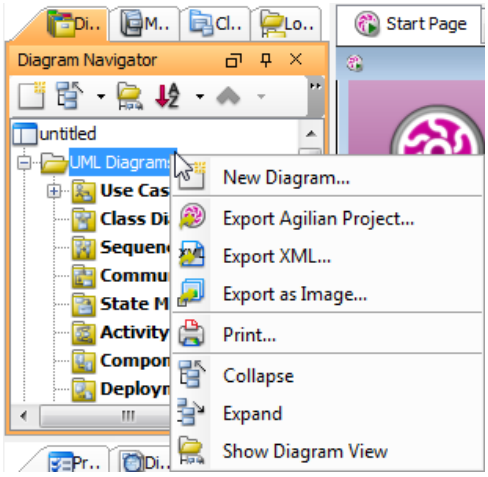


The pop-up menu of project node in diagram navigator

Menu Title	Description
No View Model Elements...	It lists the model element(s) that either has not been visualized (no view) or has no master view.
Rename...	Rename the project.
Nicknames	Configure or switch to another nickname.
New Diagram...	Create a diagram.
Configure Programming Language...	Change to another programming language or configure the type mapping for a language.
Group by Category	Categorize diagrams base on their types.
Sorting	Select the way to sort diagram elements in diagram explorer.
Teamwork	Perform teamwork activities.
Refresh	Refresh diagram navigator.
Collapse All	Collapse the project node.
Expand All	Expand the project node.

The description of pop-up menu of project node

Pop-up menu of diagram category

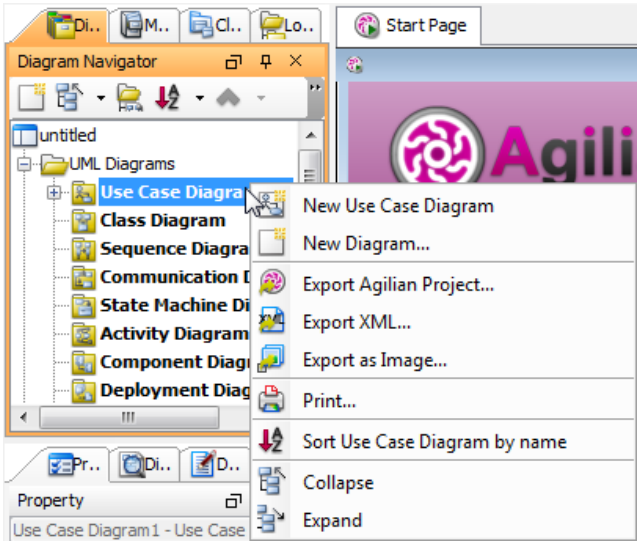


The pop-up menu of diagram category in diagram navigator

Menu Title	Description
New Diagram...	Create a diagram.
Export Agilian Project...	Export all diagrams in the selected category as Agilian project.
Export XML...	Export all diagrams in the selected category as XML.
Export as Image...	Export all diagrams in the selected category as image files.
Print...	Print the diagrams in the selected category.
Collapse	Collapse the selected diagram category node.
Expand	Expand the selected diagram category node.
Show Diagram View	Make the Diagram Navigator to expand all diagram types' nodes to show the diagrams nodes, but do not display any diagram element nodes.

The description of pop-up menu of diagram category

Pop-up menu of diagram type



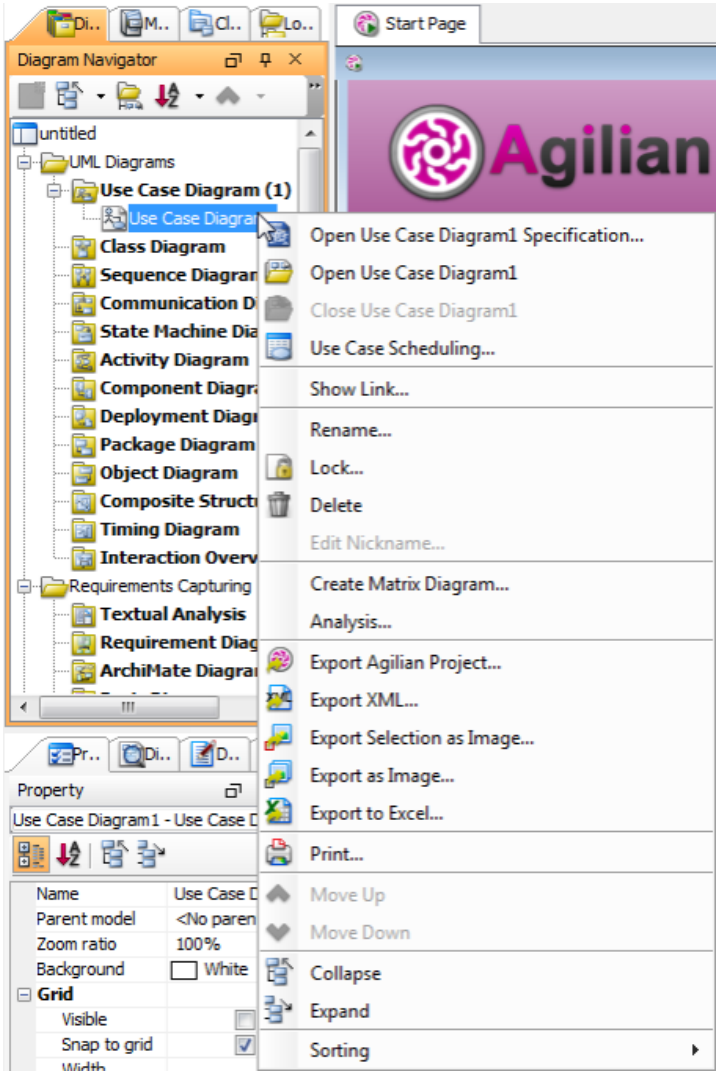
The pop-up menu of diagram type in diagram navigator

Menu Title	Description
New [diagram type]	Create a new diagram in the selected type.
New Diagram...	Create a new diagram with popup dialog box.
Export Agilian Project...	Export all diagrams in the selected type as Agilian project.
Export XML...	Export all diagrams in the selected type as XML.
Export as Image...	Export all diagrams in the selected type as image files.

Print...	Print the diagrams in the selected type.
Sort [diagram type] by name	Sort the diagram nodes of the selected type node in specific way.
Collapse	Collapse the selected diagram type node.
Expand	Expand the selected diagram type node.

The description of pop-up menu of diagram type

Pop-up menu of diagram



The pop-up menu of diagram in diagram navigator

Menu Title	Description
Open [diagram name] Specification...	Open the specification of the selected diagram.
Open [diagram name]	Open the selected diagram if closed or inactive.
Close [diagram name]	Close the selected diagram if opened.
Use Case Scheduling...	Open Use Case Scheduling dialog box.
Rename...	Rename the selected diagram.
Delete	Delete the selected diagram.
Edit Nickname...	A pop-up a dialog box for defining the nickname of model elements will appear in the diagram. This option is available only after you have configured a nickname.
Analysis	An analysis diagram will be formed to analyze the selected diagram with others and the result will be seen. Relationship like diagram transition can also be found.
Export Agilian Project...	Export the selected diagram as Agilian project.
Export XML...	Export the selected diagram as XML.

Export Selected as Image...	Export the selected diagram as image Excel.
Export as Image...	Export diagram(s) as image via the Diagram Exporter.
Export to Excel...	Export the selected diagram as image Excel.
Print...	Print the selected diagram.
Move Up	Move the selected diagram node upwards.
Move Down	Move the selected diagram node downwards.
Collapse	Collapse the selected diagram node.
Expand	Expand the selected diagram node.
Sorting	Change the way of sorting diagram elements.

The description of pop-up menu of diagram

Closing and opening the diagram navigator

Diagram navigator is opened by default. To close it, press the **X** button at the top right corner. On the other hand, it can be opened by selecting **View > Panes > Diagram Navigator** from the main menu.

Creating a diagram

Approach 1 – Direct creation

If you want to create a diagram in a quick way and don't need to supply the documentation of diagram (for time-saving), use this approach. To create a diagram:

1. Right click on the diagram type that you want to create.
2. Select **New [diagram type]** from the pop-up menu. Here, diagram type means the type of diagram you want to create.

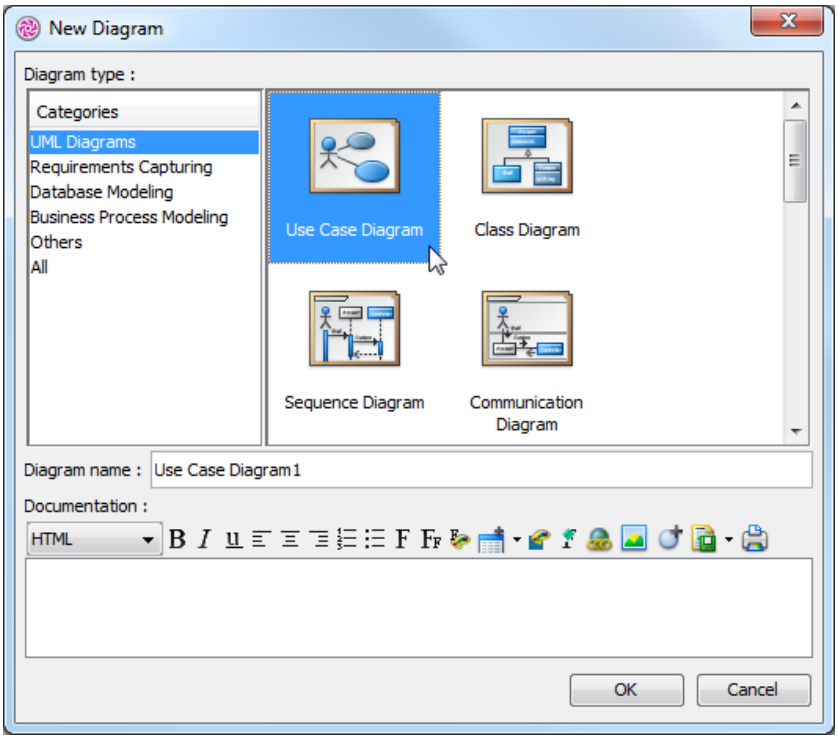
> **NOTE:** You can immediately enter the diagram name at the top left corner of diagram.

Approach 2 – Through the New Diagram dialog box

This approach let you create a diagram and enter the name and documentation simultaneously. To create a diagram:

1. Right click on the diagram type that you want to create.
2. Select **New Diagram** from the pop-up menu.

In the **New Diagram** dialog box, you can enter the name and documentation of diagram, and click **OK** to proceed.



***New Diagram** dialog box*

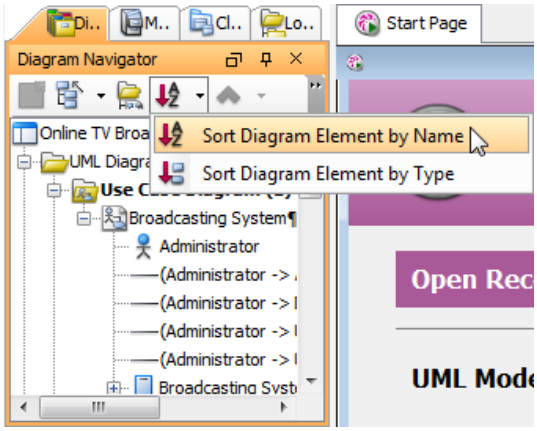
Opening a diagram

Double click the diagram you want to view in the diagram tab.

Sorting diagram elements

In diagram navigator, diagram elements are listed under diagram nodes. You can sort diagram elements by their names, or by their types (e.g. use case, package, etc.)

To sort by name, click **Sort Diagram Element by Name** button. The elements will be listed by name, in alphabetical order.



*Click **Sort Diagram Element by Name***

To sort by type, click **Sort Diagram Element by Type** button. As a result, the elements will be listed by type.

> **NOTE:** The sort function applies to the entire diagram tree instead of the selected node.

Reordering diagrams

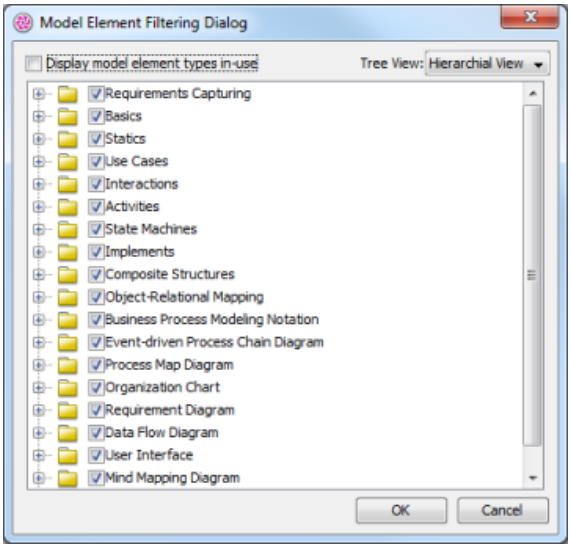
Select the diagram(s) you want to reorder. If there are multiple diagrams you want to select, just click one while pressing **Ctrl** button and make the other selection.

Click **Move Selected Diagram Up** button to move the selection upwards, or **Move Selected Diagram Down** button to move the selection downwards.

Filtering model elements

You can choose which model elements you want to be displayed or not on the diagram tree.

1. Right click on the diagram navigator's background and check **Filter Model Element...**. **Model Element Filtering Dialog** will then be pop-up.
2. Select the types of model element(s) you want to be displayed by checking to the type. Otherwise, uncheck the one you don't want to be displayed.



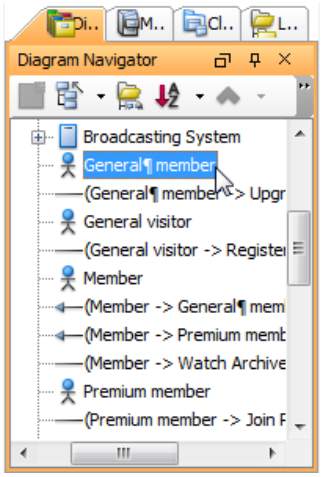
***Model Element Filtering** dialog box*

> **NOTE:** You can check **Display model element types in-use** to list only types of model elements used in project. The text box Filter enables you to filter model element type base on the type name (e.g. enter class to list only class)

Showing/hiding carriage return character

If it is the case that the name of the diagram and diagram elements is in multi-line, the character ¶ will be revealed.

When **Show Carriage Return Character** is selected, line break will be shown.



Show carriage return character

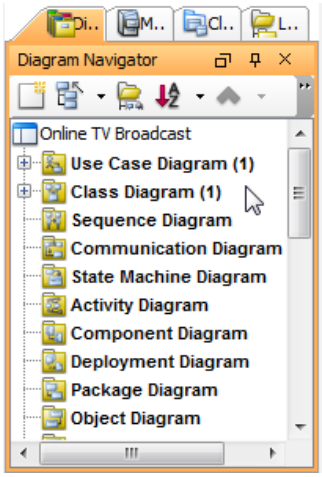
When deselected, the character ¶ is hidden.

If you want to hide it, right click on the diagram navigator's background and uncheck **Show Carriage Return Character** from the pop-up menu. As a result, the character will automatically be unshown.

Grouping or ungrouping by category

As it is said before, default groups are automatically arranged by category.
If you want to ungroup diagram:

1. Right click on the diagram navigator's background.
2. Uncheck **Group by Category** if there is a tick in the box, and then all diagram types will be ungrouped.

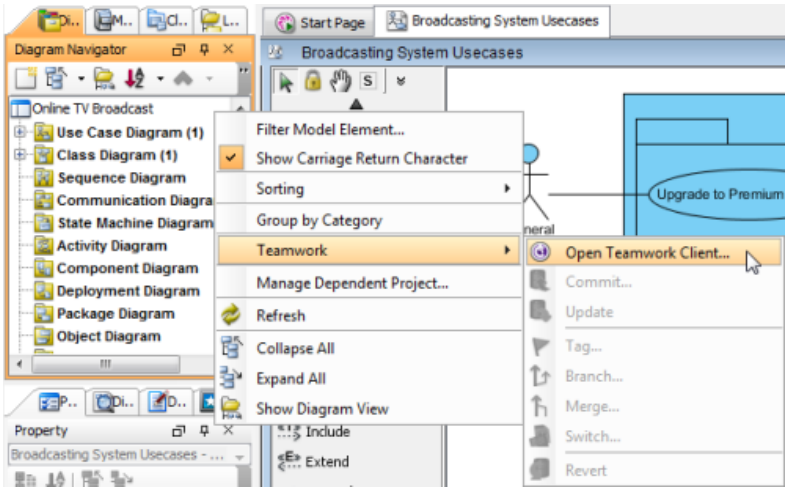


Ungrouped diagrams

Connecting to server for team collaboration

Agilian's team collaboration support enables your team members work together on projects. To connect to server and perform related activities:

1. Right click on the diagram navigator's background.
2. Select **Teamwork** and the action you want to perform (e.g. commit and update, etc) from the pop-up menu.



Displaying only diagram nodes but not diagram elements

If you only want the diagram nodes to be shown but not the diagram elements under them, just click on **Show Diagram View** button on its toolbar.

> **NOTE:** This option applies to all diagram nodes, but not just the selected one.

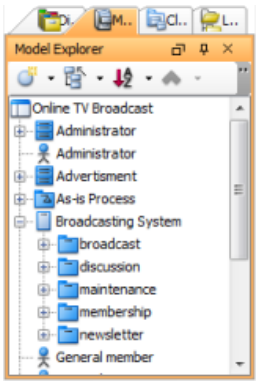
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Model Explorer

It would be much more efficient to make use of Model Explorer for your middle to large scale project which has considerable numbers of diagrams and model elements, rather than using Diagram Navigator.



*The **Model Explorer***

The toolbar

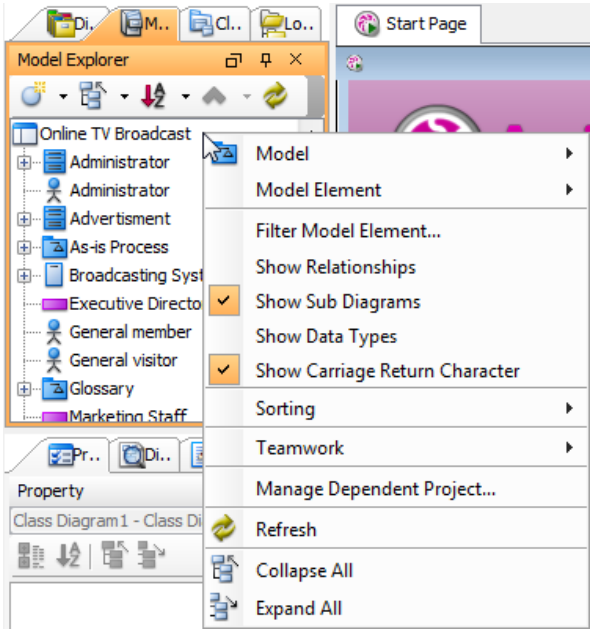
Name	Icon	Description
New Model Element		Create a new model element.
New Package		Create a new package.
Collapse		Collapse the selected model element.
Expand		Expand the selected model element.
No Sorting		Arrange model elements without grouping.
Sort By Name		Sort model elements by their names in alphabetical order.
Sort By Type		Sort model elements by their types.
Move Selected Model Element		Move the selected model elements down or up. This option only works within the same model element type when there are multiple model elements.
Refresh		Update the content of model explorer.

*The description of icons on **Model Explorer***

Pop-up menu

Pop-up menu of model explorer

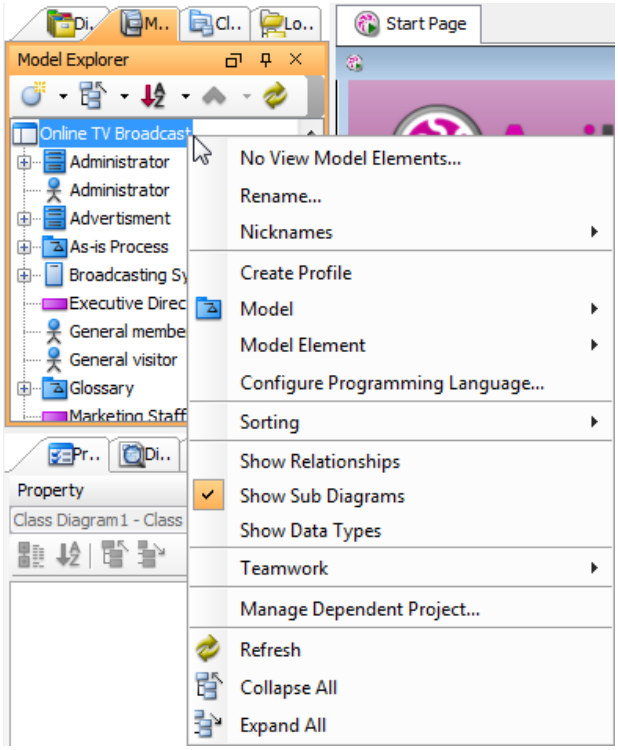
The Model Explorer lists all the model elements in the project.



The pop-up menu of Model Explorer

Menu Title	Description
Model	Create a Model.
New Model Element...	Create a new Model Element in Model Explorer without the need of creating through diagramming.
Filter Model Element...	Open the Model Element Filter dialog box to filter the diagram elements to appear in the Diagram Navigator.
Show Relationships	Show also Relationship model elements in Model Explorer (default hidden).
Show Sub Diagrams	Show Sub Diagrams in Model Explorer so that user can browse and open Sub Diagrams in Model Explorer.
Show Data Types	Show Data Types in Model Explorer (default hidden).
Show Carriage Return Character	Display line breaks of multi-lined diagram name as carriage return character.
Sorting	Select the way to sort model elements in Model Explorer.
Teamwork	Perform teamwork activities.
Manage Dependent Project...	Add or remove dependent project.
Refresh	Refresh Model Explorer content.
Collapse All	Collapse all tree nodes.
Expand All	Expand all tree nodes.

Pop-up menu of project



The pop-up menu of project node in Model Explorer

Menu Title	Description
Rename...	Rename the project.
Nicknames	Configure or switch to another nickname.
Model	Create a model.
Model Element	Create a new model element in Model Explorer without the need of creating through diagramming.
Configure Programming Language	Change to another programming language or configure the type mapping for a language.
Sorting	Select the way to sort model elements in Model Explorer.
Show Relationships	Show also Relationship model elements in Model Explorer (default hidden).
Show Sub Diagrams	Show Sub Diagrams in Model Explorer so that user can browse and open Sub Diagrams in Model Explorer.
Show Data Types	Show Data Types in Model Explorer (default hidden).
Teamwork	Perform teamwork activities.
Manage Dependent Project...	Add or remove dependent project.
Refresh	Refresh Model Explorer.
Collapse All	Collapse the project node.
Expand All	Expand the project node.

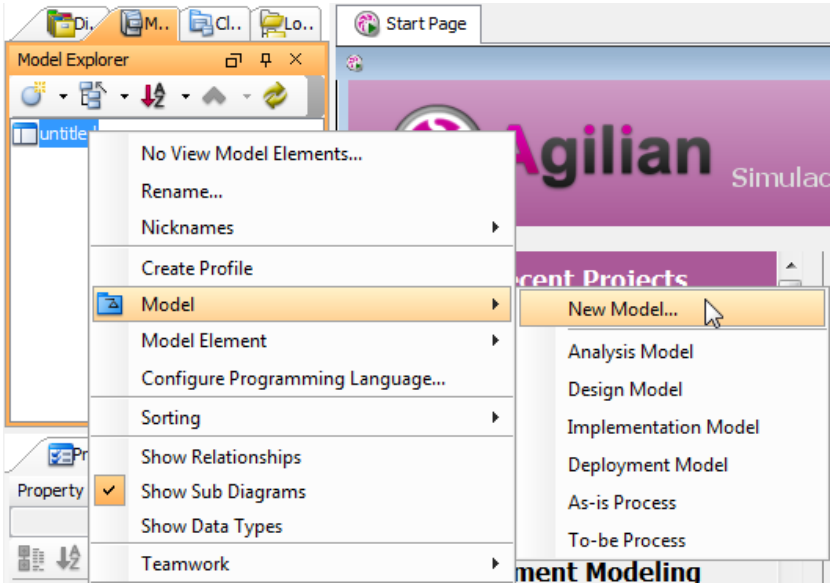
Closing and opening the model explorer

Model Explorer is opened by default. To close it, press the **X** button at the top right corner. On the other hand, it can be opened by selecting **View > Panes > Model Explorer** from the main menu.

Creating a model

A model is a package like UML element that can store model elements and diagrams. Users are recommended to structure project by using model in order to maintain a clear structure for accessing project data and improve the application performance.

Right click on the root node in **Model Explorer** and select **Model** from the popup menu. You can either create a custom model by selecting **New Model...**, or create a pre-defined model by selecting it in the list.

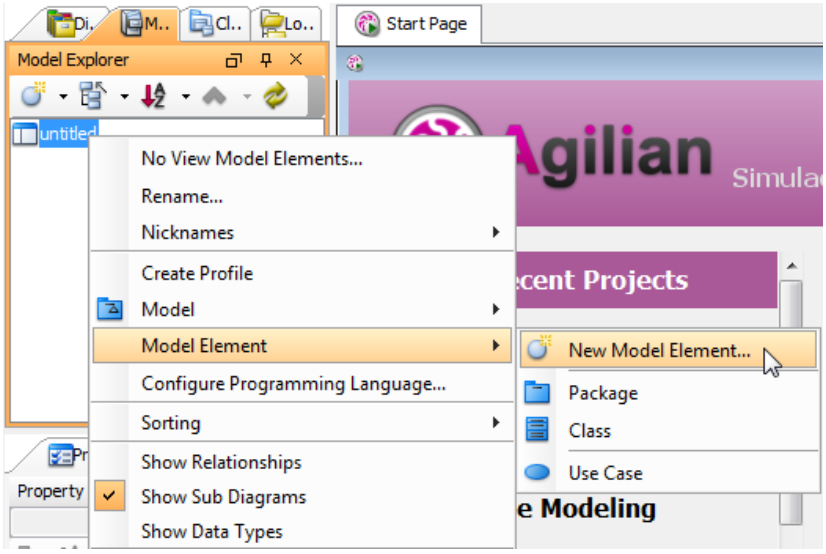


*Select **New Model...** in pop-up menu*

Creating a model element

A model element is created when you create a shape on a diagram. If you want to create a model element without visualizing it, you can create it through the **Model Explorer**. To create a model element:

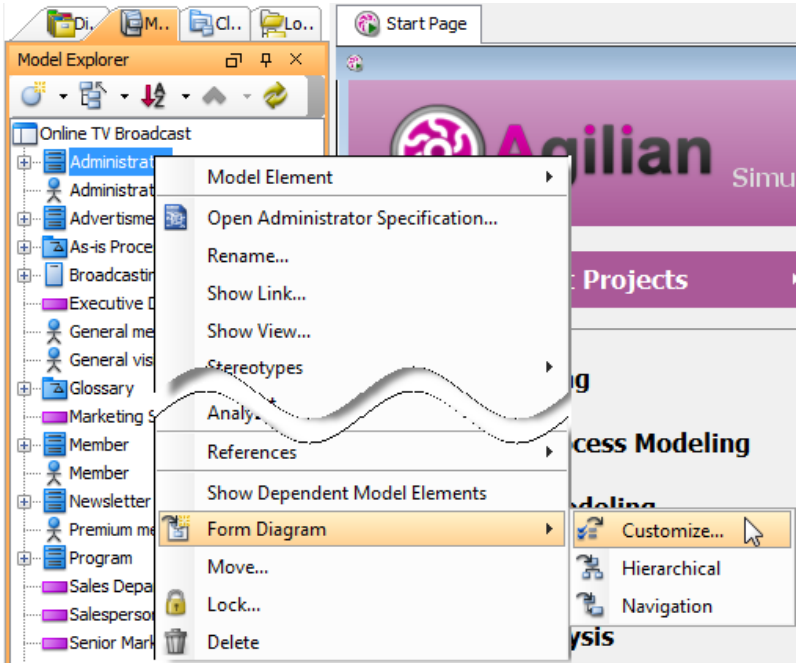
1. Right click on the root node.
2. Select **Model Element> New Model Element...**, or select a pre-defined model element from the pop-up menu.



*Select **New Model Element...** in pop-up menu*

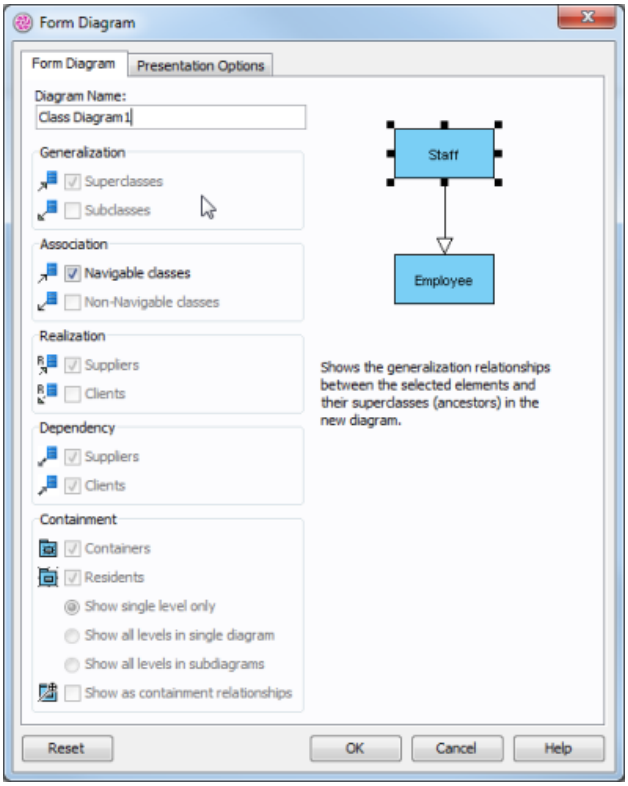
Forming class diagram from class

1. To form a class diagram, right click on a target class and select **Form Diagram > Customize...** from the pop-up menu.



*Open **Form Diagram** dialog box*

2. As a result, the **Form Diagram** dialog box prompts out. Enter diagram name and select a relationship. After you select a relationship for classes, the corresponding relationship between the selected classes and their superclasses (ancestors) will be shown in the new diagram.

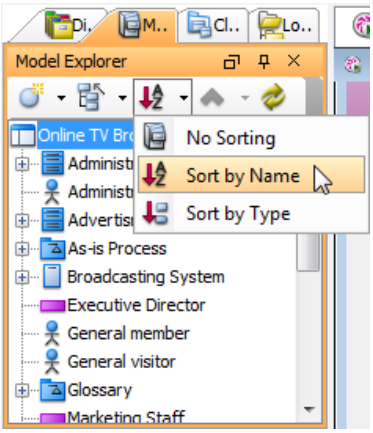


***Form Diagram** dialog box*

Sorting model elements

In model explorer, model elements are listed under root nodes. You can sort model elements by their names, by their types (e.g. use case, package, etc.) or with no sorting.

To sort by name, click **Sort by Name** button. The elements will be listed by name, in alphabetical order.



Sort model elements by name

To sort by type, click **Sort by Type** button. As a result, the elements will be listed by type. To arrange model elements without sorting, click **No sorting**.

Reordering model elements

Select the model element(s) you want to reorder. If there are multiple model elements you want to select, just click one while pressing **Ctrl** button and make the other selections.

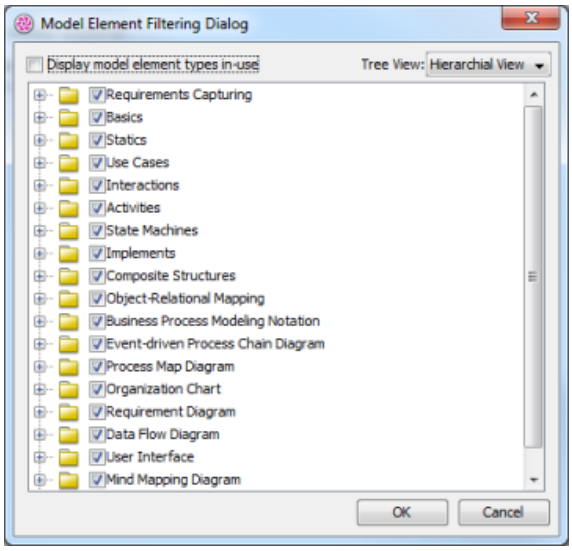
Click **Move Selected Model Element Up** button to move the selection upwards, or **Move Selected Model Element Down** button to move the selection downwards.

> **NOTE:** This option only works within the same model element type when there are multiple model elements.

Filtering model elements

You can choose which model elements you want to be displayed or not on the model explorer.

Right click on the model explorer's background and select **Filter Model Element...**. In **Model Element Filtering Dialog**, select the types of model element(s) you want to be displayed by checking the type. Otherwise, uncheck the one you don't want to be displayed.



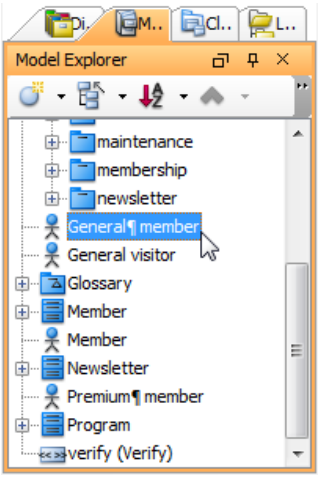
Model Element Filtering Dialog

> **NOTE:** Instead of displaying all types of model elements, checking **Display model element types in-use** to display the types of model elements used in the project.

Showing/hiding carriage return character

If it is the case that the name of model elements is in multi-line, the character ¶ will be revealed.

When **Show Carriage Return Character** is selected, line break will be shown.



***Show Carriage Return Character** is selected*

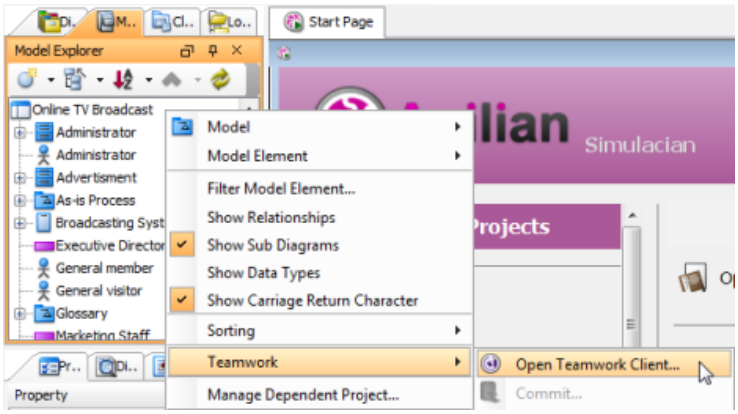
When off, the character ¶ is hidden.

If you want to hide it, right click on the diagram navigator's background and uncheck **Show Carriage Return Character** from the pop-up menu. As a result, the character will automatically be unshown.

Connecting to server for team collaboration

Agilian's team collaboration support enables your team members work together on projects. To connect to server and perform related activities:

1. Right click on the model explorer's background.
2. Select **Teamwork > Open Teamwork Client...** from the pop-up menu.



Perform teamwork

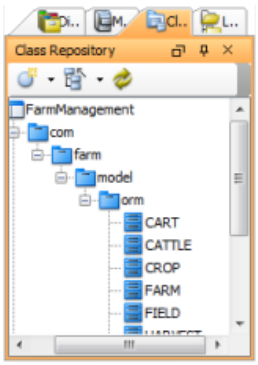
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Class repository

Class repository is a pane where classes and container that contain classes, such as packages or subsystems, are listed. To access classes, you can also form class diagram by dragging classes from class repository on class diagram.



*The **Class Repository***

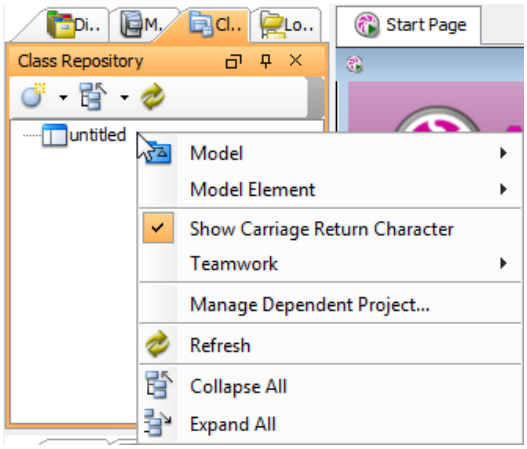
The toolbar

Name	Icon	Description
New Model Element		Create a new model element.
Collapse		Collapse the selected model element.
Expand		Expand the selected model element.
Refresh		Update the content of Class Repository.

The description of icons on Class Repository

Pop-up menu

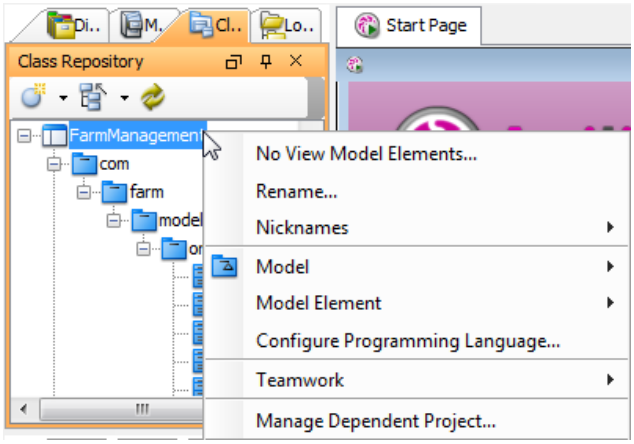
Pop-up menu of class repository



*The pop-up menu of **Class Repository***

Menu Title	Description
Model	Create a model.
Model Element...	Create a new model element in Class Repository instead of creating through diagramming.
Show Carriage Return Character	Display line breaks of multi-lined diagram name as carriage return character.
Teamwork	Perform teamwork activities.
Manage Dependent Project...	Add or remove dependent project.
Refresh	Refresh Class Repository content.
Collapse All	Collapse all tree nodes.
Expand All	Expand all tree nodes.

Pop-up menu of project node



*The pop-up menu of project node in **Class Repository***

Menu Title	Description
No View Model Elements...	It lists the model element(s) that either has not been visualized (no view) or has no master view.
Rename...	Rename the project.
Nicknames	Configure or switch to another nickname.
Model	Create a Model.
Model Element	Create a new Model Element in Class Repository without the need of creating through diagramming.
Configure Programming Language...	Change to another programming language or configure the type mapping for a language.
Teamwork	Perform teamwork activities.
Manage Dependent Project...	Add or remove dependent project.
Refresh	Refresh Class Repository.
Collapse All	Collapse the project node.
Expand All	Expand the project node.

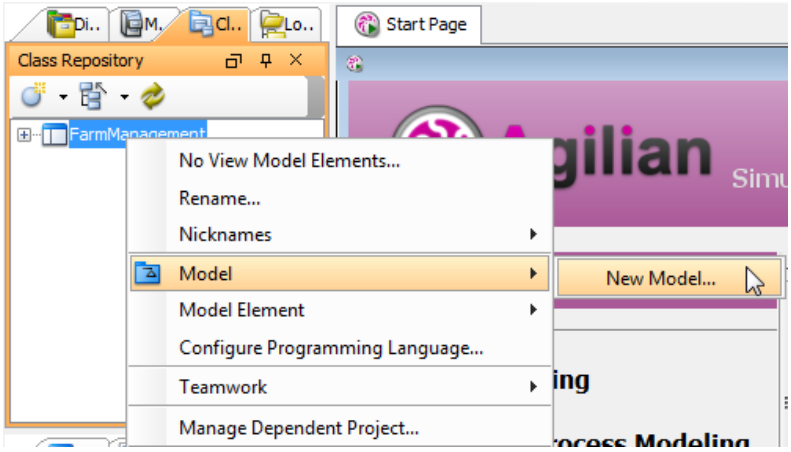
Closing and opening the Class Repository

Class Repository is opened by default. To close it, press the **X** button at the top right corner. On the other hand, it can be opened by selecting **View > Panes > Class Repository** from the main menu.

Creating a model

A model is a package like UML element that can store model elements and diagrams. Users are recommended to structure project by using model in order to maintain a clear structure for accessing project data and improve the application performance.

Right click on the root node in Class Repository and select **Model > New Model...** from the pop-up menu to create a new model.

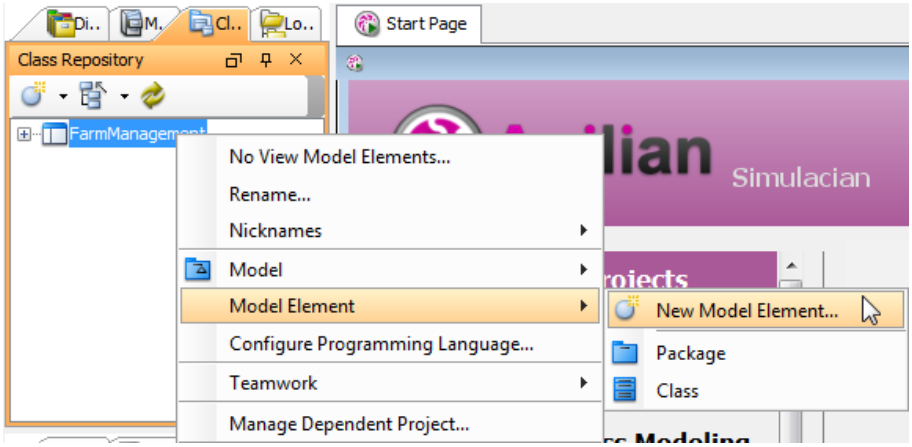


*Select **New Model** in pop-up menu*

Creating a model element

A model element is created when you create a shape on a diagram. If you want to create a model element without visualizing it, you can create it through the Class Repository. To create a model element:

1. Right click on the root node.
2. Select **Model Element> New Model Element...**, or select a pre-defined model element from the pop-up menu.

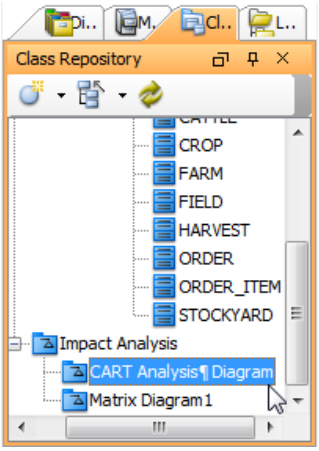


*Select **New Model Element** in pop-up menu*

Showing/hiding carriage return character

If it is the case that the name of model elements is in multi-line, the character ¶ will be revealed.

When **Show Carriage Return Character** is selected, line break will be shown.



***Show Carriage Return Character** is revealed*

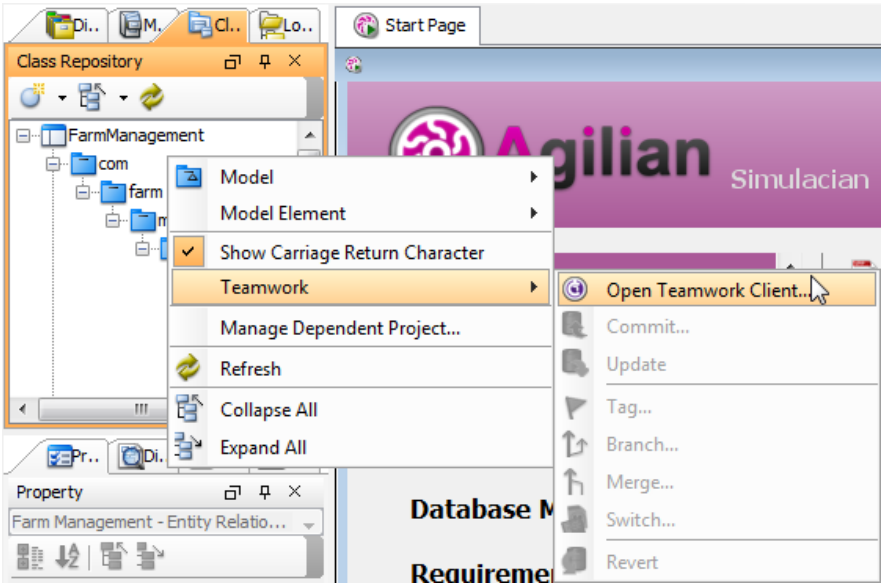
When off, the character ¶ is hidden.

If you want to hide it, right click on the diagram navigator's background and uncheck **Show Carriage Return Character** from the pop-up menu. As a result, the character will automatically be unshown.

Connecting to server for team collaboration

Agilian's team collaboration support enables your team members to work together on projects. To connect to server and perform related activities:

1. Right click on the Class Repository's background.
2. Select **Teamwork> Open Teamwork Client...** from the pop-up menu.



Perform teamwork

Related Resources

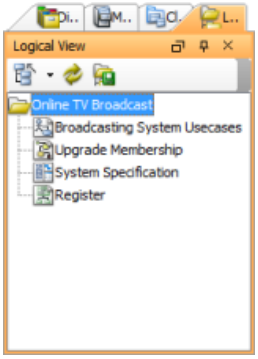
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Logical view

The logical view provides a hierarchical view of a project's structure. With the logical view, users can create and customize the diagrams in their project with meaningful categorization by adding domain specific view(s).

In addition, users can customize a default logical view for their preference, rather than re-creating a new logical view for every new project. The logical view can be exported to xml files which can be used in other projects or distributed among the development team. Different views, thereby, can be merged automatically through the teamwork server .



*The **Logical View***

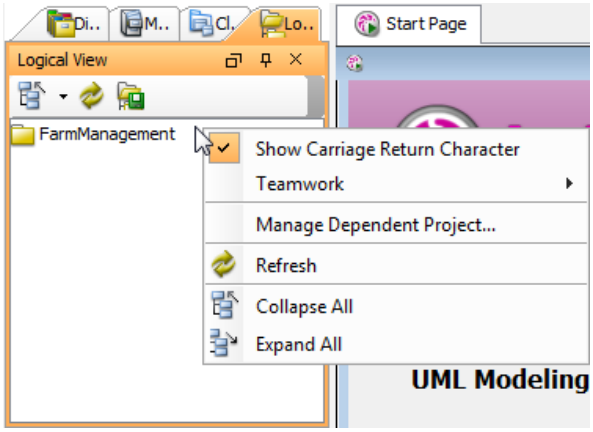
The toolbar

Name	Icon	Description
Collapse		Collapse the selected diagram.
Expand		Expand the selected diagram.
Refresh		Update the content of logical view.
Set Logical View Structure as Default		Set default structure for logical view in all projects.

The description of icons on Logical view

Pop-up menu

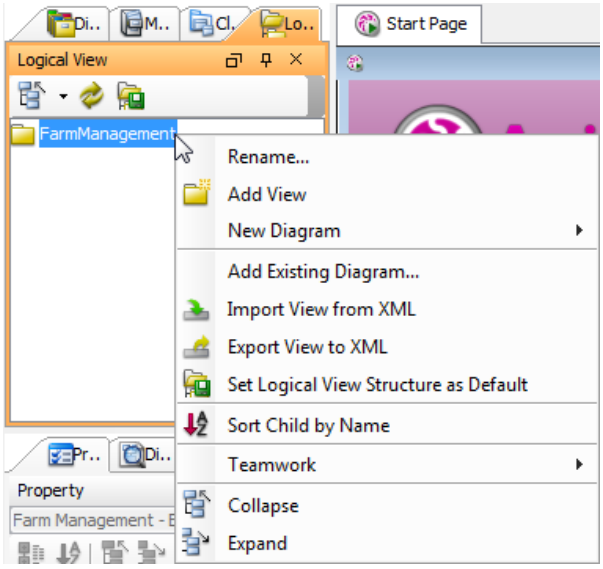
Pop-up menu of logical view



The pop-up menu of Logical View

Menu Title	Description
Show Carriage Return Character	Display line breaks of multi-lined diagram name as carriage return character.
Teamwork	Perform teamwork activities.
Manage Dependent Project...	Add or remove dependent project.
Refresh	Refresh Logical View content.
Collapse All	Collapse all tree nodes.
Expand All	Expand all tree nodes.

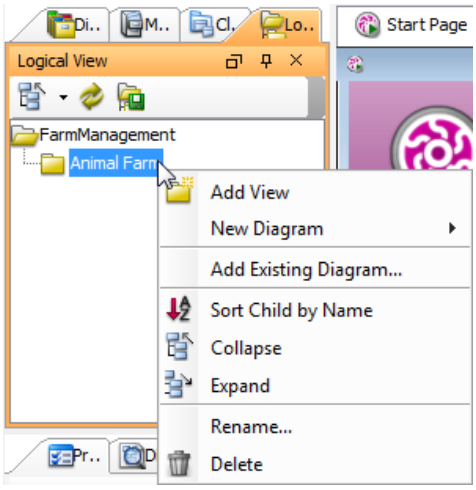
Pop-up menu of project



The pop-up menu of project node in Logical View

Menu Title	Description
Rename...	Rename the project.
Add View	Add a view under project.
New Diagram	Create a diagram under root view.
Add Existing Diagram...	Add an existing diagram under root view.
Import View from XML	Import logical view configuration file.
Export View to XML	Export logical view as configuration file.
Set Logical View Structure as Default	Set the current view structure as default so that another project that will be created under the same workspace will share the same structure.
Sort Child by Name	Sort the views by name.
Teamwork	Perform teamwork activities.
Collapse All	Collapse the project node.
Expand All	Expand the project node.

Pop-up menu of view



The pop-up menu of view in Logical View

Menu Title	Description
Add View	Add a child view under the selected view.

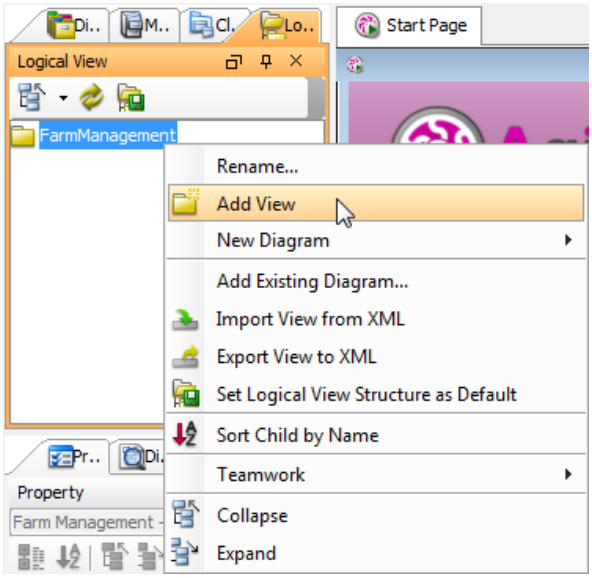
New Diagram	Create a diagram under the selected view.
Add Existing Diagram...	Add an existing diagram under the selected view.
Sort Child by Name	Sort the views/diagrams by name.
Collapse	Collapse the selected view node.
Expand	Expand the selected view node.
Rename...	Rename the selected view.
Delete	Delete the selected view.

Closing and opening the Logical view

Logical view is opened by default. To close it, press the **X** button at the top right corner. On the other hand, it can be opened by selecting **View > Panes > Logical View** from the main menu.

Creating a new view node

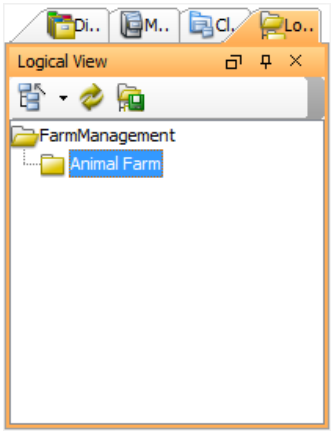
Right click a root node on **Logical View** and select **Add View** from the pop-up menu.



*Click **Add View** from the pop-up menu*

You can enter the name for the new view node in the pop-up **Input** dialog box and then click **OK** button to confirm editing and close the dialog box.

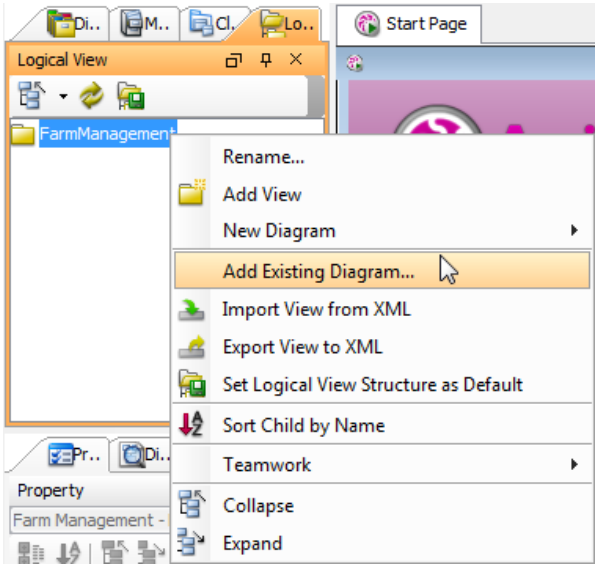
A new view node is, therefore, created under the chosen node.



Created new view node

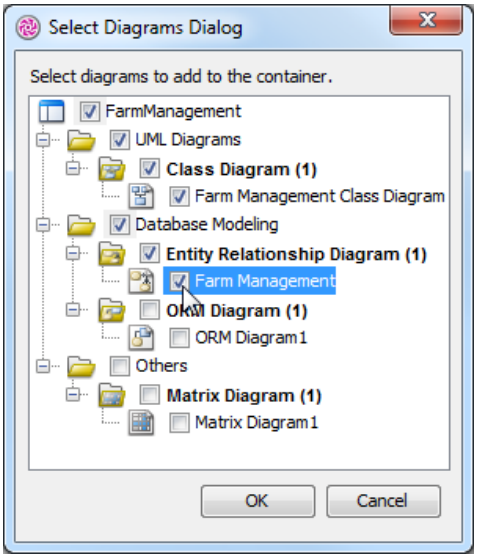
Adding diagram to view

After you create a few diagrams, right-click on a view node and select **Add Existing Diagram...** from the pop-up menu.



*Select **Add Existing Diagram...** from the pop-up menu*

In **Select Diagrams Dialog**, check the diagrams you would like to insert in the view node.

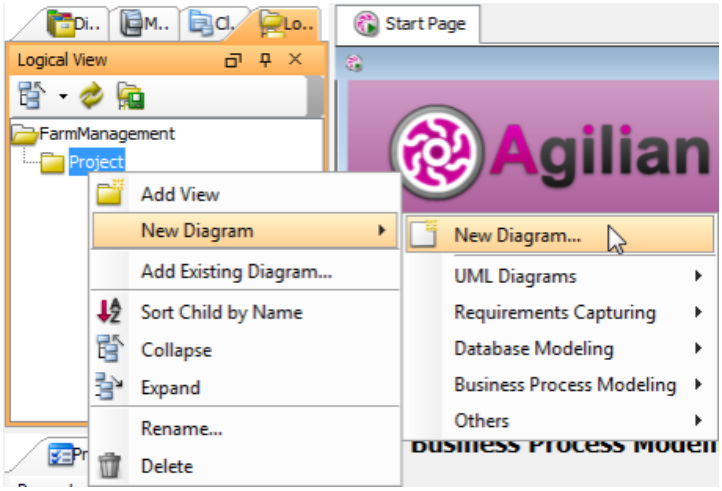


*Check diagrams in **Select Diagrams Dialog***

Click **OK** button to confirm the selection.

Creating a new diagram

Right click the newly created view node, select **New Diagram** from the pop-up menu and select **New Diagram...** or a pre-defined diagram.



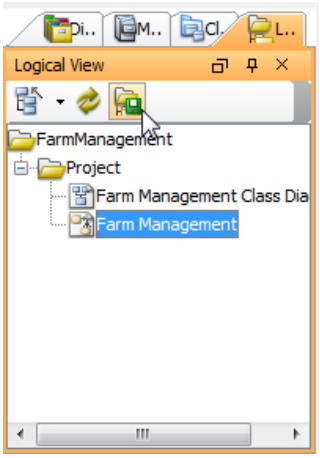
*Select **New Diagram...** from the pop-up menu*

A new diagram is, therefore, created under the view node.

Opening a diagram
Double click on the diagram you want to view in the logical view.

Setting Default View Structure
Agilian provides a feature where you can set the current logical view structure as default, therefore, you may save your time and do not have to re-create the structure every time you create a new project.

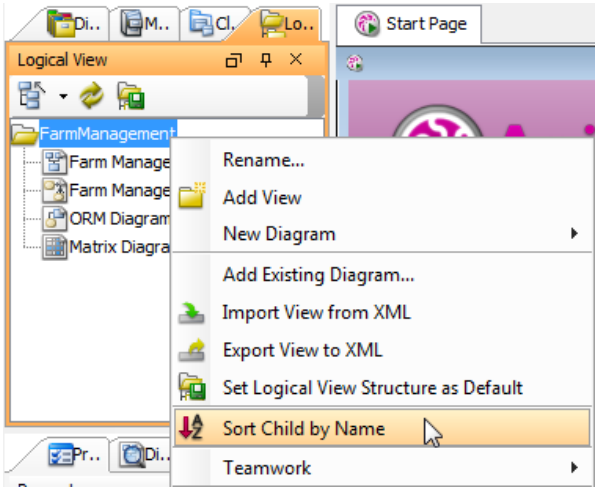
Either click **Set Logical View Structure as Default** button on the top of logical view or right click the project node to select **Set Logical View Structure as Default** from the pop-up menu.



*Click **Set Logical View Structure as Default** button*

In the pop-up **Message** dialog box, click **OK** button. The logical view structure in the new project will then follow the default style you have just customized.

Sorting diagram by name
In logical view, diagrams are listed under diagram nodes by default. You can sort child diagrams by their names as well.
To sort by name, right click on view node and select **Sort Child by Name** from the pop-up menu. The child diagrams will be listed by name, in alphabetical order.

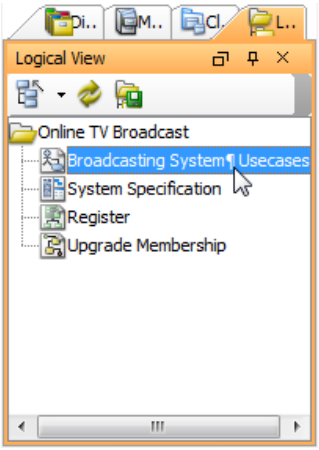


*Click **Sort Child by Name***

NOTE: The sort function applies to the entire logical view instead of the selected node.

Showing/hiding carriage return character
If it is the case that the name of the diagram is in multi-line, the character ¶ will be revealed.

When **Show Carriage Return Character** is selected, line break will be shown.



Show carriage return character

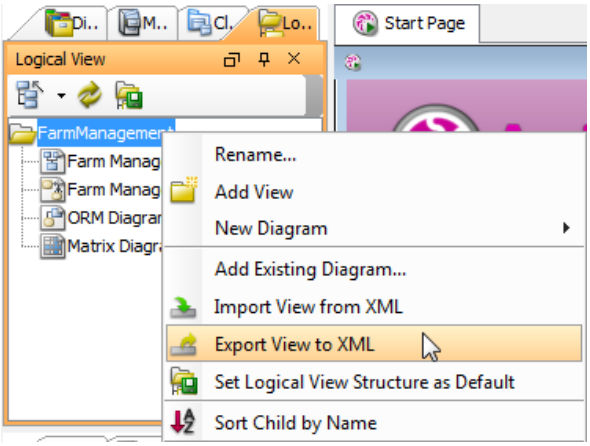
When off, the character ¶ is hidden.

If you want to hide it, right click on the diagram navigator's background and uncheck **Show Carriage Return Character** from the pop-up menu. As a result, the character will automatically be unshown.

Exporting View Structure to XML

Agilian allows you to export the current Logical View Structure as an XML file and to re-use it again on other projects.

Right click the root node and select **Export View to XML** from the pop-up menu.



*Click **Export View to XML** from the pop-up menu*

Find a location for exporting the project and enter its file name in **Save** dialog box. At last, click the **Save**.

Importing View Structure from XML

Agilian also allows you to import the existing xml file in your new project.

Right click on the root node and select **Import View to XML** from the pop-up menu.

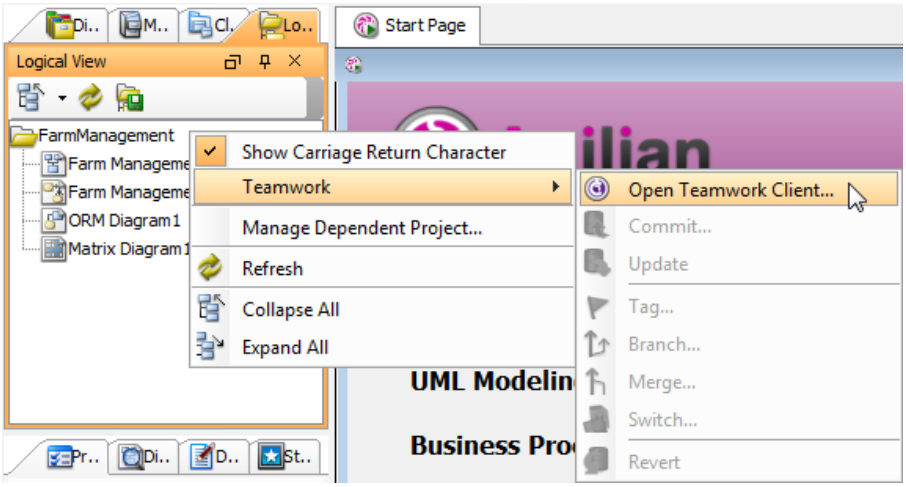
In **Open** dialog box, browse and select the xml file to be imported. You can choose one out of two following choices provided for importing a logical view structure:

1. **Append to existing structure**: the imported structure will be added to the current structure without deleting the old one.
2. **Replace existing structure**: the new imported structure will replace the current structure. Therefore, the current structure will be removed.

Connecting to server for team collaboration

Agilian's team collaboration support enables your team members work together on projects. To connect to server and perform related activities:

1. Right click on the logical view's background.
2. Select **Teamwork** and the action you want to perform from the pop-up menu.



Perform teamwork

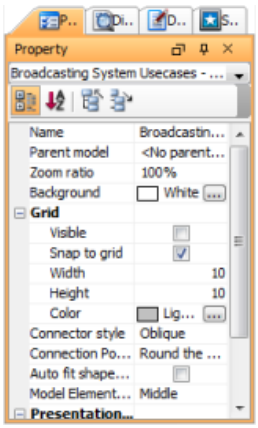
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Property pane

Property pane is the location where the properties of a diagram, model element or shape are listed. With Property pane, you can view and edit properties directly.



*The **Property pane***

The toolbar

Name	Icon	Description
Categorized View		Sort all properties by their category.
Alphabetical View		Sort all properties in ascending order base on their names.
Collapse		Collapse all properties of each subitem.
Expand		Expand all properties of each subitem.

The description of icons on Property pane

Closing and opening the Property pane

Property pane is opened by default. To close it, press the **X** button at the top right corner. On the other hand, it can be opened by selecting **View > Panes > Property** from the main menu.

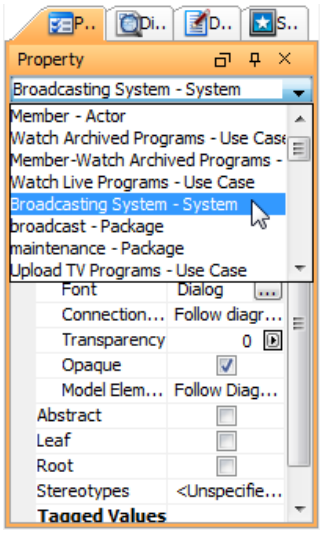
Viewing the properties

Click a shape that you would like to view its properties directly when the shape is shown on diagram pane. There are not only the properties of shapes, but also the properties of diagrams in **Diagram Navigator** and the properties of model elements in **Model Explorer** that can be viewed in property pane.

Click a diagram on **Diagram Navigator** to view the properties of diagram while click a model element in **Model Explorer** to view the properties of model element.

Viewing the properties in shortcut

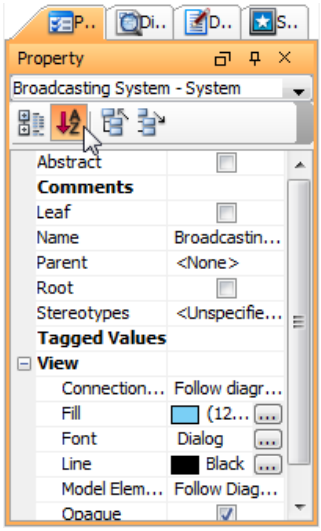
When a diagram, model element or shape is selected, its properties are shown on property pane. Meanwhile, you are allowed to view other properties, which are in the same diagram, in shortcut by selecting the diagram, model element and shape from dropdown menu at the top of the property pane.



*Select **Broadcasting System - System** in drop-down menu*

Sorting the properties

In Property pane, the properties of a diagram, model element and shape are listed. You can sort the properties by their category, or in ascending order. To sort by the name of properties, click **Alphabetical View** button. All properties will be listed in ascending order base on their names.



*Click **Alphabetical Vie w***

To sort by category, click **Categorized View** button. As a result, all properties will be listed by category.

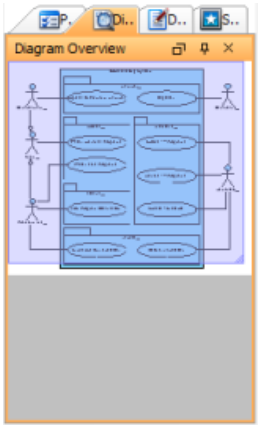
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagram overview

Diagram overview is a pane where users can view and zoom in an active diagram directly and shortly.



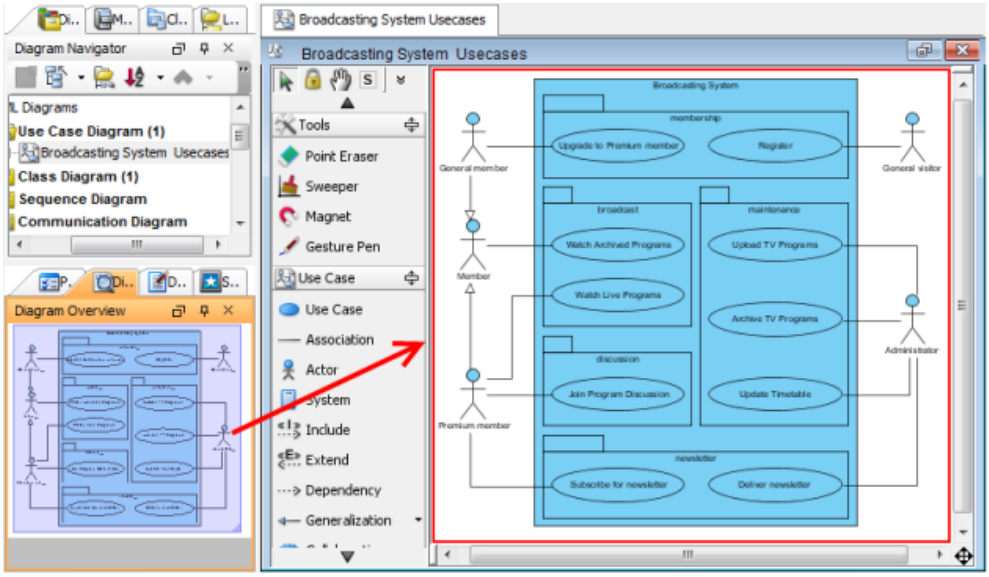
*The **Diagram overvi ew***

Closing and opening the diagram overview

Diagram overview is opened by default. To close it, press the **X** button at the top right corner. On the other hand, it can be opened by selecting **View > Panes > Diagram Overview** from the main menu.

Viewing an active diagram

An active diagram can be viewed in diagram overview automatically. As the diagram is shown on the diagram pane, it can be viewed in diagram overview.

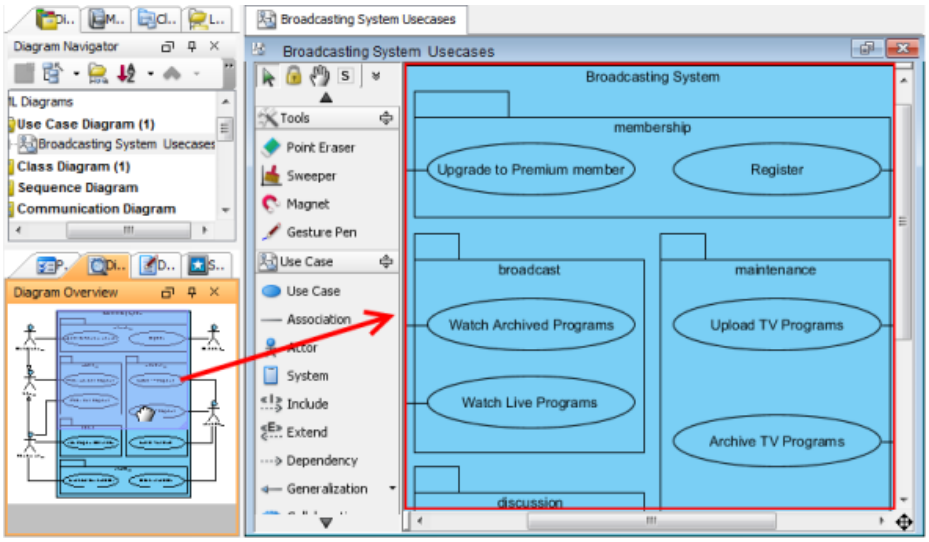


*An active diagram is shown on **Diagram Overview***

Having a quick view on a particular part of diagram

Viewing a large diagram, especially a particular part of this large diagram, is such an annoying task. To your convenience, a particular part of diagram can be navigated by moving the purple rectangle which represents the visible area of diagram in the user-friendly diagram overview.

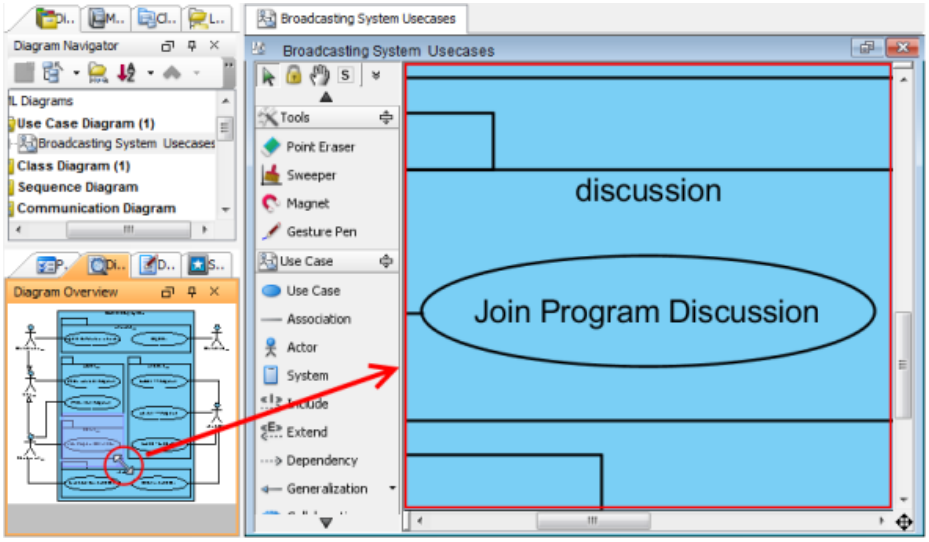
1. Press on the purple rectangle and drag it to the preferred part of diagram you would like to view.
2. As a result, the part of diagram will be shown on the diagram pane instantly.



The particular part of diagram is viewed on diagram pane

Zooming in a particular part of diagram

Drag the diagonal of purple rectangle to zoom in a particular part of diagram. The smaller you drag the purple rectangle, the more the part of diagram will be magnified.



Drag the diagonal of purple rectangle

On the contrary, the larger you drag the purple rectangle, the more the part of diagram will be dwindled.

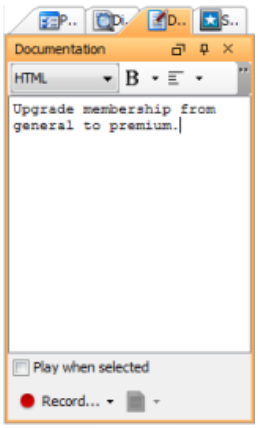
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Documentation pane

Documentation pane enables you to document project data such as model elements, shapes or diagrams either in written or verbal form. For written content, it can be a plain text or HTML text with formattings like bold, italic, table, etc.



*The **Documentation** pane*

The toolbar

Name	Icon	Description
HTML	HTML	It enables you to read and edit 3 different types of content. The 3 types of content include **HTML**, **HTML Source** and **Plain Text**. **HTML**: Read and edit the original content. **HTML Source**: Read and edit the HTML source of content. **Plain Text**: Read and edit content without formats.
Bold	B	Set the highlighted text to bold.
Italic	I	Set the highlighted text to italic.
Underline	u	Underline the highlighted text.
Left Justify		Set the alignment of highlighted text to left.
Center Justify		Set the alignment of highlighted text to center.
Right Justify		Set the alignment of highlighted text to right.
Ordered list		Add a numbered list.
Un-ordered list		Add a list with bullet points.
Font	F	Select the font family of highlighted text.
Font size	F	Select the size of highlighted text.
Font color		Select the color of highlighted text.
Table		Add a table. A few formats of insertion for rows and columns can be selected, including: **Insert Row Above**, **Insert Row Below**, **Insert Column on Left** and **Insert Column on Right**. **Insert Row Above**: Insert a row above the row you selected. **Insert Row Below**: Insert a row below the row you selected. **Insert Column on Left**: Insert a column on the left of the column you selected. **Insert Column on Right**: Insert a column on the right of the column you selected.
Background color		Select the background color of highlighted text.
Clear formats		Clear formats of whole editor to convert the content to plain text.
Link		Add a hyperlink in a specific place of documentation pane.

Image		Add an image in a specific place of documentation pane.
Add Model Element		Add a model element in a specific place of documentation pane.
Save as template...		Save the documentation as template.
Manage Template...		Preview a saved template.
Print		Print the custom content.

The description of icons on Documentation pane

Closing and opening the Documentation pane

Documentation pane is opened by default. To close it, press the **X** button at the top right corner. On the other hand, it can be opened by selecting **View > Panes > Documentation** from the main menu.

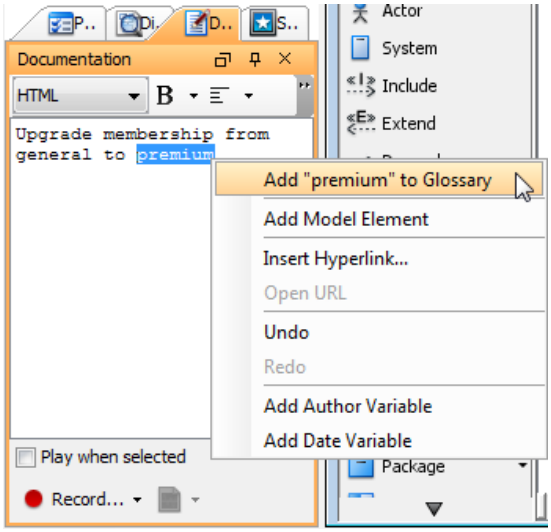
Documenting project data in text

Documentation pane enables users to type in the textual description for project data, for instance, model elements, shapes and diagrams.

Defining a glossary item

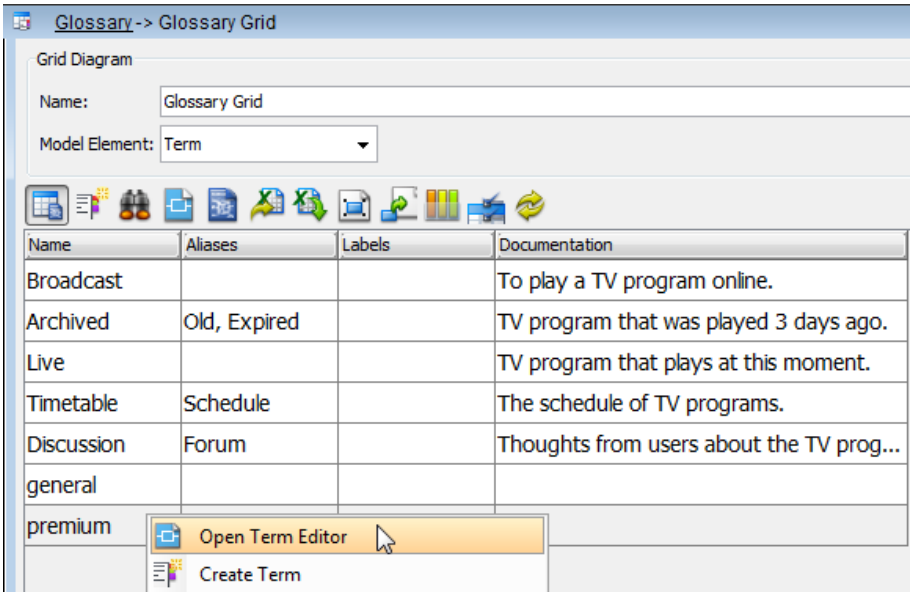
A word or a lexis can be defined as a glossary item for explication.

1. Highlight the word or the lexis you would like to be defined and then right click on it. Select **Add "[term]" to Glossary** from the pop-up menu to switch to **Glossary Grid**.



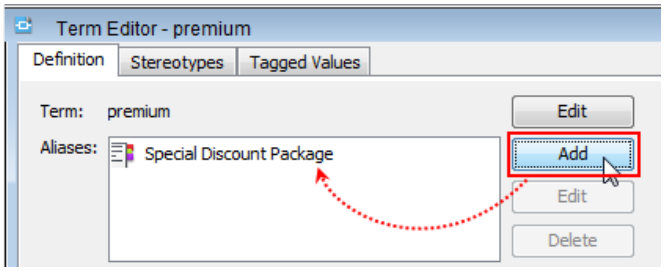
*Select **Add "premium" to Glossary** from the pop-up menu*

2. In **Glossary Grid**, select the term and then click **Open Term Editor** button in order to fill more details about the new item. Alternatively, right click on the term and select **Open Term Editor** from the pop-up menu.



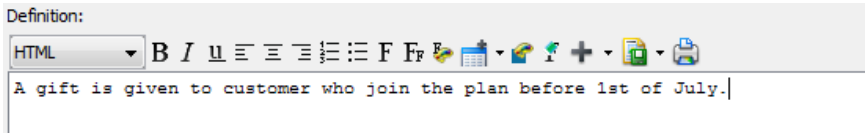
Open term editor

3. In **Term Editor** page, click **Add** button to type the alias(es) for the new item.



Type an alias for the term

4. Further information about the new term can be given by typing in the space under **Definition**.



Enter term definition

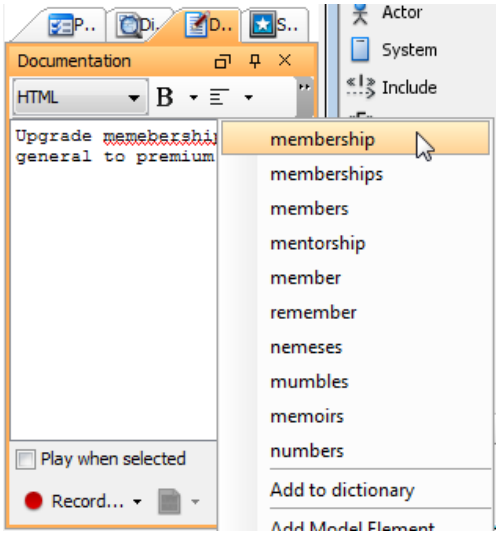
5. Finally, click **OK** to confirm editing. The window will then return to **Glossary Grid**.

6. Moreover, you can insert as many new terms as you prefer. In **Glossary Grid**, click to create another new term.

Checking the spelling

When you type an incorrect word carelessly, documentation pane can offer you a help.

For correction, right click on the incorrect word with a red curved line and select a suggested word from the pop-up menu.



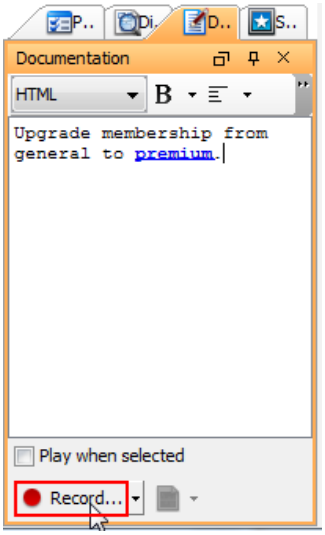
Select a correct word from the pop-up menu

Moreover, you can add a new word to the dictionary if the word you typed is a rare word or a new created word. Right click the new word and select **Add to dictionary** from the pop-up menu. When you type the word next time, it won't be marked as an incorrect word again.

Documenting a shape by voice

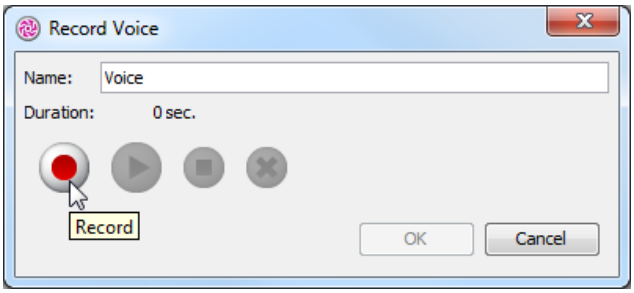
In addition to textual description, documentation pane also enables users to document for project data in verbal form .

Click **Record...** button for recording voice.



*Click **Record...** button*

In **Record Voice** dialog box, enter the name for the audio clip. Click **Record** to start recording while click **Stop** button to terminate. Click **OK** if you want to save the voice recording; click **Cancel**, and vice verse.



*Click **Record** button in **Record Voice** dialog box*

Name	Icon	Description
Record		Press it to start recording.
Play		Press it to play the voice you recorded.
Stop		Press it to stop recording.
Clear		Press it to clear the voice you recorded.

*The description of icons on **Record Voice** dialog box*

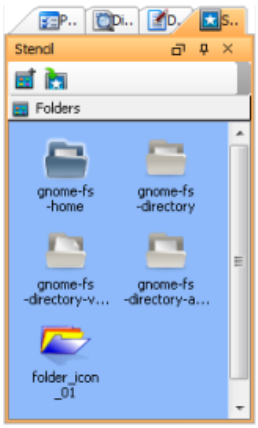
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Stencil pane

Stencil pane is a library of custom shapes or images that can be used on diagrams. With stencil pane, users can create custom shapes and display stencils by selecting stencils and drag them on diagram. Furthermore, stencils can be created in **Shape Editor**.



*The **Stencil** pane*

The toolbar

Name	Icon	Description
Add Stencil		Select a stencil to create a new shape. Different sorts of stencil are categorized into two folders: **Computers** and **Shapes** and a large amount of sub-folders are subdivided into these two folders.
Import Stencil...		Import a stencil that created externally in Microsoft Visio to Agilian.

The description of icons on stencil pane

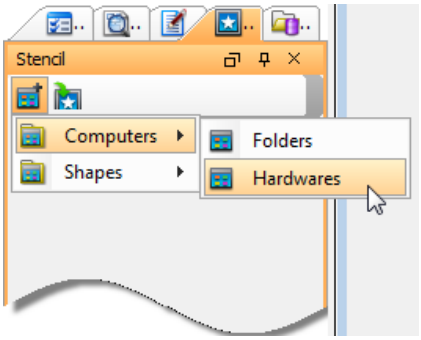
Closing and opening the stencil pane

Stencil pane is opened by default. To close it, press the **X** button at the top right corner. On the other hand, you can open it by selecting **View > Panes > Stencil** from the main menu.

Creating a stencil

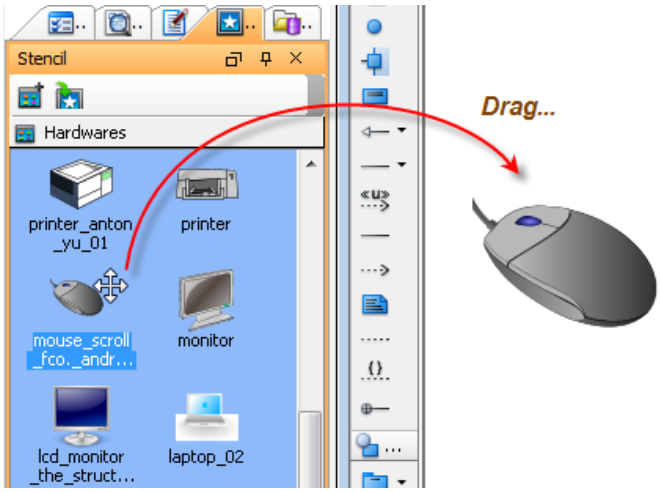
Instead of keeping the existing shapes, users can create a stencil to replace an existing one.

Click **Add Stencil** button on the toolbar of stencil pane and select a sub-folder from the pop-up menu.



*Select **Folders** from the pop-up menu*

When the sub-folder is unfolded, press the preferred stencil and drag it on the diagram pane.

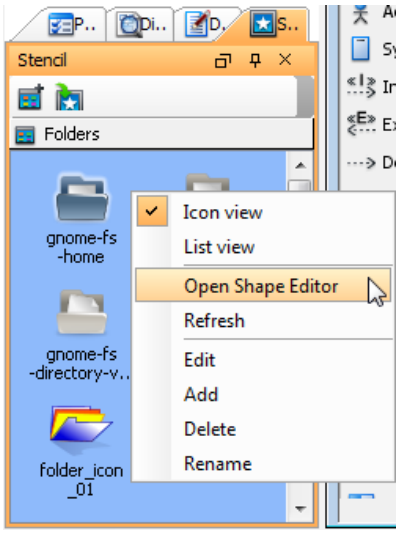


Drag a stencil on the diagram pane

Editing a stencil

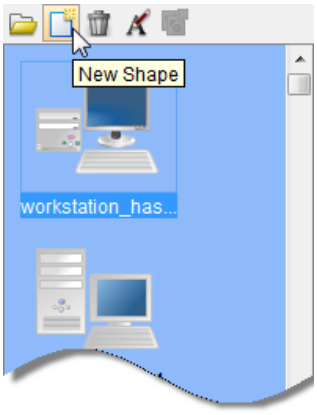
A stencil can be customized in accordance with users' preference in **Shape Editor**.

Right click on the stencil's background and select **Open Shape Editor** from the pop-up menu.



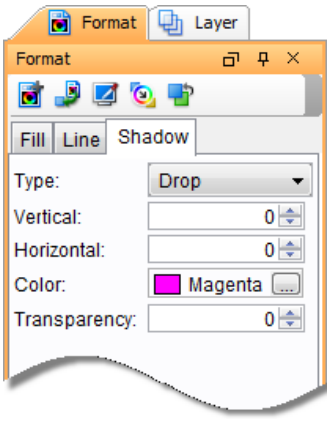
*Select **Open Shape Editor** from the pop-up menu*

In **Shape Editor** dialog box, click **New Shape** button.



*Click **New Shape** button*

Press on the stencil you want to be edited and drag it on the diagram pane.
Insert any shape(s) from the diagram toolbar with your preference and edit the format for the shape(s) in **Format** tab.



Turn the shadow of stencil into Megenta

Importing a stencil

The function of importing stencil enables you to import a stencil which is created in Microsoft Visio externally and send it to Agilian by making using of the plug-in "VisioSendToVP". Note that you are able to execute this function only after you have installed Microsoft Visio and have sent a stencil from Visio.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

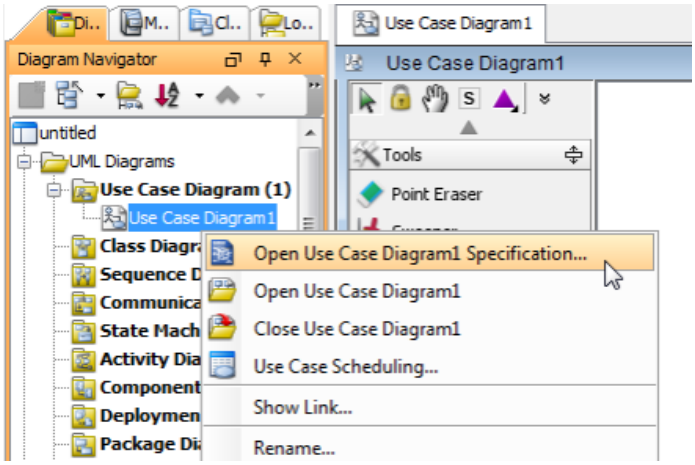
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagram Specification window

You can customize the settings of each diagram you created in diagram specification window. Those settings include: general diagram information, grid setting, references, project management and comments. Diagram specification window is similar with model element specification window in which you can also enter general information, add/ remove references, enter project management and add/ remove comments.

Opening diagram specification window

1. Right click on the target diagram on **Diagram Navigator** and select **Open [diagram name] Specification...** from the pop-up menu.



Open a diagram's specification window

2. As a result, the diagram specification window prompts out.

The overview of diagram specification window

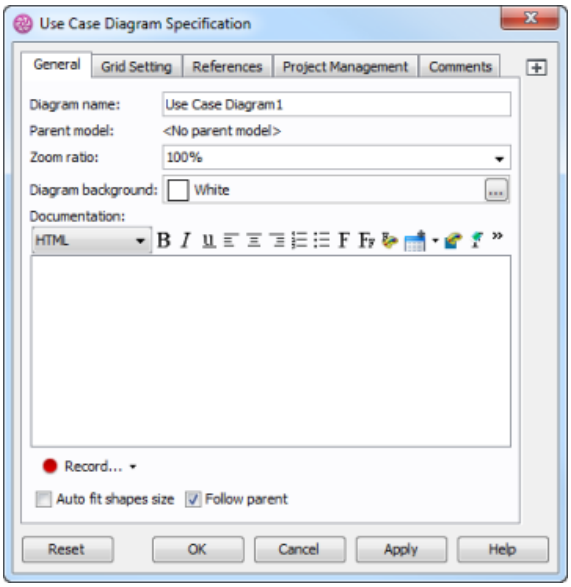


Diagram specification

A diagram specification window consists of 5 tabs: **General**, **Grid Setting**, **References**, **Project Management** and **Comments** respectively. You may notice that an icon named **Maximum** located at the right hand side of window. We'll introduce it after giving you a brief on 5 tabs.

General tab

You can specify diagram name, select zoom ratio and diagram background color, and enter documentation for the diagram. Furthermore, you can record voice documentation for the diagram. You can set all shapes including existing and future shapes to be fit size automatically.

Grid Setting tab

You can visualize the grid of diagram background by checking **Grid visible**. Moreover, you can set the size and select the color for the grid of diagram background.

References tab

You can add/ remove internal and external references for the diagram. Those references refer to file(s), folder(s), URL, diagram(s), shape(s), model element(s) and A3 resource(s).

Project Management tab

You can specify diagram process, priority, status, etc for project details.

Comments tab

You can add/ remove comment(s) for the diagram.

Maximum tab

Click to enlarge the specification window to the maximum screen size. After that, click it again to reduce it to the default size.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

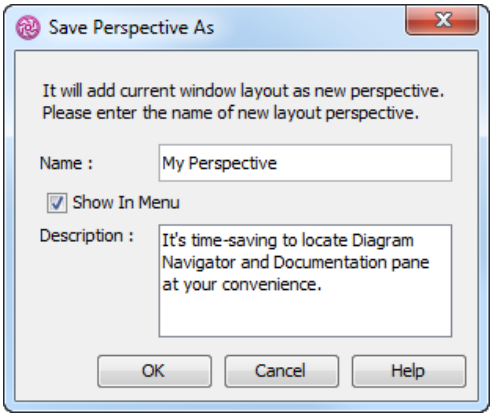
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Perspective

There are four types of pre-defined perspectives in Agilian : Default, Drawing, Informative and Resource. With Agilian, you can not only apply the pre-defined perspectives, but also customize your favorite perspective for executing tasks efficiently.

Saving your customized perspective

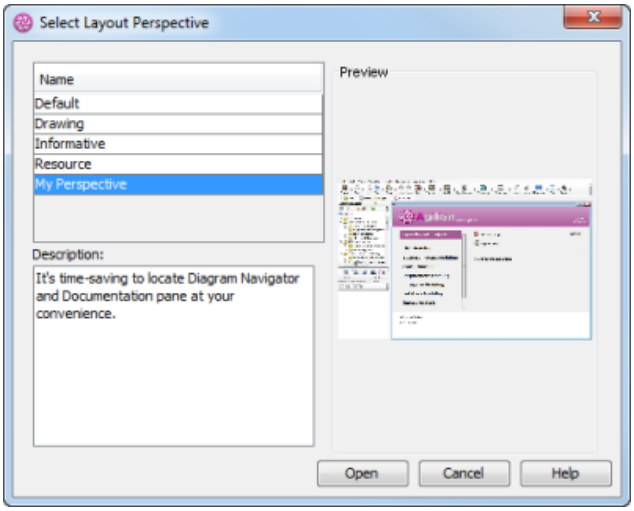
1. In your newly created perspective, select **View > Save Perspective as...** from the main menu.
2. In the **Save Perspective As** dialog box, enter name and description for the new perspective. Click **OK** button.



Enter name and description

Opening a perspective

1. Select **View > Open Perspective > Others...** from the main menu to open a perspective.
2. Besides the pre-defined perspectives, your customized perspectives are listed in the **Select Layout Perspective** dialog box.



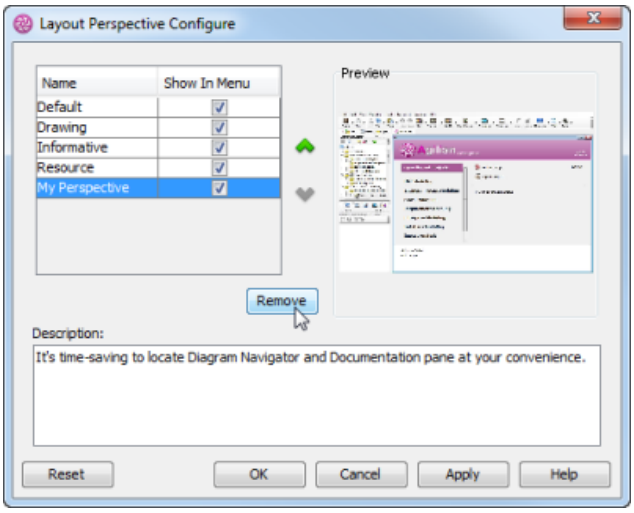
*The **Select Layout Perspective** dialog box*

3. Select a perspective under **Name** and click **Open** button.

Removing perspective

You can remove your customized perspective or even the pre-defined perspective(s) in the **Layout Perspective Configure** dialog box.

1. Select **View > Manage Perspectives...** from the main menu.
2. In the pop-up **Layout Perspective Configure** dialog box, select a perspective you want to remove and then press **Remove** button. Click **OK** button to confirm.



Remove a perspective

Related Resources

The following resources may help you learn more about the topic discussed in this page.

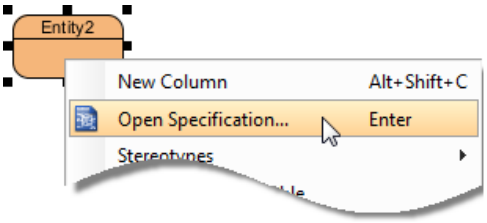
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Model's specification window

In Agilian , you can open a model's specification window to view and edit the model's details. The options, such as references, project management and comments are categorized into tabs in the specification window. Furthermore, three buttons attached on the right-hand side of window are **Pin**, **Auto open specification when select** and **Maximum**. Model specification window is similar with diagram specification window in which you can also enter general information, add/ remove references, enter project management and add/ remove comments.

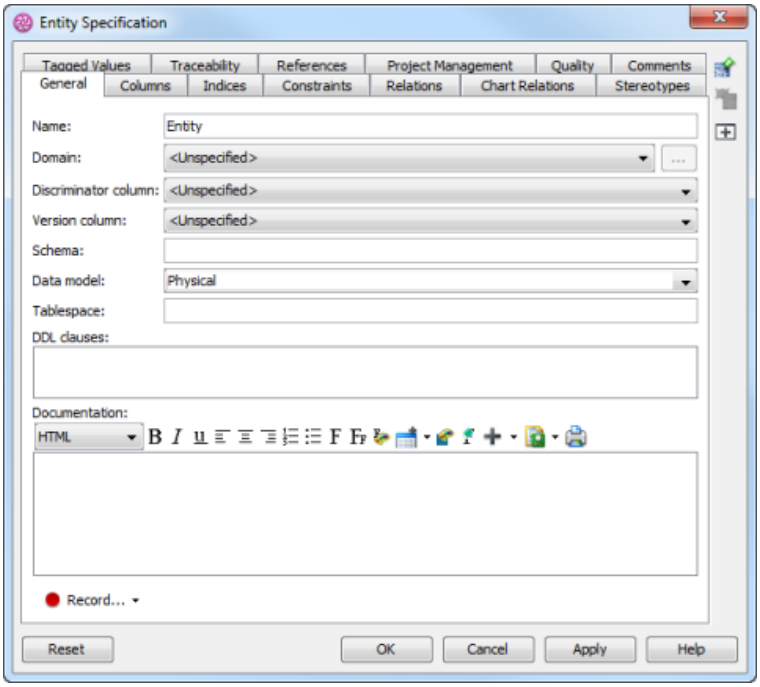
Opening a model's specification window

Right click on the target model which you want to view or edit its details and select **Open Specification...** from the pop-up menu.



Open specification window

As a result, the model's specification window pops out.

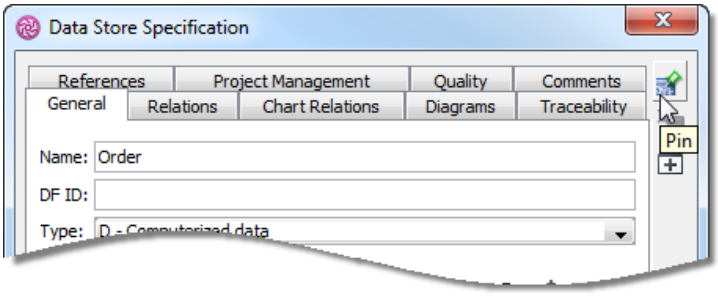


*The **Entity Specification** window*

The overview of three buttons on model's specification window

Pin

To pin the specification window, press **Pin** button.



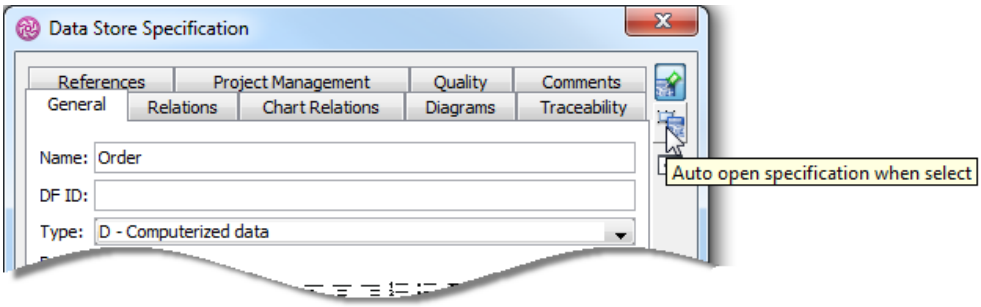
*Press **Pin** button*

NOTE: This button works in combination with **Auto open specification when select**.

Auto open specification when select

After you've pinned the specification window, press this button. After all, you don't have to close the current window in order to open another model's specification window.

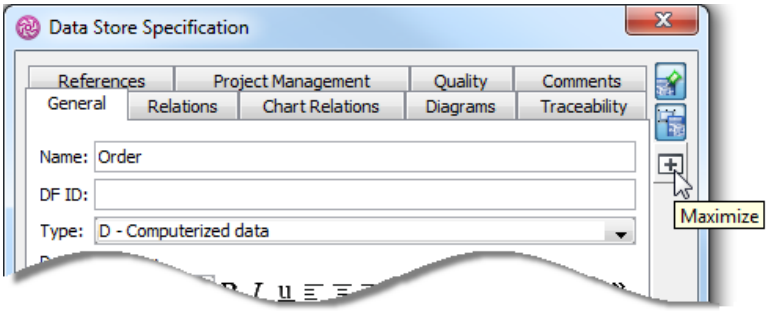
1. Press **Auto open specification when select** button.



*Press **Auto open specification when select** button*

2. Select another model to view its specification window by clicking on the target model.

Maximum
Press this button to enlarge the specification window to the maximum screen size.



*Click **Maximum** button*

After that, click it again to reduce it to the default size.

Related Resources
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Working with projects

This chapter introduces how to create and save project. You will also see how to create model and organize diagrams with model.

Creating project

This page shows you how to create a new project in Agilian.

Saving project

This page shows you how to save a project.

Organizing diagrams by model explorer

You may create models in model explorer for organizing diagrams. This page tells you how to do this in detail.

Maintaining backups

Backup files will be saved from time to time. This page tells you more about backup, and how to retrieve works from backup files.

Manage project properties dialog

Edit project properties like project name, author, company and project description.

Project template

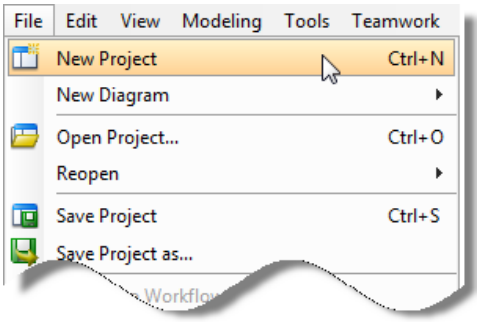
Project template enables you to specify the diagram to create by default when creating a new project.

Switch to diagram

You can open another diagram by double clicking on a tree node in Diagram Navigator. An alternative way is to open the Switch to Diagram dialog box, select diagram and click Activate Selected Diagram.

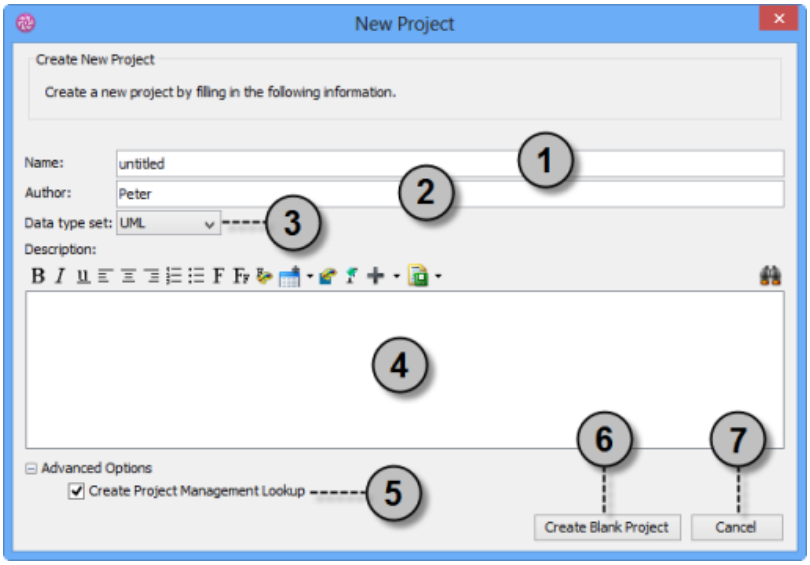
Creating project

Agilian stores information like model elements and diagrams in a project. Therefore, you need to create a project before performing modeling. To create a new project, select menu **File** > **New Project** from the main menu. When the **New Project** window pops out, click **Create Blank Project** button to create a new project.



Create a new project

Overview of New Project window



*An overview of **New Project** window*

No.	Name	Description
1	Name	The name of project.
2	Author	The person who create the project.
3	Data type set	Lets you select the programming/scripting language for the project. The language you selected mainly affects the class modeling. For example, the selectable visibilities and primitive types vary among languages.
4	Description	The project description. You can make use of the toolbar on top of the description pane to add formatted content.
5	Create Project Management Lookup	Check it to automatically fill the project management lookups such as iteration, version, etc with default lookup values.
6	Create Blank Project	Click to create the project.
7	Cancel	Click to cancel creating project, and close the window.

*Description of **New Project** window*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

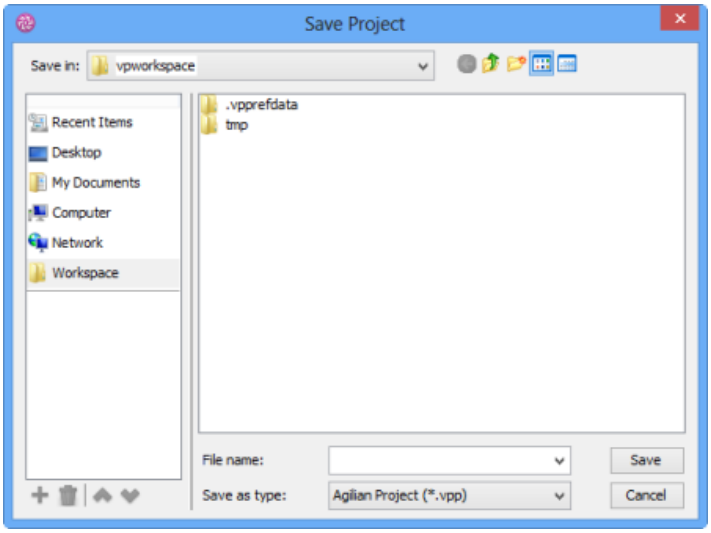
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Saving project

Agilian saves all project content to a single file, with file extension .vpp.

To save your work, select either **File > Save Project** or **File > Save Project as...** When you are saving a project for the first time, you will be asked to specify its location. If you have connected to VPository/VP Teamwork Server, you can directly import your project to the server now. Otherwise, you can save the file to a local directory.



Save Project

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Organizing diagrams by model explorer

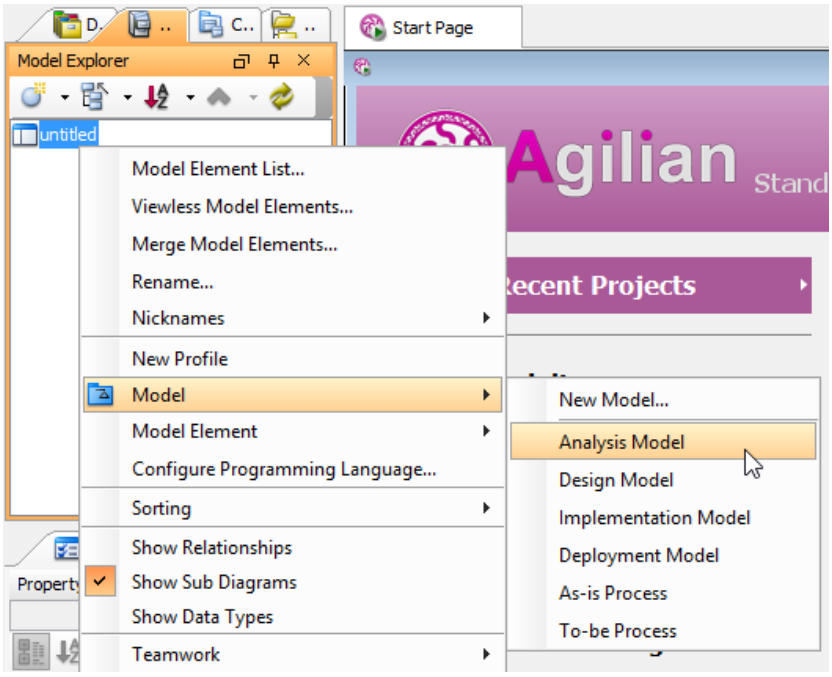
For small scale project, it would be easy to use Diagram Navigator to manage it. However, for middle to large scale project which has considerable numbers of diagrams and model elements, it would be better to use Model Explorer to organize the project.

Agilian loads diagrams and model elements only when they are used. For example, opening a diagram will load all its diagram elements, and opening the specification dialog box of a model element will cause it (and the model elements it referenced) to be loaded. Besides, selecting a tree node in the **Model Explorer** will cause the corresponding element(s) to be loaded as well.

For this reason, we recommend you to group diagrams using **Model** instead of laying them flat in the project. This can avoid accidentally loading diagrams and model elements that you never use, and thus can speed up project loading and saving.

Creating model

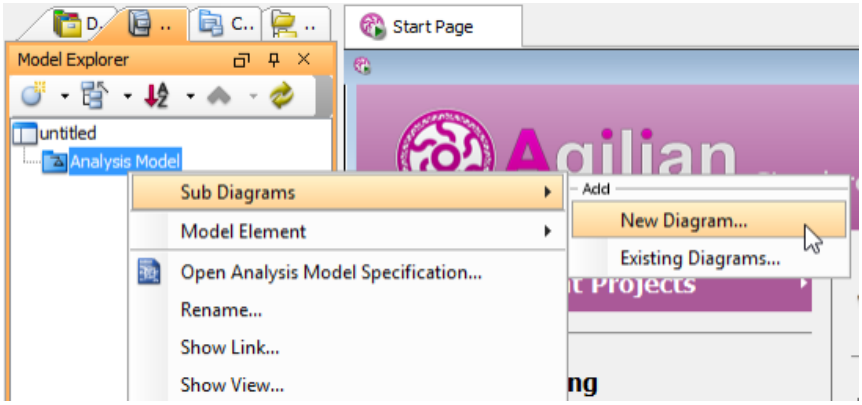
To create a Model, right-click on the project node in **Model Explorer** and select **Model** from the pop-up menu. You can either create a custom model by selecting **New Model...**, or create a pre-defined Model (e.g. Analysis Model) by selecting it in the list.



Create a model in Model Explorer

Creating diagram in model

To create diagram in model, right-click on the target model and select **Sub Diagrams > New Diagram**... from the pop-up menu. In the **New Diagram** window, select the type of diagram to create, enter its name and click **OK**.



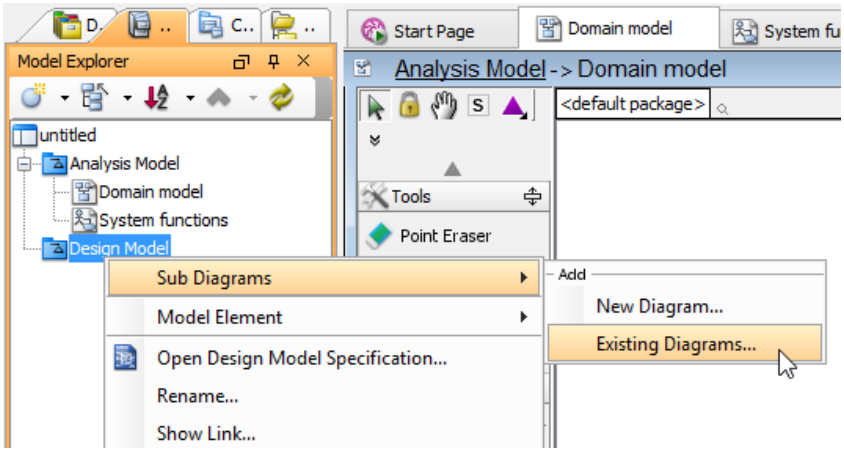
Create a diagram under Model

NOTE: When you draw a shape, its model element will be put under the same model as diagram.

Moving diagrams between models

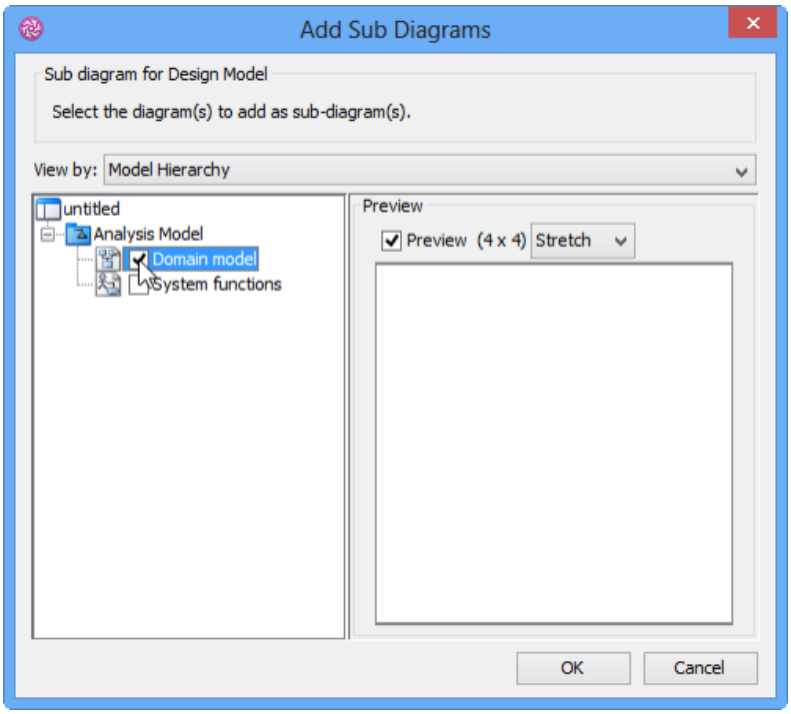
If you are not organizing project structure with model, you may want to do it now. You can move a diagram from root into a model, or transfer a diagram from one model to another.

To move diagram from one model to another, right-click on the target model in **Model Explorer** and select **Sub Diagram** > **Existing Diagram...** from the popup menu.



Add existing diagram to Model

Select the diagrams you want to move in the **Add Sub Diagrams** window and click **OK**.



Select diagram(s) to move

The selected diagrams will be moved to the target model.

NOTE: If you move a diagram which has the master view of model element(s), the model element(s) will be moved together with the diagram to the new model.

Related Resources

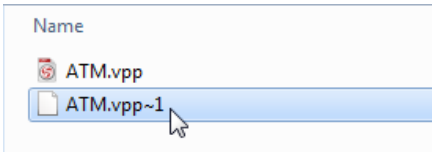
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Maintaining backups

Backup is a copy of project file. The major advantage of backup is to allow you to recover your work, in case you have made some undesired modification on your project. The backup file is usually put along with the project.

Backup is a default setting. After you save your project, it will be produced subsequently. The name of backup file is basically similar to the name of project file, but an extra ~ and a number are appended to it.

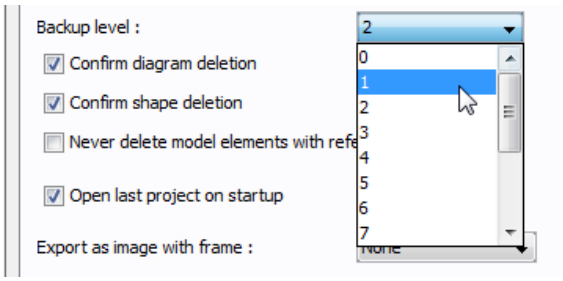


Backup copy is produced

Setting the backup level

You can set the amount of backup copy for your own reference. Select **Tools > Application Options...** from the main menu.

In **Options** dialog box, click **General** and select a number from the drop-down menu of **Backup level** under **Project** tab. The number of backup level represents the amount of backup copy that is produced after a project is saved.



*Select the amount of backup level in **Options** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Manage Project Properties window

In Agilian , you can specify the project name, main author of project, your company's name and a description of your project (in rich text format). With **Project Properties** dialog box, you can edit and review your project properties. For your convenience, when you create another new project, all the properties of previous project are set as default, except project name. However, you can modify those default properties in accordance with your preference.

1. Open **Project Properties** dialog box by selecting **File > Project Properties...** from the main menu.
2. Enter project name, author, company and project description. Click **OK** button to confirm and close the dialog box. Note that you can enter formatted text for project description.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

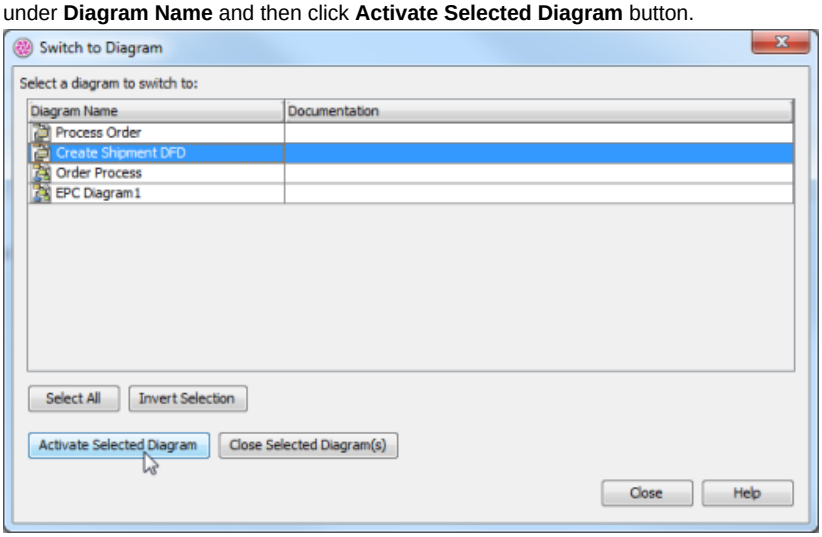
Switch to Diagram

After you've opened a few diagrams, you can switch to your target diagram easily with the feature of **Switch to Diagram**. Moreover, you can close diagram(s) with this feature as well.

1. After you've opened a few diagrams, select **Window > Switch to Diagram...** from the main menu.

> **NOTE:** Alternatively, you can right click on any diagram tab and select **Switch to Diagram...** from the pop-up menu.

2. In the **Switch to Diagram** dialog box, you can select a diagram to activate or select a diagram to close. To activate a diagram, select a diagram under **Diagram Name** and then click **Activate Selected Diagram** button.



Select a diagram to activate

On the other hand, select a diagram under **Diagram Name** and then click **Close Selected Diagram(s)** button to close the selected diagram.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Use case diagram

Use case diagram lets you model system functions (i.e. goals) as well as the actors that interaction with those functions. In this chapter, you will learn how to draw a use case diagram, how to record the events behind use cases using use case details and flow of events as well as to have a quick look on actor and use case grid.

Creating use case diagram

Shows you how to create use case diagram and provides you with information about those frequently used use case diagram elements.

Documenting use case details

Document the details of use case with the use case details editor.

Actor grid

Create, list and manage actor through the use of actor grid.

Use case grid

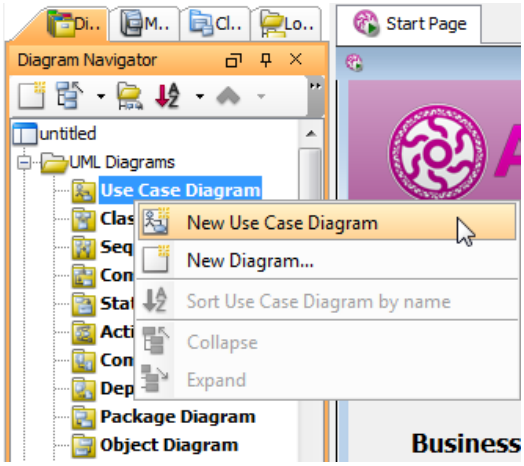
Create, list and manage use case through the use of use case grid.

Creating use case diagram

Use case diagram lets you model system functions (i.e. goals) as well as the actors that interact with those functions. You can draw use case diagrams in VP-UML as well as to document the event flows of use cases using the flow-of-events editor. In this page, you will see how to draw use case diagram. Flow of events will be mentioned in coming pages.

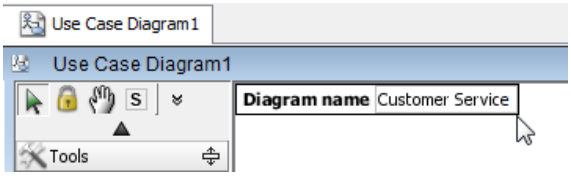
Creating a use case diagram

- Click on **UML** on toolbar and select **Use Case Diagram** from the drop down menu **.**
- Right click on **Use Case Diagram** in **Diagram Navigator** and select **New Use Case Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Use Case Diagram** from the main menu.



Create a use case diagram

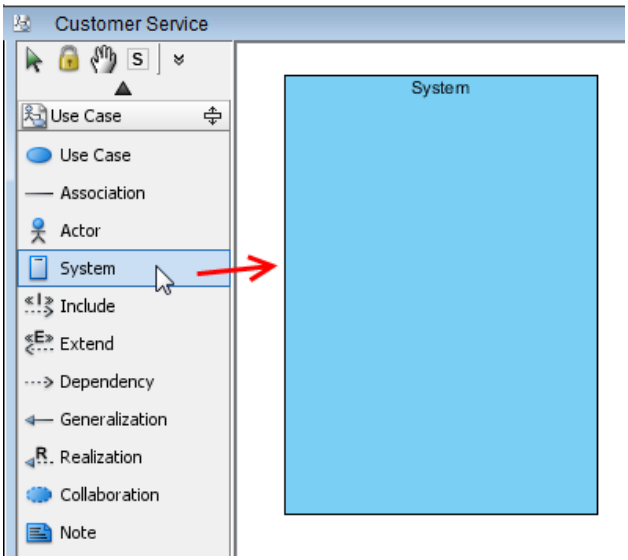
Enter name for the newly created use case diagram in the text field of pop-up box on the top left corner.



Enter name for the newly created use case diagram

Drawing a system

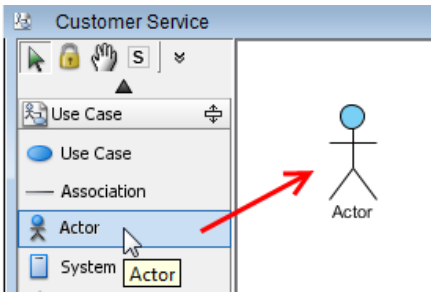
To create a system, select **System** on the diagram toolbar and then click it on the diagram pane. Finally, name the newly created system when it is created.



Create a system

Drawing an actor

To draw an actor, select **Actor** on the diagram toolbar and then click it on the diagram pane. Finally, name the newly created actor when it is created.

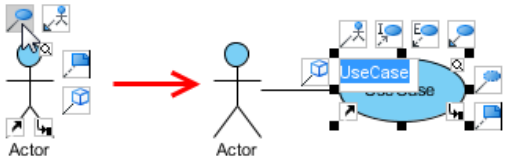


Create an actor

Drawing a use case

Besides creating a use case through diagram toolbar, you can also create it through resource icon.

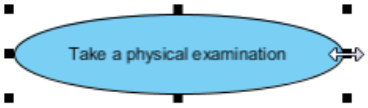
Move the mouse over a shape and press a resource icon that can create use case. Drag it and then release the mouse button until it reaches to your preferred place. The source shape and the newly created use case are connected. Finally, name the newly created use case.



Create a use case through resource icon

Line wrapping use case name

If a use case is too wide, for a better outlook, you may resize it by dragging the filled selectors. As a result, the name of use case will be line-wrapped automatically.

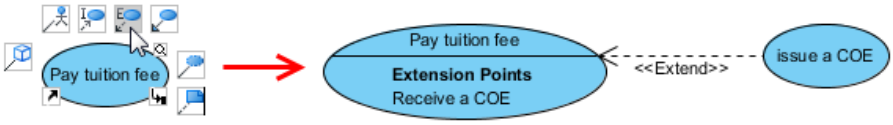


Resize a use case

NOTE: Alternatively, you can press **Alt** + **Enter** to force a new line.

Drawing <<Extend>> relationship

To create an extend relationship, move the mouse over a use case and press its resource icon **Extend -> Use Case**. Drag it to your preferred place and then release the mouse button. The use case with extension points and a newly created use case are connected. After you name the newly created use case, a pop-up dialog box will ask whether you want the extension point to follow the name of use case. Click **Yes** if you want it to do so; click **NO** if you want to enter another name for extension point.



Create an extend relationship

Drawing <<Include>> relationship

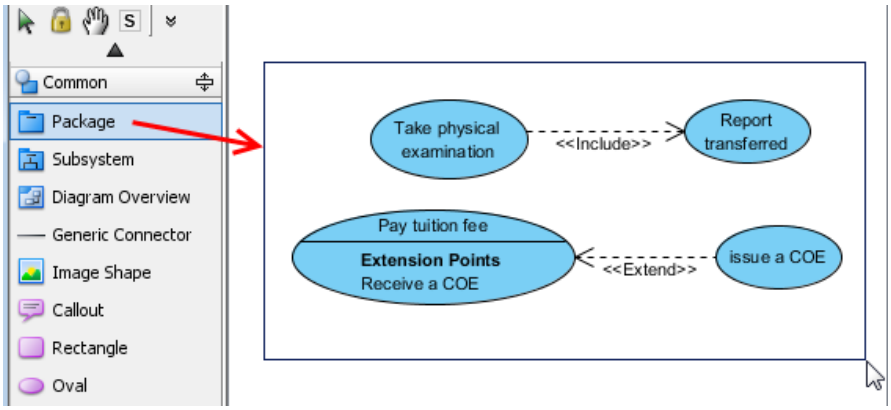
To create an include relationship, mouse over a use case and press its resource icon **Include -> Use Case**. Drag it to your preferred place and then release the mouse button. A new use case together with an include relationship is created. Finally, name the newly created use case.

Include relationship is created

Structuring use cases with package

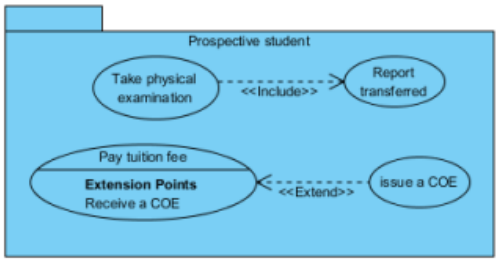
You can organize use cases with package when there are many of them on the diagram.

Select **Package** on the diagram toolbar (under **Common** category) and then drag the mouse to create a package surrounding those use cases.



Create a package

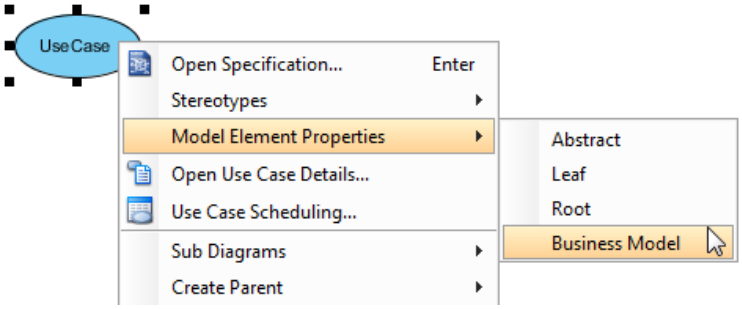
Finally, name the package.



Name the package

Drawing business use case

1. Right click on a use case and select **Model Element Properties > Business Model** from the pop-up menu.



*Click **Business Model***

2. After selected, an extra slash will be shown on the left edge of the use case.

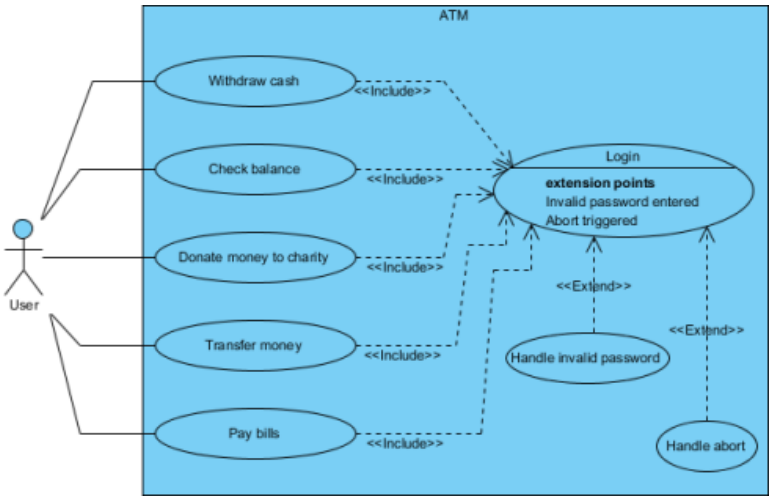
Business model

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- The 10-tips to create a professional use case diagram
- User's Guide - Use case diagram notations guide
- Visual Paradigm on YouTube
- Contact us if you need any help or have any suggestion

UML Use case diagram notations guide



Sample UML use case diagram

Here is a list of Unified Modeling Language (UML) notations supported in a UML use case diagram:

Icon	Name
	Use Case
	Association
	Actor
	System
	Include
	Extend
	Dependency
	Generalization
	Realization
	Collaboration

List of UML notations available in UML use case diagram

Use Case

UML use case

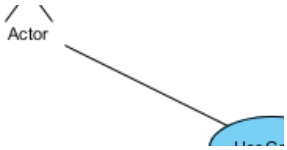
A use case represent a user goal that can be achieved by accessing the system or software application. In Agilian, you can make use of the sub-diagram feature to describes the interaction between user and system, within a use case by creating a sub-sequence diagram under a use case. You can also describe the use case scenario using the Flow of Events editor.

OMG UML Specification

What is a use case in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 606), use case is:

> A use case is the specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more actors or other stakeholders of the system.

Association



UML association

Actor and use case can be associated to indicate that the actor participates in that use case. Therefore, an association correspond to a sequence of actions between the actor and use case in achieving the use case.

OMG UML Specification

What is an association in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 36), association is:

> An association describes a set of tuples whose values refer to typed instances. An instance of an association is called a link. A link is a tuple with one value for each end of the association, where each value is an instance of the type of the end.
> ...
> An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type.
> An end property of an association that is owned by an end class or that is a navigable owned end of the association indicates that the association is navigable from the opposite ends; otherwise, the association is not navigable from the opposite ends.

Actor

UML actor

Actors are the entities that interact with a system. Although in most cases, actors are used to represent the users of system, actors can actually be anything that needs to exchange information with the system. So an actor may be people, computer hardware, other systems, etc.

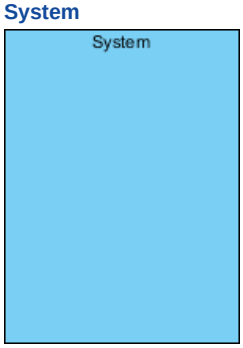
Note that actor represent a role that a user can play, but not a specific user. So in a hospital information system you may have doctor and patient as actors, but not Dr. John, Mrs. Brown as actors.

OMG UML Specification

What is an actor in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1), actor is:

> An actor specifies a role played by a user or any other system that interacts with the subject. (The term "role" is used informally here and does not necessarily imply the technical definition of that term found elsewhere in this specification.)
> ...
> An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances.

System



UML system

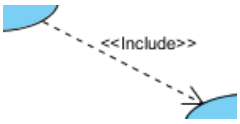
The scope of a system can be represented by a system (shape), or sometimes known as a system boundary. The use cases of the system are placed inside the system shape, while the actor who interact with the system are put outside the system. The use cases in the system make up the total requirements of the system.

OMG UML Specification

What is a system in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 608), system is:

> If a subject (or system boundary) is displayed, the use case ellipse is visually located inside the system boundary rectangle. Note that this does not necessarily mean that the subject classifier owns the contained use cases, but merely that the use case applies to that classifier.

Include



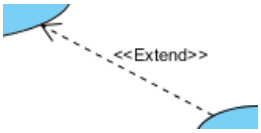
UML include

An include relationship specifies how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.

OMG UML Specification
What is an include in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 604), include is:

> An include relationship defines that a use case contains the behavior defined in another use case.

Extend



UML extend

An extend relationship specifies how the behavior of the extension use case can be inserted into the behavior defined for the base use case.

OMG UML Specification
What is an extend in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 601), extend is:

> A relationship from an extending use case to an extended use case that specifies how and when the behavior defined in the extending use case can be inserted into the behavior defined in the extended use case.
> ...
> This relationship specifies that the behavior of a use case may be extended by the behavior of another (usually supplementary) use case. The extension takes place at one or more specific extension points defined in the extended use case. Note, however, that the extended use case is defined independently of the extending use case and is meaningful independently of the extending use case. On the other hand, the extending use case typically defines behavior that may not necessarily be meaningful by itself. Instead, the extending use case defines a set of modular behavior increments that augment an execution of the extended use case under specific conditions.
> Note that the same extending use case can extend more than one use case. Furthermore, an extending use case may itself be extended.

Dependency

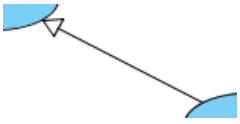
UML dependency

A dependency relationship represent that a model element relies on another model element for specification and/or implementation.

OMG UML Specification
What is a dependency in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 61), dependency is:

> A dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation. This means that the complete semantics of the depending elements is either semantically or structurally dependent on the definition of the supplier element(s).

Generalization



UML generalization

A generalization relationship is used to represent inheritance relationship between model elements of same type. The more specific model element share the same specification with the more general model element, but carries more details in extra.

OMG UML Specification
What is a generalization in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 70), generalization is:

A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. Each instance of the specific classifier is also an indirect instance of the general classifier. Thus, the specific classifier inherits the features of the more general classifier.

Realization

UML realization

A realization is a relationship between a specification and its implementation.

OMG UML Specification

What is a realization in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 131), realization is:

Realization is a specialized abstraction relationship between two sets of model elements, one representing a specification (the supplier) and the other represents an implementation of the latter (the client). Realization can be used to model stepwise refinement, optimizations, transformations, templates, model synthesis, framework composition, etc.

Collaboration

UML collaboration

OMG UML Specification

What is a collaboration in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 174), collaboration is:

A collaboration describes a structure of collaborating elements (roles), each performing a specialized function, which collectively accomplish some desired functionality. Its primary purpose is to explain how a system works and, therefore, it typically only incorporates those aspects of reality that are deemed relevant to the explanation. Thus, details, such as the identity or precise class of the actual participating instances are suppressed.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

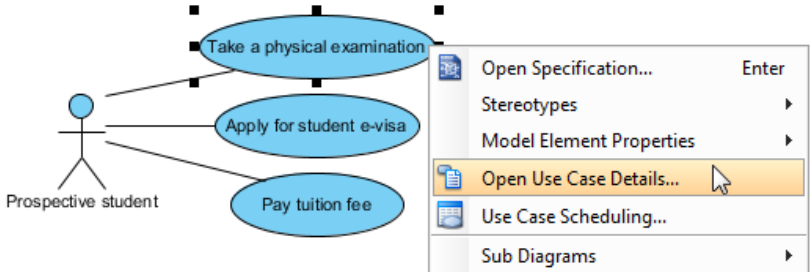
- The 10-tips to create a professional use case diagram
- User's Guide - Creating a use case diagram
- Visual Paradigm on YouTube
- Contact us if you need any help or have any suggestion

Documenting use case details

The feature of use case details refers to the basic information, flow of events , requirements and test plan of a use case. Documenting use case details is essential in recording meaningful and important information for a use case.

Opening use case details

To start editing and viewing use case details, right click on the target use case in use case diagram and select **Use Case Details...** from the pop-up menu.

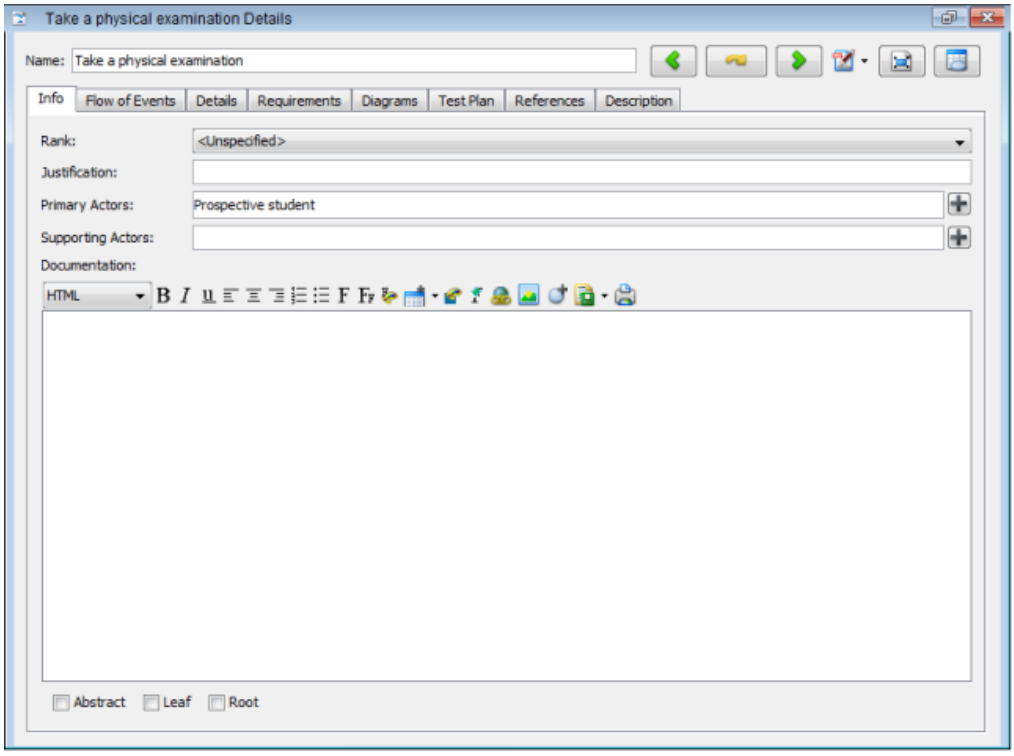


*Select **Open Use Case Details...***

Entering basic information

Basic information refers to all general information of a use case. Rank and justification determine the importance of use case. Select a rank from the drop-down menu and enter the text in **Justification** text field.

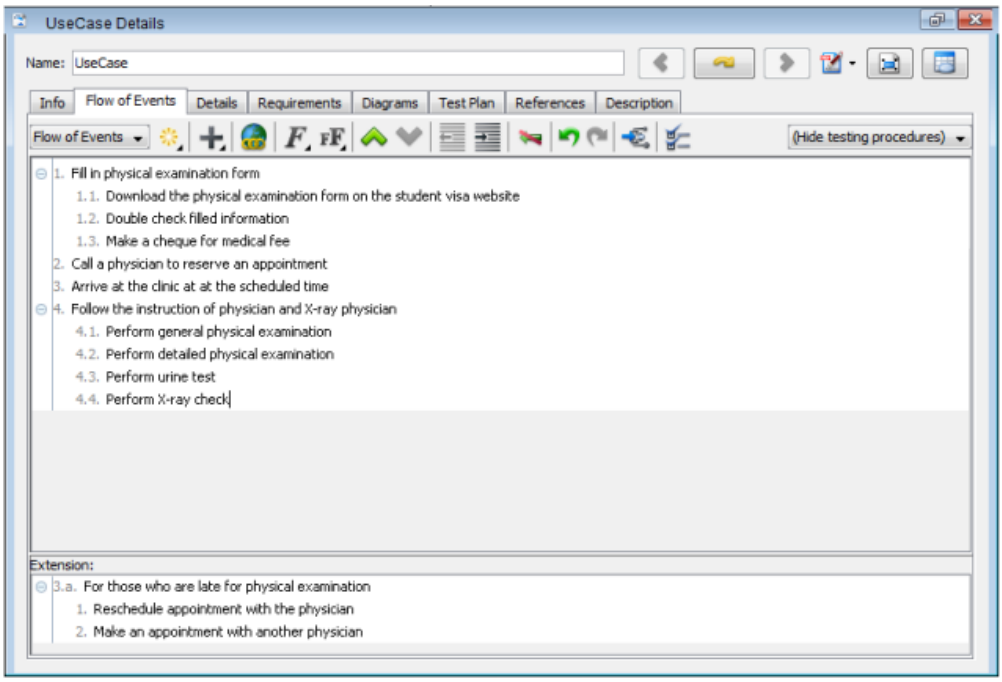
Primary actors list the actors being involved in a use case. Actors that are connected to a use case are automatically defined as primary actors. Supporting actor are actors who are beneficial from the system, but without direct interaction. Both primary and supporting actors can be added manually by pressing the **Plus** button and select the actors in the pop-up dialog box.



Basic information of use case

Entering flow of events

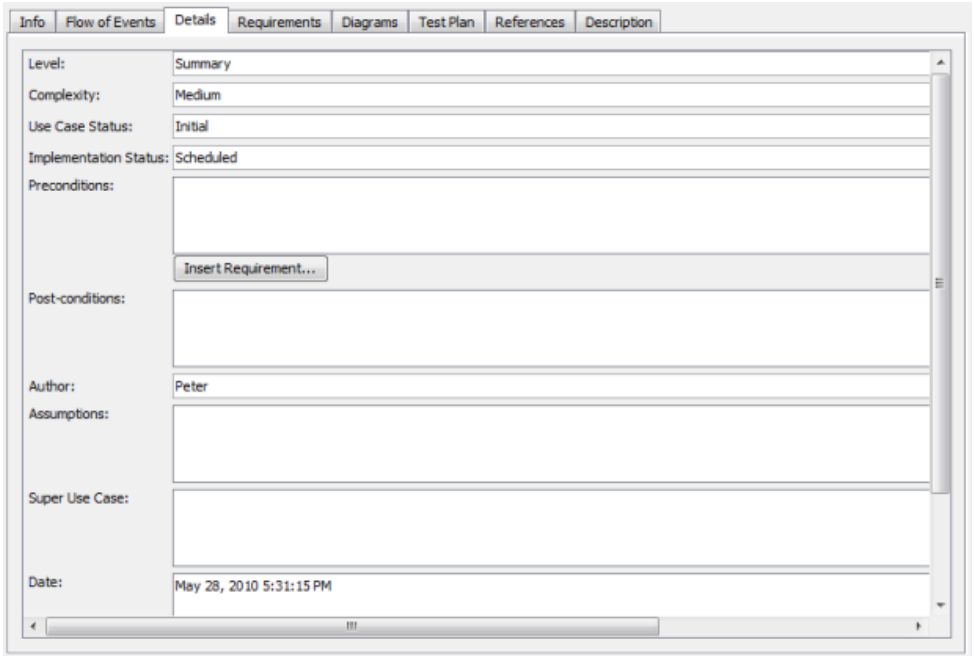
Flow of events refers to the steps required to go through and fulfill a use case. You may define multiple flows of events under a use case and add extension to an event as well. For more information about documenting flow of events, read the next chapter **Documenting Flow of Events**.



Flow of events of use case

Entering details

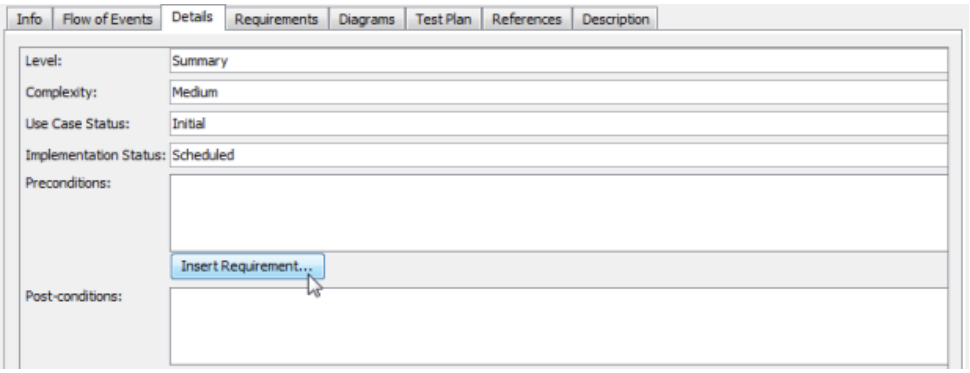
Details are predefined and detailed fields of a use case, which includes level, complexity, status, implementation status, preconditions and post-conditions, author and assumptions. Select an option for **Level**, **Complexity**, **Use Case Status** and **Implementation Status** from the drop-down menu.



Details of use case

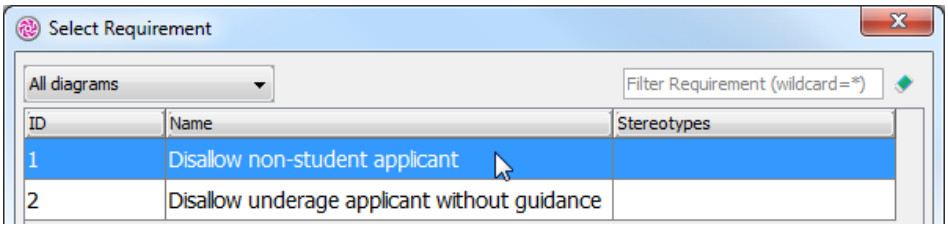
Inserting requirement links

1. Click in the text field where you want to insert a requirement link. Click the **Insert Requirement...** button when it pops out. Note that only fields where support multiple line are allowed to add requirement links.



*Click **Insert Requirement...** button*

2. When the **Select Requirement** dialog box pops out, select the requirement you want to link to and click **OK** to confirm. The searching scope of selecting requirement may be narrow down if you find too many requirements in your project. Select a specific diagram from the drop-down menu at the top left corner of dialog box, or enter its name at the **Filter** field directly at the top right corner.

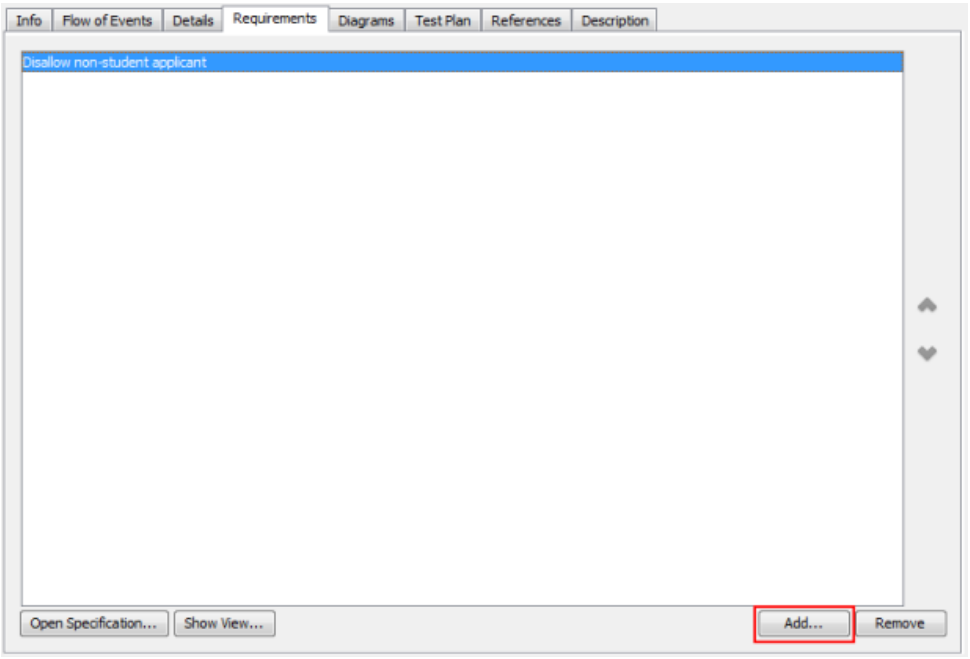


Select a requirement

3. Once the link is inserted in the text field, you can right click to navigate it through its pop-up menu.

Adding requirements

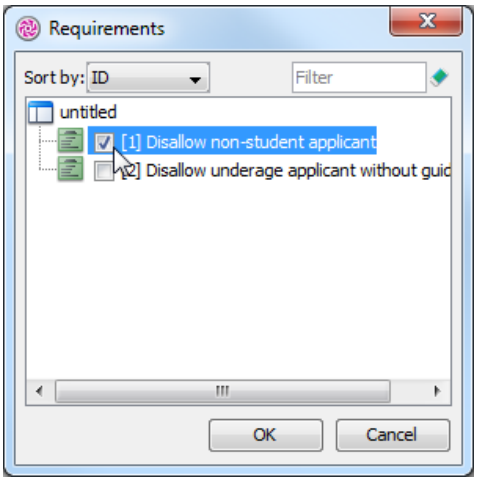
Requirements of a use case can be added in the **Requirements** page.



Requirements of use case

To add requirement(s) to a use case:

1. Click the **Add...** button at the bottom right of dialog box.
2. In the **Requirements** dialog box, look for and select the requirements to add, and click **OK** to confirm the selection.

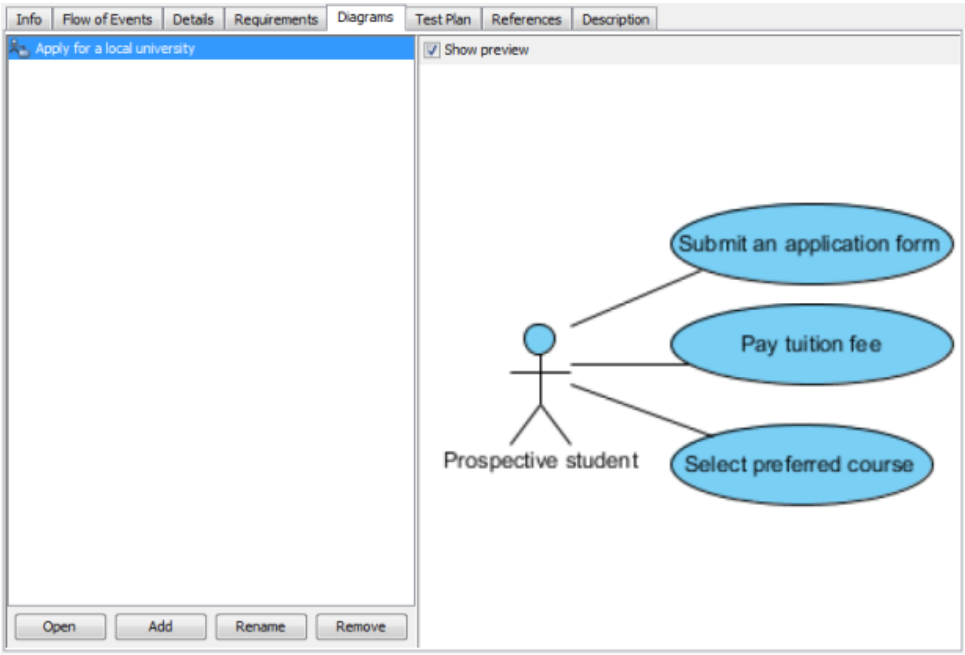


Select a requirement

NOTE: The **Requirements** page is for adding existing requirements as requirements. If you want to define a new requirement, read the next section *Adding a sub-diagram*. Information about how to add a requirement diagram as sub-diagram and define the requirements in the diagram is provided. Requirements are made in **Diagrams** page will be automatically added to the use case's requirements.

Managing sub-Diagrams

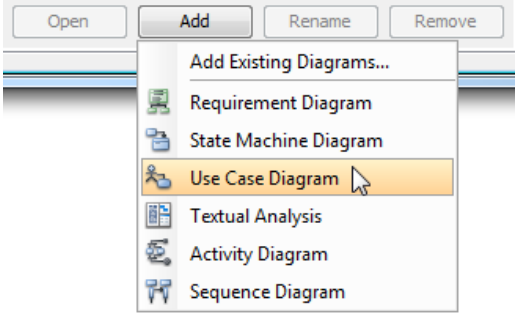
You can make use of another diagram for elaborating a use case. The **Diagrams** page enables you to add and open sub-diagrams of a use case. When you select a diagram on the list on the left, you may preview it on the right if **Show preview** is checked.



Diagrams of use case

Adding a sub-diagram

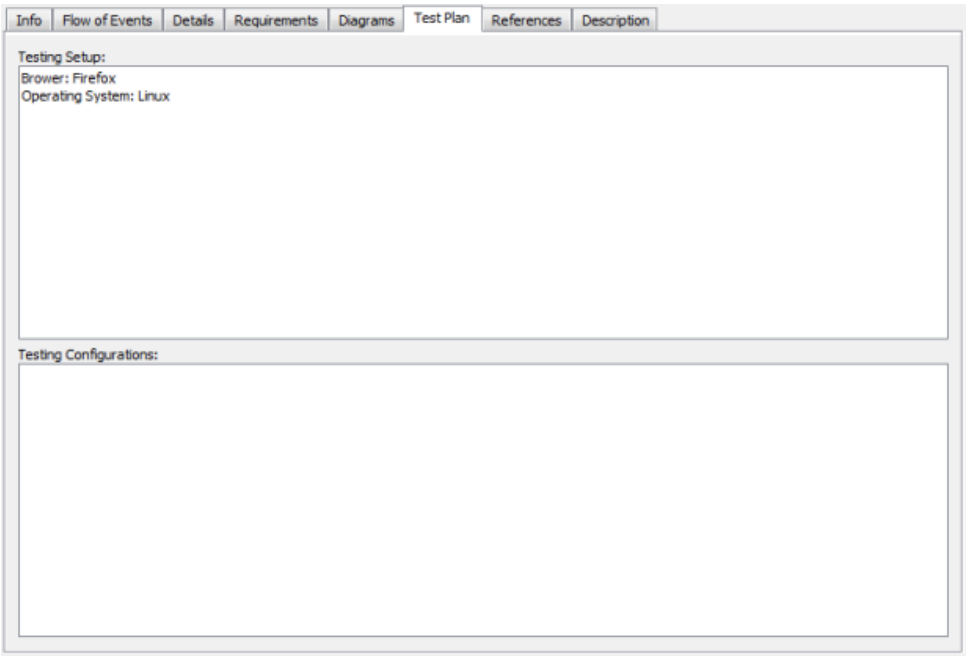
1. Click the **Add** button at the bottom of **Diagrams** page, select a type of diagram from the pop-up menu if you want to add a new diagram as sub-diagram. On the other hand, select **Add Existing Diagrams...** if you want to add an existing diagram in your current project.



Add a sub-diagram

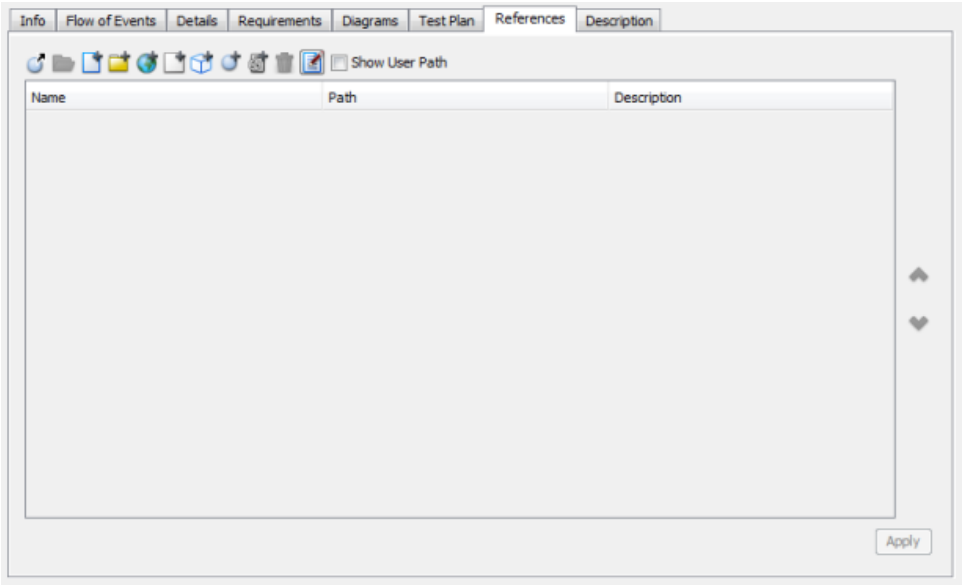
Opening a sub-diagram
Select a sub-diagram on the list to open and click the **Open** button at the bottom of **Diagrams** page.

Writing test plan
While the detailed testing procedure can be documented in flow of events, the testing setup and configurations can be documented in the **Test Plan** tab.



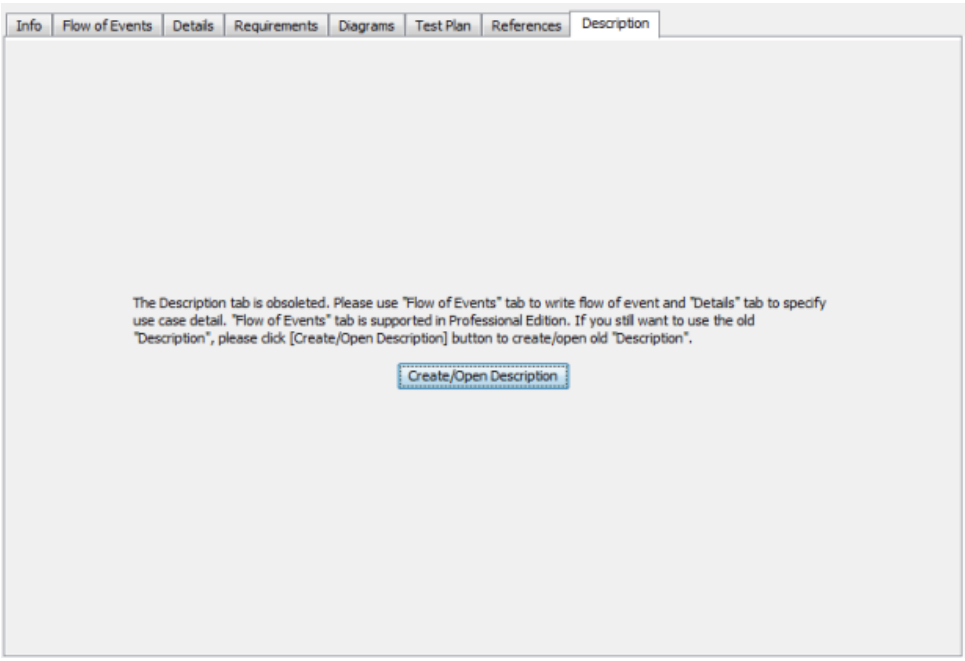
Test Plan of use case

Adding references
You may add references to both internal and external artifacts, such as shapes, diagrams, files, folders and URLs for describing the use case in various views.



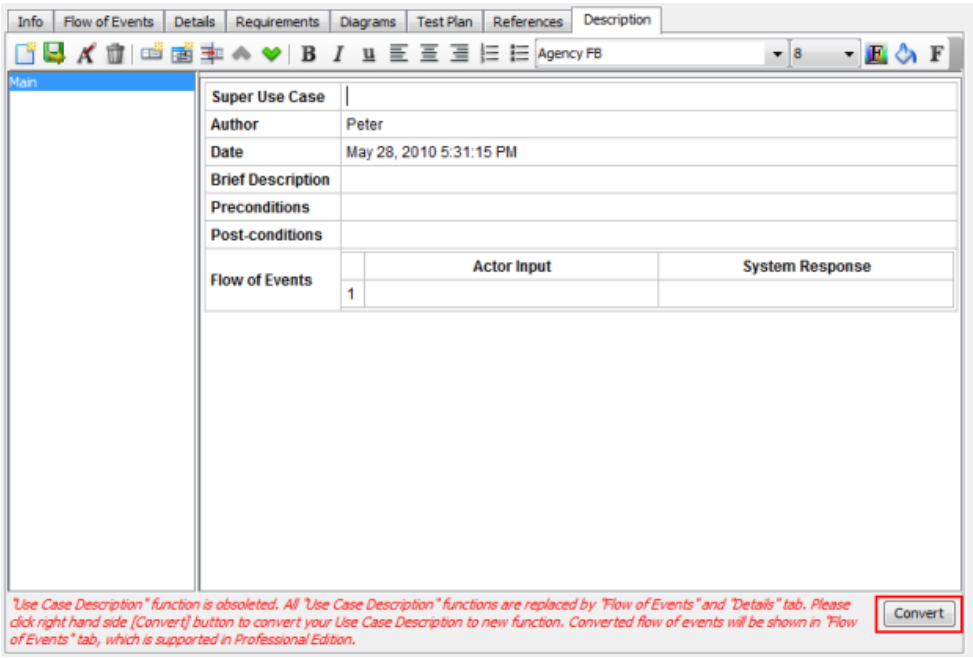
References of use case

Opening obsolete use case description
Use Case Description was an obsolete feature removed in Visual Paradigm Suite version 4.1. We recommend users to use the flow of events tool to document the internal flow of use case rather than making use of use case description. But for the old version users, they can still activate the **Description** pane to access the saved data.



Obsolete use case description

You can convert the obsolete use case details to flow of events by clicking on the **Convert** button at the bottom right corner.



Convert use case description

Related Resources

The following resources may help you learn more about the topic discussed in this page.

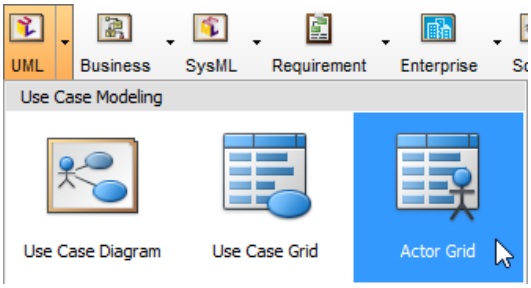
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Managing actors with Actor Grid

Actor Grid is a table with actors listed in it. It lets you to access all actors in a project or diagram, check their related use cases, lookup and create actors.

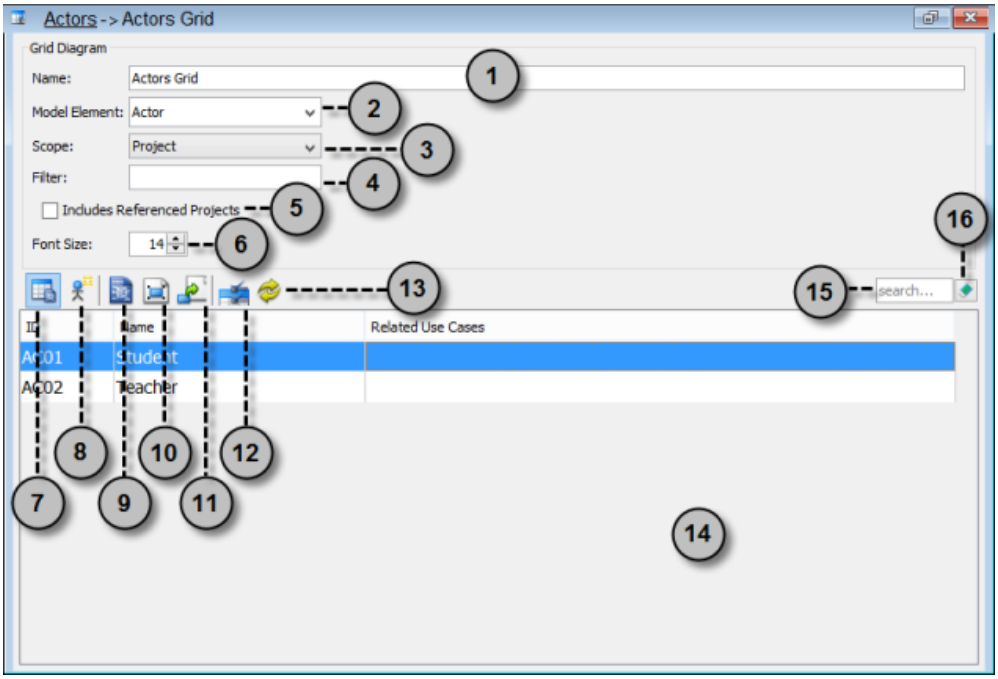
Creating the Actor Grid

Click on **UML** on toolbar and select **Actor Grid** from the drop down menu **.**



*Open **Actor Grid***

The overview of Actor Grid



The Actor Grid

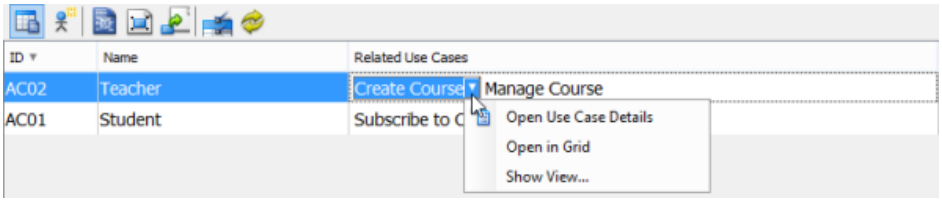
No.	Name	Description
1	Name	The name of this **Actor Grid**.
2	Model Element	Change the type of model element to be listed in grid. Although **Actor** is selected by default, you may change it to other types.
3	Scope	The location to look for the actors to list in grid. By default, actors are found from the whole project. You can change to find actors from specific model or package, or to find only actors right at the root level. You can also restrict the scope to all diagrams, to within a specific diagram or to all actors that has not been visualized in any diagram.
4	Filter	Apply filter to grid content. Text entered here is matched against the **Name** property of actors listed in grid. Actors that do not contain the entered text in their name are hidden.
5	Include Referenced Projects	Check it to list also actors in referenced projects, in **Actor Grid**.
6	Font Size	Click to adjust the font size of text in **Actor Grid**.
7	Configure Grid	Click to show/hide the grid configuration panel, which allows you to enter the name of grid, the model element to be listed in grid, the scope and to apply filter to grid content.
8	New Actor	Click to create an actor.
9	Open Specification...	Select an actor in **Actor Grid** and click this button to open its specification.

10	Show View...	Select an actor in **Actor Grid** and click this button to list the diagrams that contains the view of the selected actor.
11	Visualize...	Select an actor in **Actor Grid** and click this button to show it in a new or existing diagram.
12	Configure Columns...	Click to select the property(ies) of actors to be listed in the grid, as columns.
13	Refresh	Click to refresh the grid content by showing the most updated information of actors listed.
14	List of actors	Actors are listed here.
15	Search	Find actor(s) by entering search criteria.
16	Clear	Click to clear the text entered in **Search** box.

The fields in Actor Grid

Accessing use case from actor

Click on the name of use case in the **Related Use Cases** cell of the actor that you want to access its related use case. Select the way to access the use case.

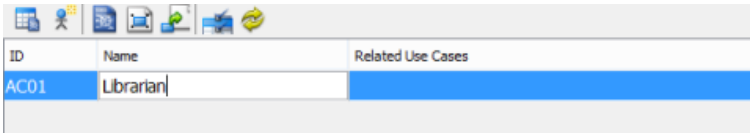


Access use case from actor

Creating actor in Actor Grid

To create an actor in **Actor Grid**:

1. Click on **New Actor** above the **Actor Grid**.
2. Enter the name of actor.



Creating actor in Actor Grid

3. Press **Enter** to confirm editing.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

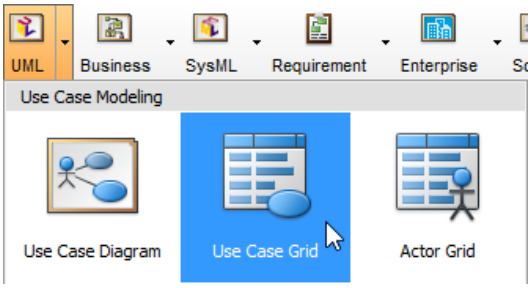
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Managing use cases with Use Case Grid

Use case grid is a table with use cases listed in it. It lets you to access all use cases in a project or diagram, check their related actors, lookup and create use case.

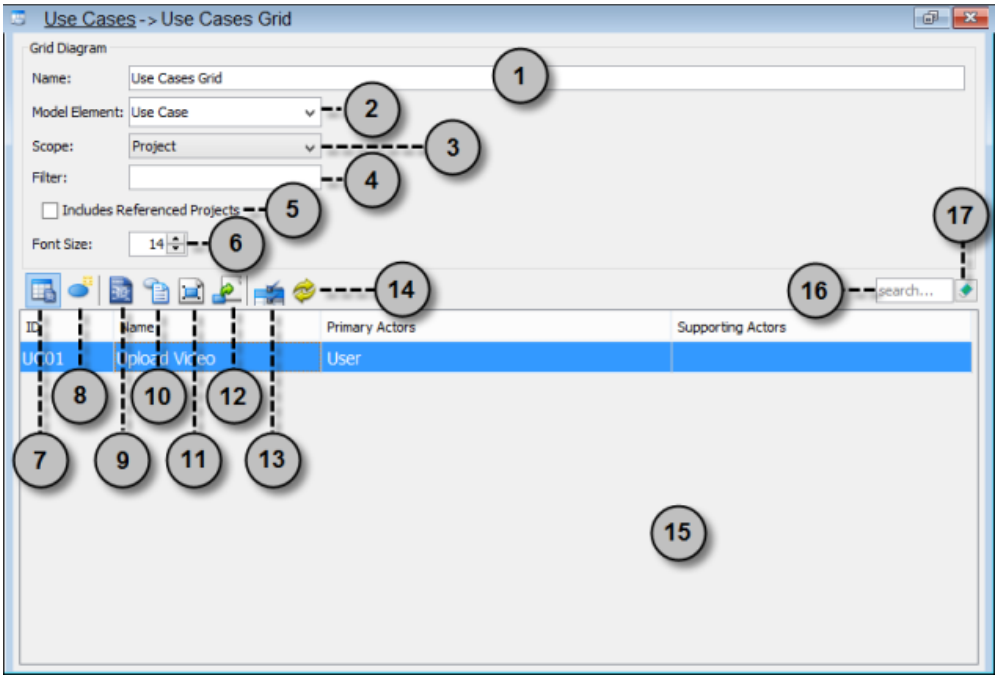
Creating the Use Case Grid

Click on **UML** on toolbar and select **Use Case Grid** from the drop down menu **.**



*Open **Use Case Grid***

The overview of Use Case Grid



The Use Case Grid

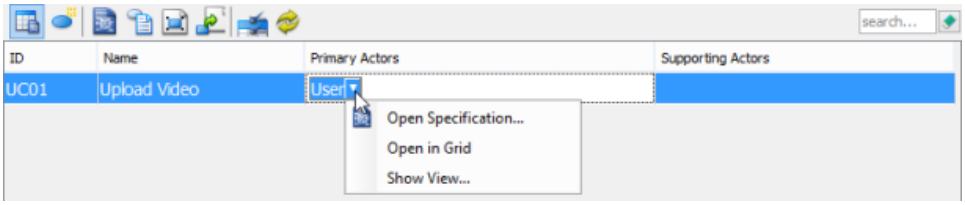
No.	Name	Description
1	Name	The name of this **Use Case Grid**.
2	Model Element	Change the type of model element to be listed in grid. Although **Use Case** is selected by default, you may change it to other types.
3	Scope	The location to look for the use cases to list in grid. By default, use cases are found from the whole project. You can change to find use cases from specific model or package, or to find only use cases right at the root level. You can also restrict the scope to all diagrams, to within a specific diagram or to all use cases that has not been visualized in any diagram.
4	Filter	Apply filter to grid content. Text entered here is matched against the **Name** property of use cases listed in grid. Use cases that do not contain the entered text in their name are hidden.
5	Include Referenced Projects	Check it to list also use cases in referenced projects, in **Use Case Grid**.
6	Font Size	Click to adjust the font size of text in **Use Case Grid**.
7	Configure Grid	Click to show/hide the grid configuration panel, which allows you to enter the name of grid, the model element to be listed in grid, the scope and to apply filter to grid content.
8	New Use Case	Click to create a use case.
9	Open Specification...	Select a use case in **Use Case Grid** and click this button to open its specification.
10	Open Use Case Details	Select a use case in **Use Case Grid** and click this button to acces its details in **Use Case Details** editor.

11	Show View...	Select a use case in **Use Case Grid** and click this button to list the diagrams that contains the view of the selected use case.
12	Visualize...	Select a use case in **Use Case Grid** and click this button to show it in a new or existing diagram.
13	Configure Columns...	Click to select the property(ies) of use cases to be listed in the grid, as columns.
14	Refresh	Click to refresh the grid content by showing the most updated information of use cases listed.
15	List of use cases	Use cases are listed here.
16	Search	Find use case(s) by entering search criteria.
17	Clear	Click to clear the text entered in **Search** box.

The fields in Use Case Grid

Accessing actor from use case

Click on the name of actor in the **Primary Actors or Supporting Actors** cell of the use case that you want to access its related actor. Select the way to access the actor.

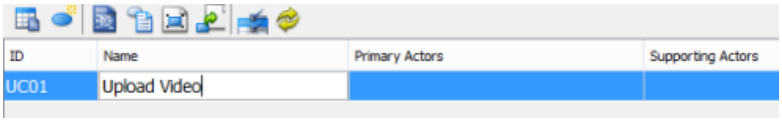


Access actor from use case

Creating use case in Use Case Grid

To create a use case in **Use Case Grid**:

1. Click on **New Use Case** above the **Use Case Grid**.
2. Enter the name of use case.



Creating use case in Use Case Grid

3. Press **Enter** to confirm editing.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Class diagram

A class diagram models the classes of a system/application going to develop. In this chapter, you will learn how to create class diagram.

Creating class diagram

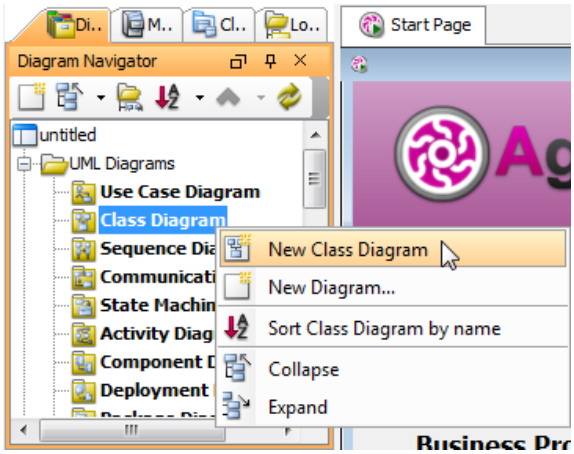
Create and draw class diagram. You will learn how to create class, attribute, operation and other common class diagram constructs.

Creating class diagram

A class diagram models the classes of a system/application going to develop. Since it provides detailed information about the properties and interfaces of the classes, it can be considered as the main model and regard the other diagrams as supplementary models.

Creating class diagram

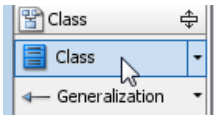
- Click on **UML** on toolbar and select **Class Diagram** from the drop down menu **.**
- Right click on **Class Diagram** in **Diagram Navigator** and select **New Class Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Class Diagram** from the main menu.



Create new class diagram

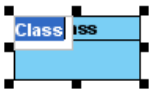
Creating class

To create class, click **Class** on the diagram toolbar and then click on the diagram.



Create class

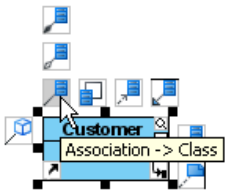
A class will be created.



Class created

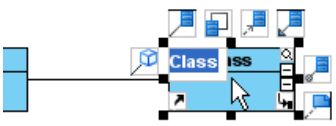
Creating association

To create association from class, click the **Association** -> **Class** resource beside it and drag.



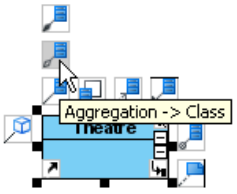
Create association

Drag to empty space of the diagram to create a new class, or drag to an existing class to connect to it. Release the mouse button to create the association.



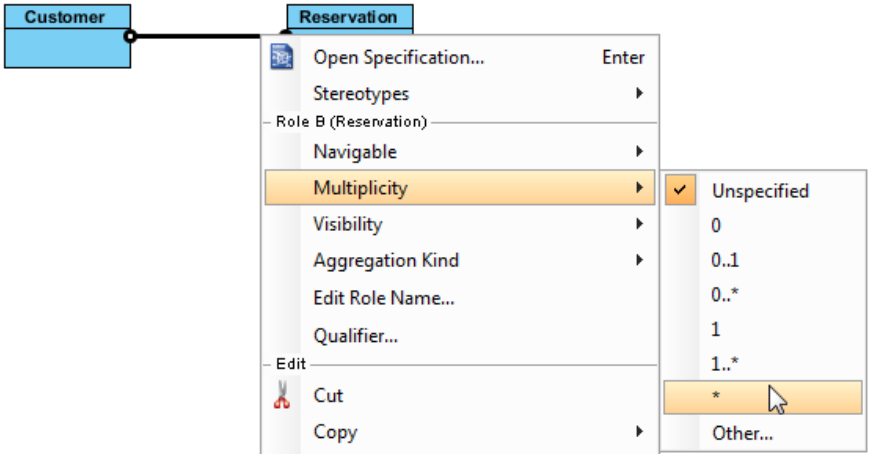
Association created

To create aggregation, use the **Aggregation** -> **Class** resource instead.



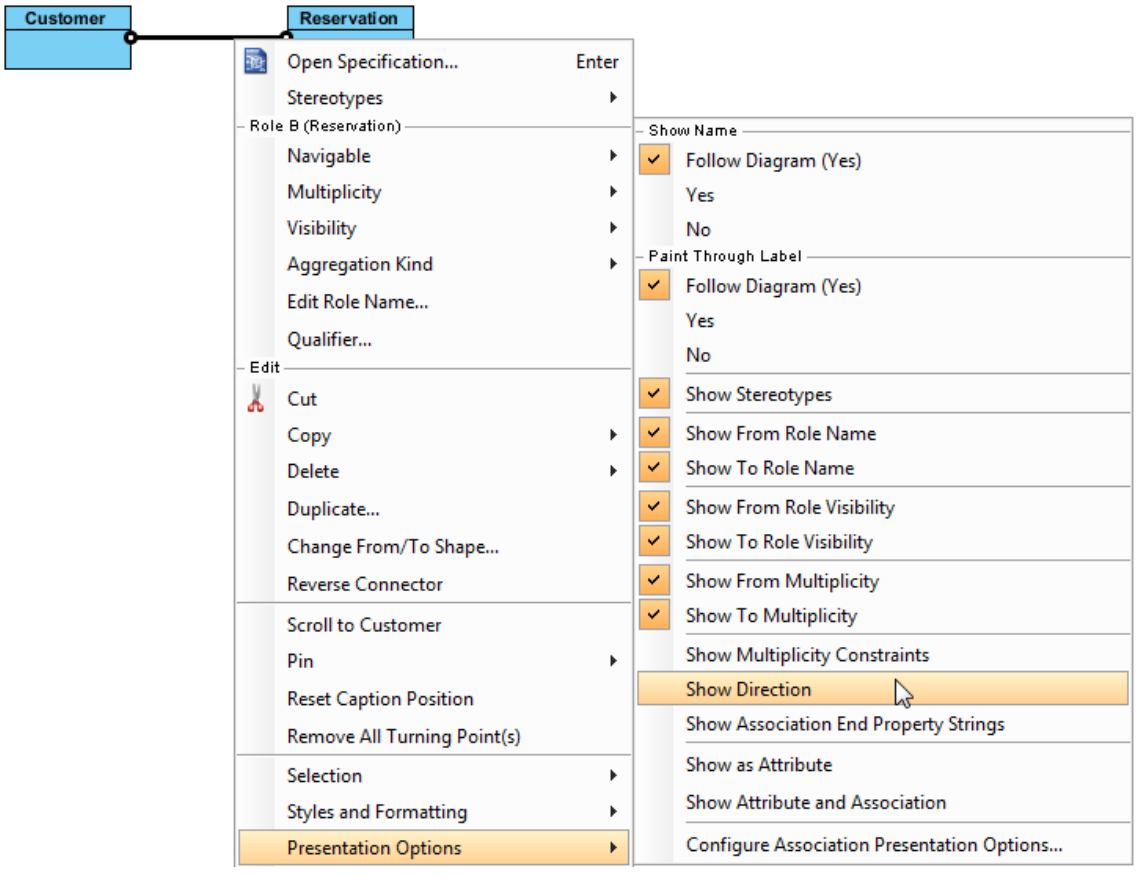
Create aggregation

To edit multiplicity of an association end, right-click near the association end, select **Multiplicity** from the popup menu and then select a multiplicity.



Edit multiplicity

To show the direction of an association, right click on it and select **Presentation Options** > **Show Direction** from the pop-up menu.



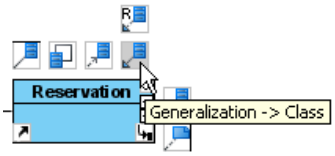
Show direction

The direction arrow is shown beside the association.

Direction shown

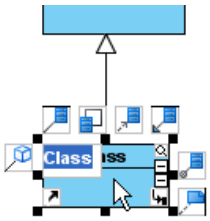
Creating generalization

To create generalization from class, press its **Generalization** -> **Class** resource icon.



Create generalization

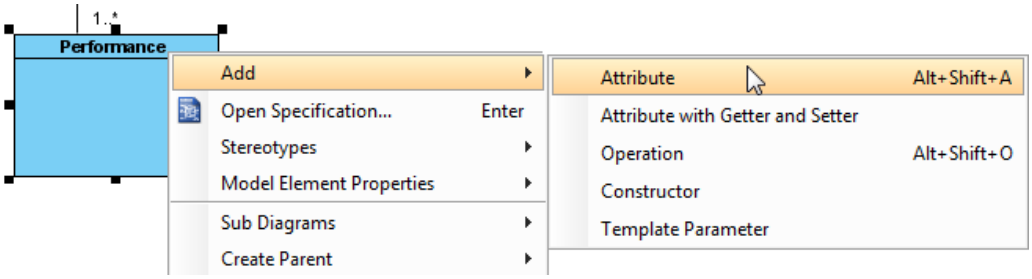
Drag to empty space of the diagram to create a new class, or drag to an existing class to connect to it. Release the mouse button to create the generalization.



Generalization created

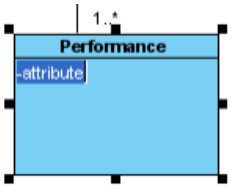
Creating attribute

To create attribute, right-click the class and select **Add** > **Attribute** from the popup menu.



Create attribute

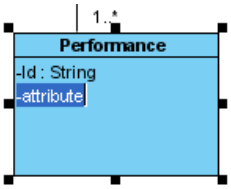
An attribute is created.



Attribute created

Creating attribute with enter key

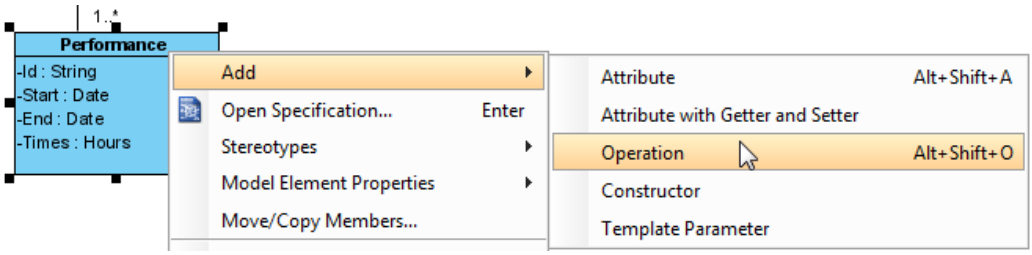
After creating an attribute, press the Enter key, another attribute will be created. This method lets you create multiple attributes quickly and easily.



Create attribute with Enter key

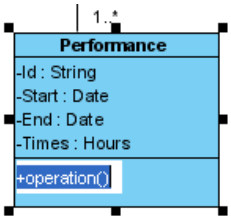
Creating operation

To create operation, right-click the class and select **Add** > **Operation** from the popup menu.



Create operation

An operation is created.

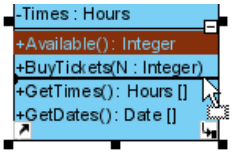


Operation created

Similar to creating attribute, you can press the Enter key to create multiple operations continuously.

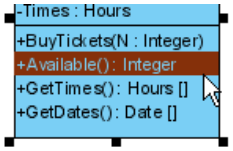
Drag-and-Drop reordering, copying and moving of class members

To reorder a class member, select it and drag within the compartment, you will see a thick black line appears indicating where the class member will be placed.



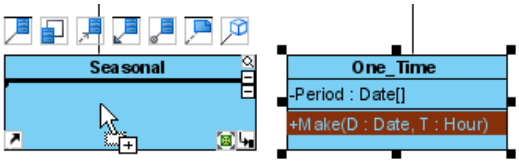
Reorder class member

Release the mouse button, the class member will be reordered.



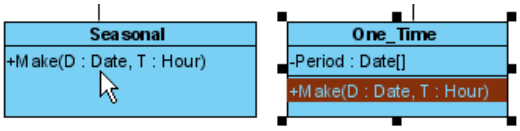
Class member reordered

To copy a class member, select it and drag to the target class while keep pressing the Ctrl key, you will see a thick black line appears indicating where the class member will be placed. A plus sign is shown beside the mouse cursor indicating this is a copy action.



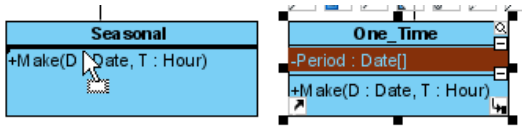
Copy class member

Release the mouse button, the class member will be copied.



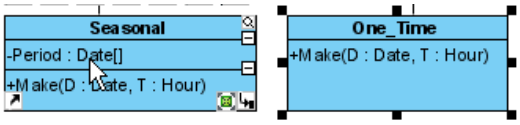
Class member copied

To move a class member, select it and drag to the target class, you will see a thick black line appears indicating where the class member will be placed. Unlike copy, do not press the Ctrl key when drag, the mouse cursor without the plus sign indicates this is a move action.



Move class member

Release the mouse button, the class member will be moved.

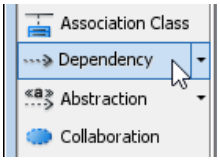


Class member moved

Relating class members

Relationships such as dependency and generic connectors can be added between attribute and operation of classes. To do this:

1. Select the type of relationship to be created, under the diagram toolbar.

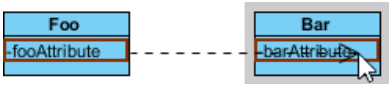


Selecting Dependency

2. Move the mosue pointer over the source member.

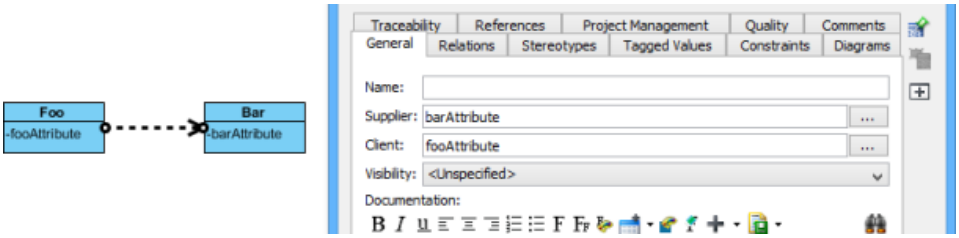
Create a relationship from a class member

3. Press on it and hold the mouse button.
4. Drag to the target member.



Release mouse button on target class member

5. Release the mouse button to create the connector. While it looks like the connector is connecting the classes but not the members, if you check its specification you can see that the connector is indeed connecting the members.



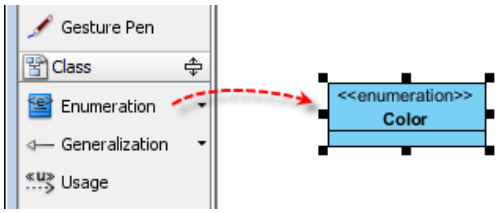
Relationship created

Creating enumeration and adding enumeration literal

An enumeration is a special data type, which consists of a pre-defined set of values, known as enumeration literals. Here are some of the common examples:

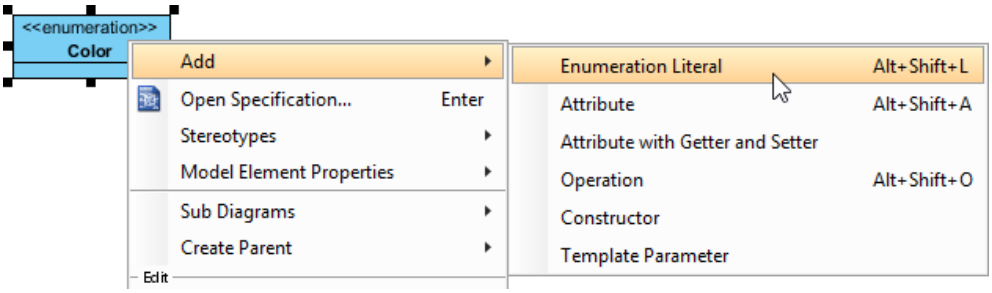
- Color (RED, GREEN, BLUE)
- Orientation (NORTH, SOUTH, EAST, WEST)
- Switch (ON, OFF)

To create an enumeration, select **Enumeration** from the diagram toolbar and click on the diagram to create one.



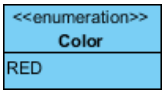
Create an enumeration

To add an enumeration literal, right click on the enumeration class and select **Add > Enumeration Literal** from the popup menu.



Add an enumeration literal

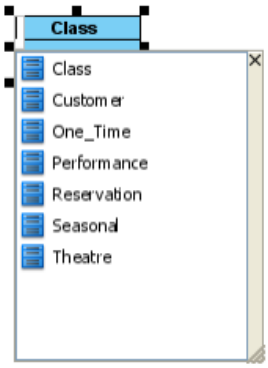
Then, enter the name of the literal and confirm editing.



Enumeration literal entered

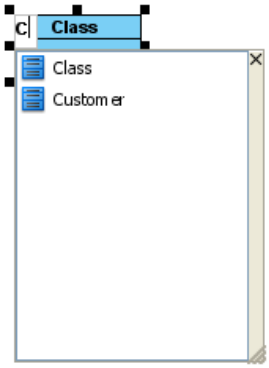
Model name completion for class

The model name completion feature enables quick creation of multiple views for the same class model. When create or rename class, the list of classes shows.



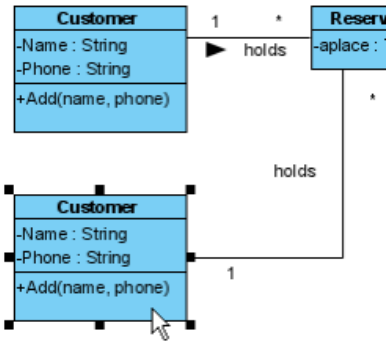
Model name completion

Type text to filter classes in the list.



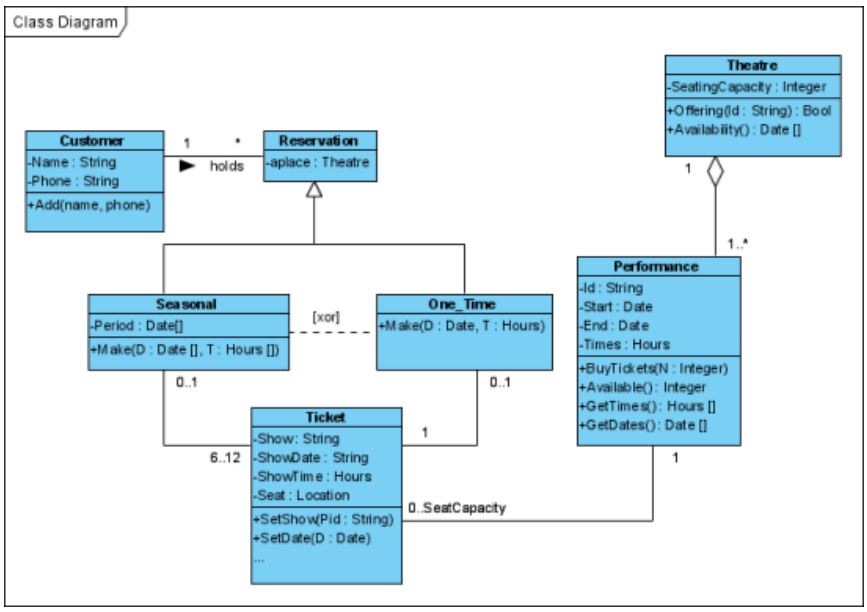
List filtered by typed text

Press up or down key to select class in the list, press Enter to confirm. Upon selecting an existing class, all class members and relationships are shown immediately.



Multiple views of the same model

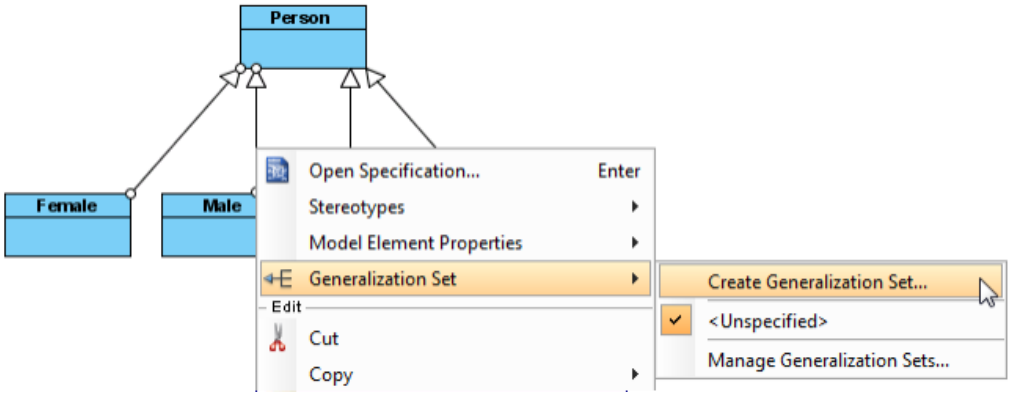
Continue to complete the diagram.



Completed diagram

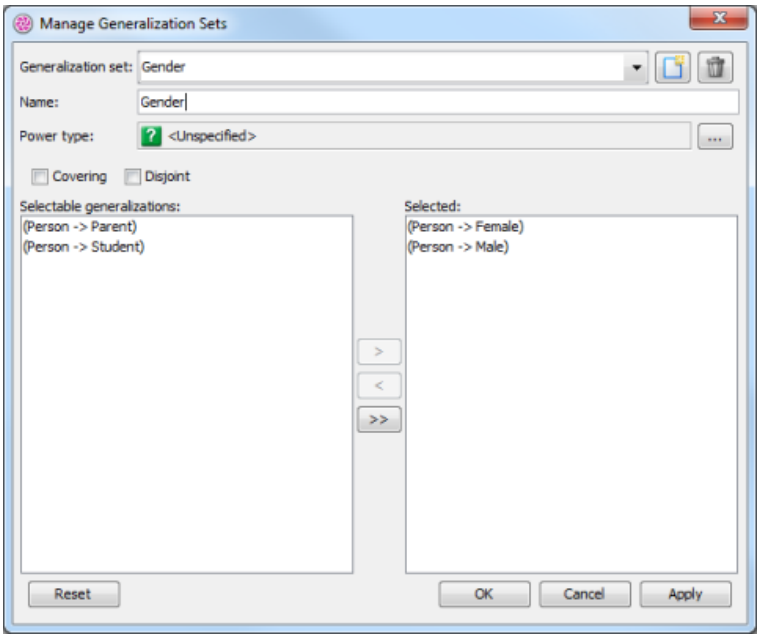
Generalization set

A generalization set defines a particular set of generalization relationships that describe the way in which a general classifier (or superclass) may be divided using specific subtypes. To define a generalization set, select the generalizations to include, right click and select **Generalization set > Create Generalization Set...** from the pop-up menu.



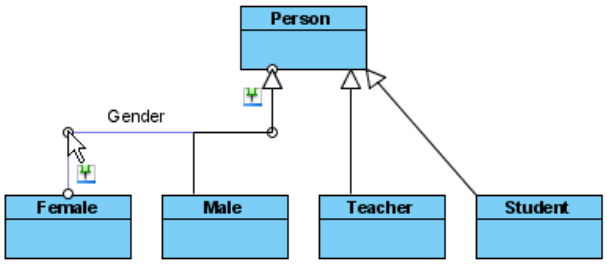
Create a generalization set

Name the set in the Manage Generalization Sets dialog box, and confirm by pressing OK.



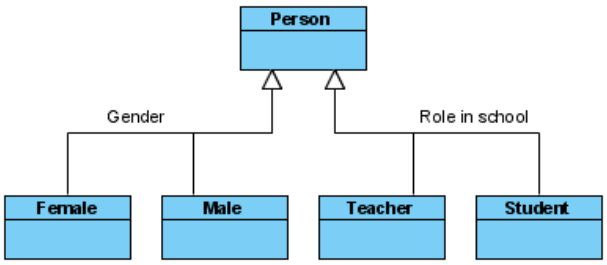
Name the generalization set

The selected generalizations are grouped. Adjust the connector to make the diagram tidy.



Adjust connector

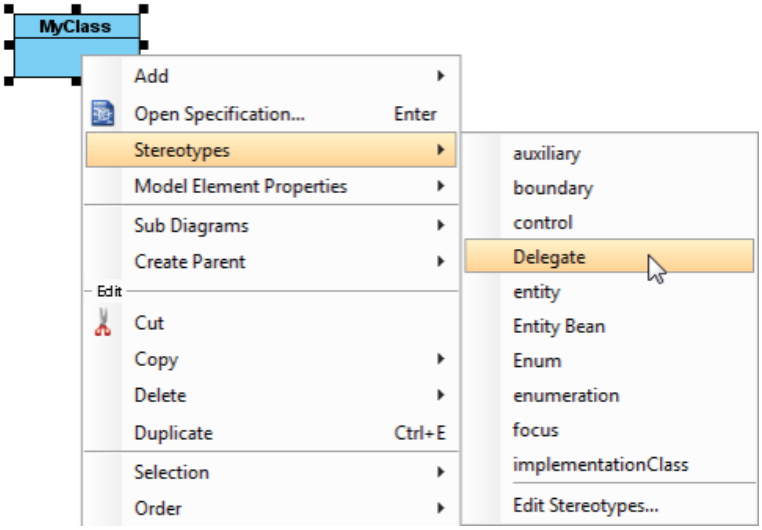
Repeat the steps for other generalizations.



Generalization sets defined

Defining delegate method for class

When project's programming language is set to be Visual Basic or C#, it is possible to define delegate method for classes. To define delete method, right click on the class and select **Stereotypes > Delegate** from the popup menu.



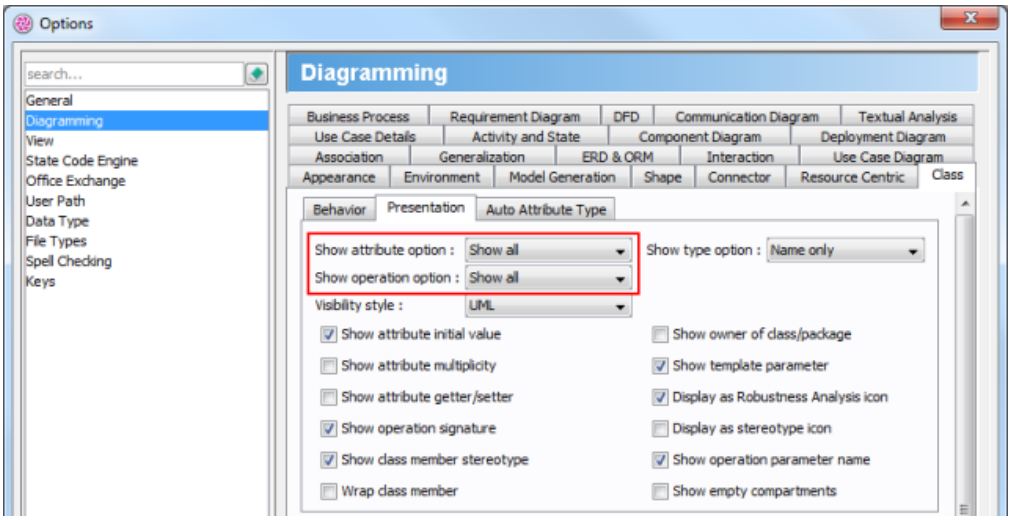
*Stereotype class as **Delegate***

Hiding (and showing) attributes and operations

Per workspace

This applies to new classes that will be created in a project opened in specific workspace. To change the setting:

1. Select **Tools > Project Options** from the main menu to open the **Options** dialog box.
2. Click **Diagramming** on the list.
3. Open the **Class** tab.
4. Click **Presentation** tab.
5. Change the settings for **Show attribute option** and/or **Show operation option**.

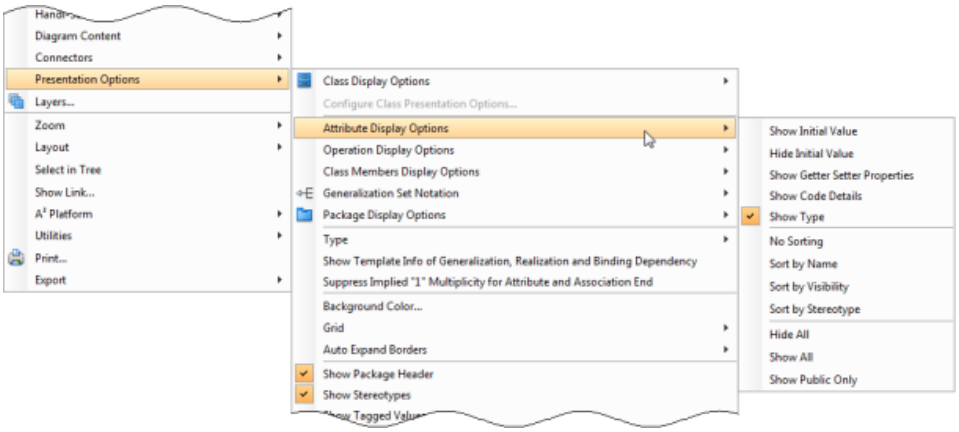


Show or hide operations

Per diagram

This applies to classes in specific diagram. To change the setting:

1. Right click on the class diagram to set the option.
2. Select **Presentation Options > Attribute Display Options / Operation Display Options** from the popup menu.
3. Select Hide All / Show All / Show Public Only.

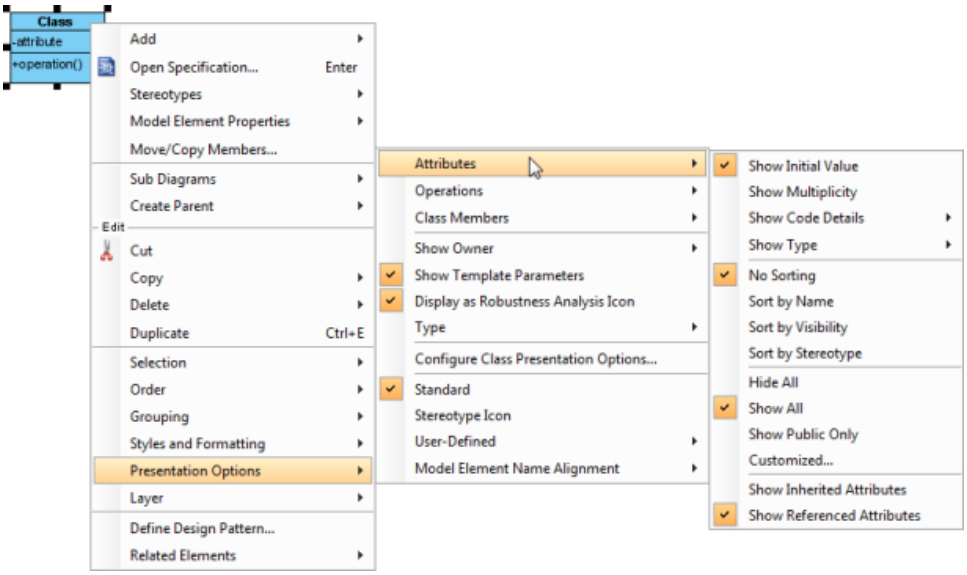


Change the attribute's presentation options for classes in diagram

Per class

This applies to specific class. To change the setting:

1. Right click on the class to set the option.
2. Select **Presentation Options > Attributes / Operations** from the popup menu.
3. Select Hide All / Show All / Show Public Only.

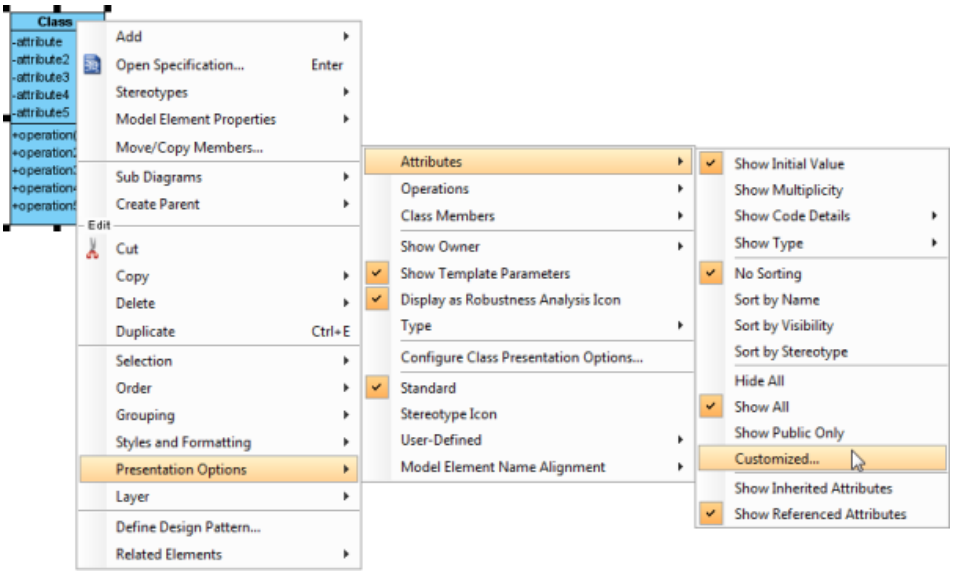


Change the attribute's presentation options for a class

For specific attribute/operation

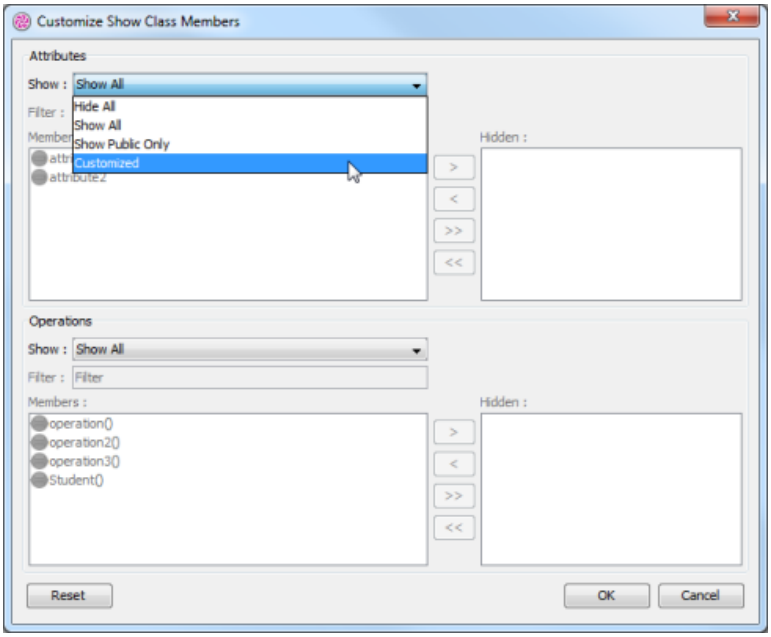
Instead of showing or hiding all members or public members, you may show/hide specific class member per class. To do this:

1. Right click on the class to set the option.
2. Select **Presentation Options > Attributes / Operations > Customized...** from the pop-up menu.



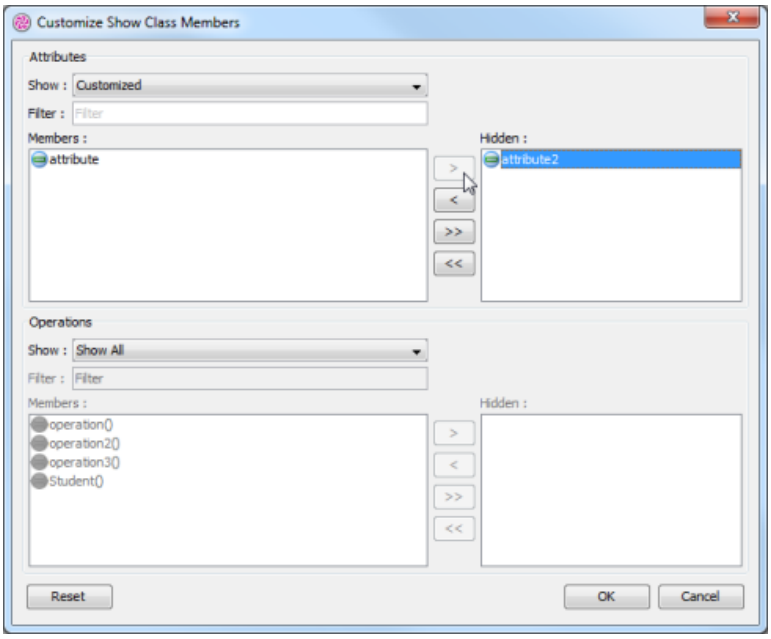
Show or hide specific class member

3. Select **Customized** under the drop-down menu of **Show**.



Select Customized in dialog box

4. Select the member(s) to hide and click **>** to hide them.



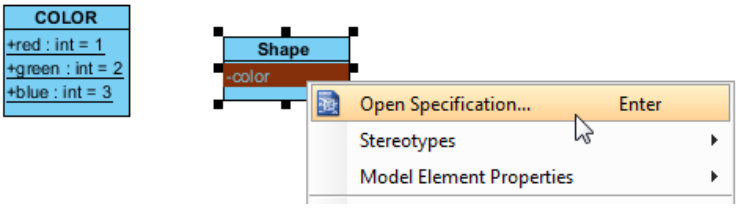
Select attributes to hide

5. Click **OK** button to confirm.

Setting initial (default) value for attribute

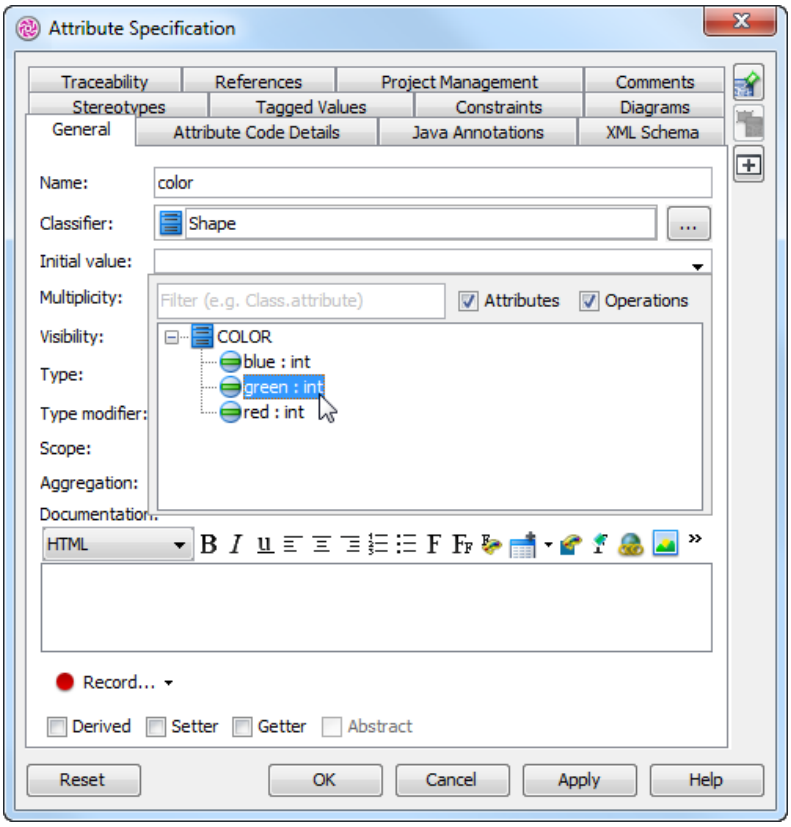
Initial value can be set to an attribute, indicating the default value of the attribute when the owning object is instantiated. You can give a text value for initial value, or select an attribute of another class. To set initial value to an attribute:

1. Open the specification dialog box of attribute by right clicking on the attribute and selecting **Open Specification...** from the popup menu.



Opening the attribute specification

2. In the **General** page of the specification dialog box, enter the initial value in initial value field if it is a text value, or pop up the drop-down menu to select a public and static field of any class to be the value.



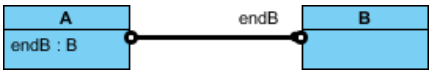
Selecting an initial value

NOTE: In order to select the attribute of another class to be the default value, make sure the attribute you want to select is static (i.e. set to be in classifier scope) and is public (so that other classes can access).

3. Click **OK** to confirm.

Setting the ownership of association end

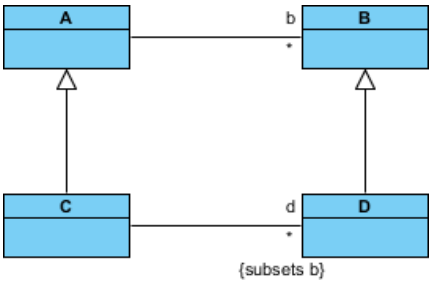
Ownership of association ends by an associated class may be indicated by a small dot. To set the ownership, right click at the association end where you want to set ownership, select **Owned by** in the popup menu, then select either the association or the class at the opposite end. By selecting class, the small dot will be shown.



Association end with ownership set

Subsetting on association end

Take a look at the sample below. The subset on d indicates that the collection d, which is an instance of class C, is a subset of the collection b, instance of class A.



Subsetting on association end

To define a subset on an association end:

1. Right click on the association (where the subset end exist) and select **Open Specification...** from the popup menu.
2. In the **General** tab, locate the association end where you want to define a subset. Click on **...** for the Role property of the association end.
3. In the **Association End Specification**, open the **Subsetted Association Ends** tab.
4. From the list on the left hand side, click on the role you want to define subset for. Click **>** to select it. If you do not see any role listing there, make sure your model respect the pattern similar to the class diagram above - The class of both association ends are subclasses, and there is an association connecting their superclasses.

5. Click **OK** to confirm and close the **Association End Specification** window.
6. Click **OK** to confirm and close the **Association Specification** window.
7. Right click on the association end and select **Presentation Options > Show Association End Property Strings** from the popup menu to show the subset.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Sequence diagram

A sequence diagram is used primarily to show the interactions between objects that are represented as lifelines in a sequential order. In this chapter, you will learn how to draw a sequence diagram.

Creating sequence diagram

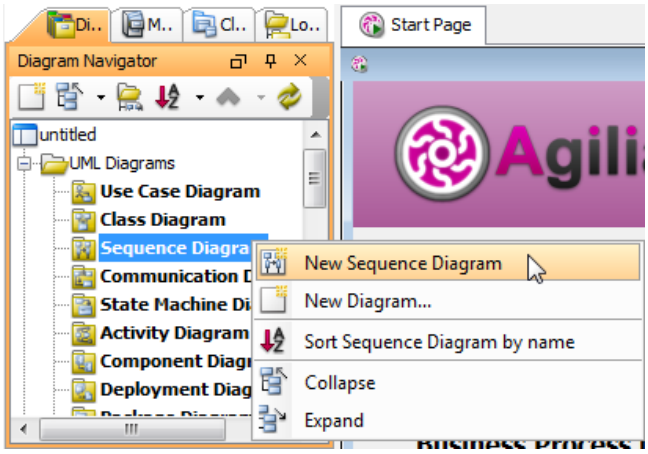
Teaches you how to create sequence diagram through the diagram and through the editor at the bottom of diagram.

Creating sequence diagram

A sequence diagram is used primarily to show the interactions between objects that are represented as lifelines in a sequential order.

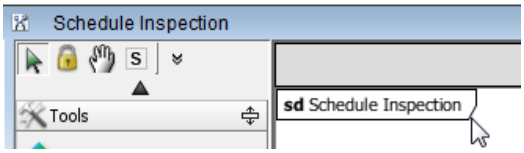
Creating a sequence diagram

- Click on **UML** on toolbar and select **Sequence Diagram** from the drop down menu **.**
- Right click on **Sequence Diagram** in **Diagram Navigator** and select **New Sequence Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Sequence Diagram** from the main menu.



Create sequence diagram

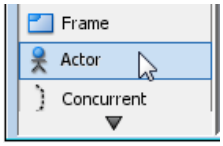
Enter name for the newly created sequence diagram in the text field of pop-up box on the top left corner.



Enter name for sequence diagram

Creating actor

To create actor, click **Actor** on the diagram toolbar and then click on the diagram.

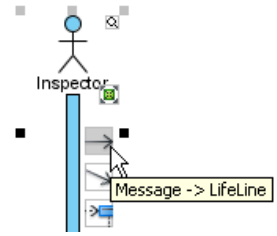


Create actor

Creating lifeline

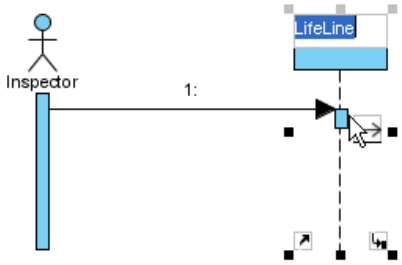
To create lifeline, you can click **LifeLine** on the diagram toolbar and then click on the diagram.

Alternatively, a much quicker and more efficient way is to use the resource-centric interface. Click on the **Message** -> **LifeLine** resource beside an actor/lifeline and drag.



Create lifeline

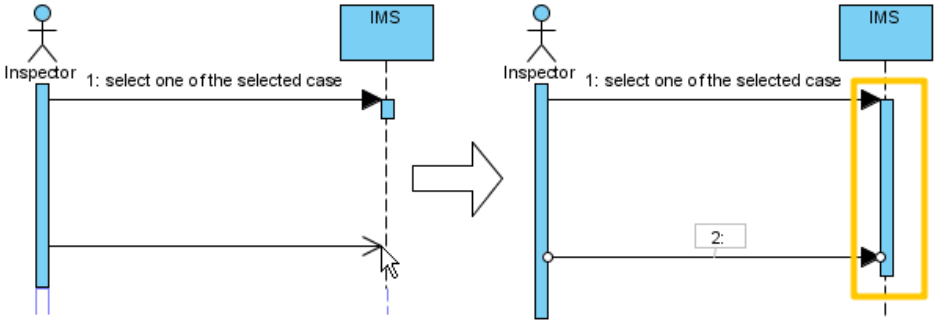
Move the mouse to empty space of the diagram and then release the mouse button. A new lifeline will be created and connected to the actor/lifeline with a message.



Lifeline and message created

Auto extending activation

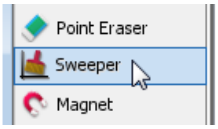
When create message between lifelines/actors, activation will be automatically extended.



Auto extend activation

Using sweeper and magnet to manage sequence diagram

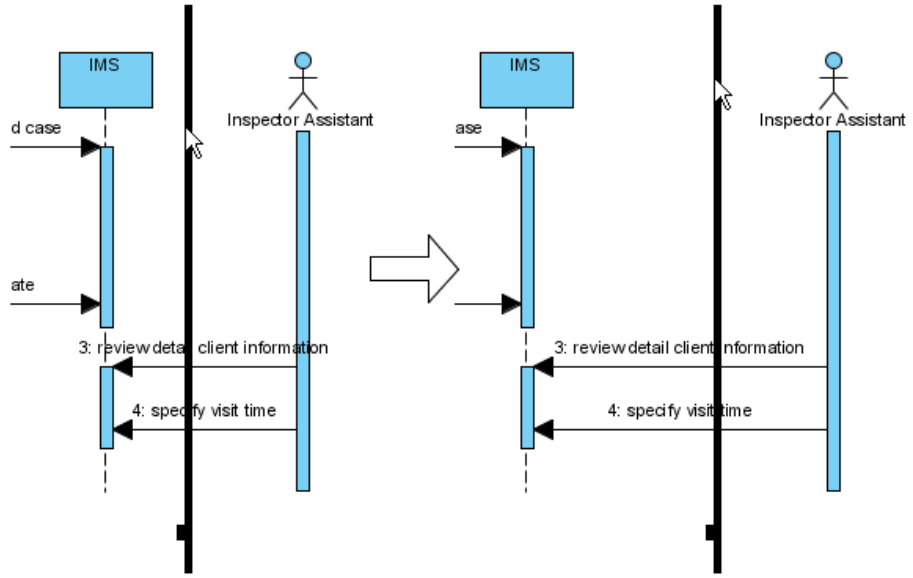
Sweeper helps you to move shapes aside to make room for new shapes or connectors. To use sweeper, click **Sweeper** on the diagram toolbar (under the **Tools** category).



*Click **Sweeper***

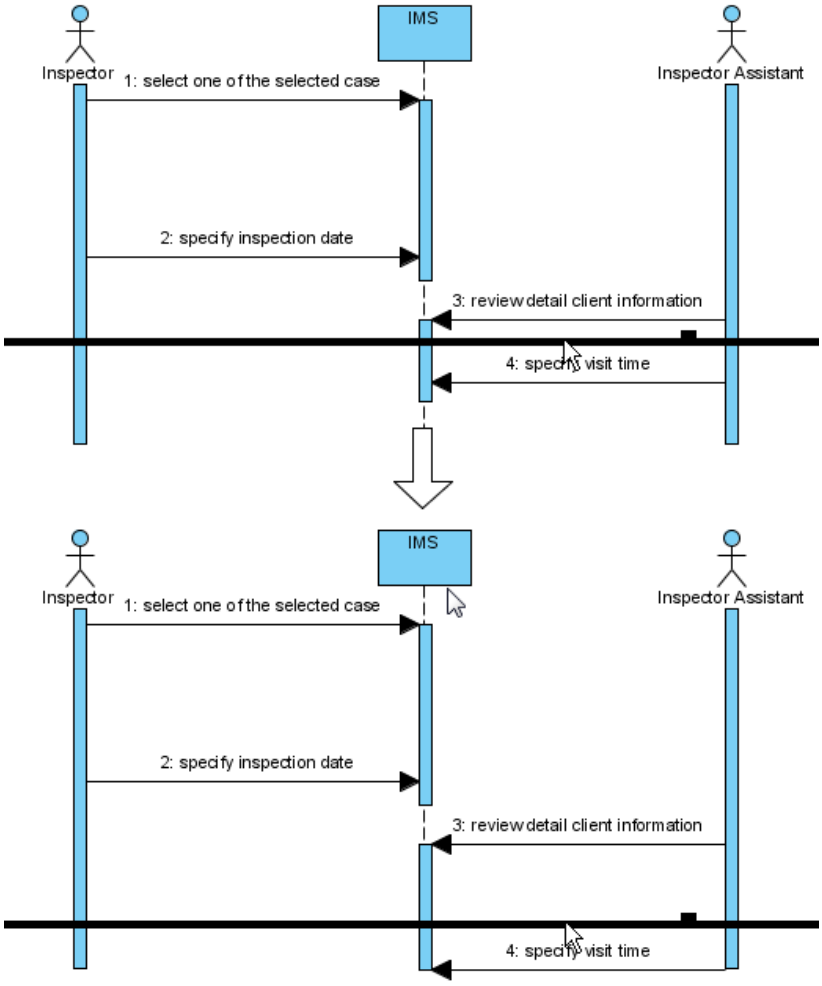
Click on empty space of the diagram and drag towards top, right, bottom or left. Shapes affected will be swept to the direction you dragged.

The picture below shows the actor *Inspector Assistant* is being swept towards right, thus new room is made for new lifelines.



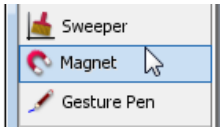
Sweep towards right

The picture below shows the message *specify visit time* is being swept downwards, thus new room is made for new messages.



Sweep downwards

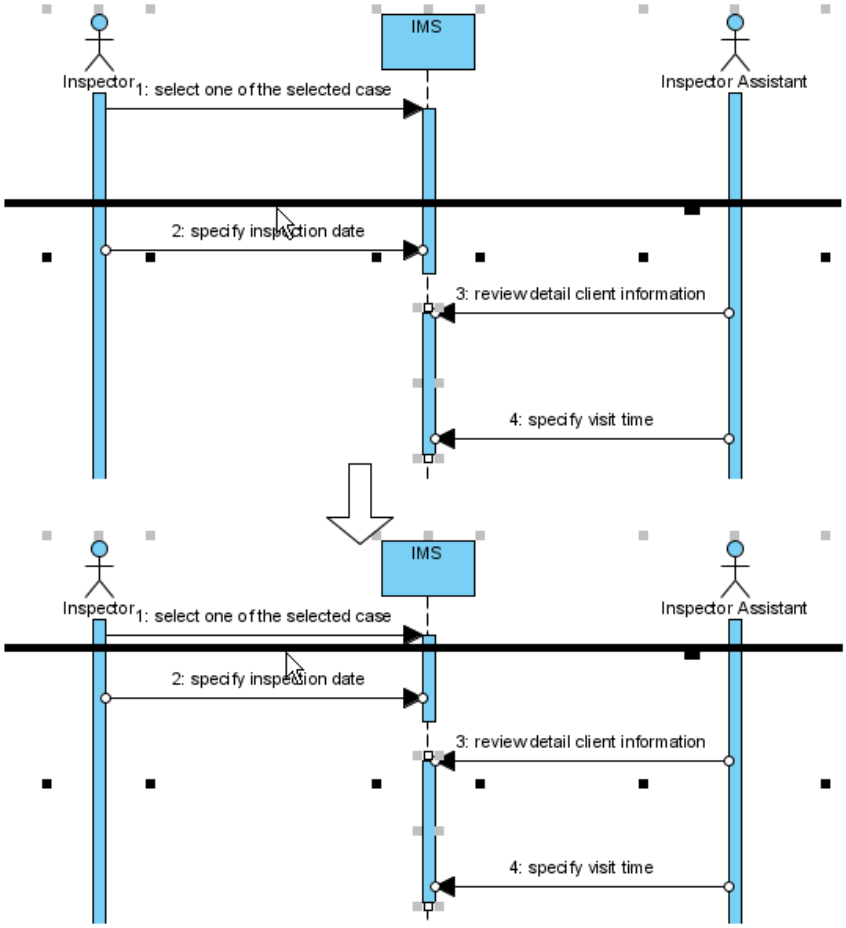
You can also use magnet to pull shapes together. To use magnet, click **Magnet** on the diagram toolbar (under the **Tools** category).



*Click **Magnet***

Click on empty space of the diagram and drag towards top, right, bottom or left. Shapes affected will be pulled to the direction you dragged.

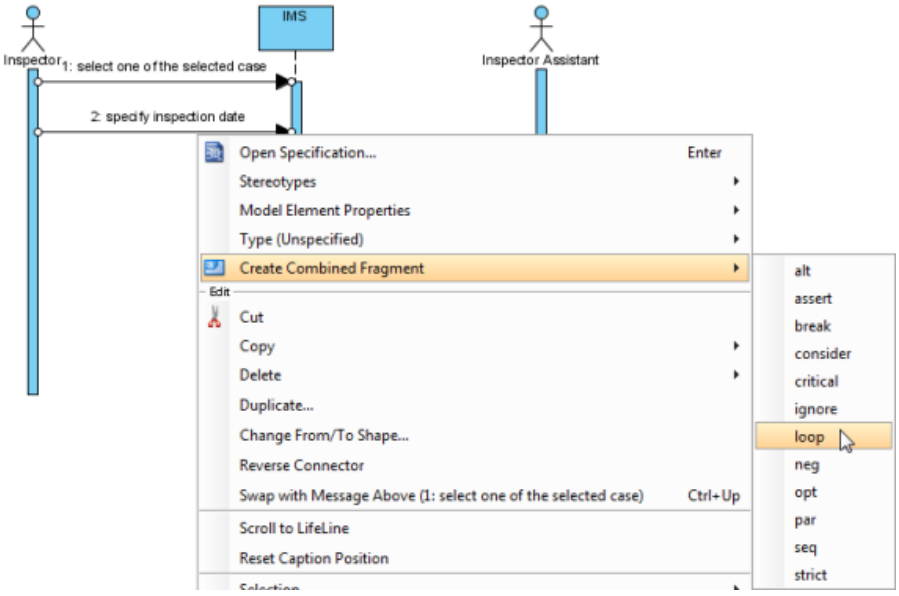
The picture below shows when drag the magnet upwards, shapes below dragged position are pulled upwards.



Pull shapes upwards using magnet

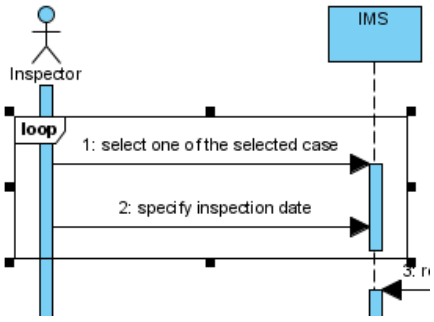
Creating combined fragment for messages

To create combined fragment to cover messages, select the messages, right-click on the selection and select **Create Combined Fragment**, and then select a combined fragment type (e.g. loop) from the popup menu.



Create combined fragment for messages

A combined fragment of selected type will be created to cover the messages.

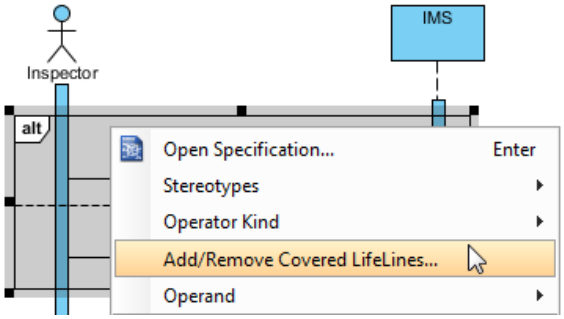


Combined fragment created

Adding/removing covered lifelines

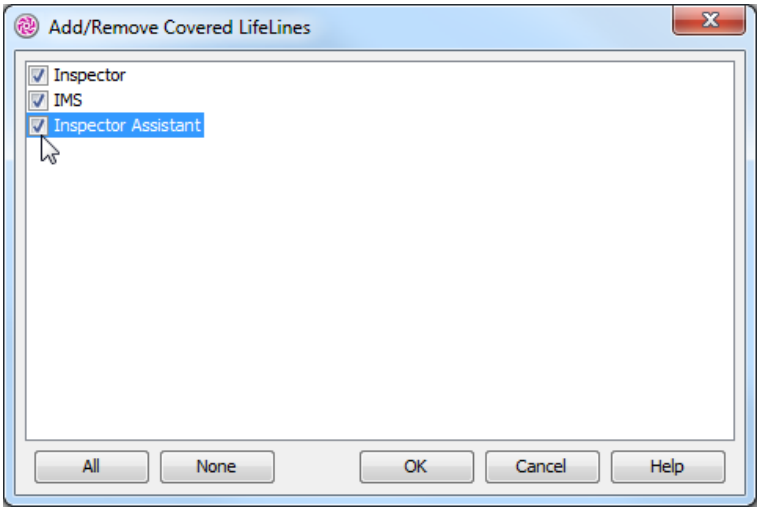
After you've created a combined fragment on the messages, you can add or remove the covered lifelines.

1. Move the mouse over the combined fragment and select **Add/Remove Covered Lifeline...** from the pop-up menu.



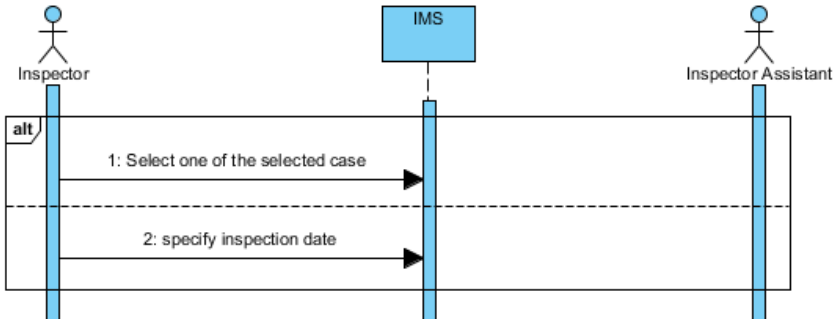
Add/Remove covered lifelines

2. In the **Add/Remove Covered Lifelines** dialog box, check the lifeline(s) you want to cover or uncheck the lifeline(s) you don't want to cover. Click **OK** button.



*Check **Inspector Assistant***

As a result, the area of covered lifelines is extended or narrowed down according to your selection.

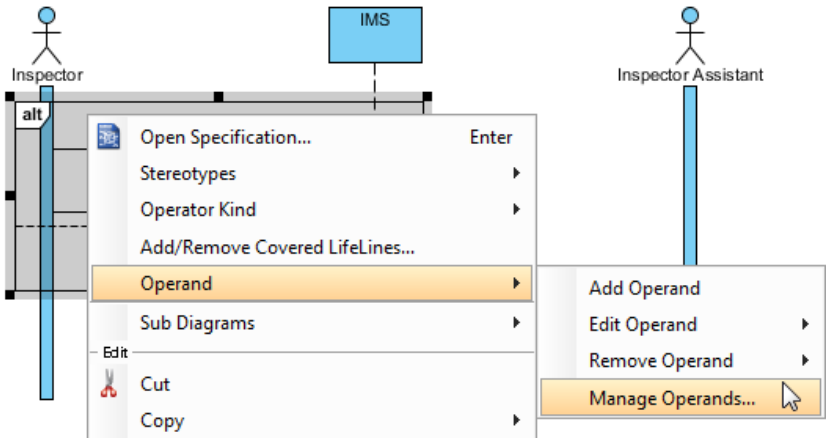


The area of covered lifelines is extended

Managing Operands

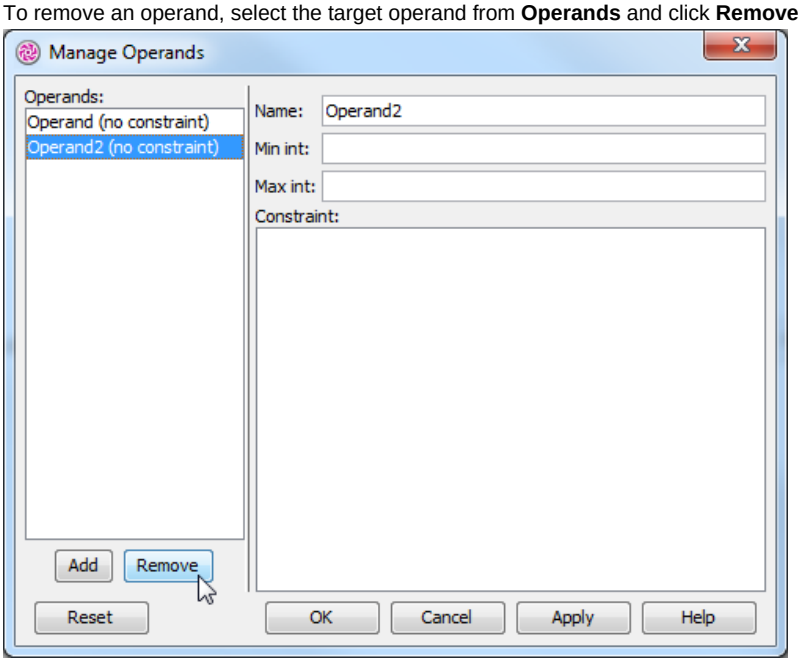
After you've created a combined fragment on the messages, you can also add or remove operand(s).

1. Move the mouse over the combined fragment and select **Operand > Manage Operands...** from the pop-up menu.



Manage operands

2. To remove an operand, select the target operand from **Operands** and click **Remove** button. Click **OK** button.

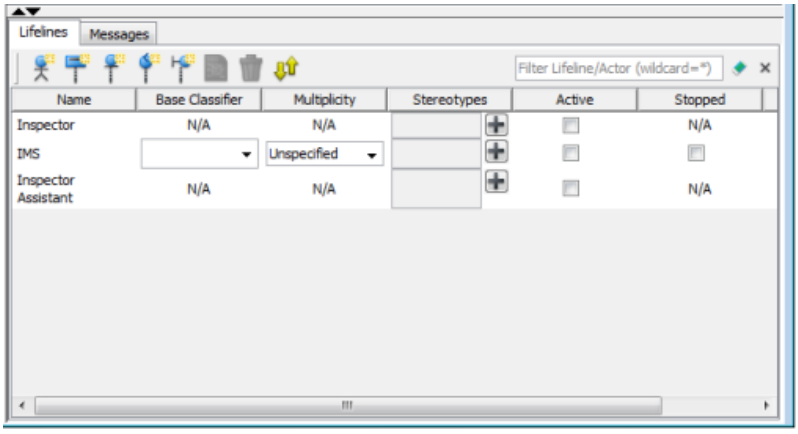


Remove Operand2

Otherwise, click **Add** button to add a new operand and then name it. Click **OK** button.

Developing sequence diagram with quick editor or keyboard shortcuts

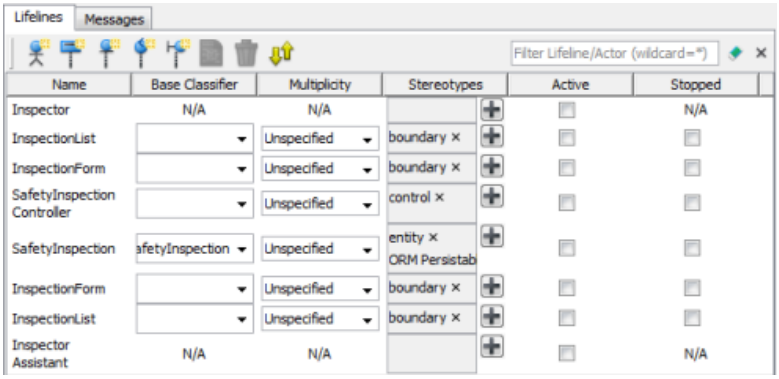
In sequence diagram, an editor appears at the bottom of diagram by default, which enables you to construct sequence diagram with the buttons there. The shortcut keys assigned to the buttons provide a way to construct diagram through keyboard. Besides constructing diagram, you can also access diagram elements listing in the editor.



The quick editor

Editing lifelines

There are two panes, **Lifelines** and **Messages**. The **Lifelines** pane enables you to create different kinds of actors and lifelines.



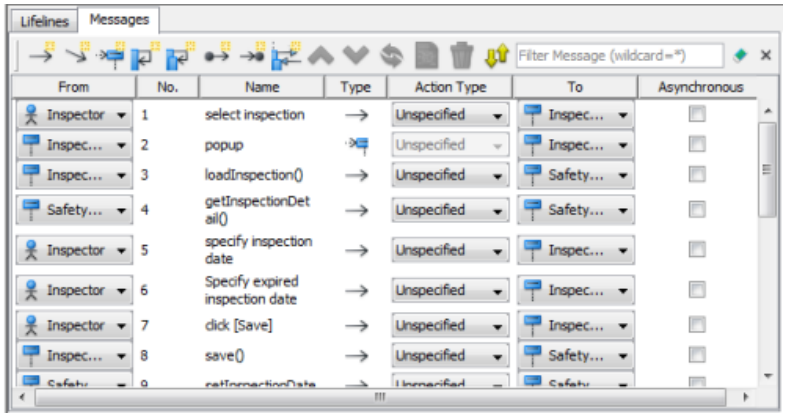
Lifelines pane in quick editor

Button	Shortcut	Description
	Alt-Shift-A	To create an actor.
	Alt-Shift-L	To create a general lifeline.
	Alt-Shift-E	To create an <<entity>> lifeline.
	Alt-Shift-C	To create a <<control>> lifeline.
	Alt-Shift-B	To create a <<boundary>> lifeline.
	Alt-Shift-O	To open the specification of the element chosen in quick editor.
	Ctrl-Del	To delete the element chosen in quick editor.
	Ctrl-L	To link with the diagram, which cause the diagram element to be selected when selecting an element in editor, and vice versa.

Buttons in Lifelines pane

Editing messages

The **Messages** pane enables you to connect lifelines with various kinds of messages.



Messages pane in quick editor

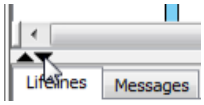
Button	Shortcut	Description
	Alt-Shift-M	To create a message that connects actors/lifelines in diagram.
	Alt-Shift-D	To create a duration message that connects actors/lifelines in diagram.
	Alt-Shift-C	To create a create message that connects actors/lifelines in diagram.
	Alt-Shift-S	To create a self message on an actor/lifeline in diagram.
	Alt-Shift-R	To create a recursive message on an actor/lifeline in diagram.

	Alt-Shift-F	To create a found message that connects to an actor/lifeline.
	Alt-Shift-L	To create a lost message from an actor/lifeline.
	Alt-Shift-E	To create a reentrant message that connects actors/lifelines in diagram.
	Ctrl-Shift-Up	To swap the chosen message with the one above.
	Ctrl-Shift-Down	To swap the chosen message with the one below.
	Ctrl-R	To revert the direction of chosen message.
	Alt-Shift-O	To open the specification of the message chosen in quick editor.
	Ctrl-Del	To delete the message chosen in quick editor.
	Ctrl-L	To link with the diagram, which cause the message to be selected when selecting a message in editor, and vice versa.

Buttons in Messages pane

Expanding and collapsing the editor

To hide the editor, click on the down arrow button that appears at the bar on top of the quick editor. To expand, click on the up arrow button.



Collapse the quick editor

Setting different ways of numbering sequence messages

You are able to set the way of numbering sequence messages either on diagram base or frame base.

Diagram-based sequence message

Right click on the diagram's background, select **Sequence Number** and then either **Single Level** or **Nested Level** from the pop-up menu.

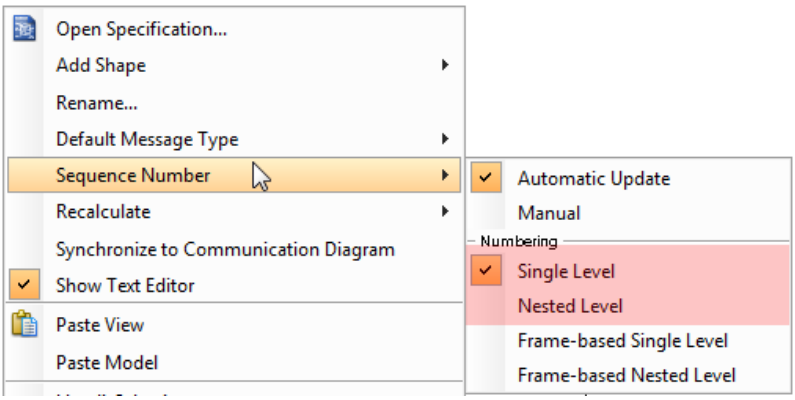
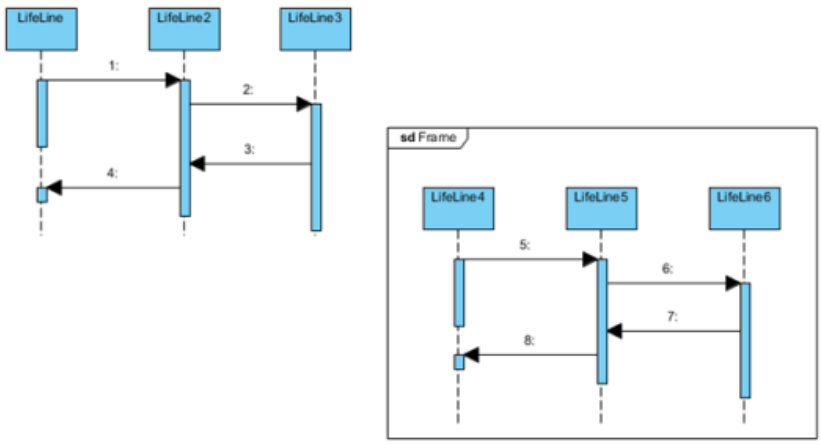


Diagram-based pop-up menu

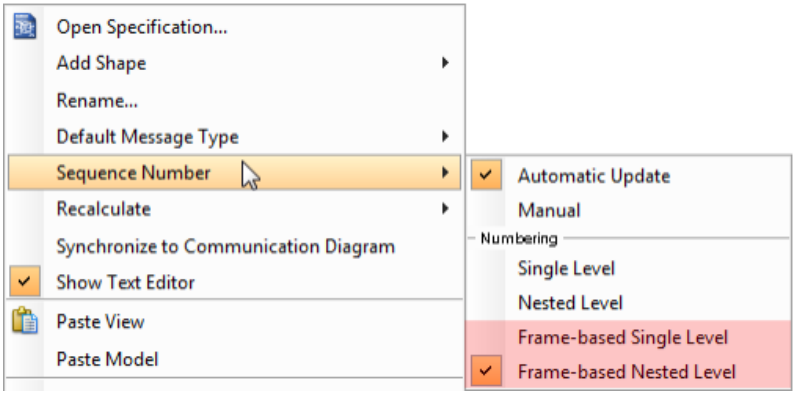
If you choose **Single Level**, all sequence messages will be ordered with integers on diagram base. On the other hand, if you choose **Nested Level**, all sequence messages will be ordered with decimal place on diagram base.



Single level

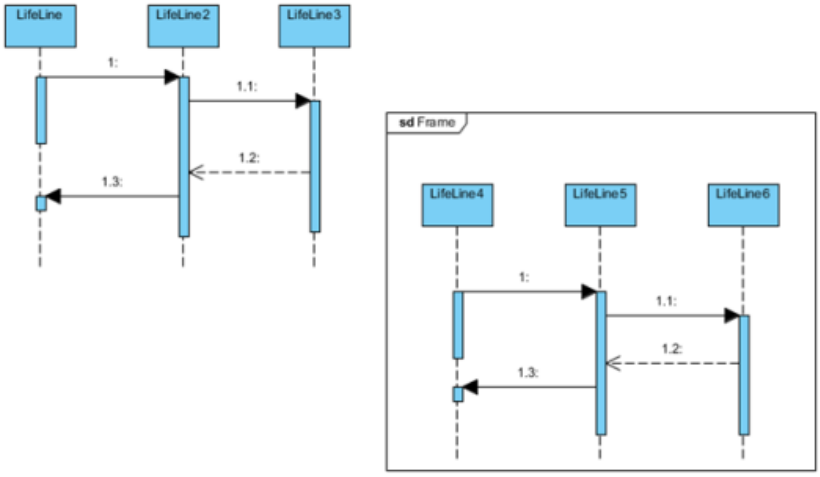
Frame-based sequence message

Right click on the diagram's background, select **Sequence Number** and then either **Frame-based Single Level** or **Frame-based Nested Level** from the pop-up menu.



Frame-based pop-up menu

When you set the way of numbering sequence messages on frame base, the sequence messages in frame will restart numbering sequence message since they are independent and ignore the way of numbering sequence message outside the frame.



Frame-based nested level

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Communication diagram

Communication diagram was designed for presenting the interactions between objects (i.e. lifelines). You will learn how to create communication diagram in this chapter.

Creating communication diagram

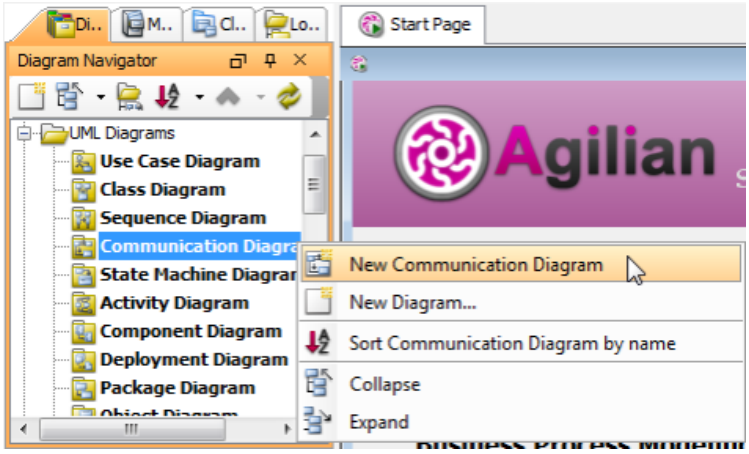
Learn how to create and draw communication diagram.

Creating communication diagram

Communication diagram was designed for presenting the interactions between objects (i.e. lifelines). It emphasizes the structural organization of the objects' send and receive messages.

Creating communication diagram

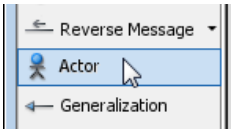
- Click on **UML** on toolbar and select **Communication Diagram** from the drop down menu **.**
- Right click on **Communication Diagram** in **Diagram Navigator** and select **New Communication Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Communication Diagram** from the main menu.



Create communication diagram

Creating actor

To create actor, click **Actor** on the diagram toolbar and then click on the diagram.

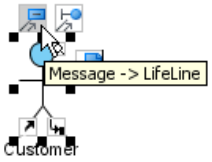


Create actor

Creating lifeline

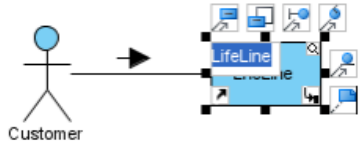
To create lifeline, you can click **LifeLine** on the diagram toolbar and then click on the diagram.

Alternatively, a much quicker and more efficient way is to use the resource-centric interface. Click on actor/lifeline's **Message** -> **LifeLine** resource and drag.



Create lifeline

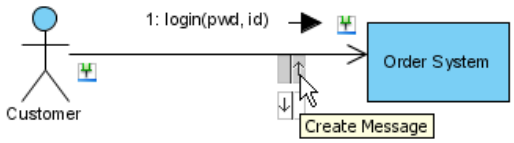
Move the mouse to your preferred space of the diagram and then release the mouse button. A new lifeline will be created and connected to the actor/ lifeline with a link (the line) and a message (the arrow).



Lifeline, link and message created

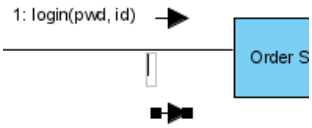
Creating message on link

To create message on link, move the mouse over the target link and then click its **Create Message** resource icon.



Create message on link

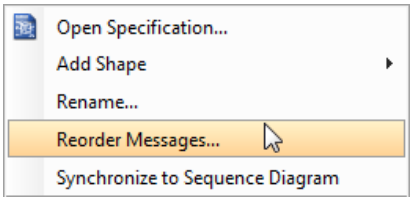
A message will be created on the link.



Message created on link

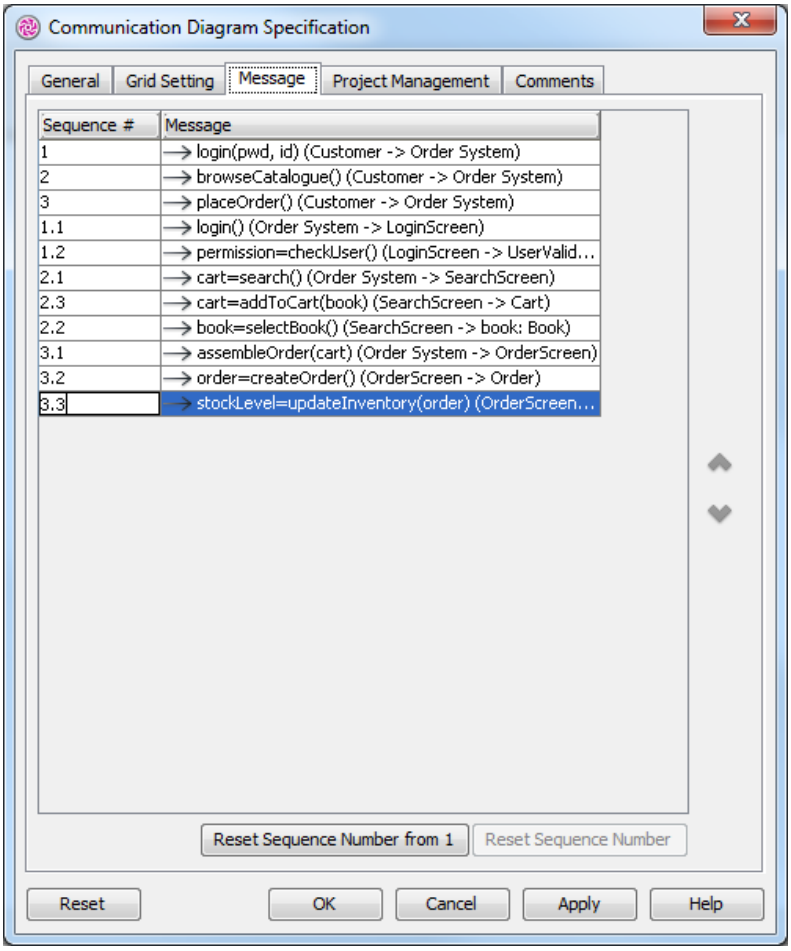
Editing sequence number of messages

To edit sequence number of messages, for example, to show certain messages are in nested level of interaction, right click the diagram's background and select **Reorder Messages ...** from the popup menu.



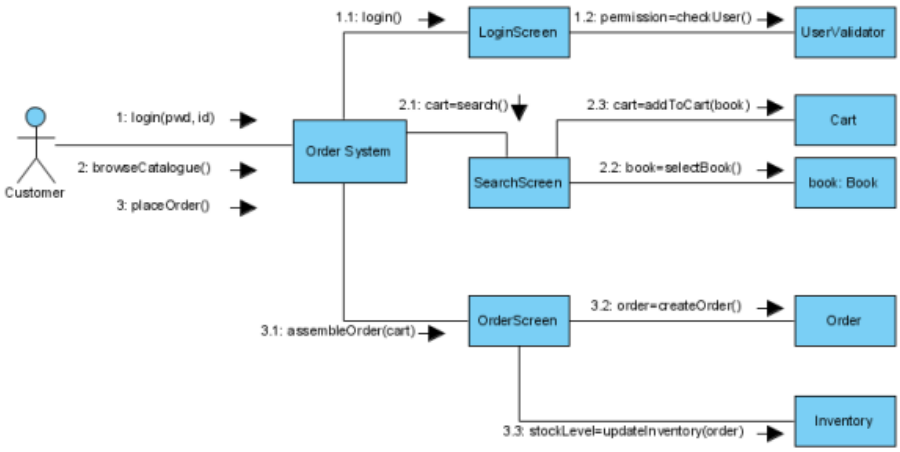
Reorder messages

When the **Communication Diagram Specification** dialog box appears, the **Message** tab is opened by default. Double click on the **Sequence #** cell of a message to edit it. Click **OK** button to apply the changes.



Edit sequence number of messages

The sequence number of messages is updated.



Sequence number of messages updated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

State machine diagram

State machine diagram shows flow of control from state to state within single object. You will learn how to create a state machine diagram in this chapter.

Creating state machine diagram

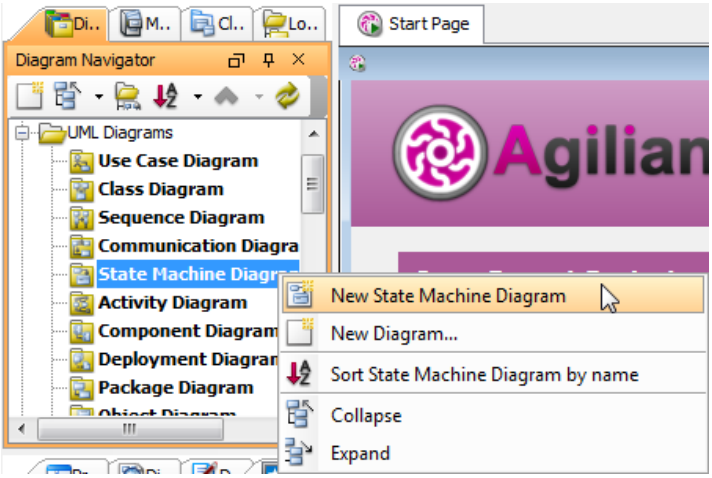
Learn how to create state machine diagram and configure state properties.

Creating state machine diagram

State machine diagram shows flow of control from state to state within single object. It usually contains simple states, composite states, composite states, transitions, events and actions.

Creating state machine diagram

- Click on **UML** on toolbar and select **State Machine Diagram** from the drop down menu **.**
- Right click on **State Machine Diagram** in **Diagram Navigator** and select **New State Machine Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > State Machine Diagram** from the main menu.



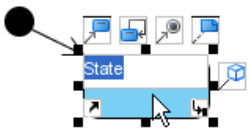
Create state machine diagram

Creating states and transitions

After creating a state machine diagram, an initial pseudo state appears by default. Move the mouse over it and click its resource icon **Transition** -> **State**.

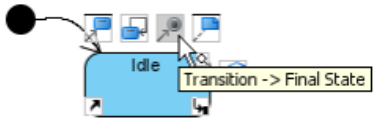
Create state from initial pseudo state

Drag it to your preferred place and then release the mouse to confirm the place. As a result, a state is created and is connected to the initial pseudo state with a transition.



State and transition created

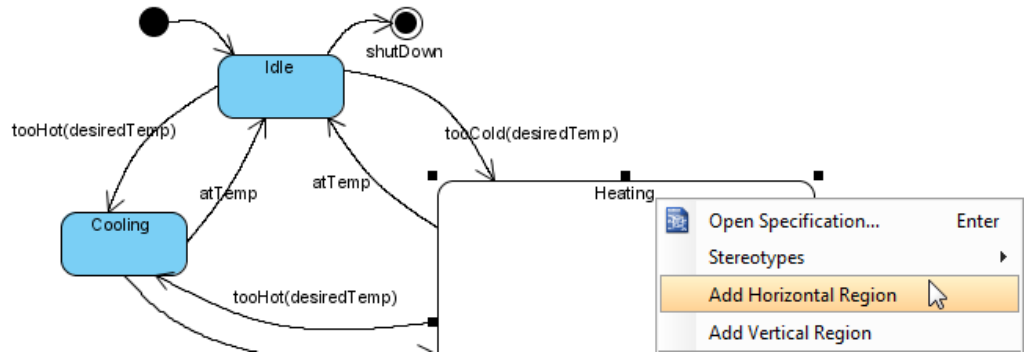
Similarly, you can use the **Transition** -> **Final State** resource to create a final state.



Create final state

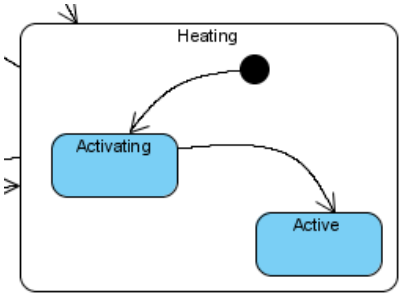
Adding region to state

To model substates of a composite state, you need to add one or more regions to it. To add a region, right click the state and select either **Add Horizontal Region** or **Add Vertical Region** from the pop-up menu.



Add region to state

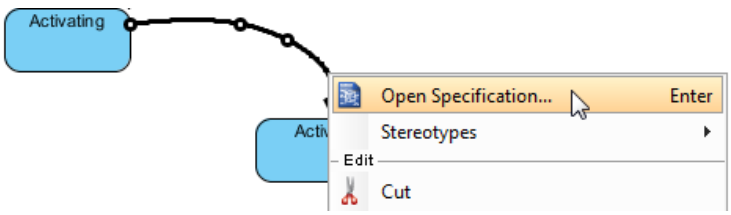
Next, you can draw the substates inside the region.



Substates in a composite state

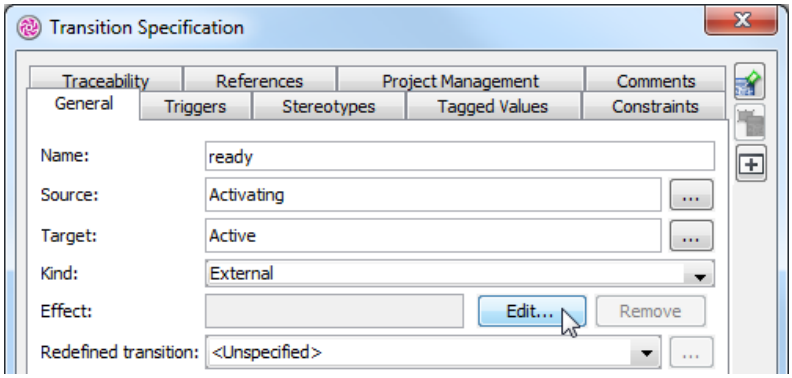
Modeling properties of transition

To model properties of transition such as effect and guard, right click on the target transition and select **Open Specification...** from the pop-up menu.



Open specification of transition

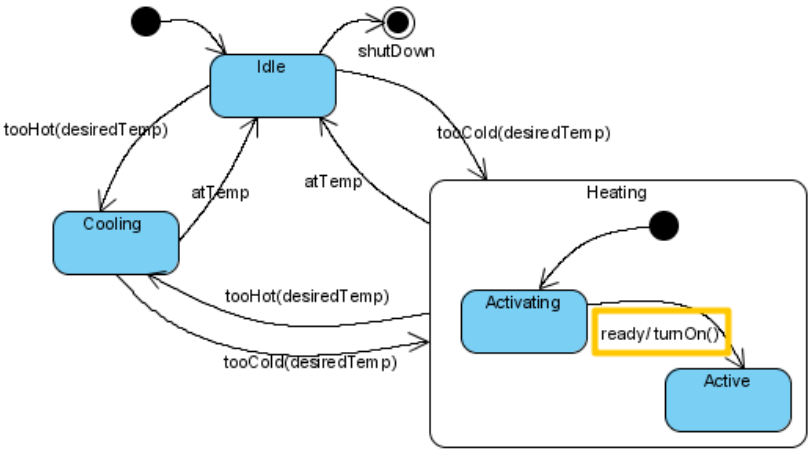
When the **Transition Specification** dialog box pops out, you can edit its name, effect and guard. Next, click **Edit...** button of the **Effect** property.



***Transition Specification** dialog box*

In **Activity Specification (Effect)** dialog box, change its name, and then click **OK** button to apply the change.

Click **OK** button in the **Transition Specification** dialog box to close it. Finally, the name and effect are shown on the transition caption.



Name and effect shown in caption of transition

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Activity diagram

Activity diagram is a flowchart-based diagram showing flow of control from activity to activity. It shows concurrency, branch, control flow and object flow. You will learn how to create activity diagram in this chapter.

Creating activity diagram

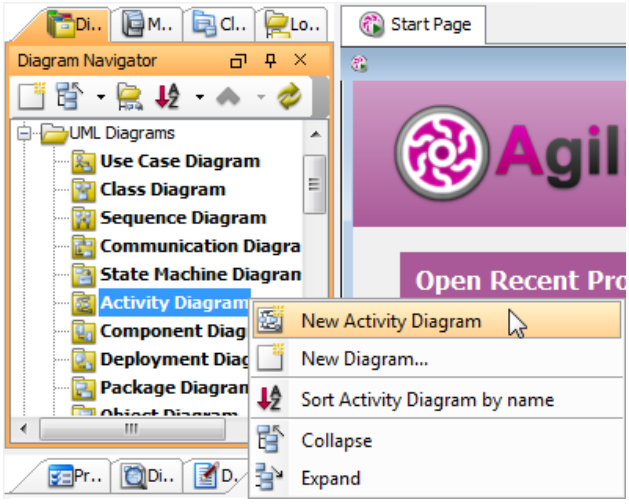
Learn how to create activity diagram.

Creating activity diagram

Activity diagram is a flowchart-based diagram showing flow of control from activity to activity. It shows concurrency, branch, control flow and object flow. Swimlane, furthermore, is used for partitioning the activity states.

Creating activity diagram

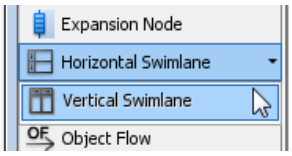
- Click on **UML** on toolbar and select **Activity Diagram** from the drop down menu **.**
- Right click on **Activity Diagram** in **Diagram Navigator** and select **New Activity Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Activity Diagram** from the main menu.



Create activity diagram

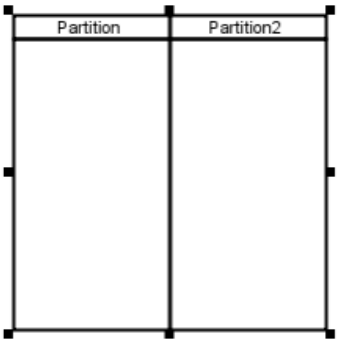
Creating swimlane

You can click either **Horizontal Swimlane** or **Vertical Swimlane** on the diagram toolbar.



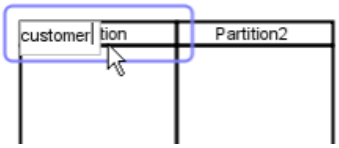
Create swimlane

Next, click on the diagram to create it.



Swimlane is created

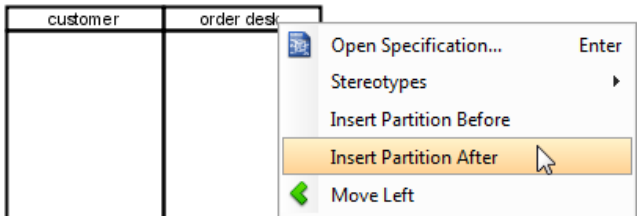
Double click at the top of partition to rename it.



Rename partition

Inserting partition to swimlane

To insert partition to swimlane, right click on a partition and select either **Insert Partition Before** or **Insert Partition After** from the pop-up menu. The selection depends on the location you want to insert.



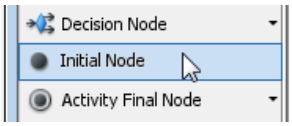
Insert partition to swimlane

A partition is inserted.

Partition inserted

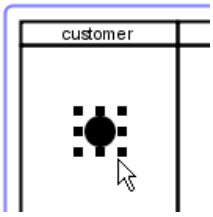
Creating initial node

Click **Initial Node** on the diagram toolbar.



Create initial node

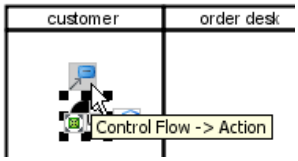
Click inside the partition to create the initial node there.



Initial node created

Creating action

Move the mouse over the initial node, press its resource icon **Control Flow** -> **Action**.

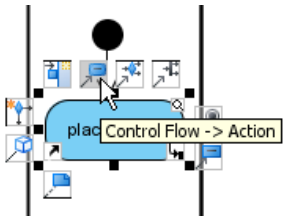


Create action

Drag it to your preferred place and then release the mouse to confirm creating. An action is created and is connected to the initial node with a control flow.

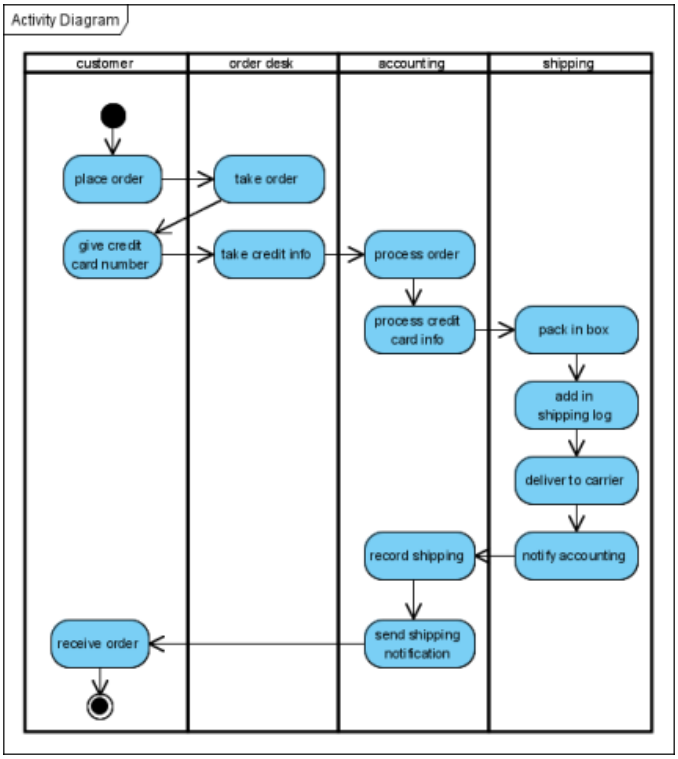
Action created

Similarly, you can create a new action from the existing action by pressing its **Control Flow** -> **Action** resource icon.



Create a new action from an action

Create as much as actions you need. Finally, when you want to finish the activity diagram, press **Control Flow -> Activity Final Node**.



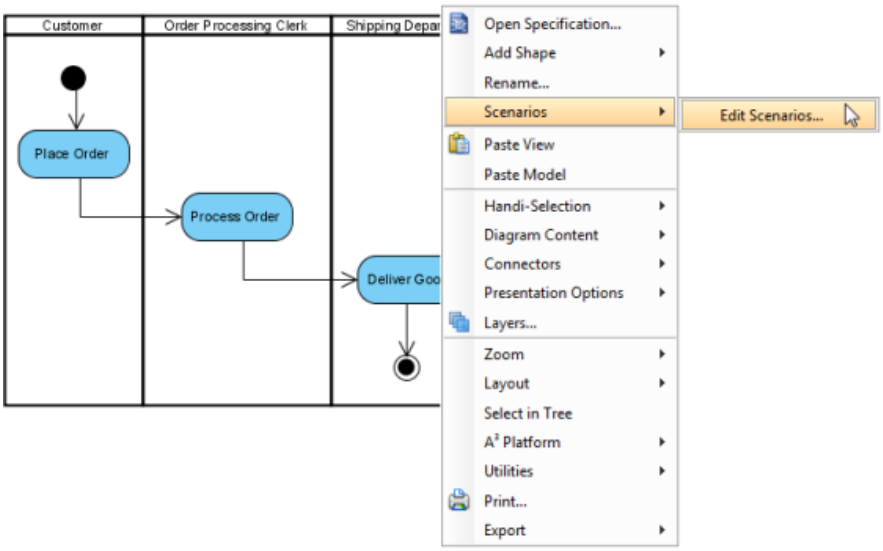
Completed activity diagram

Working with scenario

A scenario is a diagram formed by the internal interaction of a sequence of action, modeled by their sub-diagrams. With scenario, you can produce a diagram which presents an overview of an execution path in activity diagram, so as to know how user and system communicate with each other in order to complete the flow.

Producing scenario from activity diagram

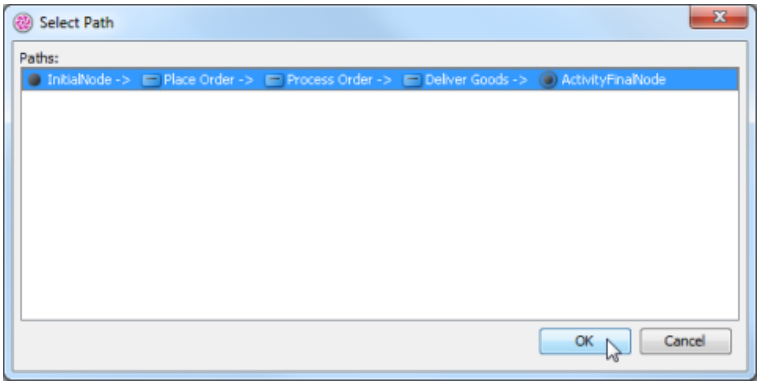
1. Right click on the activity diagram's background that contains the flows that you want to produce a scenario, and select **Scenarios > Edit Scenarios...** from the pop-up menu.



Edit scenario

2. In the **Edit Scenarios** dialog box, click **Add...** button at the bottom left corner to add a scenario.

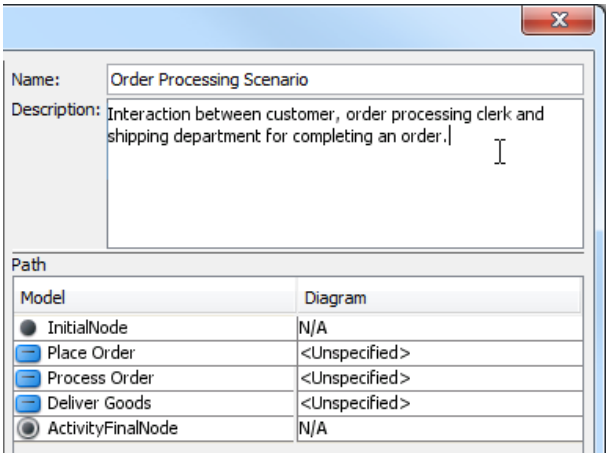
3. Select a path for generating scenario. Click **OK** button to confirm.



Select a path for generating scenario

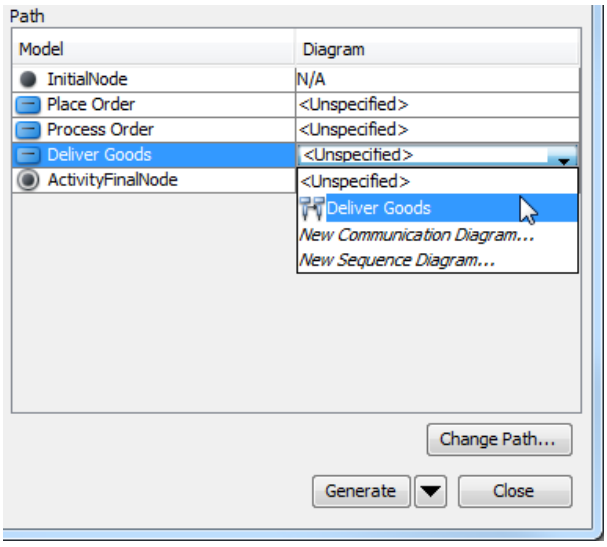
NOTE: A path is a continuous flow of actions in the diagram, with an initial node placed at the beginning of the actions. Multiple paths are obtained by determining the existence of decision nodes within the flow.

4. Name the scenario. Add description if necessary.



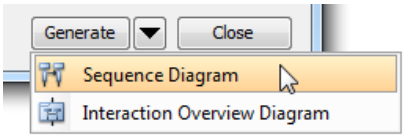
Name and describe scenario

5. The actions being involved in the flow are listed in the **Path** table. For actions that have sub-diagram(s), pick up the sub-diagram in **Diagram** column, or just create a new one. You may, however, leave it unspecified, which cause that action to be ignored when producing scenario.



Select diagram for action

6. Click the upside down arrow of **Generate** button and select the type of diagram of the scenario.



Updating scenario

Whenever the sub-diagram(s) of action(s) are updated, you can update the scenario to make it represents the latest information of interaction. To update scenario, right click on the activity diagram that have scenario produced before, select **Scenarios**, then the name of scenario from the popup menu.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Component diagram

Component diagram shows the physical aspect of an object-oriented software system. You will learn how to create component diagram in this chapter.

Creating component diagram

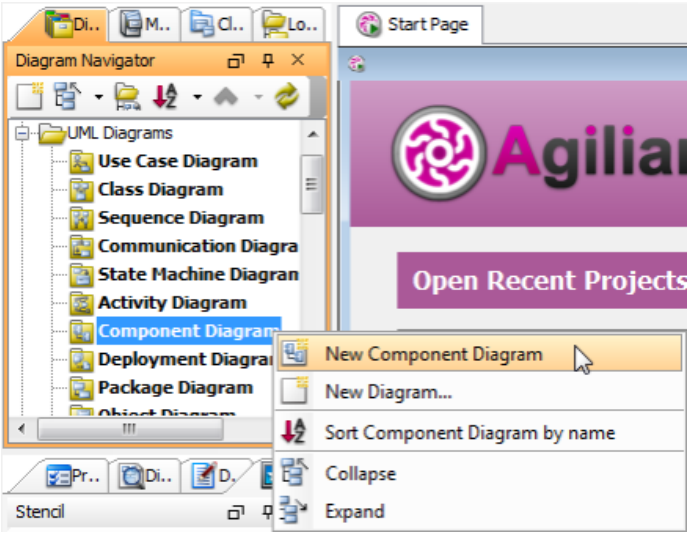
Learn how to create component diagram.

Creating component diagram

Component diagram shows the physical aspect of an object-oriented software system. It illustrates the architectures of the software components and dependencies between them.

Creating component diagram

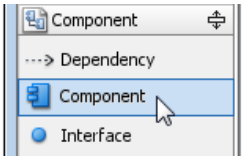
- Click on **UML** on toolbar and select **Component Diagram** from the drop down menu **.**
- Right click on **Component Diagram** in **Diagram Navigator** and select **New Component Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Component Diagram** from the main menu.



Create component diagram

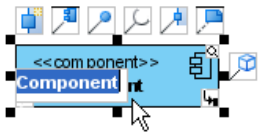
Creating component

To create component, click **Component** on the diagram toolbar and then click on the diagram.



Create component

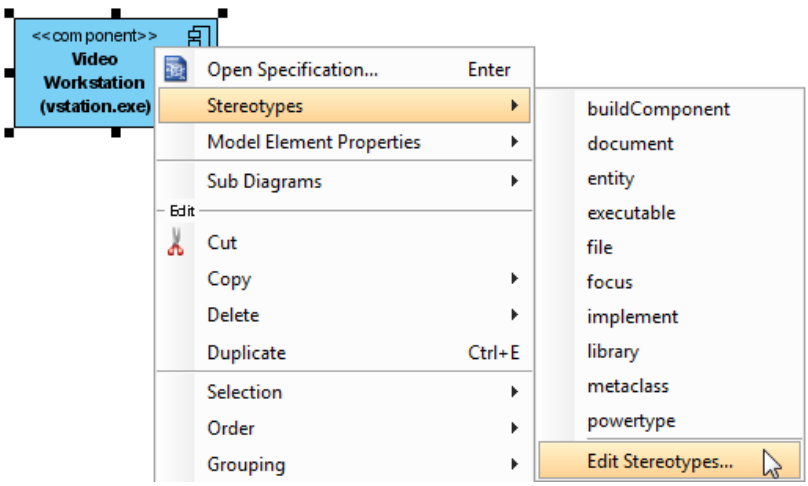
A component will be created.



Component created

Assigning stereotypes

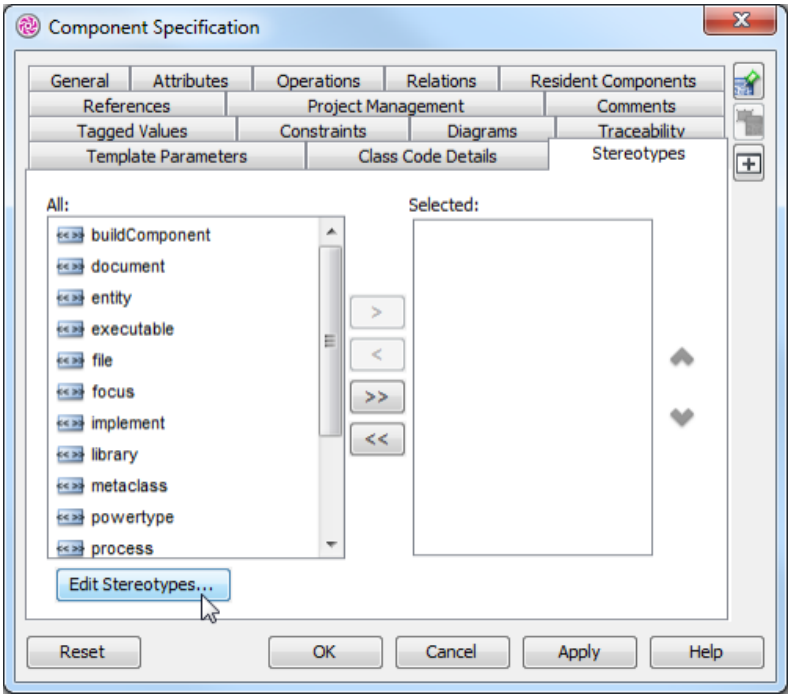
Right click on the package and select **Stereotypes** > **Edit Stereotypes**... from the pop-up menu.



Assign stereotypes

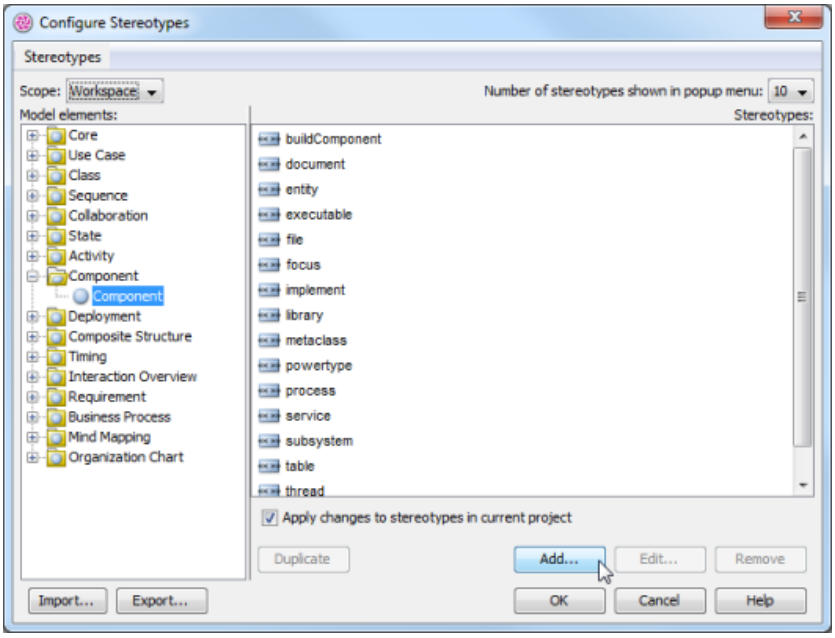
When the **Component Specification** dialog box pops out, the **Stereotypes** tab is opened by default. The list on the left shows the selectable stereotypes.

If the stereotype you want to use is not on the list, click **Edit Stereotypes...** button.



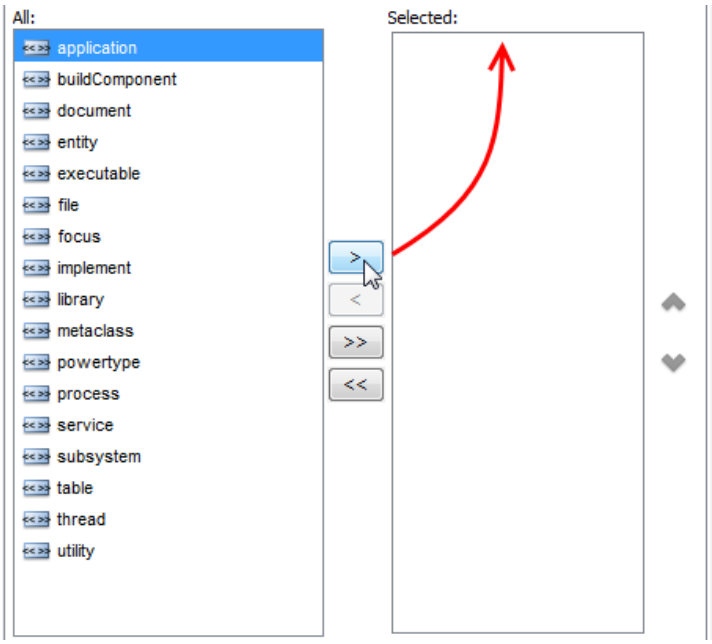
Edit stereotypes

Click **Add...** button in the **Configure Stereotypes** dialog box.



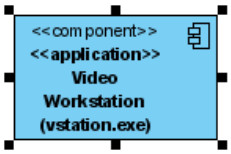
Add stereotype

Name the stereotype (e.g. *application*) in the **Stereotype Specification** dialog box and then click **OK** button to close it. Click **OK** button in the **Configure Stereotypes** dialog box. The added stereotype will then be shown on the list in the **Component Specification** dialog box. Select it and click **Add Selected** button. Finally, click **OK** button to confirm.



Add selected stereotypes

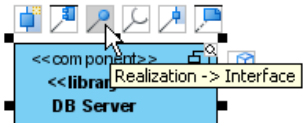
Close the specification dialog box. Stereotypes will be applied to the package.



Stereotypes assigned

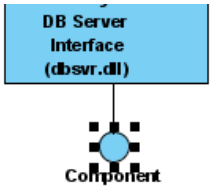
Creating provided interface

To create provided interface for a component, move the mouse over the target component and press its resource icon **Realization** -> **Interface**.



Create provided interface

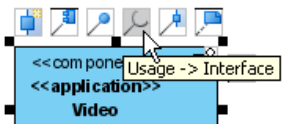
Drag it to your preferred place and then release the mouse button to confirm creating. The provided interface is created.



Provided interface created

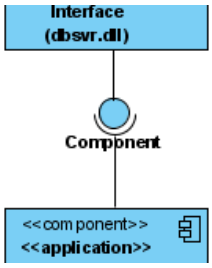
Creating required interface

To create required interface for a component, move the mouse over the target component and press its resource icon **Usage**-> **Interface**.



Create required interface

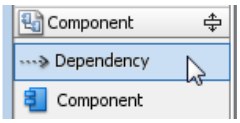
Drag it to your preferred place to create a new interface, or drag to an existing interface to connect to it. Release the mouse button to create the required interface.



Required interface created

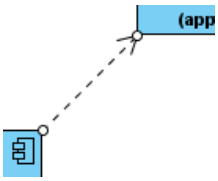
Creating dependency

To create dependency, click **Dependency** on the diagram toolbar.



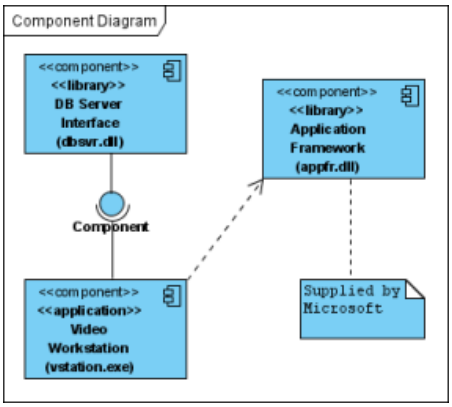
Create dependency

Drag from the source shape to the target shape and then release the mouse button to create the dependency.



Dependency created

Continue to complete the diagram



Completed diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Deployment diagram

Deployment diagram shows the physical aspects of an object-oriented system. You will learn how to create a deployment diagram in this chapter.

Creating deployment diagram

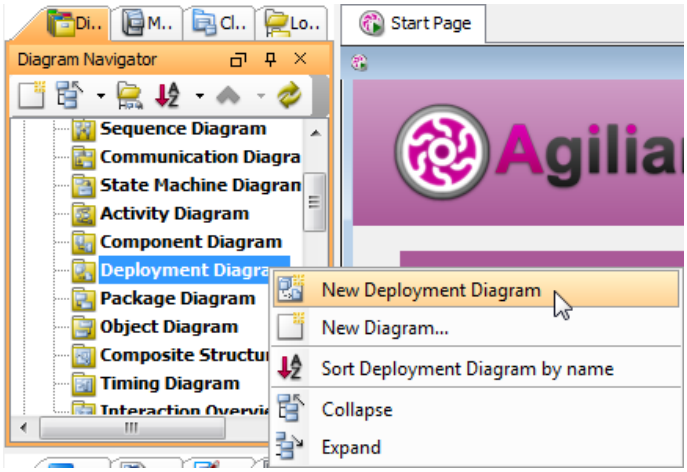
Learn how to create deployment diagram.

Creating deployment diagram

Deployment diagram shows the physical aspects of an object-oriented system. It also shows the configuration of run time processing nodes and artifacts.

Creating deployment diagram

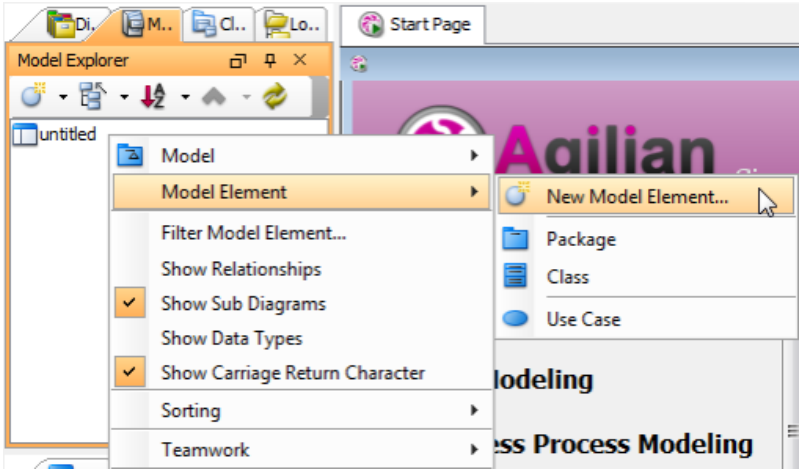
- Click on **UML** on toolbar and select **Deployment Diagram** from the drop down menu **.**
- Right click on **Deployment Diagram** in **Diagram Navigator** and select **New Deployment Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Deployment Diagram** from the main menu.



Create deployment diagram

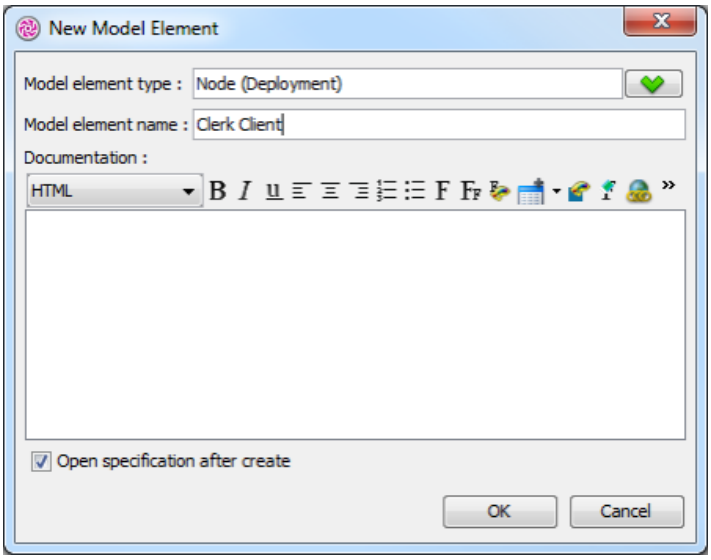
Creating node model element

To create node model element, right click on the background of **Model Explorer** and select **Model Element** > **New Model Element** from the pop-up menu.



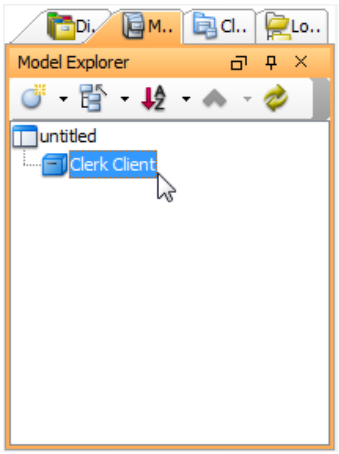
Create model element

In the **New Model Element** dialog box, type *Node (Deployment)* in **Model element type**, enter the node name in **Model element name**. Click **OK** button to confirm.



Create node

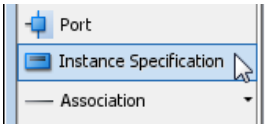
As a result, the Node Specification dialog box will pop out when the node is created.



Node is created

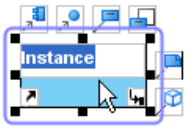
Creating instance of node

To create instance of node, click **Instance Specification** on the diagram toolbar and then click on the diagram.



Create instance specification

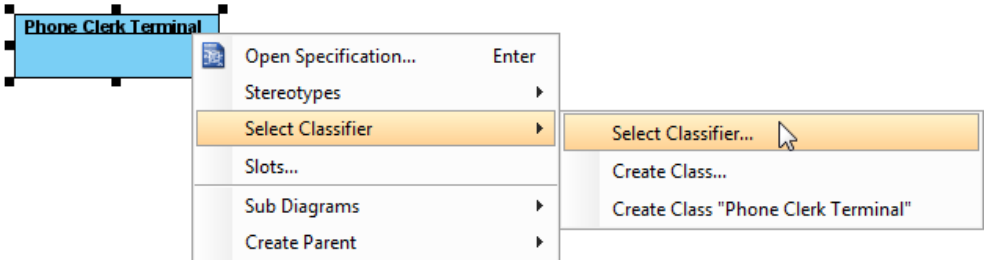
As a result, an instance specification will be created.



Instance specification created

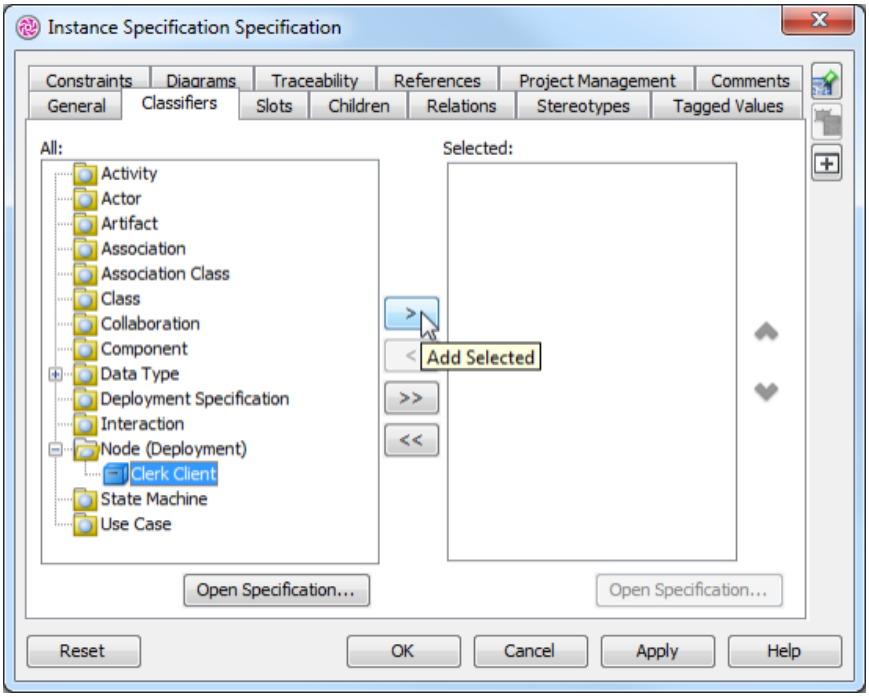
Selecting classifiers

To specify classifiers for an instance specification, right click on the target instance specification and select **Select Classifier** > **Select Classifier...** from the pop-up menu.



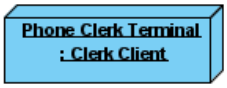
Select classifier

When the **Instance Specification Specification** dialog box pops out, the **Classifiers** tab is opened by default. Select the classifier(s) on the left and click **Add Selected** button to add them.



Add selected classifiers

Click **OK** button to close the specification dialog box. Finally, the selected classifier is assigned to the instance specification.



Classifier is assigned

Creating link

To create link from instance specification, move the mouse over the target shape and press its resource icon **Link** -> **Instance Specification**.

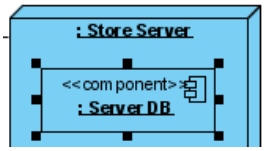
Create link

Drag to empty space of the diagram to create a new instance specification, or drag to an existing instance specification to connect to it. Release the mouse button to create the link.

Link created

Creating instance of component

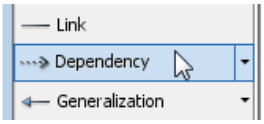
Similar to creating instance of node, you first create a component model element, and then create an instance specification, but in this case, you assign a component to the instance specification as classifier. After that, the instance specification will be displayed as a component.



Instance of component

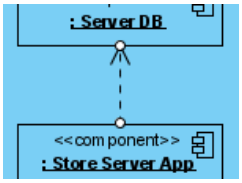
Creating dependency

To create dependency, click **Dependency** on the diagram toolbar.



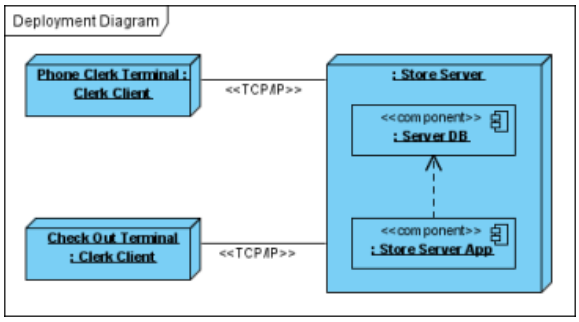
Create dependency

Drag from the source shape, move the mouse over the target shape and then release the mouse button to create the dependency.



Dependency created

Continue to complete the diagram.



Completed diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Package diagram

Package diagram shows the arrangement and organization of model elements in middle to large scale project. You will learn how to create package diagram in this chapter.

Creating package diagram

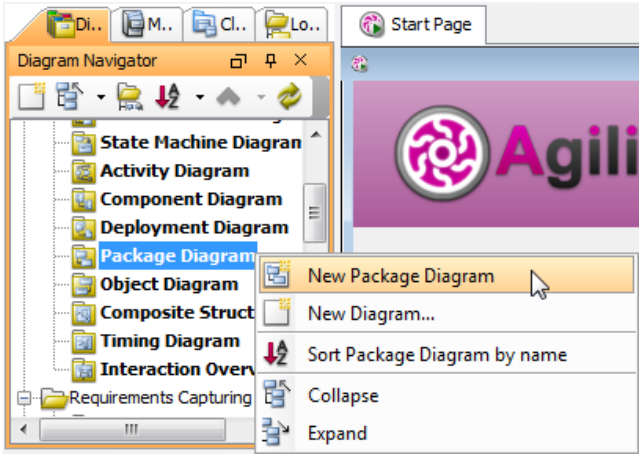
Learn how to create package diagram.

Creating package diagram

Package diagram shows the arrangement and organization of model elements in middle to large scale project. It can show both structure and dependencies between sub-systems or modules.

Creating package diagram

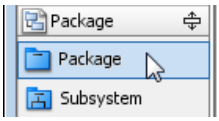
- Click on **UML** on toolbar and select **Package Diagram** from the drop down menu **.**
- Right click on **Package Diagram** in **Diagram Navigator** and select **New Package Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Package Diagram** from the main menu.



Create package diagram

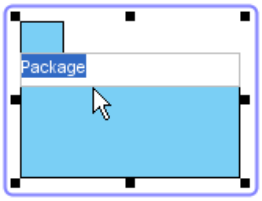
Creating package

To create package, click **Package** on the diagram toolbar and then click on the diagram.



Create package

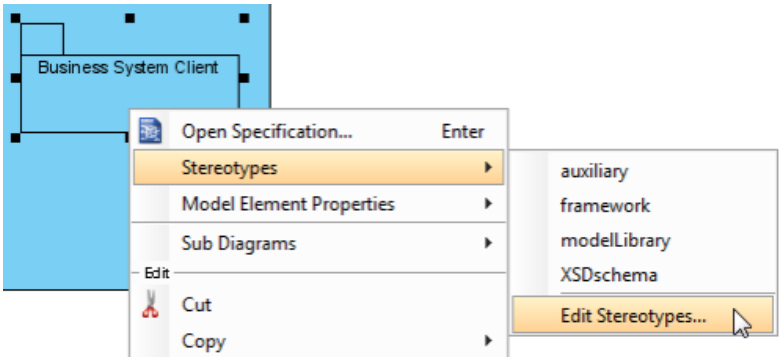
A package will be created.



Package created

Assigning stereotypes

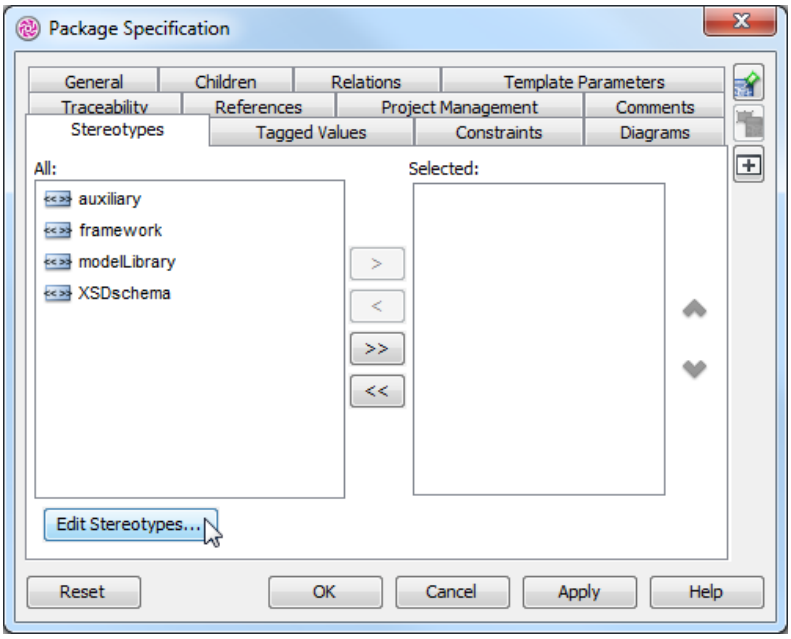
Right click on the package and select **Stereotypes** > **Edit Stereotypes**... from the pop-up menu.



Assign stereotypes

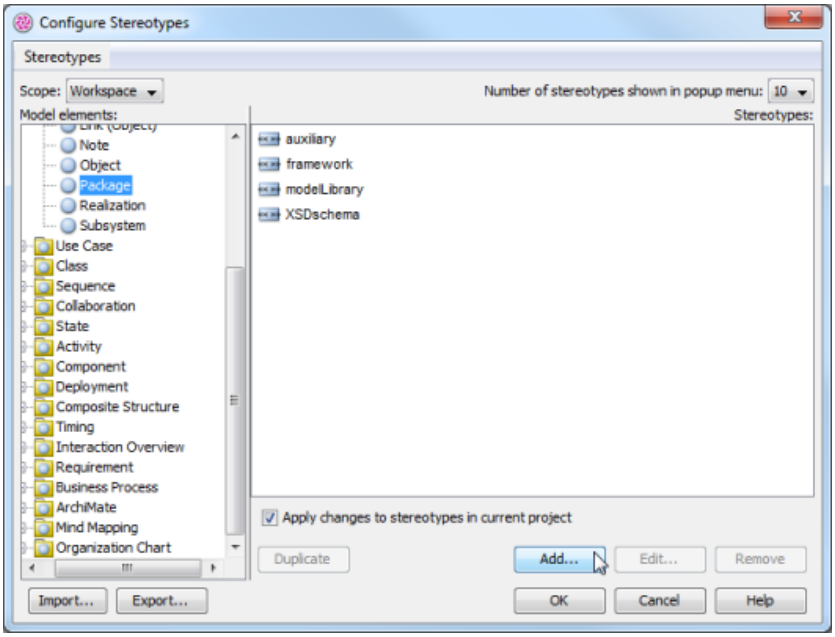
When the **Package Specification** dialog box pops out, the **Stereotypes** tab is opened by default. The list on the left shows the selectable stereotypes.

If the stereotype you want to use is not on the list, click **Edit Stereotypes...** button.



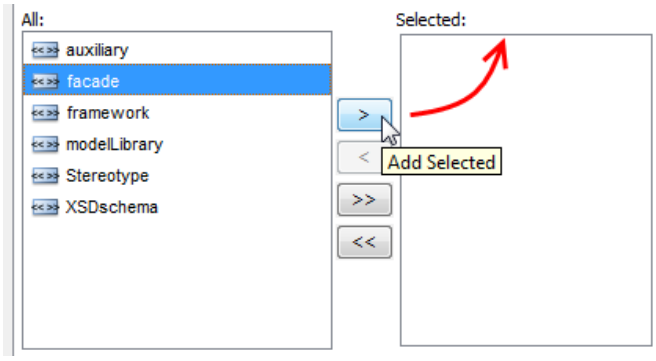
*Click **Edit Stereotypes...** button*

Click **Add...** in the **Configure Stereotypes** dialog box.



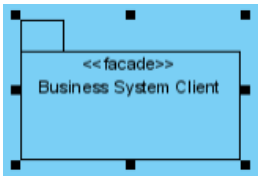
Add stereotype

Enter name for the new stereotype (e.g. *facade*). Click **OK** button in **Stereotype Specification** dialog box and the **Configure Stereotypes** dialog box. You will see the added stereotype appears on the list in **Package Specification** dialog box. Select it and click **Add Selected** button. Next, click **OK** button to proceed.



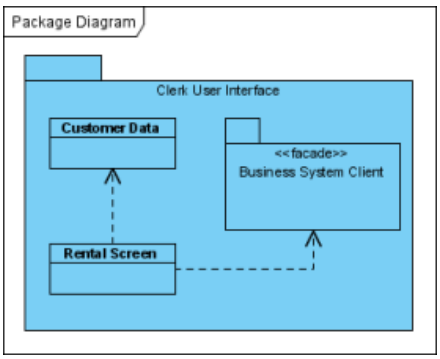
Add selected stereotype

Close the specification dialog box. Stereotypes will be applied to the package.



Stereotypes assigned

Continue to complete the diagram.



Completed diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Object diagram

Object diagram shows a snapshot of instances of things in class diagram. You will learn how to create an object diagram in this chapter.

Creating object diagram

Learn how to create an object diagram.

Creating object diagram

Object diagram shows a snapshot of instances of things in class diagram. Similar to class diagram, it shows the static design of system from the real or prototypical perspective.

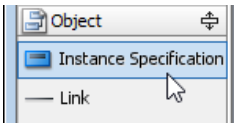
Creating object diagram

- Click on **UML** on toolbar and select **Object Diagram** from the drop down menu **.**
- Right click on **Object Diagram** in **Diagram Navigator** and select **New Object Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Object Diagram** from the main menu.

Creating instance specification

To create instance specification:

1. Select **Instance Specification** from the diagram toolbar.



Create instance specification

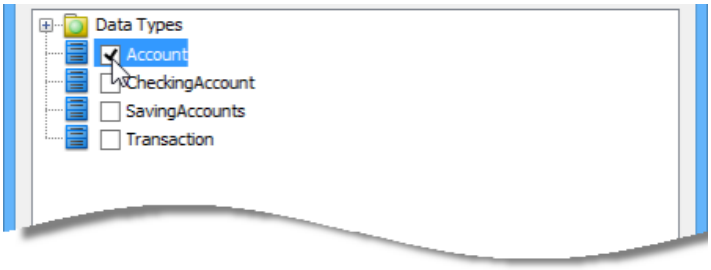
2. Click on the diagram to create an instance specification shape. Name it.

Instance specification created

Selecting classifiers

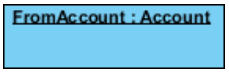
To specify classifiers for an instance specification:

1. Right-click on the desired instance specification shape and select **Select Classifier** > **Select Classifier...** from the pop-up menu.
2. This opens the **Classifiers** tab. Click **Add...** in it.
3. In the **Select Classifier** window, select the class(es) to be the classifier of the instance specification. If you are referencing another project, you can select its model element to be the classifier. Just change the **from project** selection at the top of the window.



Selecting classifier

4. Click **OK** to return to the **Instance Specification Specification** window.
5. Click **OK** to return to the diagram.

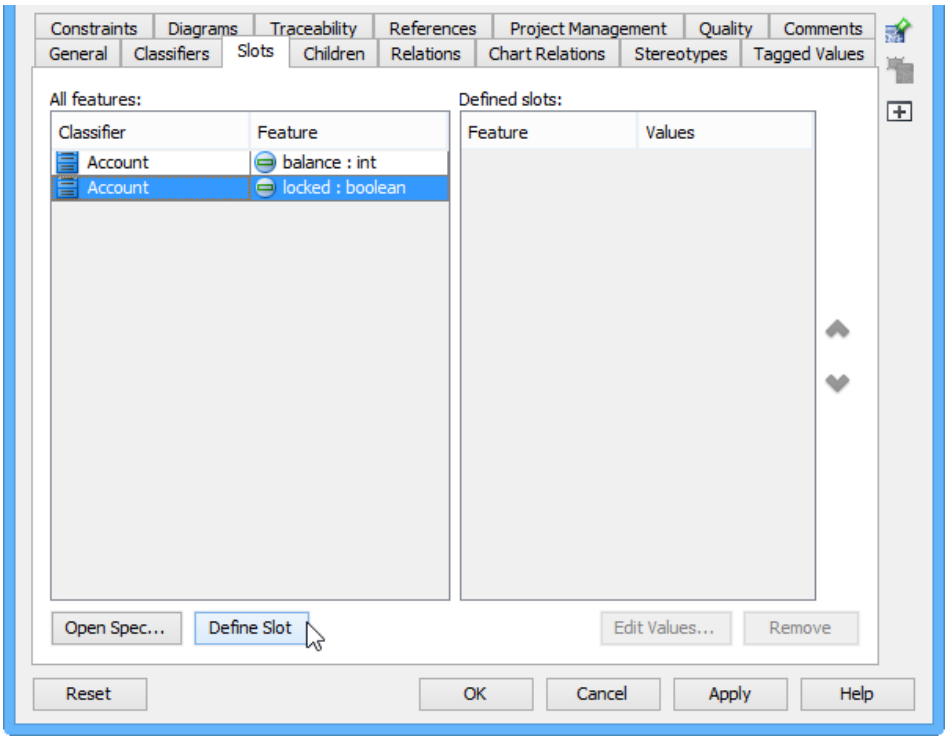


Classifier selected

Defining slots

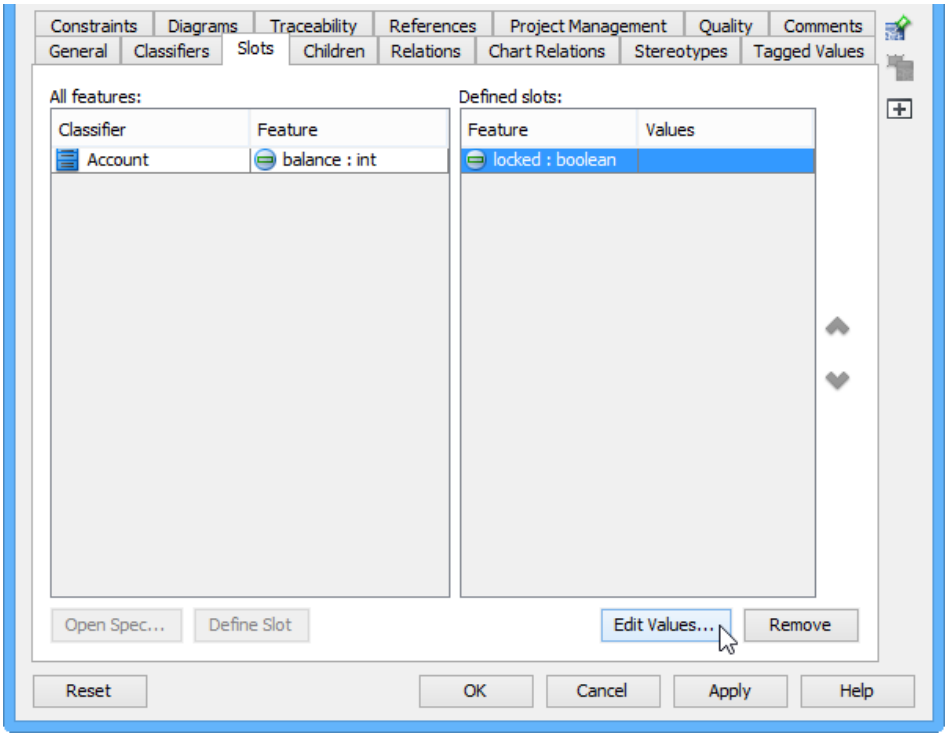
To define slots for an instance specification:

1. Right-click on the desired instance specification shape and select **Slots. ..** from the pop-up menu.
2. The **Instance Specification Specification** dialog box appears with the **Slots** tab selected. Select the features that you want to define slots on the left and click **Define Slot**.



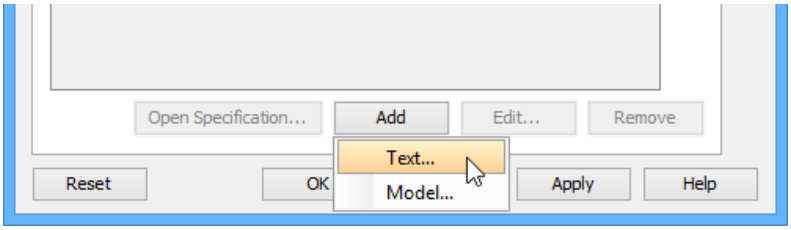
Defining slot

3. Select a defined slot and click **Edit Values...** at bottom right.



Edit values

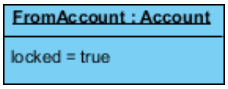
4. The **Slot Specification** window pops out, the **Values** tab is opened by default. Click **Add** button and select **Text** from the pop-up menu.



Add values to defined slot

5. Enter the slot value and click **OK** to confirm.

6. Click **OK** again in the **Instance Specification Specification** window to return to the diagram.

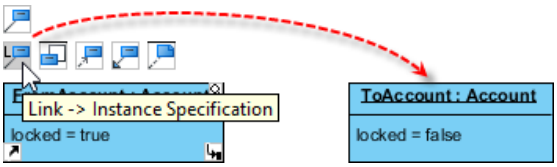


Instance specification with slot defined

Creating link

To create link between instance specifications:

1. Move the mouse pointer over the source instance specification.
2. Press on the **Link** -> **Instance Specification** resource and hold the mouse button.



Create a link

3. Drag to the target instance specification and release the mouse button.

Link created

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Composite structure diagram

Composite structure diagram visualizes the internal structure of a class or collaboration. You will learn how to create composite structure diagram in this chapter.

Creating composite structure diagram

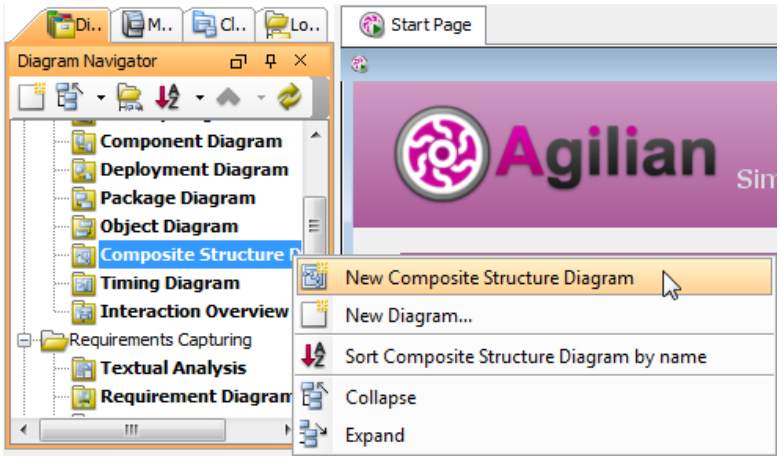
Learn how to create composite structure diagram.

Creating composite structure diagram

Composite structure diagram visualizes the internal structure of a class or collaboration. It is a kind of component diagram mainly used in modeling a system at micro point-of-view.

Creating composite structure diagram

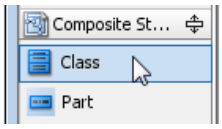
- Click on **UML** on toolbar and select **Composite Structure Diagram** from the drop down menu **.**
- Right click on **Composite Structure Diagram** in **Diagram Navigator** and select **New Composite Structure Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Composite Structure Diagram** from the main menu.



Create composite structure diagram

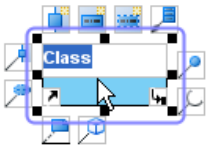
Creating class

To create class, select **Class** on the diagram toolbar and then click on the diagram.



Create class

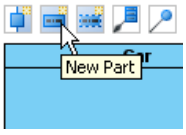
A class will be created.



Class created

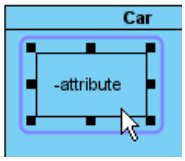
Creating part

To create part, move the mouse over the target class, press its resource icon **New Part**.



Create part

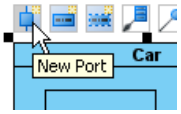
A part will be created.



Part created

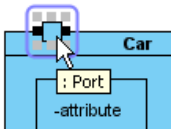
Creating port

To create port, move the mouse over the target class, press its resource icon **New Port**.



Create port

A port will be created.

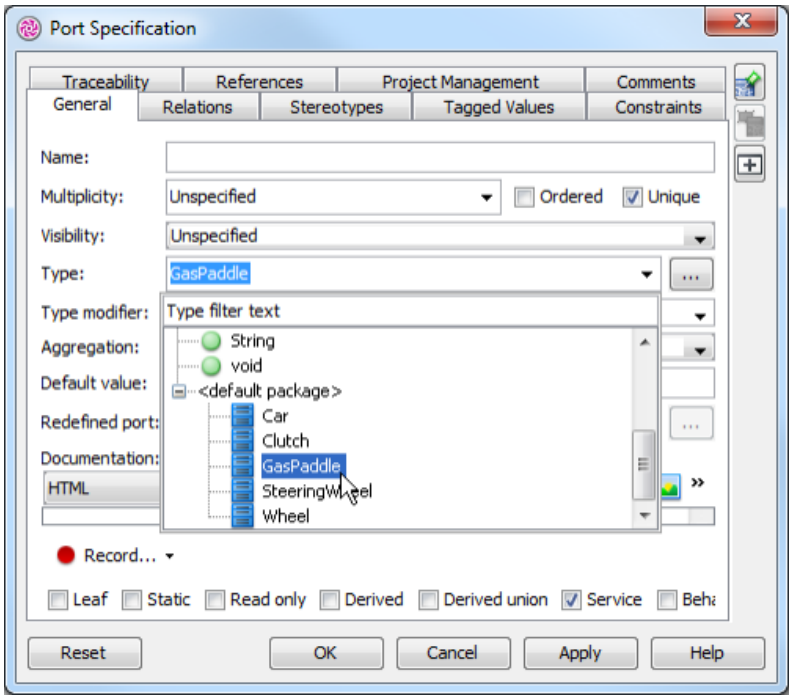


Port created

Specifying type of port

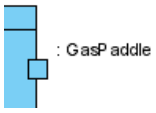
Right click on the target port and select **Open Specification...** from the pop-up menu. Subsequently, **Port Specification** dialog box pops out.

Click the combo box of **Type** and select a class.



Select type

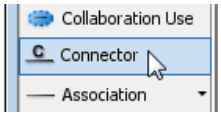
Click **OK** button to apply the changes. Type will be shown on the caption of the port.



Type shown on port

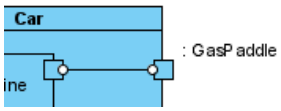
Creating connector

To create connector, select **Connector** on the diagram toolbar.



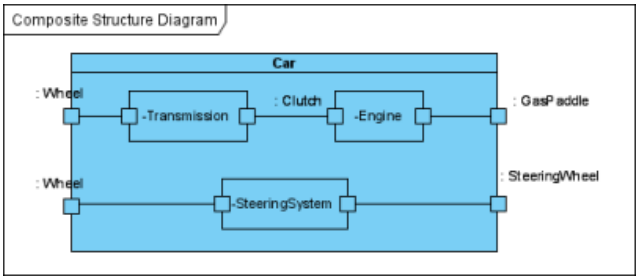
Create connector

Drag from the source shape to the target shape and then release the mouse button to create the connector.



Connector created

Continue to complete the diagram.



Completed diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Timing diagram

Timing diagram shows time, event, space and signal for real-time and distributed system. You will learn how to create timing diagram in this chapter.

Creating timing diagram

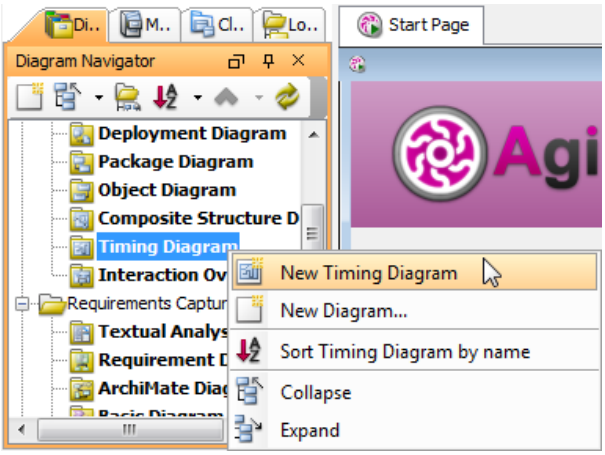
Learn how to create timing diagram.

Creating timing diagram

Timing diagram shows time, event, space and signal for real-time and distributed system.

Creating timing diagram

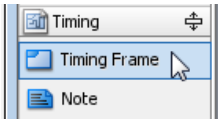
- Click on **UML** on toolbar and select **Timing Diagram** from the drop down menu **.**
- Right click on **Timing Diagram** in **Diagram Navigator** and select **New Timing Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Timing Diagram** from the main menu.



Create timing diagram

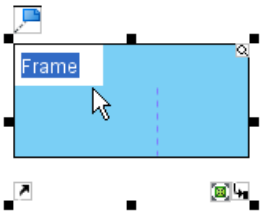
Creating timing frame

To create timing frame, click **Timing Frame** on the diagram toolbar and then click on the diagram.



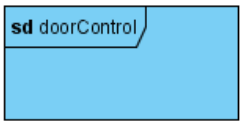
Create timing frame

Double click on the top left corner of the frame to rename it.



Rename frame

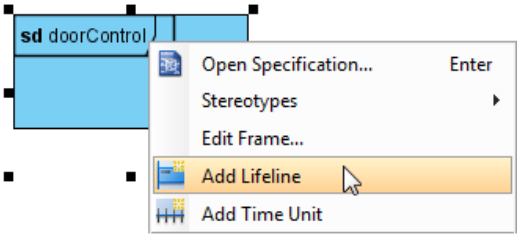
The name of a timing frame is usually preceded by the **sd** keyword.



Frame renamed

Adding lifeline to frame

To add lifeline to frame, right click the frame and select **Add Lifeline** from the pop-up menu.

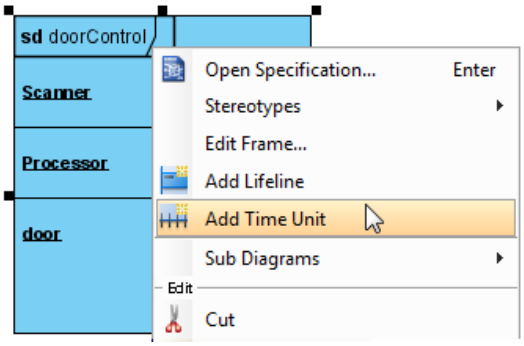


Add lifeline

Double-click on the name of the lifeline to rename it.

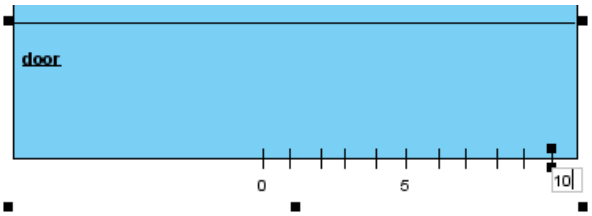
Adding time unit to frame

To add time unit to frame, right click the frame and select **Add Time Unit** from the pop-up menu.



Add time unit

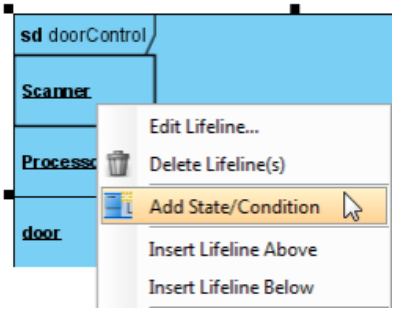
Repeat the step to add as many as time units you need. Double click on a time unit to rename it.



Rename time unit

Adding state/condition to lifeline

To add state/condition to lifeline, right click the lifeline and select **Add State/Condition** from the pop-up menu.

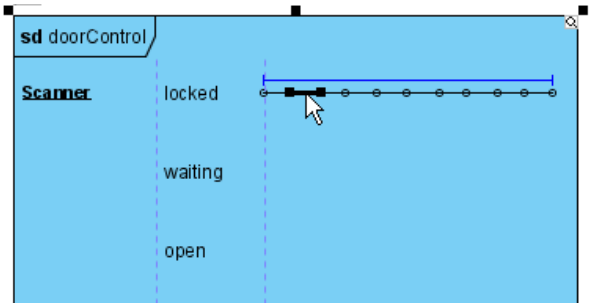


Add state/condition

Double click on the name of the state/condition to rename it.

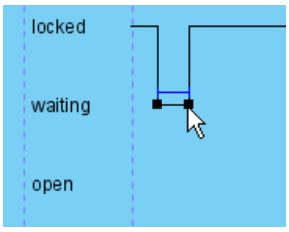
Dragging time instance

Mouse over the line segment of a time instance, click and drag it.



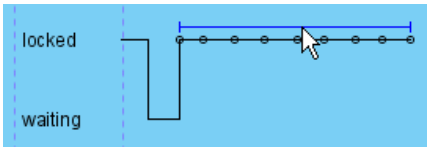
Drag time instance

Release the mouse button when reached the target state/condition.



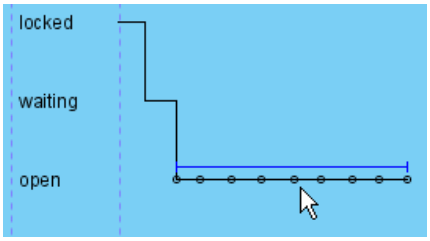
Dragged time instance

You can also move a group of time instances that are at the same state/condition. Mouse over the time instances and you will see a blue line above them, click and drag on the blue line.



Move a group of time instances

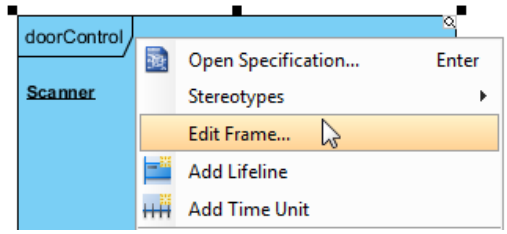
Release the mouse button when reached the target state/condition. The group of time instances is moved at once.



Moved group of time instances

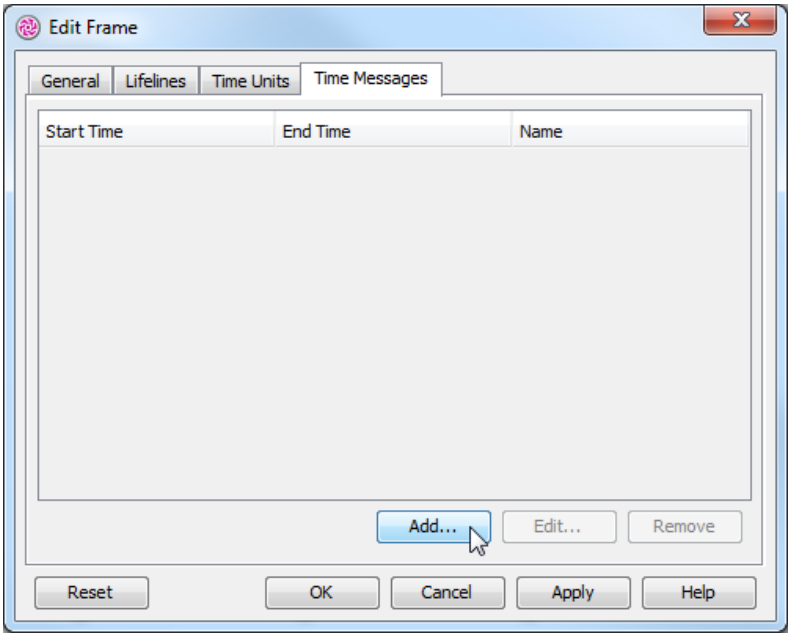
Adding time messages to frame

To add time messages to frame, right-click the frame and select **Edit Frame...** from the pop-up menu.



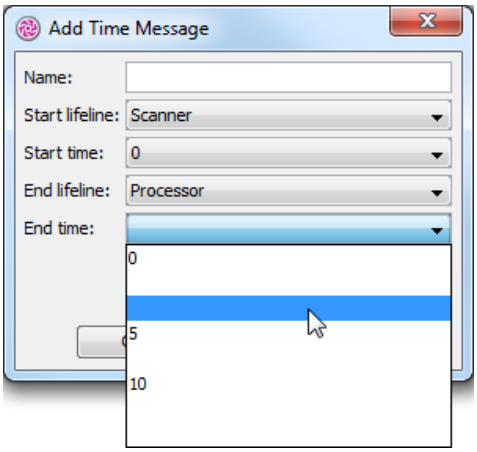
Edit frame

In the **Edit Frame** dialog box, open the **Time Messages** tab and click **Add...** button.



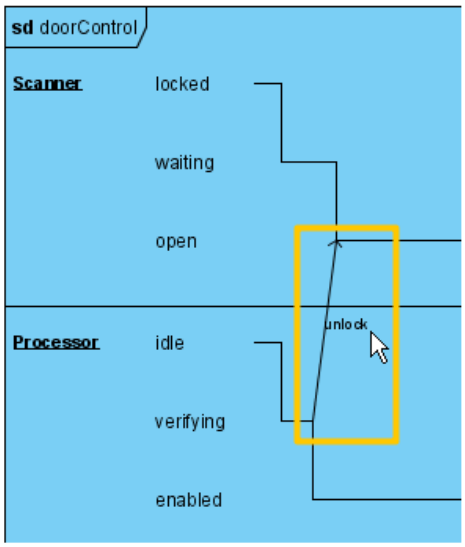
Add time message

When the **Add Time Message** dialog box pops out, enter name and select the start lifeline, start time, end lifeline and end time for this time message. Note that as time units may be unnamed, when selecting start/end time you should check the relative position of the time unit in the list.



Select end time of time message

The time message is shown on the frame.

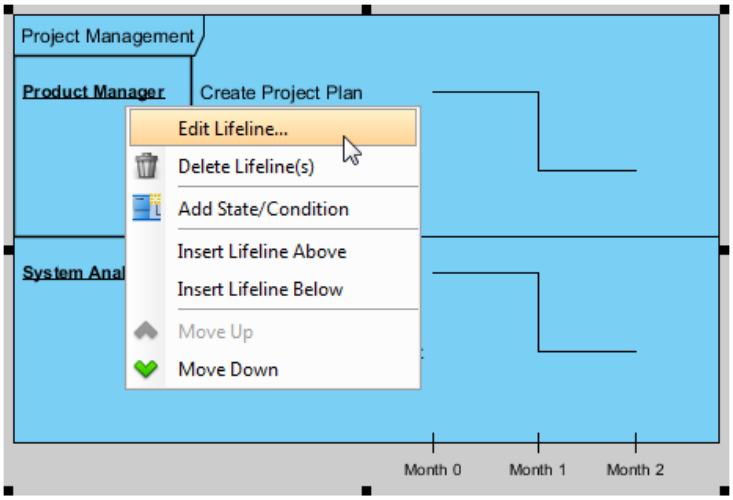


Time message

Adding duration constraint

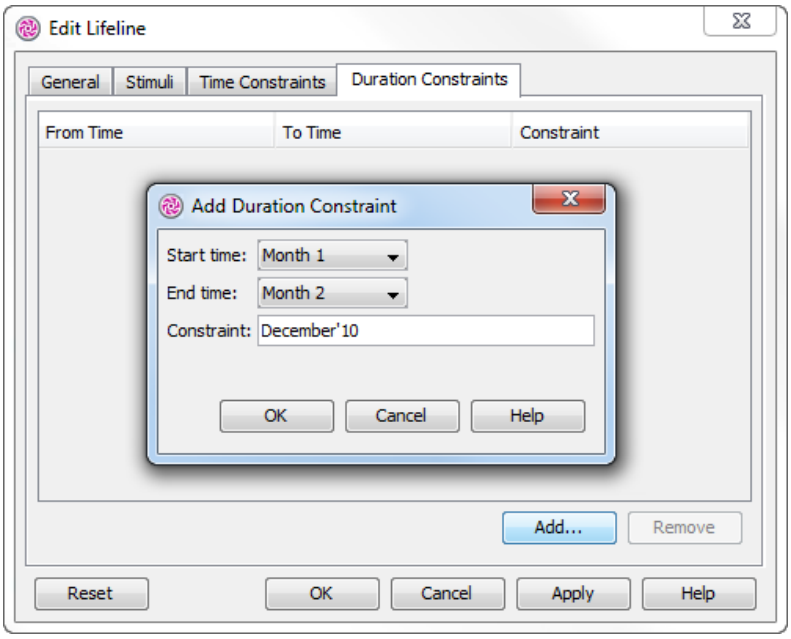
Duration constraint is used to show the duration limitation of a particular lifeline over a period of time.

1. To set the duration constraints of a lifeline, right-click on the lifeline and select **Edit Lifeline...** from the pop-up menu.



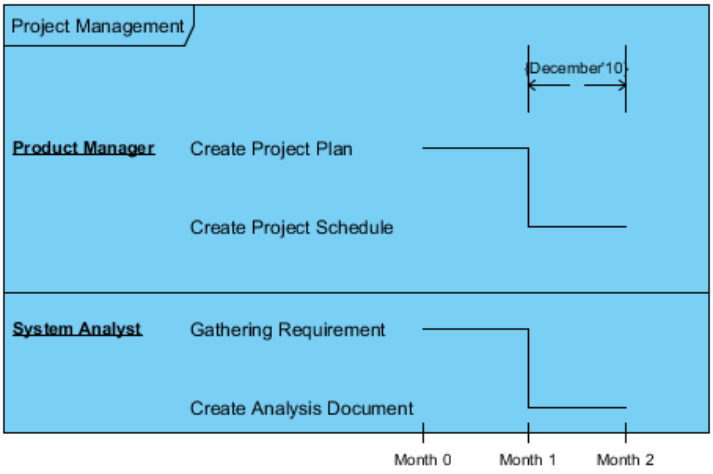
Edit lifeline

2. In the **Duration Constraints** tab, click on the **Add...** button. In the **Add Duration Constraint** dialog box, select the appropriate **Start time** and **End time** from the drop down menu. Fills in the duration constraint of the selected time on the **Constraint** field. Click on the **OK** button to close the dialog box.



Add duration constraint

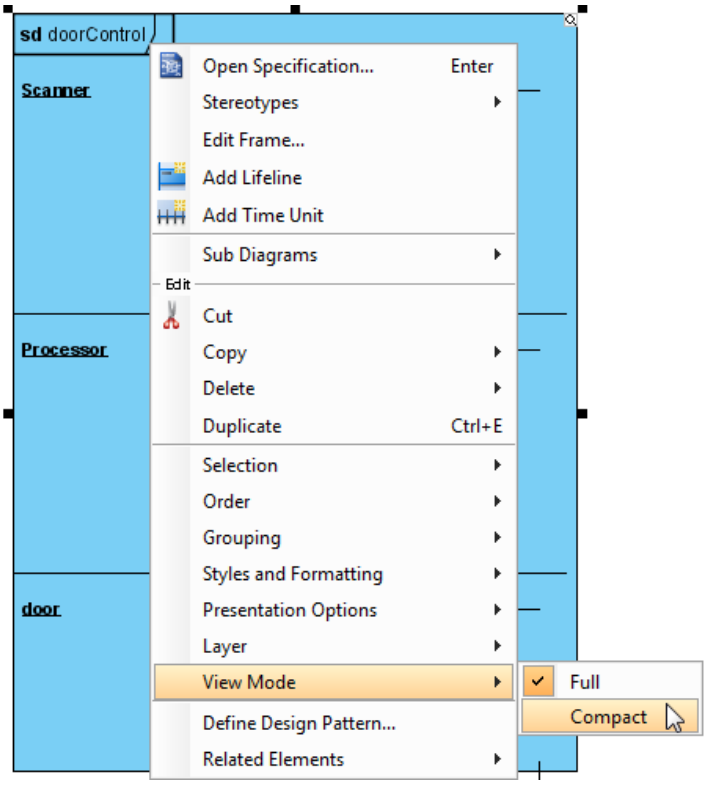
3. Click **OK** to return to diagram.



Duration constraint is added

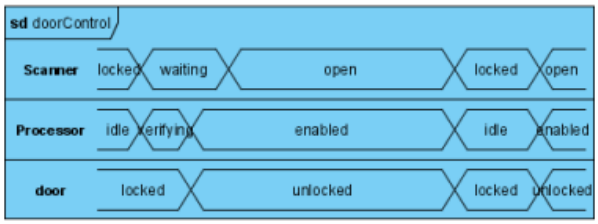
Switching to compact view mode

To switch to compact view mode, right-click the frame and select **View Mode** > **Compact** from the popup menu.



Switch to compact view mode

The frame will be shown in compact mode.



Frame shown in compact mode

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Interaction overview diagram

Interaction overview diagram is the variant of activity diagram in which a control flow with nodes represents interaction diagrams. You will learn how to create interaction overview diagram in this chapter.

Creating interaction overview diagram

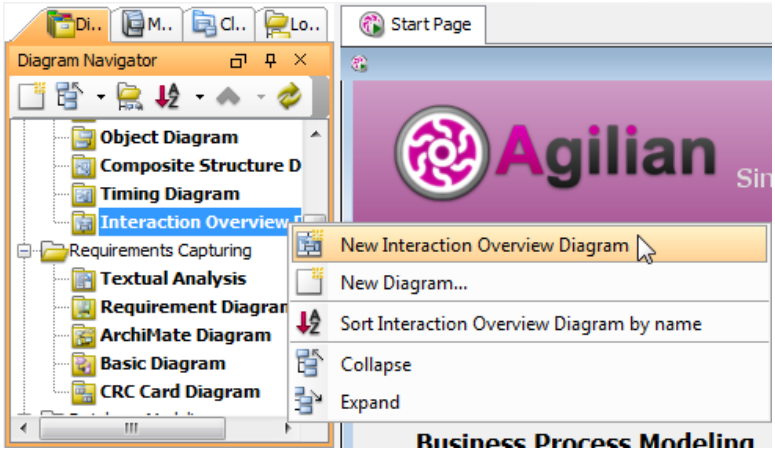
Learn how to create interaction overview diagram.

Creating interaction overview diagram

Interaction overview diagram is the variant of activity diagram in which a control flow with nodes represents interaction diagrams. Interaction diagrams are constructed with blocks, such as sequence, communication, interaction overview and timing diagrams. Most of the notation for interaction overview diagrams is the same as activity diagrams, for example, initial, final, decision, merge, fork and join nodes. However, interaction overview diagram introduces two new elements: interaction occurrences and interaction elements.

Creating interaction overview diagram

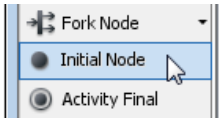
- Click on **UML** on toolbar and select **Interaction Overview Diagram** from the drop down menu **.**
- Right click on **Interaction Overview Diagram** in **Diagram Navigator** and select **New Interaction Overview Diagram** from the popup menu.
- Select **File > New Diagram > UML Diagrams > Interaction Overview Diagram** from the main menu.



Create interaction overview diagram

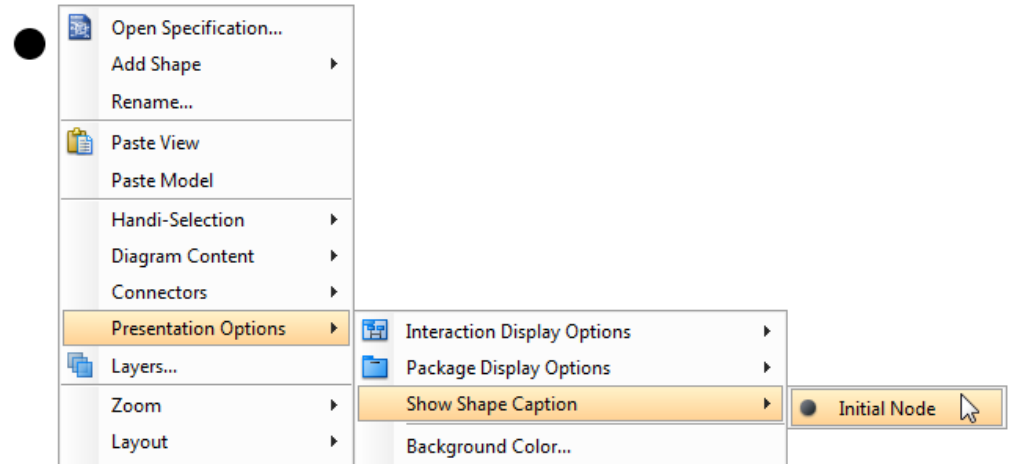
Creating initial node

To create initial node, click **Initial Node** on the diagram toolbar and then click on the diagram.



Create initial node

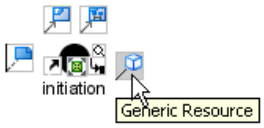
An initial node is created. The caption of initial node is hidden by default. If you want to show it, right click on the diagram background and select **Presentation Options** > **Show Shape Caption** > **Initial Node** from the pop-up menu.



Show caption of initial node

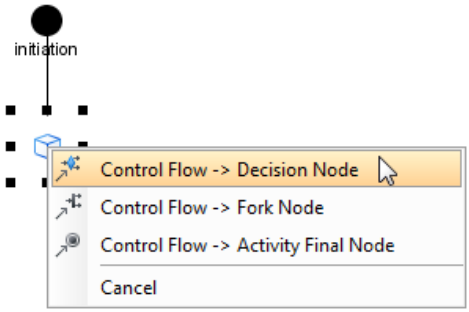
Creating decision node

To create a decision node from an initial node, move the mouse over the initial node and press its resource icon **Generic Resource**.



Generic resource

Drag it to your preferred place and then release the mouse button. Select **Control Flow** -> **Decision Node** from the pop-up menu.

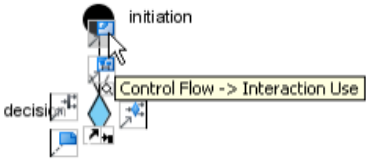


Create decision node

If you want to show the caption of decision node, right click on the diagram background and select **Presentation Options** > **Show Shape Caption** > **Decision Node** from the pop-up menu.

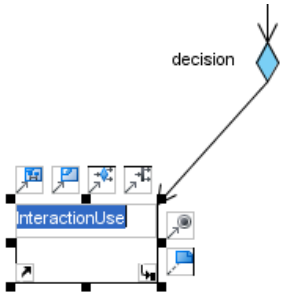
Creating interaction use

To create an interaction use, move the mouse over a shape and press its resource icon **Control Flow** -> **Interaction Use**.



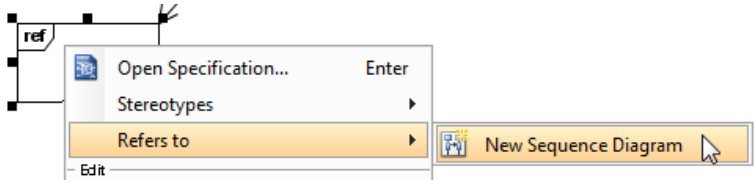
Create interaction use

Drag it to your preferred place and then release the mouse button. An interaction use is created and connected to the shape you selected with a control flow.



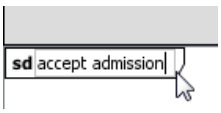
Interaction use and control flow created

You can make the interaction use refer to a diagram by right clicking on it and select **Refers to** > **New Sequence Diagram** from the pop-up menu.



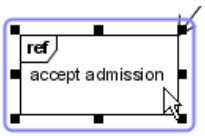
Interaction use refers to diagram

When a new sequence diagram is created, rename it.



Rename sequence diagram

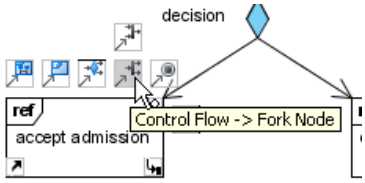
When you return to the interaction overview diagram, you can see the interaction use caption shows the diagram it refers to.



Interaction use caption updated

Creating fork node

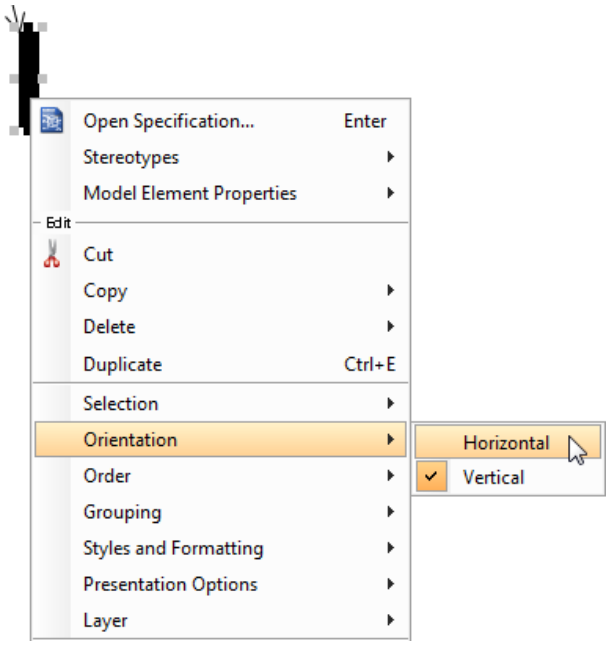
To create a fork node, move the moue over a shape and press its resource icon **Control Flow** -> **Fork Node**.



Create fork node

Drag it to your preferred place and then release the mouse button. A fork node is created and connected to the shape you selected with a control flow.

The fork node created is vertical by default. If you want to change it to horizontal, right click on the fork node and select **Orientation** > **Horizontal** from the pop-up menu.

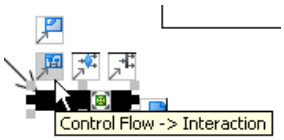


Change fork node orientation

Moreover, if you want to show the caption of fork node, right click the diagram and select **Presentation Options** > **Show Shape Caption** > **Fork Node** from the pop-up menu.

Creating interaction

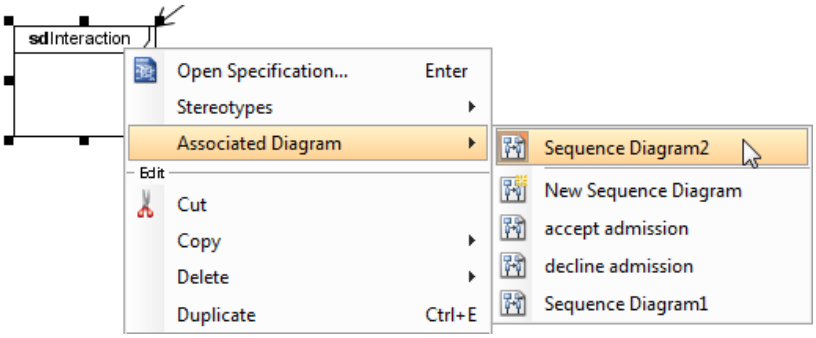
To create an interaction, move the mouse over a shape and press its resource icon **Control Flow** -> **Interaction**.



Create interaction

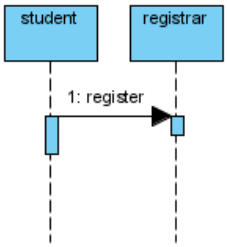
Drag it to your preferred place and then release the mouse button. An interaction is created and connected to the shape you selected with a control flow.

A new sequence diagram is created and associated with an interaction by default. To open it, right click on the interaction and select **Associated Diagram** > (target diagram name).



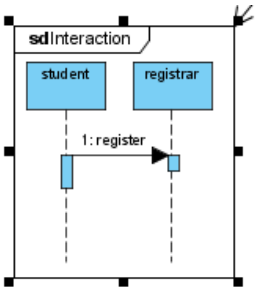
Open associated diagram of interaction

Draw the sequence diagram.



Sequence diagram

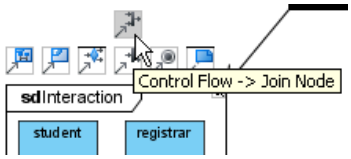
When you return to the interaction overview diagram, you can see the interaction shows the thumbnail of the sequence diagram.



Updated diagram thumbnail in interaction

Creating join node

To create a join node, move the mouse over a shape and press its resource icon **Control Flow** -> **Join Node**.



Create join node

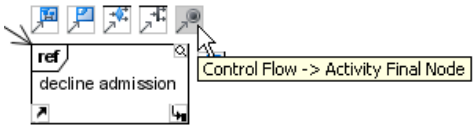
Drag it to your preferred place and then release the mouse button. A join node is created and connected to the shape you selected with a control flow.

The join node created is vertical by default, to change it to horizontal, right click on the join node and select **Orientation** > **Horizontal** from the pop-up menu.

Moreover, if you want to show the caption of join node, right click the diagram background and select **Presentation Options** > **Show Shape Caption** > **Join Node** from the pop-up menu.

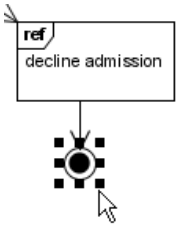
Creating activity final node

To create an activity final node, move the mouse over a shape and press its resource icon **Control Flow** -> **Activity Final Node**.



Create activity final node

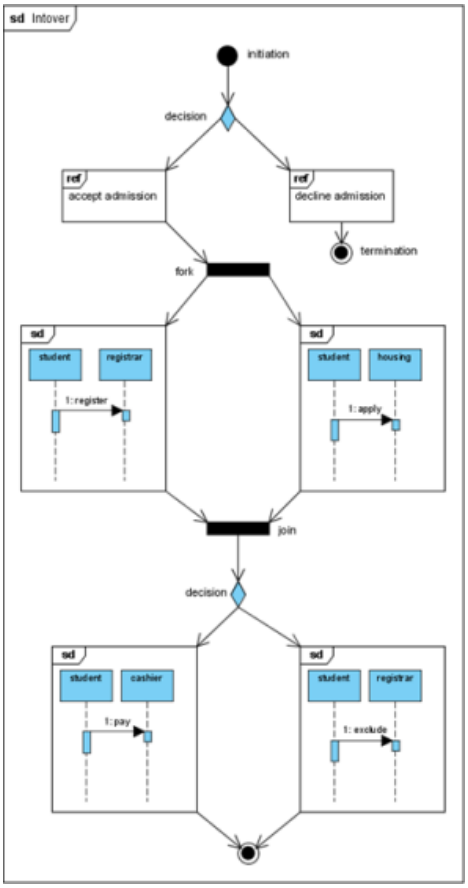
Drag it to your preferred place and then release the mouse button. An activity final node is created and connected to the shape you selected with a control flow.



Activity final node and control flow created

If you want to show the caption of activity final node, right click on the diagram background and select **Presentation Options** > **Show Shape Caption** > **Activity Final Node** from the pop-up menu.

Continue to complete the diagram.



Completed diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Zachman Framework

Zachman Framework provides structured and disciplined way of defining an enterprise.

Creating Zachman Framework

How to create Zachman Framework

Editing cell in Zachman Framework

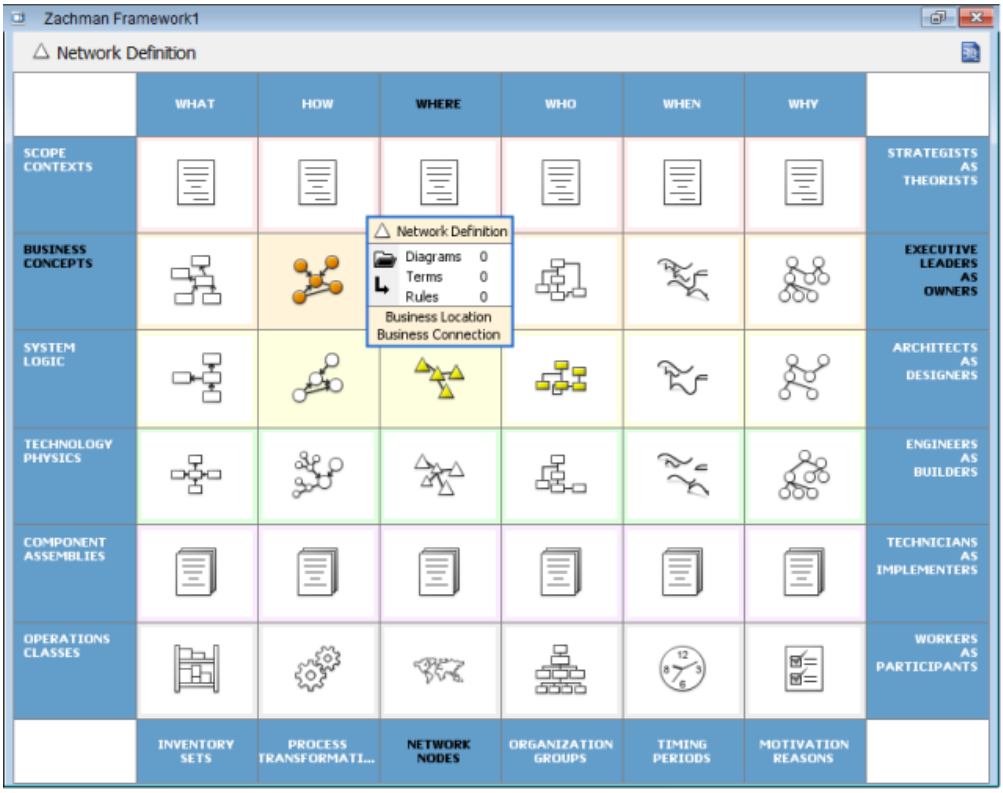
Diagrams, terms and business rules can be added to cells in Zachman Framework to describe enterprise from different perspectives.

Collapsing/Expanding rows or columns

Collapse the non-related rows and columns to make the remaining cell be focused.

Creating Zachman Framework

Zachman Framework provides structured and disciplined way of defining an enterprise. It has a matrix representation, with six rows (scope contexts, business concepts, system logic, technology physics, component assemblies, operations classes) and six columns (what, how, where, who, when, why). By adding proper diagrams, terms or business rules into cells, enterprise can be defined.

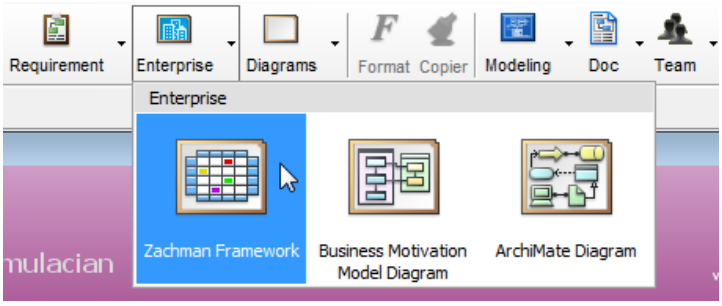


Zachman Framework

Creating Zachman Framework

To create a Zachman Framework, take any of the steps below:

- Click on **Enterprise** on toolbar and select **Zachman Framework**
- Right click on **Zachman Framework** in **Diagram Navigator** and select **New Zachman Framework** from the popup menu.
- Select **File > New Diagram > Enterprise Modeling > Zachman Framework** from the main menu.



Create a Zachman Framework through toolbar

Related Resources

The following resources may help you learn more about the topic discussed in this page.

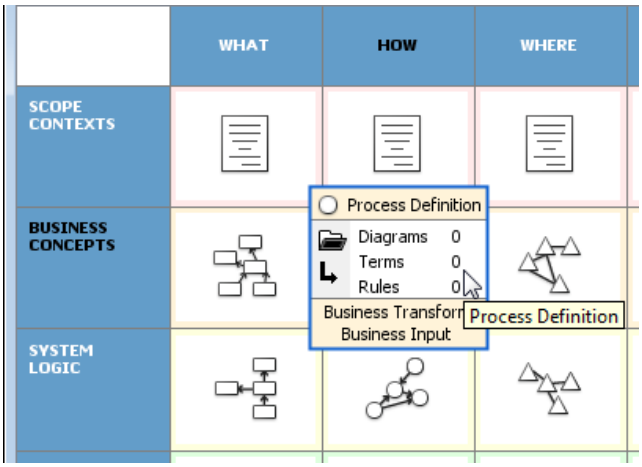
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Editing cell in Zachman Framework

Diagrams, terms and business rules can be added to cells in Zachman Framework to describe enterprise from different perspectives. To edit a cell, click on it. Then, click on appropriate link to add things to cell.

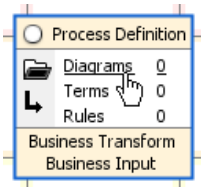
Add/Edit diagrams

1. Click on the cell you want to edit.



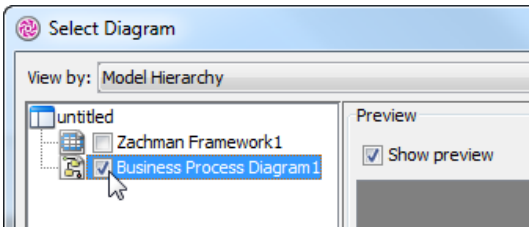
To edit the Process Definition cell

2. Click on the **Diagrams** link.



Click on the Diagrams link

3. In the window popped up, click **Add...** under the **Diagrams** tab.
4. In the **Select Diagram** window, select the diagram(s) to add to cell and click **OK**.



Select diagram(s) to add

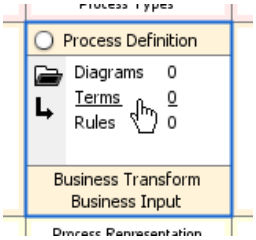
> **NOTE:** You can check **Add as sub diagram** to make the selected diagrams be added to the sub-diagrams of the cell element. When unchecked, the selected diagram will have their parent elements unchanged.

5. Click **OK** to return to diagram. You can see that the symbol in the edited cell is highlighted.

Add/Edit terms

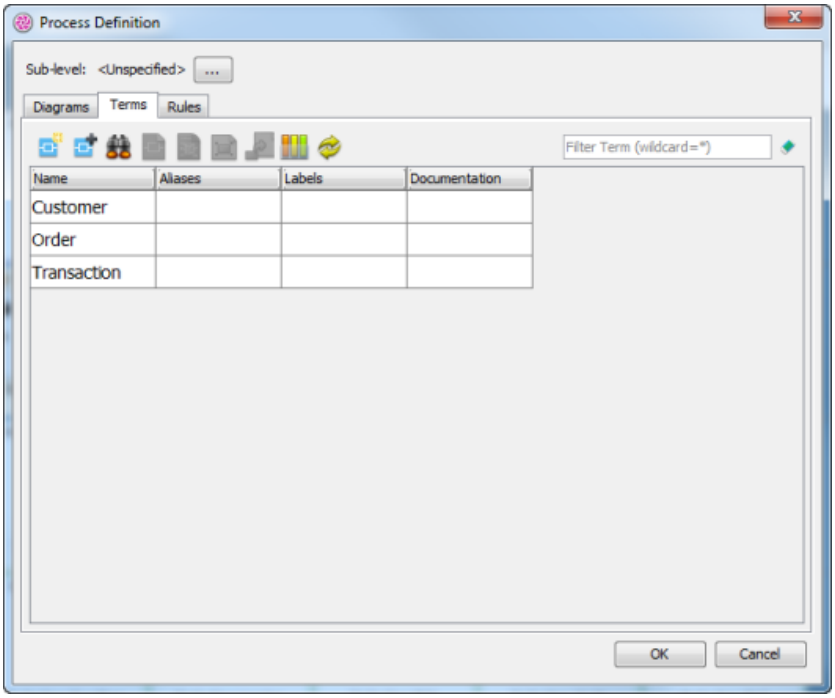
Here 'terms' refers to glossary terms. You can add terms to a cell.

1. Click on the cell you want to edit.
2. Click on the **Terms** link in the cell.



Click on the Terms link

3. This pop up a window with **Terms** tab selected. If you want to define a term here, click **New Term** in toolbar, which is the first button. Then, enter the name of the term. If you want to add reference to an existing term, click on the **Add Existing Terms** button, which is the second button. Then, select the terms to add an in the popup window and click **OK** to confirm.



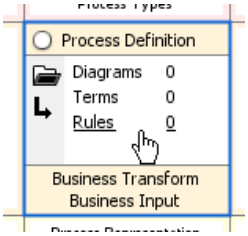
Terms added to cell

4. Click **OK** to return to diagram. You can see that the symbol in the edited cell is highlighted.

Add/Edit rules

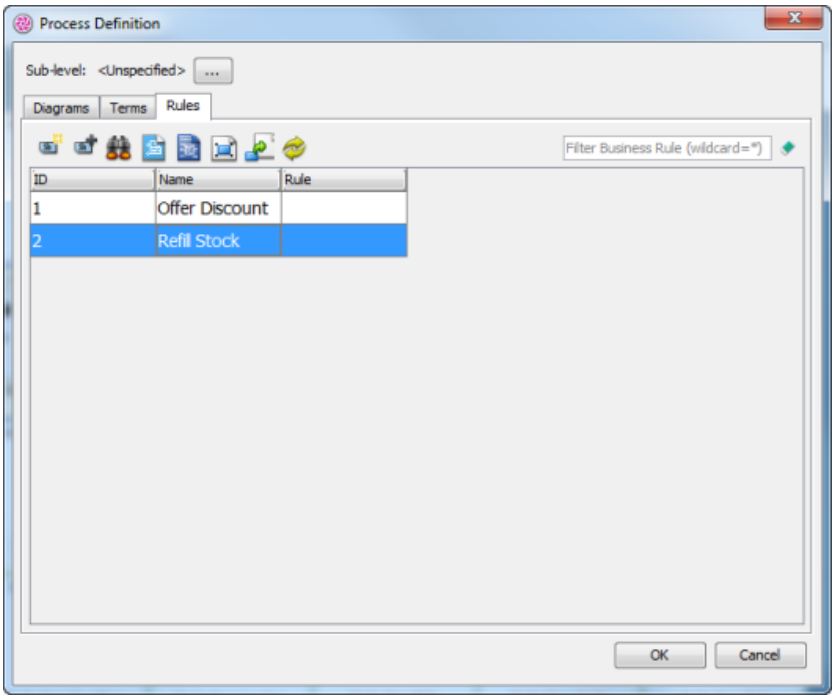
Business rules can also be added to cell.

1. Click on the cell you want to edit.
2. Click on the **Rules** link in the cell.



Click on the Rules link

3. This pop up a window with **Rules** tab selected. If you want to define a rule here, click **New Business Rule** in toolbar, which is the first button. Then, enter the name of therule . If you want to add reference to an existing rule, click on the **Add Existing Rules** button, which is the second button. Then, select the rules to add an in the popup window and click **OK** to confirm.

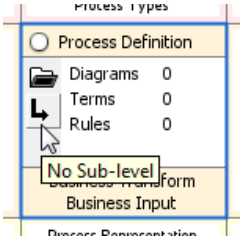


Rules added to cell

4. Click **OK** to return to diagram. You can see that the symbol in the edited cell is highlighted.

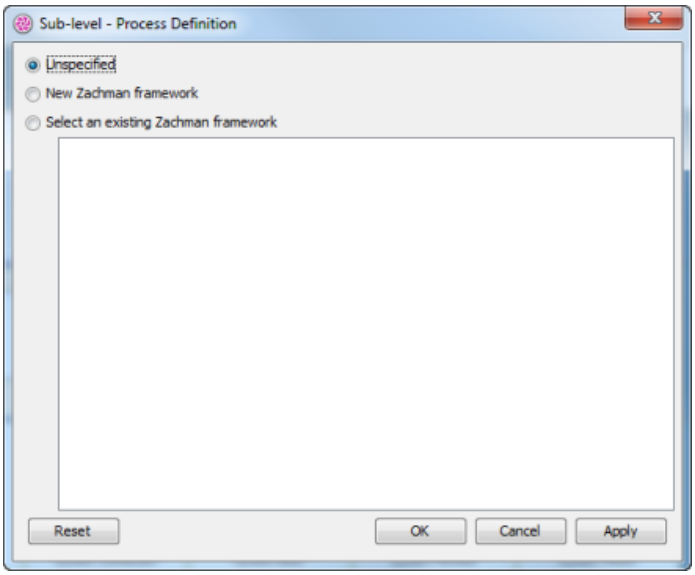
Forming sub-level

1. Click on the cell you want to edit.
2. Click on the button at the left hand side of the cell for adding sub-level.



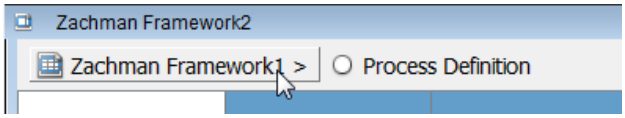
Form sub-level of cell

3. In the **Sub-level** window, select either of the following and click **OK**.
   - **Unspecified** - Nothing will happen (same as clicking **Cancel** directly)
   - **New Zachman framework** - Create a new Zachman Framework and add it as the sub-level of the editing cell.
   - **Select an existing Zachman framework** - Select a Zachman Framework created before to be the sub-level of the editing cell.



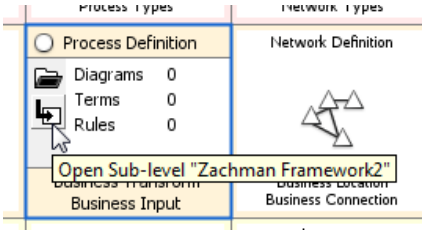
Select the Zachman Framework to add as sub-level

4. Click **OK**. Now, you are on the sub-level. To go back to the previous level, you may click on the link at the top of Zachman Framework. Cells with sub-level added will have their background painted.



To go back to the parent level

You can open the sub-level by clicking on the same button as clicked in step 2.



Open sub-level

Related Resources

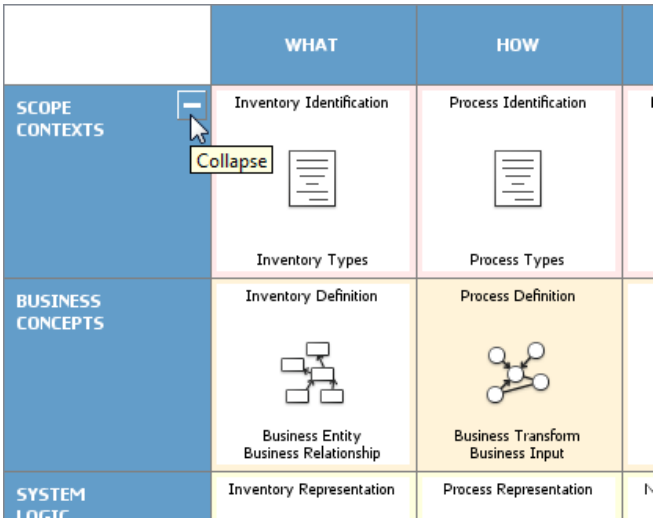
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Collapsing/Expanding rows or columns

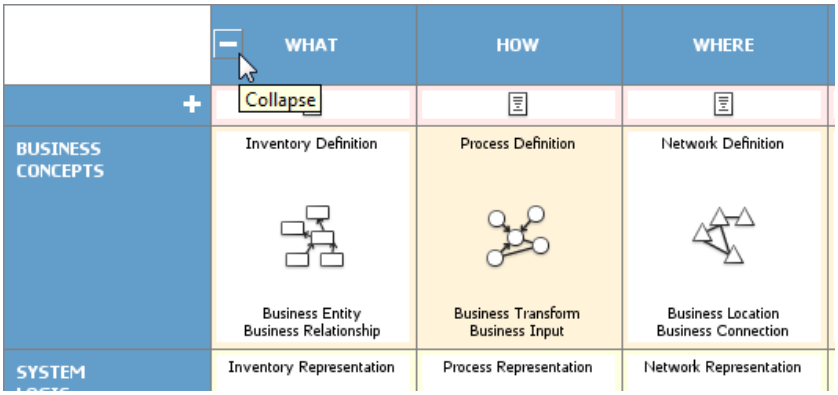
By default, you can see the names, symbols and descriptions in all cells in Zachman Framework. If you want to focus on specific cell, you can collapse the non-related rows and columns to make the remaining cell be focused. Collapsed cells show only the tiny symbol without showing any name and description.

1. To collapse a row, click on its **Collapse** button.



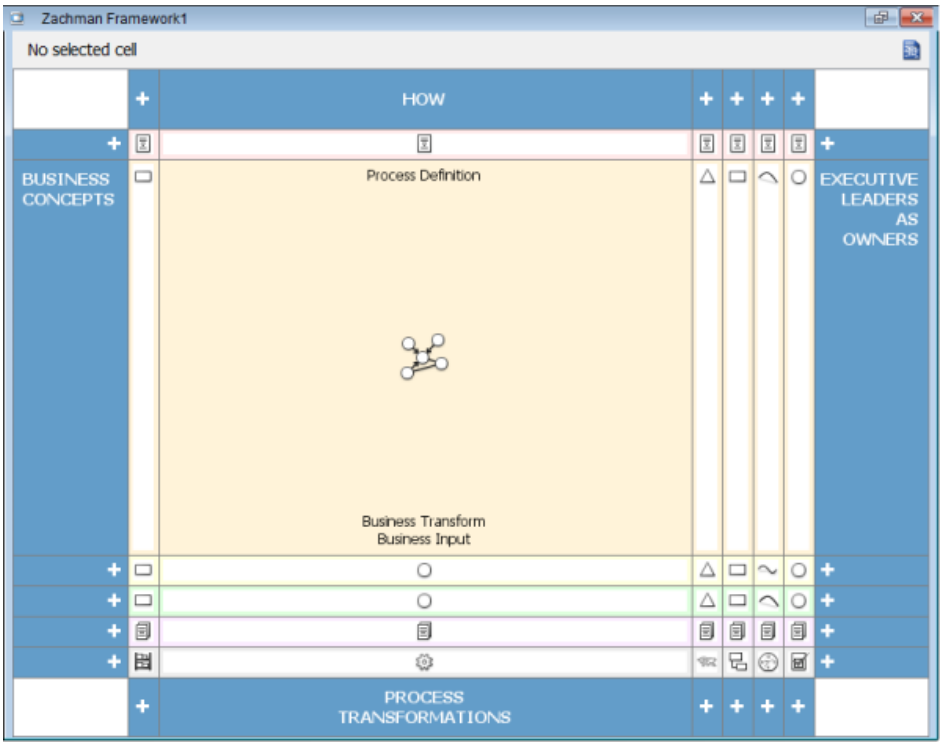
Collapse the Scope Contexts row

Similarly, you can click on the **Collapse** button of column to collapse it.



Collapse the What column

2. Collapse the non-interested rows and columns to make the interested cell remain expanded and dominate the matrix.



Cells collapsed

On the contrary, you can click on the **Expand** button (+) to expand rows/columns.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Business Motivation Model diagram

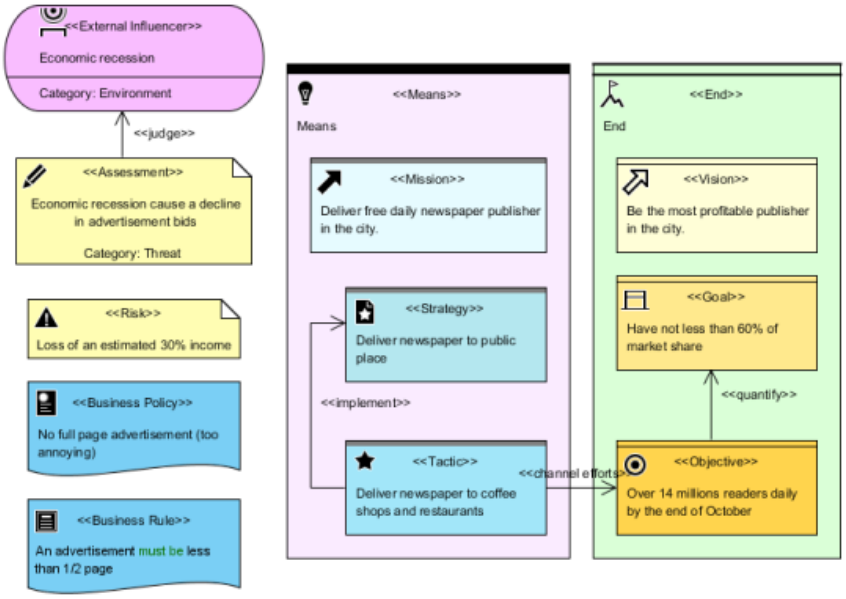
A Business Motivation Model provides business enterprises a set of notations for forming business plans.

Creating Business Motivation Model diagram

Learn how to create a BMM.

Creating Business Motivation Model diagram

A Business Motivation Model provides business enterprises a set of notations for forming business plans. It models things that the enterprise wishes to achieve, how to achieve, potential impacts, resources and etc.

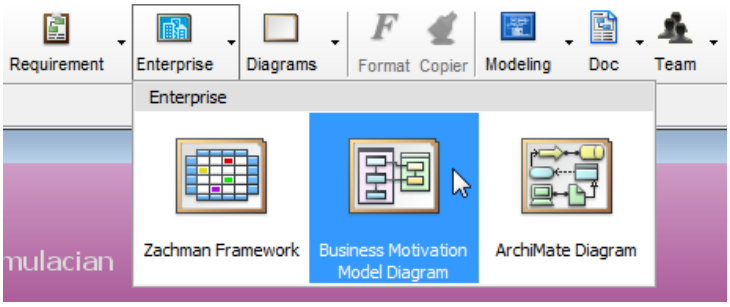


A sample Business Motivation Model diagram

Creating Business Motivation Model (BMM) diagram

To create a BMM diagram, take any of the steps below:

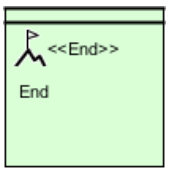
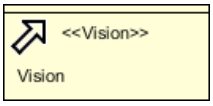
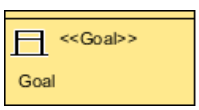
- Click on **Enterprise** on toolbar and select **Business Motivation Model Diagram**
- Right click on **Business Motivation Model Diagram** in **Diagram Navigator** and select **New Business Motivation Model Diagram** from the popup menu.
- Select **File > New Diagram > Enterprise Modeling > Business Motivation Model Diagram** from the main menu.

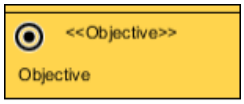
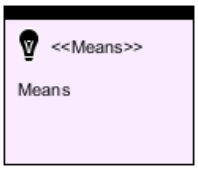
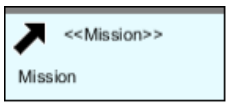
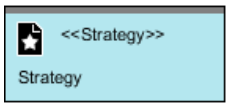
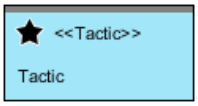
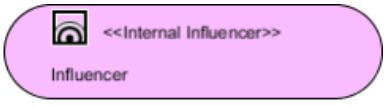


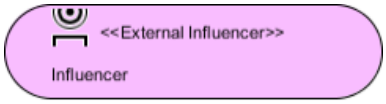
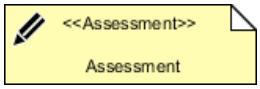
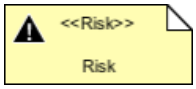
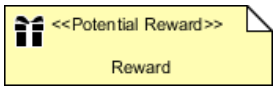
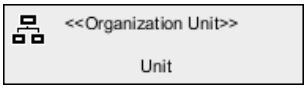
Create a BMM through toolbar

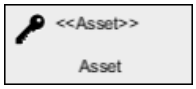
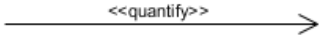
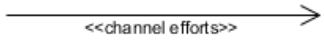
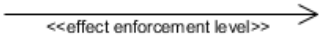
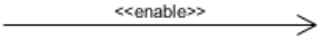
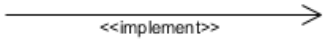
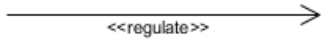
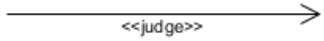
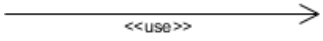
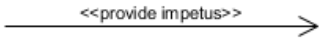
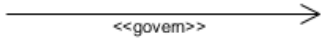
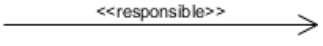
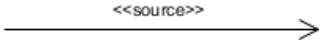
Notations

The description of notations is taken from OMG BMM Specification v1.1.

Name	Representation	Description
End		Ends are about what an enterprise wants to be. Ends can be about changing what the enterprise is (e.g., developing new lines of business, moving into new markets) orabout maintaining its current position relative its market and competition. The definition of an end does not say how it will be achieved.
Vision		A Vision describes the future state of the enterprise, without regard to how it is to be achieved. A Vision is the ultimate, possibly unattainable, state the enterprise would like to achieve. A Vision is often compound, rather than focused toward one particular aspect of the business problem. A Goal, in contrast, should generally be attainable and should be more specifically oriented to a single aspect of the business problem.
Goal		A Goal is a statement about a state or condition of the enterprise to be brought about or sustained through appropriate Means. A Goal amplifies a Vision; that is, it indicates what must be satisfied on a continuing basis to effectively attain the Vision.

Objective		An Objective is a statement of an attainable, time-targeted, and measurable target that the enterprise seeks to meet in order to achieve its Goals.
Means		Means are about what an enterprise has decided to do in order to become what it wants to be. A Means is some "device, capability, regime, technique, restriction, agency, instrument, or method that may be called upon, activated, or enforced to achieve Ends." It does not include either the tasks (business processes and workflow) necessary to exploit it, nor responsibility for such tasks. In the Business Motivation Model, Means are organized into Mission, Courses of Action, and Directives. A Mission indicates the ongoing operational activity of the enterprise. Its definition should be broad enough to cover all Strategies and the complete area of operations. An enterprise can use the Business Motivation Model without defining a Mission explicitly.
Mission		A Mission indicates the ongoing operational activity of the enterprise. The Mission describes what the business is or will be doing on a day-to-day basis. A Mission makes a Vision operative; that is, it indicates the ongoing activity that makes the Vision a reality. A Mission is planned by means of Strategies.
Strategy		A Strategy is one component of the plan for the Mission. A Strategy represents the essential Course of Action to achieve Ends (Goals in particular). A Strategy usually channels efforts towards those Goals. A Strategy is more than simply a resource, skill, or competency that the enterprise can call upon; rather, a Strategy is accepted by the enterprise as the right approach to achieve its Goals, given the environmental constraints and risks.
Tactic		A Tactic is a Course of Action that represents part of the detailing of Strategies. A Tactic implements Strategies. For example, the Tactic "Call first-time customers personally" implements the Strategy "Increase repeat business."
Business Policy		A Business Policy is a Directive that is not directly enforceable whose purpose is to govern or guide the enterprise. Business Policies provide the basis for Business Rules. Business Policies also govern Business Processes. The formulation of a Business Policy, which is always under the enterprise’s control, is by some party who is authorized to manage, control, or regulate the enterprise by selecting from a variety of alternatives in response to one or more Assessments. Business Policies that exist merely to enable a Strategy in a direct and trivial manner should be avoided. For example, suppose the enterprise has the Strategy "Encourage repeat business." A Business Policy that says "Repeat business should be encouraged" is trivial and does not need to be expressed. In general Business Policies exist to govern; that is, control, guide, and shape the Strategies and Tactics. For example, the Business Policy "We will not make on-site visits" governs the Strategy "Encourage repeat business," as well as the specific Tactics that might be selected to implement the Strategy. Specifically, no Tactic requiring on-site visits will be permitted to support the Strategy; even though on-site visits would probably be effective in that regard. On the other hand, a Tactic involving sending coupons by mail would be acceptable under the Business Policy since it involves no onsite visits.
Business Rule		A Business Rule is a Directive, intended to govern, guide, or influence business behavior, in support of Business Policy that has been formulated in response to an Opportunity, Threat, Strength, or Weakness. It is a single Directive that does not require additional interpretation to undertake Strategies or Tactics. Often, a Business Rule is derived from Business Policy. Business Rules guide Business Processes. Formally, a Business Rule is a rule that is under business jurisdiction. A rule always introduces an obligation or necessity.
Internal Influencer		An Influencer is something that can cause changes that affect the enterprise in its employment of its Means or achievement of its Ends. Alternatively, it might confirm that there are no changes where changes might have been expected. Influencers may be Internal (from within the enterprise) or External (from outside the enterprise boundary). If the enterprise being modeled is an

		Organization Unit within a larger organization, it may choose to treat the larger organization as an External Influencer. The Business Motivation Model provides an example set of categories of Influencer. In practice, enterprises define their own set.
External Influencer		An Influencer is something that can cause changes that affect the enterprise in its employment of its Means or achievement of its Ends. Alternatively, it might confirm that there are no changes where changes might have been expected. Influencers may be Internal (from within the enterprise) or External (from outside the enterprise boundary). If the enterprise being modeled is an Organization Unit within a larger organization, it may choose to treat the larger organization as an External Influencer. The Business Motivation Model provides an example set of categories of Influencer. In practice, enterprises define their own set.
Assessment		An influence (a change caused by an Influencer) is neutral. It is more or less simply just 'there' until the enterprise decides how to react to it. An Assessment is a judgment about the influence on the enterprise’s ability to employ its Means or achieve its Ends. The decisions are reflected in changes to the Ends and/or Means. Different people might make different Assessments of a given influence on the same Ends and Means, perhaps even the same people at different points in time. The model supports a record of which people made which Assessments and when, providing an audit trail for future reference. The Business Motivation Model suggests SWOT (Strength, Weakness, Opportunity, Threat) as an example of an approach for making assessments. In practice, enterprises can substitute different approaches. The model also includes Potential Impacts that can be identified to support Assessments. Potential Impacts are categorized as Risk and Potential Reward. As well as more general associations between Assessment, Ends and Means, there is a direct association "Directive is motivated by Potential Impact." This is one of the minor enhancements in version 1.1 of the Business Motivation Model, based on experience of using the model in risk management.
Risk		An Assessment records judgments about the impact (or potential for impact) of some Influencer on Ends and/or Means in terms of Potential Impacts. In other words, an Assessment identifies some Potential Impact(s) that is/are significant to that Assessment. Each Potential Impact is an evaluation that quantifies or qualifies some aspect of an Assessment in specific terms, types, or dimensions. A Potential Impact significant to an Assessment can provide the impetus for Directives that govern Courses of Action or support the achievement of Ends. An Influencer may lead to the creation of a Business Policy only through an Assessment having been made that identifies some Potential Impact. Potential Impacts are categorized as follows: Risk, Potential Reward. Typically, Risks are regarded to be negative impacts, whereas Rewards are considered positive.
Potential Reward		An Assessment records judgments about the impact (or potential for impact) of some Influencer on Ends and/or Means in terms of Potential Impacts. In other words, an Assessment identifies some Potential Impact(s) that is/are significant to that Assessment. Each Potential Impact is an evaluation that quantifies or qualifies some aspect of an Assessment in specific terms, types, or dimensions. A Potential Impact significant to an Assessment can provide the impetus for Directives that govern Courses of Action or support the achievement of Ends. An Influencer may lead to the creation of a Business Policy only through an Assessment having been made that identifies some Potential Impact. Potential Impacts are categorized as follows: Risk, Potential Reward. Typically, Risks are regarded to be negative impacts, whereas Rewards are considered positive.
Organization Unit		Organization Unit has two roles: 1. It is a concept in the Business Motivation Model, participating in the following associations: • defines Ends, • establishes Means, • makes Assessments, • recognizes Influencers, • may be defined by a Strategy, and • may be responsible for Business Processes.

		2. It is usually the basis for defining the boundaries of the enterprise being modeled. The decomposition of Business Policies, Courses of Action, and Desired Results and assignment of responsibilities within the enterprise is often guided by (or, at least, consistent with) the definition of units within the organization structure.
Asset		When Courses of Action are being defined, 'things' that are used in operating the enterprise often have to be considered. They are represented in the Model as Assets, of two kinds: • Fixed Assets - things that are kept long-term, maintained, reused, and perhaps eventually replaced. They can be tangible, such as equipment and buildings, or intangible, such as patents and licenses. • Resources - things that are consumed and replenished, such as raw materials, parts, finished goods, and cash.
Amplify		N/A
Quantify		N/A
Channel Efforts		N/A
Effect Enforcement Level		N/A
Enable		N/A
Implement		N/A
Regulate		N/A
Judge		N/A
Use		N/A
Provide Impetus		N/A
Govern		N/A
Responsible		N/A
Source		N/A
Composition		N/A

A list of supported notations in Business Motivation Model (BMM) diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

ArchiMate diagram

ArchiMate is a modeling technique for describing enterprise architectures. You will learn in this chapter how to create archimate diagram, and learn the notations supported by archimate diagram.

Creating ArchiMate diagram

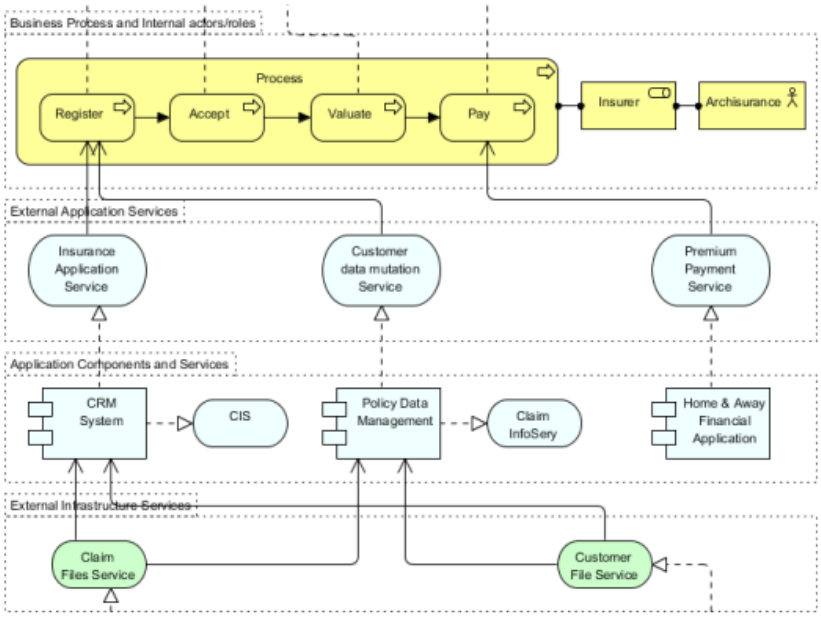
Teaches you how to create and draw an archimate diagram.

Working with Viewpoints

Teaches you how to create standard and user-defined viewpoints

Creating ArchiMate diagram

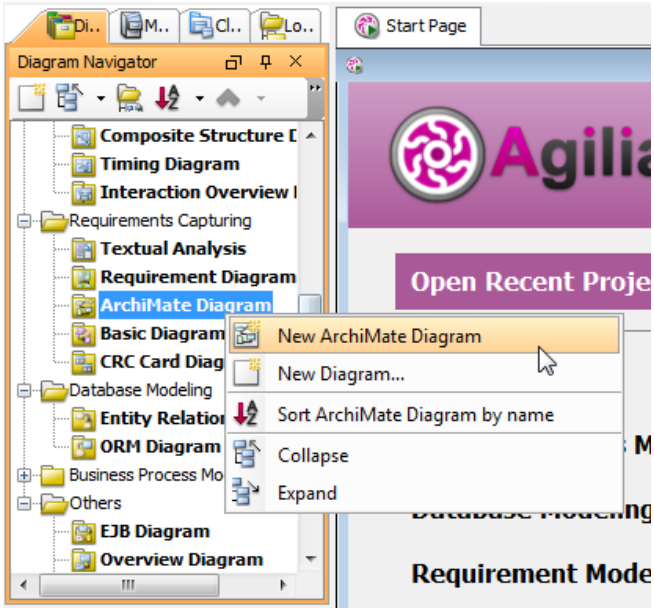
ArchiMate is a modeling technique for describing enterprise architectures. It divides architecture into three layers - business layer, application layer and technology layer. The business layer offers products and services to external customers. The application layer supports business layer and the technology layer offers infrastructural services for application layer.



A part of a sample ArchiMate diagram

Creating ArchiMate diagram

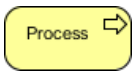
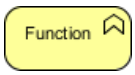
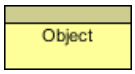
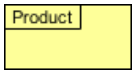
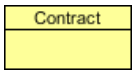
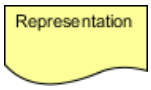
To create an ArchiMate diagram, right click on **ArchiMate Diagram** in **Diagram Navigator** and select **New ArchiMate Diagram** from the popup menu.



*To create an ArchiMate diagram through **Diagram Navigator***

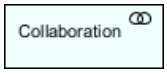
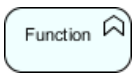
Notations - Business Layer

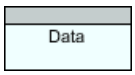
Name	Representation
Business actor	Actor
Business role	Role
Business collaboration	Collaboration

Business process	
Business function	
Business interaction	
Business event	
Business service	
Business interface	
Location	
Business object	
Product	
Contract	
Representation	
Meaning	
Value	

A list of supported notations in ArchiMate diagram, for business layer

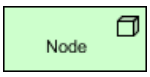
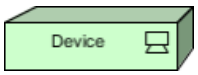
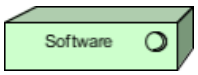
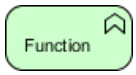
Notations - Application Layer

Name	Representation
Application collaboration	
Application component	
Application service	
Application function	
Application interaction	
Application interface	

Data object	

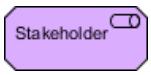
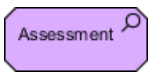
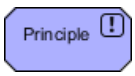
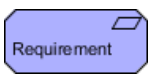
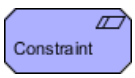
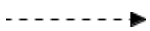
A list of supported notations in ArchiMate diagram, for application layer

Notations - Technology Layer

Name	Representation
Node	
Device	
System software	
Infrastructure interface	
Infrastructure function	
Infrastructure service	
Artifact	
Communication path	
Network	

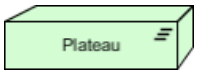
A list of supported notations in ArchiMate diagram, for technology layer

Notations - Motivation Layer

Name	Representation
Stakeholder	
Driver	
Assessment	
Goal	
Principle	
Requirement	
Constraint	
Influence	

A list of supported notations in ArchiMate diagram, for motivation layer

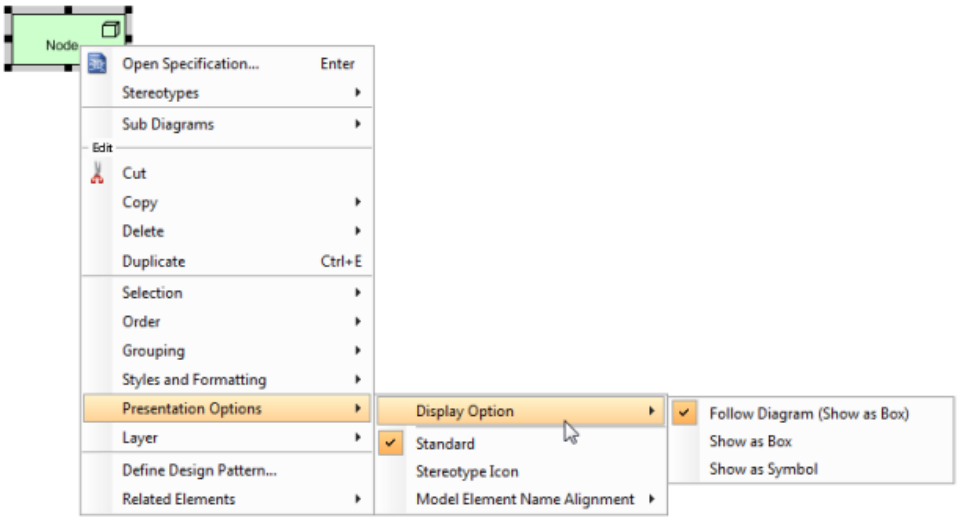
Notations - Implementation & Migration Layer

Name	Representation
Plateau	
Gap	
Deliverable	
Work Package	

A list of supported notations in ArchiMate diagram, for implementation & migration layer

Changing the appearance of some notations

Some of the ArchiMate notations support different ways of presentation. Take node in technology layer as example, to change to another presentation, right click on the node and select **Presentation Options > Display Option > Show as Box/Symbol** from the pop-up menu.



To change the presentation of a node

Node shown as box or symbol

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Working with Viewpoint

To develop and maintain an enterprise architecture requires the cooperation between different people, teams and even organizations. These stakeholders have different backgrounds, expertise and responsibilities, and have different interests, goals and needs. If there is a means to allow stakeholders to focus on particular aspects of an enterprise architecture, conerns can be better addressed, and this is what ArchiMate's viewpoint aimed to to achieve.

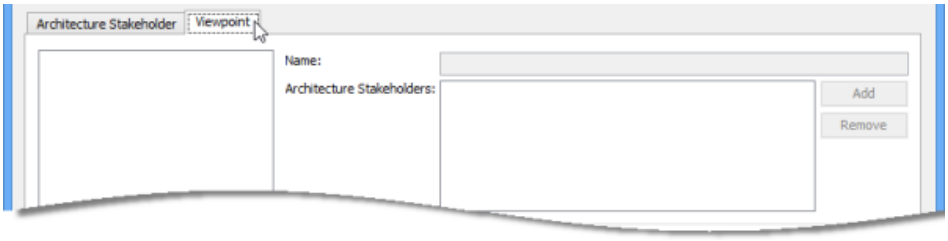
Viewpoint allows enterprise architects, designers and any other stakeholders involved in building, maintaining and participating in an enterprise architecture to define their own views on the architecture. Stakeholders views the portions they are interested in and avoid reading information that they do not care as much about.

In this page, you will learn how to add a standard viewpoint that is suggested by OpenGroup ArchiMate 2.0 specification, and how to create your own viewpoint.

Adding a Standard Viewpoint

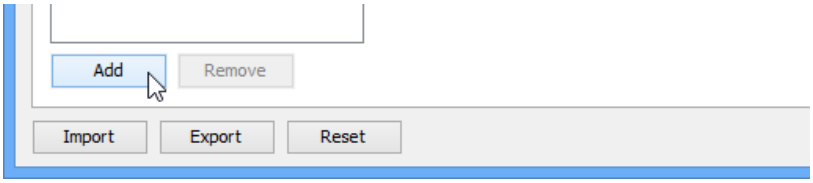
A set of standard viewpoints have been suggested by OpenGroup. In this section, you will learn how to add a standard viewpoint.

1. Select **Modeling > Manage Viewpoints...** from the main menu.
2. In the **Viewpoint Management** window, select the **Viewpoint** tab.



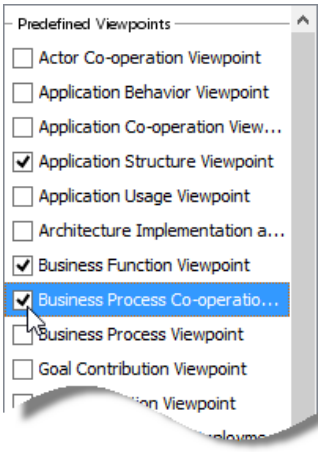
Manage viewpoints

3. At the bottom left, click **Add**.



Add a Viewpoint

4. A list of standard viewpoints are listed, known as the predefined viewpoints. Check the ones that are required by your enterprise architecture (project).



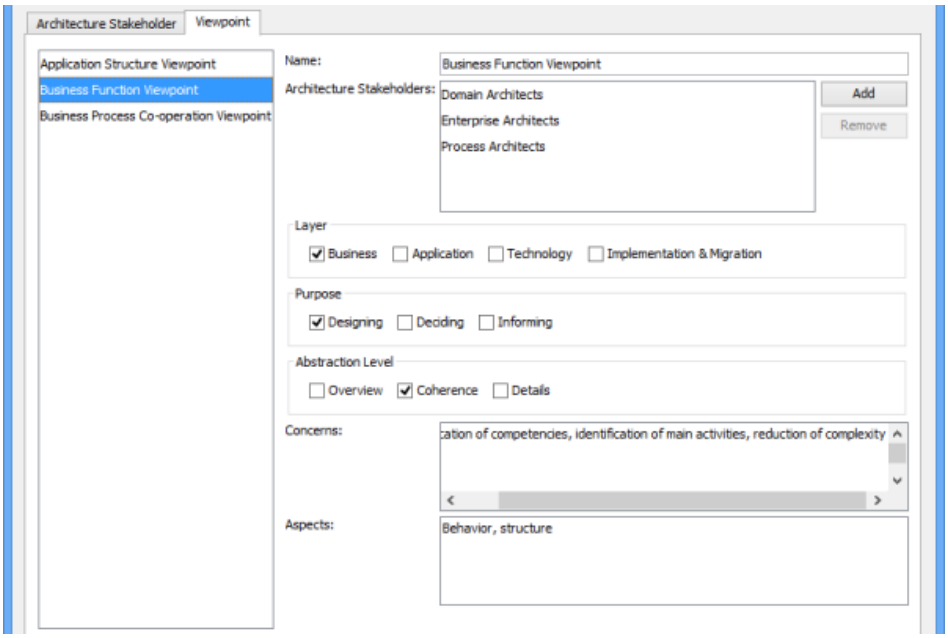
Add standard viewpoints to enterprise architecture

5. Click **Done** at bottom left to confirm your selection.
6. Each predefined viewpoint comes with a set of preset properties such as stakeholders, layers, purpose, abstraction level, concerns and aspects. You may change them by first selecting the viewpoint you want to change on the left hand side and editing the fields on the right hand side. Here is a description of properties:

Property	Description
Name	Name of the viewpoint.
Architecture Stakeholders	People who view the enterprise architecture through the viewpoint.
Layer	The perspective involved. The selection affects the visibility of tools that are available in diagram toolbar in ArchiMate diagram.

Purpose	One of the two dimensions in classifying a viewpoint. The Purpose dimension allows the classification of viewpoint base on the purpose of view. Designing - Support architectures and designers in the design process. Deciding - Assists stakeholders in decision-making. Informing - Inform stakeholders who have a need in understanding the enterprise architecture.
Abstraction Level	One of the two dimensions in classifying a viewpoint. The Abstract Level dimension classify viewpoint base on the level of detail of a view. Details - Typically consider one layer and one aspect from the ArchiMate framework. Coherence - Multiple layers or multiple aspects are spanned. Overview - Addresses both multiple layers and multiple aspects.
Concerns	Conerns of stakeholders to be addressed by the viewpoint.
Aspects	Certain aspects of enterprise architecture.

Caption Here



Properties of a viewpoint

7. Click **OK** to confirm viewpoint management and close the window.

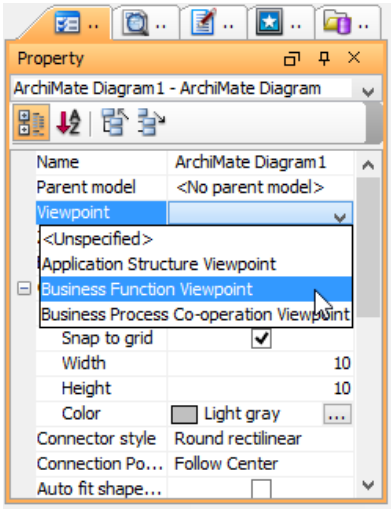
Creating Your Own Viewpoint

Instead of using standard viewpoints, you may also define your own viewpoints, with your own stakeholders. To create your own viewpoint:

1. Select **Modeling > Manage Viewpoints...** from the main menu.
2. In the **Viewpoint Management** window, select the **Viewpoint** tab.
3. At the bottom left, click **Add**.
4. Click **New**.
5. Enter the name of the stakeholder.
6. You may optionally add the stakeholder to existing viewpoints now.
7. Specify the documentation.
8. At the bottom left, click **Done** to confirm the creation of stakeholder.

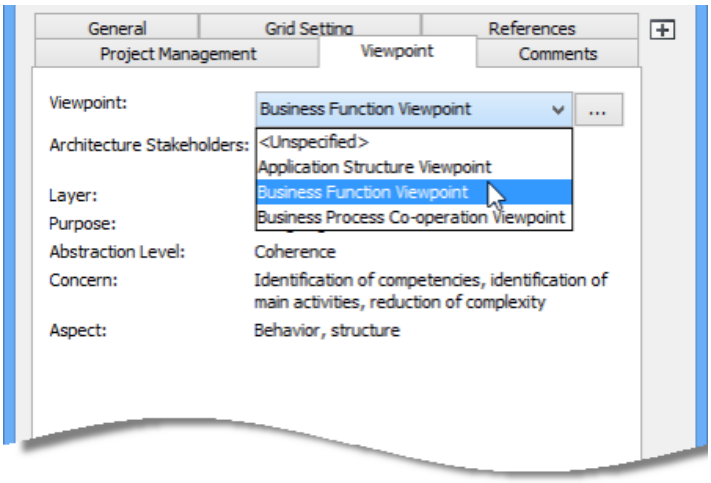
Assign a Viewpoint to Diagram (i.e. View in ArchiMate)

Once you have added or created a viewpoint, you can assign the viewpoint to diagram(s). The diagram with viewpoint specified is supposed to be designed for the stakeholders listed in that viewpoint. There are two possible ways to set viewopoint. The first way is to set via the **Property Pane**.



Set viewpoint via Property Pane

The second way is to set via the diagram specification window. Open the diagram specification, open its **Viewpoint** tab and set viewpoint there.



Set viewpoint via diagram specification window

Creating a Stakeholder

When you add a standard viewpoint, the stakeholders involved will be added to your project automatically. In addition to these 'default stakeholders', you can create your own to suit your business and problem domain.

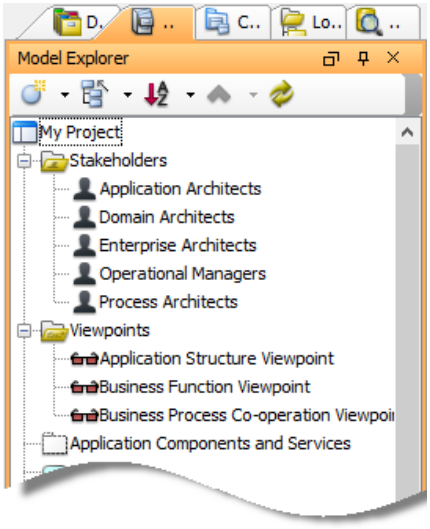
To create a stakeholder:

1. Select **Modeling > Manage Viewpoints...** from the main menu.
2. In the **Viewpoint Management** window, select the **Architecture Stakeholder** tab.
3. At the bottom left, click **Add**.
4. Click **New**.

Browsing an Enterprise Architecture with Viewpoint

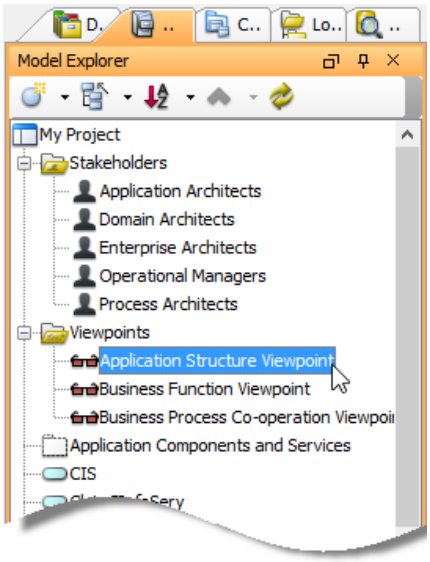
You can list the diagrams that have been assigned with certain viewpoint, or to list diagrams based on certain stakeholder. To do these:

1. Open the **Model Explorer**. You should see two nodes - Stakeholders and Viewpoints, with stakeholders and viewpoints listed.



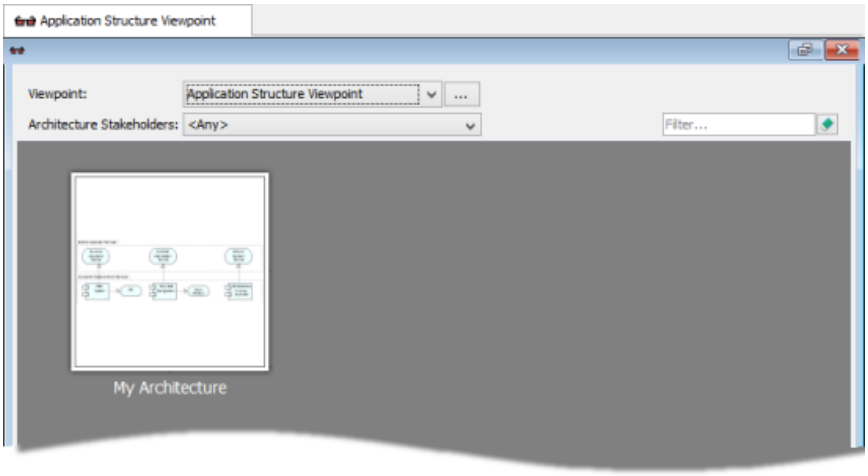
Model Explorer

2. If you want to browse enterprise architecture base on a stakeholder or viewpoint, double click on it.



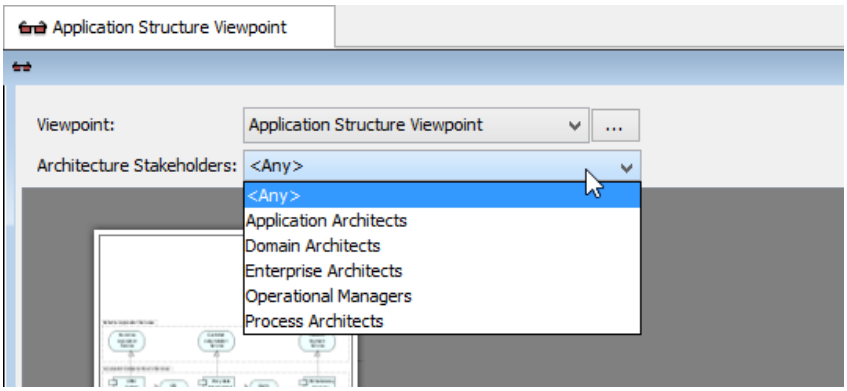
Browse an enterprise architecture from viewpoint

You will see the diagram(s) that is associated with the chosen stakeholder/viewpoint will be displayed.



Browsing enterprise architecture

3. If you want to browse with another stakeholder/viewpoint, you can update the selection in the drop down at the top of the panel.



View another stakeholder

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Service interface diagram

The interface-based approach of SoaML involves the use of simple interfaces and service interface. Simple interface focuses mainly on one-way service delivery that requires no protocol between parties.

Creating service interface diagram

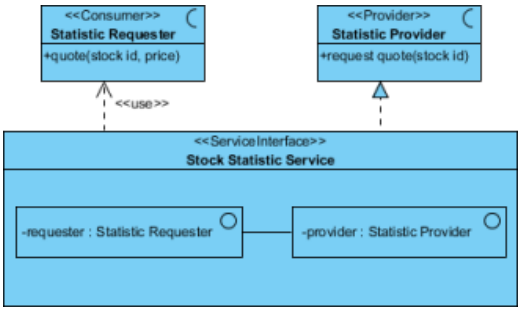
Learn how to create a service interface diagram. Have a look at the notations supported.

Creating service interface diagram

SoaML supports both a contract and an interface-based approach to SOA. They differ in the way services are specified.

The interface-based approach involves the use of simple interfaces and service interface. Simple interface focuses mainly on one-way service delivery that requires no protocol between parties. Service interface allows for bi-directional services. Provider and consumer work together to complete a service.

Service interface diagram is a type of SoaML diagram specialized for the definition and specification of both simple interface and service interface.



A sample service interface diagram

Creating service interface diagram

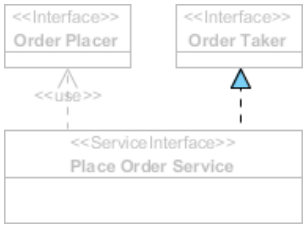
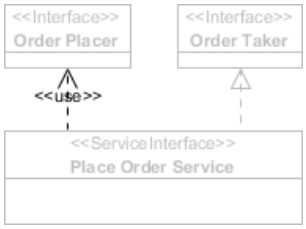
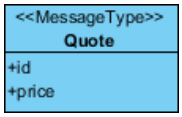
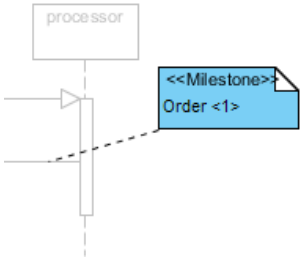
To create a SoaML service interface diagram, take any of the steps below:

- Click on **SoaML** on toolbar and select **Service Interface Diagram**
- Right click on **Service Interface Diagram** in **Diagram Navigator** and select **New Service Interface Diagram** from the popup menu.
- Select **File > New Diagram > SoaML > Service Interface Diagram** from the main menu.

Notations

The description of notations is either extracted or derived from the OMG SoaML Specification v1.0.1.

Name	Representation	Description
Service Interface	<<ServiceInterface>> **Place Order Service**	A ServiceInterface defines the interface and responsibilities of a participant to provide or consume a service. It is used as the type of a Service or Request Port. A ServiceInterface is the means for specifying how a participant is to interact to provide or consume a Service. A ServiceInterface may include specific protocols, commands, and information exchange by which actions are initiated and the result of the real world effects are made available as specified through the functionality portion of a service. A ServiceInterface may address the concepts associated with ownership, ownership domains, actions communicated between legal peers, trust, business transactions, authority, delegation, etc.
Interface	<<Interface>> **Stock Quote**	Simple interfaces define one-way services that do not require a protocol. Such services may be defined with only a single UML interface and then provided on a "Service" port and consumed on a "Request" port.
Role	-consumer : Order Taker	A ServiceInterface is a UML Class and defines specific roles each participant plays in the service interaction. These roles have a name and an interface type. The interface of the provider (which must be the type of one of the parts in the class) is realized (provided) by the ServiceInterface class. The interface of the consumer (if any) must be used by the class.
Connector	-consumer : Order Taker -provider : Order Taker	Connect roles in a service interface.
Capability	<<Capability>> **Shipping** +requestShipping()	A Capability models the ability to act and produce an outcome that achieves a result that may provide a service specified by a ServiceContract or ServiceInterface irrespective of the Participant that might provide that service. A ServiceContract, alone, has no dependencies or expectation of how the capability is realized – thereby separating the concerns of 'what" vs. "how." The Capability may specify dependencies or internal process to detail how that capability is provided including dependencies on other Capabilities. Capabilities are shown in context using a service dependencies diagram.

Expose		The Expose dependency provides the ability to indicate what Capabilities that are required by or are provided by a participant should be exposed through a Service Interface.
Dependency		The Provider may also have a uses dependency on the consumer interface, representing the fact that the provider may call the consumer as part of a bi-directional interaction. These are also known as "callbacks" in many technologies.
Realization		A ServiceInterface specifies the receptions and operations it receives through InterfaceRealizations. A ServiceInterface can realize any number of Interfaces. Some platform specific models may restrict the number of realized interfaces to at most one.
Usage		A ServiceInterface specifies its required needs through Usage dependences to Interfaces.
Message Type		A MessageType is a kind of value object that represents information exchanged between participant requests and services. This information consists of data passed into, and/or returned from, the invocation of an operation or event signal defined in a service interface. A MessageType is in the domain or service-specific content and does not include header or other implementation or protocol-specific information.
Milestone		A Milestone depicts progress by defining a signal that is sent to an abstract observer. The signal contains an integer value that intuitively represents the amount of progress that has been achieved when passing a point attached to this Milestone.

A list of supported notations in service interface diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Service participant diagram

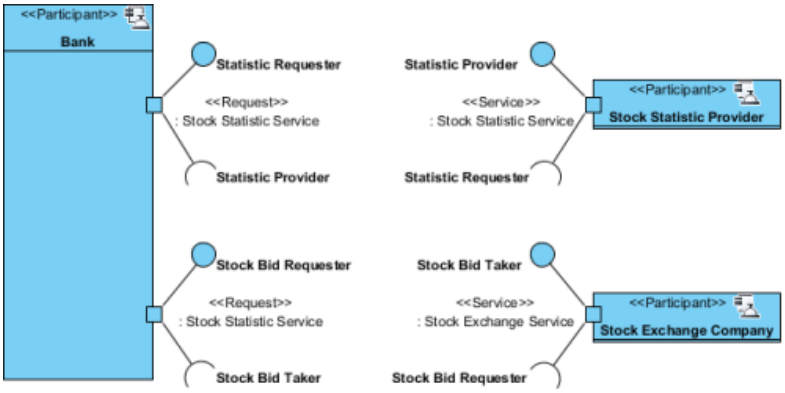
Modelers use SoaML service participant diagram to represent these participants as well as the interfaces they required or provided in accomplishing services.

Creating service participant diagram

Learn how to create a service participant diagram. Have a look at the notations supported.

Creating service participant diagram

SoaML service participant diagram focuses on the person, organization, system or anyone who take part in a services architecture. Modelers use service participant diagram to represent these participants as well as the interfaces they required or provided in accomplishing services. Note that the way how participants interact is not modeled in service participant diagram.



A sample service participant diagram

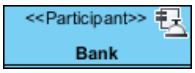
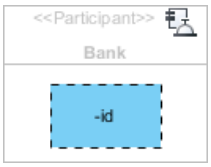
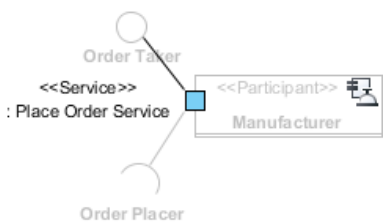
Creating service participant diagram

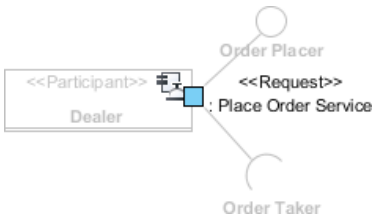
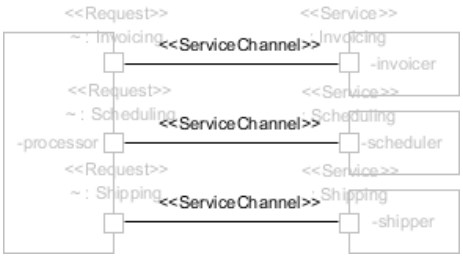
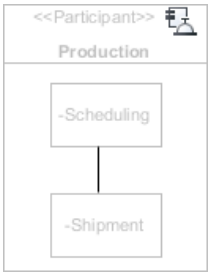
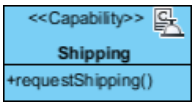
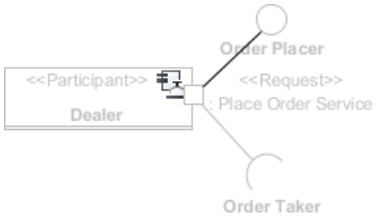
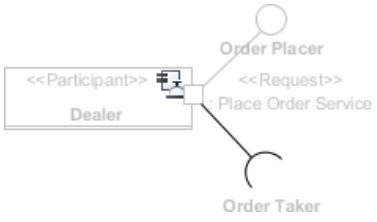
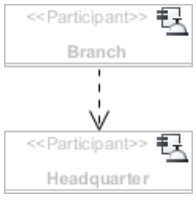
To create a SoaML service participant diagram, take any of the steps below:

- Click on **SoaML** on toolbar and select **Service Participant Diagram**
- Right click on **Service Participant Diagram** in **Diagram Navigator** and select **New Service Participant Diagram** from the popup menu.
- Select **File > New Diagram > SoaML > Service Participant Diagram** from the main menu.

Notations

The description of notations is either extracted or derived from the OMG SoaML Specification v1.0.1.

Name	Representation	Description
Participant		A Participant represents some (possibly concrete) party or component that provides and/or consumes services (participants may represent people, organizations, or systems that provide and/or use services). A Participant is a service provider if it offers a service. A Participant is a service consumer if it uses a service. A participant may provide or consume any number of services. Service consumer and provider are roles Participants play: the role of providers in some services and consumers in others, depending on the capabilities they provide and the needs they have to carry out their capabilities. Since most consumers and providers have both services and requests, Participant is used to model both.
Agent		In general, agents can be software agents, hardware agents, firmware agents, robotic agents, human agents, and so on. While software developers naturally think of IT systems as being constructed of only software agents, a combination of agent mechanisms might in fact be used from shop-floor manufacturing to warfare systems.
Part		Used as a composite component for the participant.
Property		A property is a structural feature. It relates an instance of the class to a value or collection of values of the type of the feature. A property may be designated as an identifier property, a property that can be used to distinguish or identify instances of the containing classifier in distributed systems.
Service Port		A service port is the feature that represents the point of interaction on a Participant where a service is actually provided. On a service provider this can be thought of as the "offer" of the service (based on the service interface). In other words, the service port is the point of interaction for engaging participants in a service via its service interfaces.

Request Port		A consumer of a service specifies the serviceinterface they require using a request port.
Port		Port is extended with a connectorRequired property to indicate whether a connector is required on this port, or the containing classifier may be able to function without anything connected.
Service Channel		A ServiceChannel provides a communication path between consumer Requests and provider services.
Connector		Connect parts in a participant, if any.
Capability		A Capability models the ability to act and produce an outcome that achieves a result that may provide a service specified by a ServiceContract or ServiceInterface irrespective of the Participant that might provide that service. A ServiceContract, alone, has no dependencies or expectation of how the capability is realized – thereby separating the concerns of "what" vs. "how." The Capability may specify dependencies or internal process to detail how that capability is provided including dependencies on other Capabilities. Capabilities are shown in context using a service dependencies diagram.
Realization		Connects service/request port and interface to represents the interface provided by the participant, at specific port.
Usage		Connects service/request port and interface to represents the interface required by the participant, at specific port.
Dependency		Shows that a participant/interface/port depends on a participant/interface/port.

A list of supported notations in service participant diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Service contract diagram

Service contract diagram is designed for the specification of service contract. Modelers usually combines the use of sequence diagram and activity diagram in representing the choreography of a service contract.

Creating service contract diagram

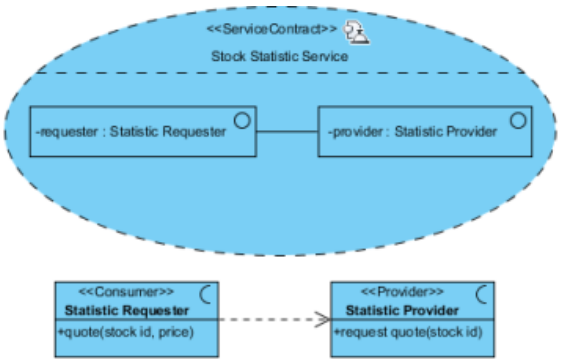
Learn how to create a service contract diagram. Have a look at the notations supported.

Creating service contract diagram

SoaML supports both a contract and an interface-based approach to SOA. They differ in the way services are specified.

The service contract approach defines the contract that specify how providers and consumers work together to achieve a goal, through the use of service. The service contract represents an agreement between parties for how the service is to be provided and consumed. Such agreement includes the interfaces, choreography and other terms and conditions.

Service contract diagram is designed for the specification of service contract. Modelers usually combines the use of sequence diagram and activity diagram in representing the choreography of a service contract.



A sample service contract diagram

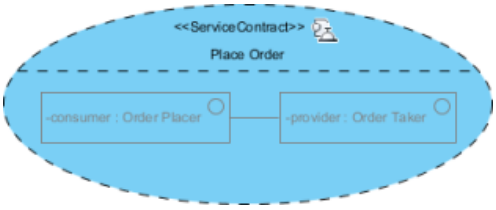
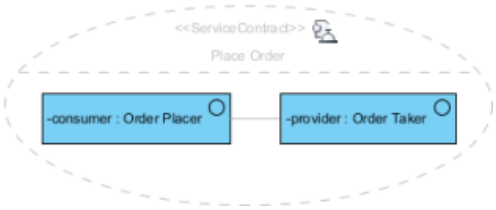
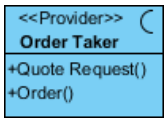
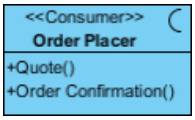
Creating service contract diagram

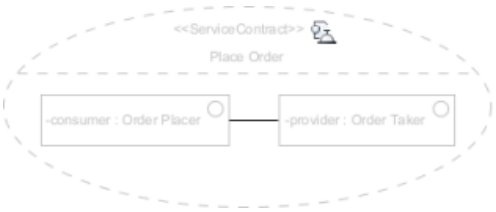
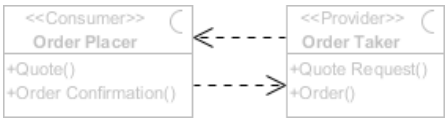
To create a SoaML service contract diagram, take any of the steps below:

- Click on **SoaML** on toolbar and select **Service Contract Diagram**
- Right click on **Service Contract Diagram** in **Diagram Navigator** and select **New Service Contract Diagram** from the popup menu.
- Select **File > New Diagram > SoaML > Service Contract Diagram** from the main menu.

Notations

The description of notations is either extracted or derived from the OMG SoaML Specification v1.0.1.

Name	Representation	Description
Service Contract		A ServiceContract is the specification of the agreement between providers and consumers of a service as to what information, products, assets, value, and obligations will flow between the providers and consumers of that service. It specifies the service without regard for realization, capabilities, or implementation. A ServiceContract does not require the specification of who, how, or why any party will fulfill their obligations under that ServiceContract, thus providing for the loose coupling of the SOA paradigm. In most cases a ServiceContract will specify two roles (provider and consumer) but other service roles may be specified as well. The ServiceContract may also own a behavior that specifies the sequencing of the exchanges between the parties as well as the resulting state and delivery of the capability. The owned behavior is the choreography of the service and may use any of the standard UML behaviors such as an interaction, timing, state, or activity diagram.
Role		ServiceContract captures an agreement between the roles played by consumers and providers of the service, their capabilities and needs, and the rules for how the consumers and providers must interact. The roles in a ServiceContract are typed by Interfaces that specify Operations and events which comprise the choreographed interactions of the services.
Provider		A "Provider" models the interface provided by the provider of a service. The provider of the service delivers the results of the service interaction. The provider will normally be the one that responds to the service interaction. Provider interfaces are used in as the type of a "ServiceContract" and are bound by the terms and conditions of that service contract.
Consumer		A «Consumer» models the interface provided by the consumer of a service. The consumer of the service receives the results of the service interaction. The consumer will normally be the one that initiates the service interaction. Consumer interfaces are used as the type of a «ServiceContract» and are bound by the terms and conditions of that service contract.

Connector		Connect roles, if any, in a participant.
Dependency		The Provider may also have a uses dependency on the consumer interface, representing the fact that the provider may call the consumer as part of a bi-directional interaction. These are also known as "callbacks" in many technologies.

A list of supported notations in service participant diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Services architecture diagram

Services architecture diagram is a SoaML diagram that represents services architecture.

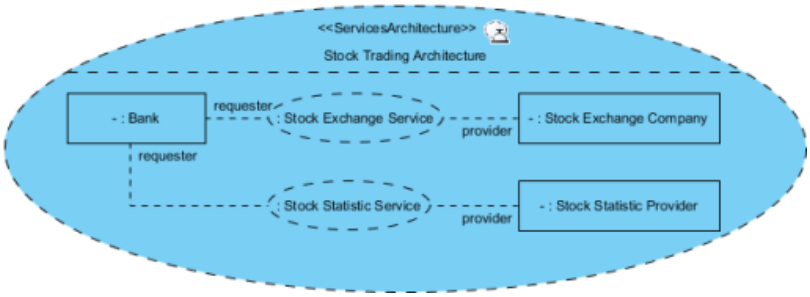
Creating services architecture diagram

Learn how to create a services architecture diagram. Have a look at the notations supported.

Creating services architecture diagram

Understanding how people, team and organizations work together for a goal enables them to work more cohesively using services, without getting overly coupled. SoaML enables modelers to build a services architecture model for this purpose. The services architecture put together the service specification and participants, and shows how they work together to achieve a goal.

Services architecture diagram is a SoaML diagram that represents services architecture.



A sample services architecture diagram

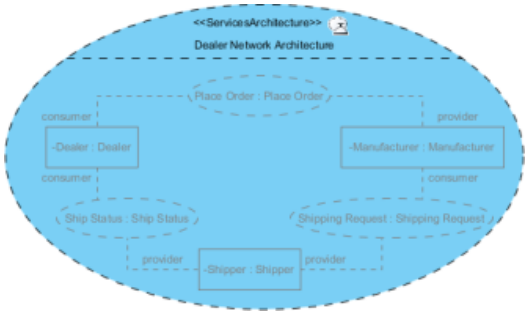
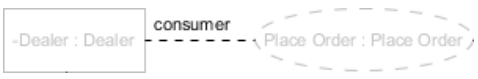
Creating services architecture diagram

To create a SoaML services architecture diagram, take any of the steps below:

- Click on **SoaML** on toolbar and select **Services Architecture Diagram**
- Right click on **Services Architecture Diagram** in **Diagram Navigator** and select **New Services Architecture Diagram** from the popup menu.
- Select **File > New Diagram > SoaML > Services Architecture Diagram** from the main menu.

Notations

The description of notations is either extracted or derived from the OMG SoaML Specification v1.0.1.

Name	Representation	Description
Services Architecture		A ServicesArchitecture (a SOA) describes how participants work together for a purpose by providing and using services expressed as service contracts. By expressing the use of services, the ServicesArchitecture implies some degree of knowledge of the dependencies between the participants in some context. Each use of a service in a ServicesArchitecture is represented by the use of a ServiceContract bound to the roles of participants in that architecture.
Internal Participant		Participants who work together in an architecture.
External Participant		External participant uses the "shared aggregation" feature of UML to show that these roles are external to the architecture while the other roles are internal.
Service Contract Use (CollaborationUse)		A CollaborationUse explicitly indicates the ability of an owning Classifier to fulfill a ServiceContract or adhere to a ServicesArchitecture. A Classifier may contain any number of CollaborationUses that indicate what it fulfills. The CollaborationUse has roleBindings that indicate what role each part in the owning Classifier plays. If the CollaborationUse is strict, then the parts must be compatible with the roles they are bound to, and the owning Classifier must have behaviors that are behaviorally compatible with the ownedBehavior of the CollaborationUse’s Collaboration type.
Role Binding		A CollaborationUse contains roleBindings that binds each of the roles of its Collaboration to a part of the containing Classifier.

A list of supported notations in services architecture diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube

- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Service categorization diagram

The SoaML service categorization diagram allows categorizing SoaML elements with catalog, categories and category values.

Creating service categorization diagram

Learn how to create a service categorization diagram. Have a look at the notations supported.

Creating service categorization diagram

In order to allow model elements to be used for multiple purposes and viewed from different perspectives, we need a way to organize the content of model. Categorization is available for such purpose.

The SoaML service categorization diagram allows categorizing SoaML elements with catalog, categories and category values.

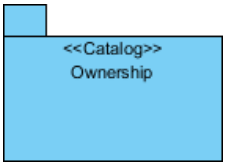
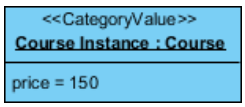
Creating service categorization diagram

To create a SoaML service categorization diagram, take any of the steps below:

- Click on **SoaML** on toolbar and select **Service Categorization Diagram**
- Right click on **Service Categorization Diagram** in **Diagram Navigator** and select **New Service Categorization Diagram** from the popup menu.
- Select **File > New Diagram > SoaML > Service Categorization Diagram** from the main menu.

Notations

The description of notations is either extracted or derived from the OMG SoaML Specification v1.0.1.

Name	Representation	Description
Catalog		A named collection of related elements, including other catalogs characterized by a specific set of categories. Applying a Category to an Element using a Categorization places that Element in the Catalog.
Category		A Category is a piece of information about an element. A Category has a name indicating what the information is about, and a set of attributes and constraints that characterize the Category. An Element may have many Categories, and the same Category can be applied to many Elements. Categories may be organized into Catalogs hierarchies.
Category Value		A CategoryValue provides values for the attributes of a Category. It may also be used to categorize model elements providing detailed information for the category.

A list of supported notations in service categorization diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Business process diagram

This chapter briefly talks about how to create a business process diagram, with description to most of the common notations, about their usage and tips when creating or editing them.

Creating business process diagram

Shows you how to create a business process diagram as well as to update the format of ID.

Pool and lane

Pool and lane are used to model roles in a process. This part will cover some of the common operations with pool and lane, such as to change their orientation, to define a black box and to move lanes up and down, left and right (depending on the orientation of pool).

Task and sub-process

Task and sub-process are used to model the activity needed to do in a process. This part will cover the use of marker, a description of various types of task, how to define working procedure for task and how to create a sub-process.

Event

An event in a business process refers to something that happens and affects the flow of process. This part will description the various types of start, intermediate and end event.

Gateway

Gateway is a kind of flow objects which is used to direct sequence flows within a process, base on certain condition(s). There are several kinds of gateway for different kinds of control behavior. We will go through each of them in detail.

Sequence and message flow

There are two types of connectors for modeling flows in a process - Sequence flow and Message flow. You will see how to correct invalid flow, and how to visualize message that pass between pools.

Choreography task and sub-process

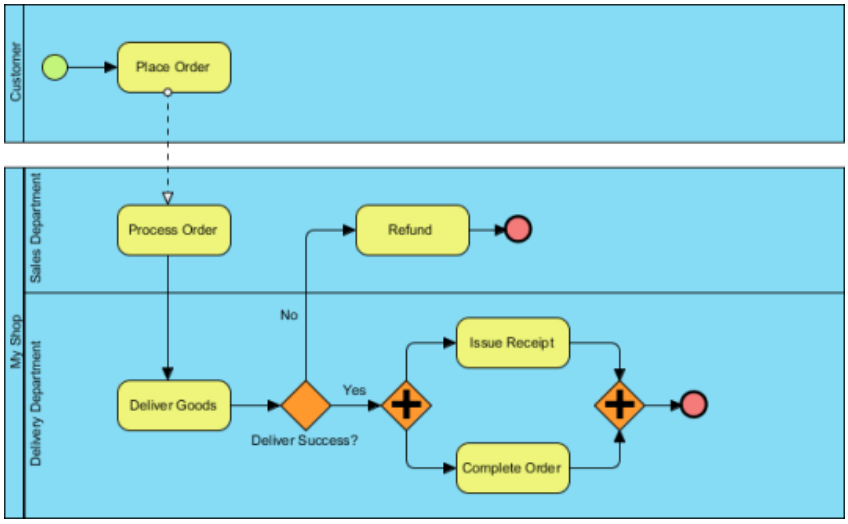
A choreography is a type of process which defines the sequence of interaction between participant. You will learn how to set participants to choreography task and sub-process.

Data object

You can use data objects to model data within process flow. You will learn how to define states for data object.

Creating business process diagram

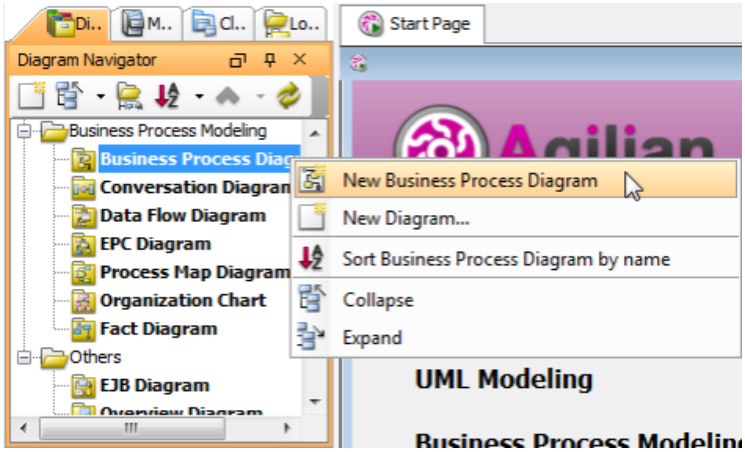
Business Process Modeling Notation (BPMN) is a graphical representation for designing and modeling business processes visually. It is a standard for business process modeling, and provides a graphical notation for specifying business processes in a business process diagram (BPD).



A sample business process diagram

Creating business process diagram

To create a business process diagram, right click on **Business Process Diagram** in **Diagram Navigator** and select **New Business Process Diagram** from the popup menu.



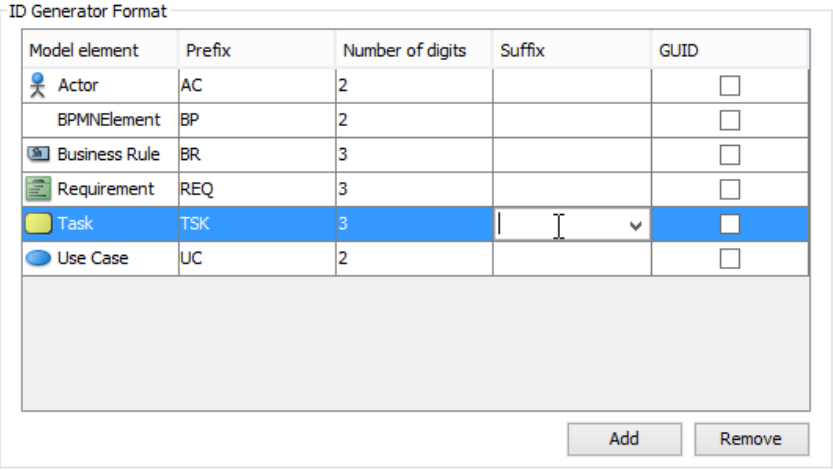
*Create a business process diagram through **Diagram Navigator***

Assigning IDs to model elements

It is possible to assign IDs to objects in business process diagram. By default, IDs are assigned by following the order of object creation, starting from number 1. However, you can define the format, or to enter an ID manually.

Defining the format of ID

To define the format of ID, open the **Project Options** window by selecting **Tools > Project Options** from the main menu. Select **Diagramming** from the list on the left hand side, and open the **Model Generation** tab. Click Add and then select the type of model element that you want to change its ID format (E.g. Task). Then, you can adjust the format say to specify the prefix, the number of digits and suffix.



Defining format of ID

Below is a description of options.

Option	Description
Prefix	Text to add before the number
Num of digits	The number of digits of the number. For example, when digit is 3, ID "1" will become "001"
Suffix	Text to append to the number
GUID	A randomly generated string. Note that the string will be very long. And by selecting this option, the prefix, num of degits and suffix options will be ignored.

Options for formatting ID

Controlling how ID is shown on new diagram

By default, ID is just a text property that won't appear on diagram. However, you can make it appear either near or within a shape.

Different looks of a task when ID is not shown, ID is shown as label and ID is shown below caption

To define the way ID is shown in BPD, open the **Project Options** window by selecting **Tools > Project Options** from the main menu. Select **Diagramming** from the list on the left hand side, and open the **Business Process > Behavior** tab.

In the **ID Generator Format** section, select whether or not to show ID on diagram, and whether to show it as label that attach to a shape, or as text below caption. If you select **Show as Customized**, you can edit the in table of model element the way ID is shown.

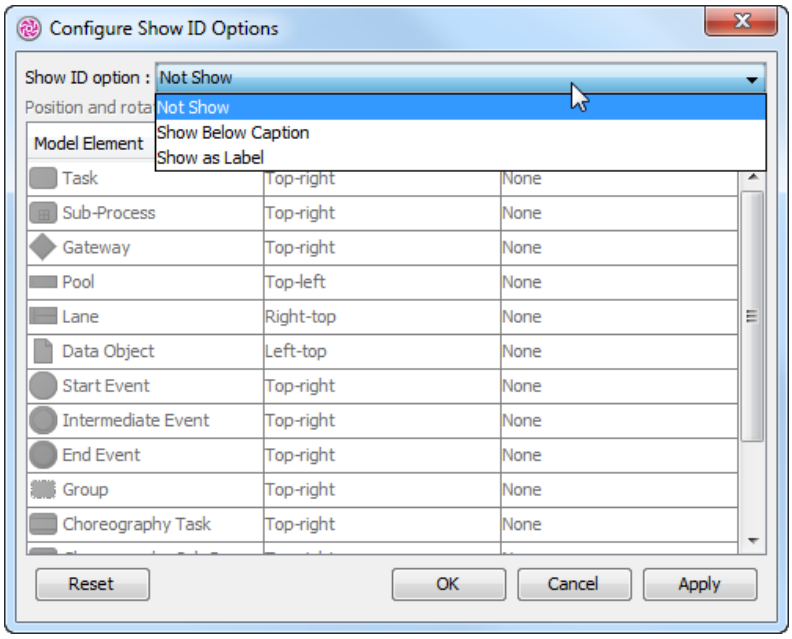
Note that the setting is effective only in newly created diagrams.

Controlling how ID is shown on an existing diagram

The previous section described how to set the appearance of ID in new diagrams. Here you can see how to change the ID appearance of an existing diagram.

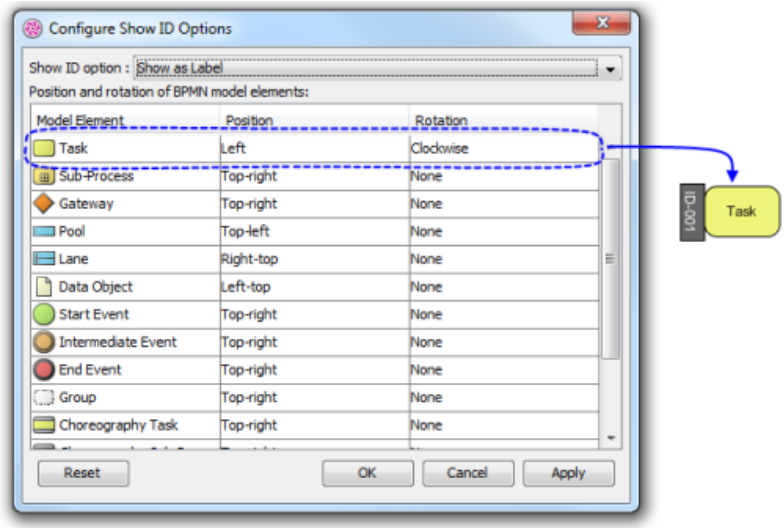
To do this, right click on the BPD you want to edit and select **Presentation Options >Configure Show ID Options...** from the pop-up menu.

In the **Configure Show ID Options** window, click on the drop down menu **Show ID Options** and select if you want to show IDs, and if yes, where to show - below caption or as a label.



To configure the whether or not to show ID

If you selected to show as label, you can adjust in further the position of ID, relative to the shape (e.g. Top right of task) and the rotation.



To make ID of task show as label, position at the left of shape

ID assignment

There are several ways that you can assign an ID to an element, including:

- Through the specification dialog box (Right click on it and select **Open Specification...** from the popup menu)
- Through the ID label (available only when ID is shown as label on diagram)
- Through the **Property Pane**

Nested ID

When you draw something in a lane/pool, or to drill down a sub-process into another level and draw something, this forms a nested element hierarchy such like a task is contained by its parent pool. ID is formed nested according to the hierarchy. For instance, while a pool has '3' as ID, its children have 3.1, 3.2... as IDs. You can turn this function on and off.

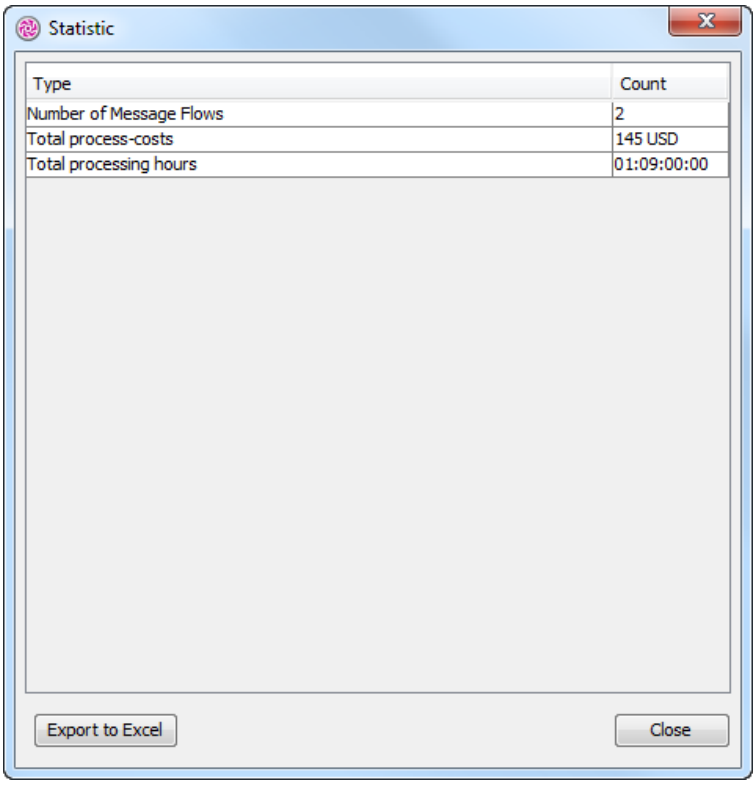
To turn on or off, right click on a BPD and select **Diagram Content > Edit IDs...** from the popup menu. At the bottom of the dialog box, check or uncheck the option **Sub-Level ID**.

Showing process statistic

Process statistic refers to the results of the statistical analysis that can be conducted upon your process. There are three types of figures: number of message flows, total process-costs and the total processing hours.

To show process statistic:

1. Right click on the background of the business process diagram.
2. Select **Utilities > Show Statistic...** from the popup menu. This opens the **Statistic** window, like this:



Process statistic

Below is a description of figures.

Figure	Description
Number of Message Flows	The number of message flows that exist in the current diagram.
Total process-costs	A summation of costs specified for tasks and sub-processes in the current diagram.
Total processing hours	A summation of duration specified for tasks and sub-processes in the current diagram.

Description of figures in process statistic

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

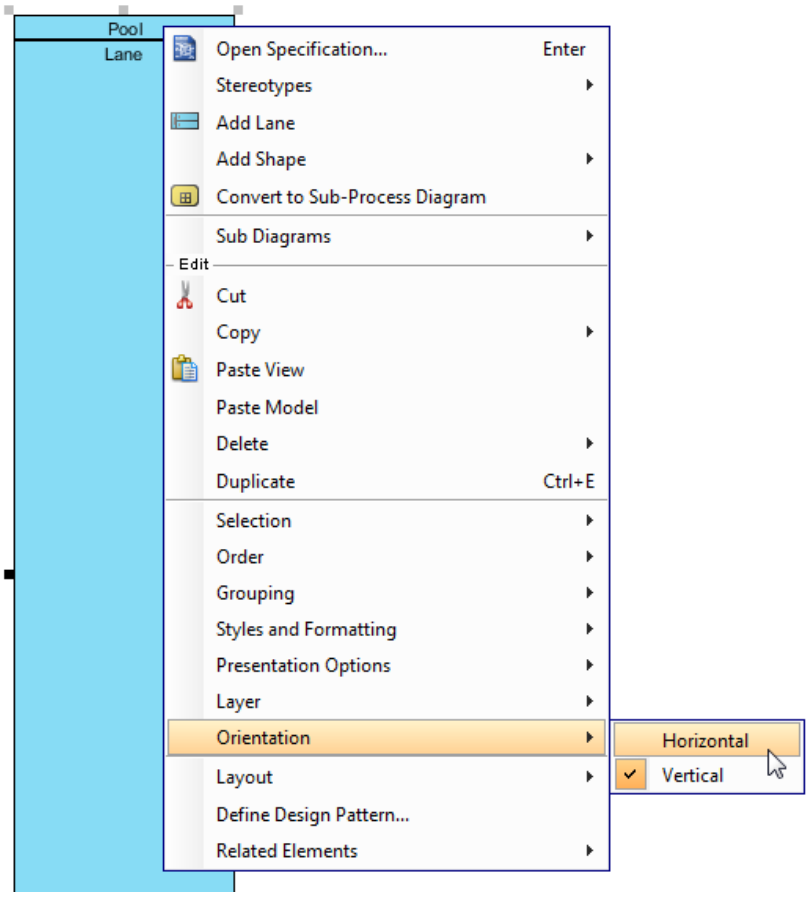
Pool and lane

A pool represents a participant who takes part in a process. It is visually a rectangular box that can contain flow objects like task and activity. A lane is a sub-partition in a pool. It is often used to represent internal roles or a department under the role represented by pool.

Horizontal pool that contains two lanes

Orientation of pool

In a business process diagram, pool and lane can show either vertically or horizontally. To change the orientation of pool and contained lane, right click on the pool header and select **Orientation > Vertical/Horizontal** from the popup menu. Note that this function is available when the pool is empty.

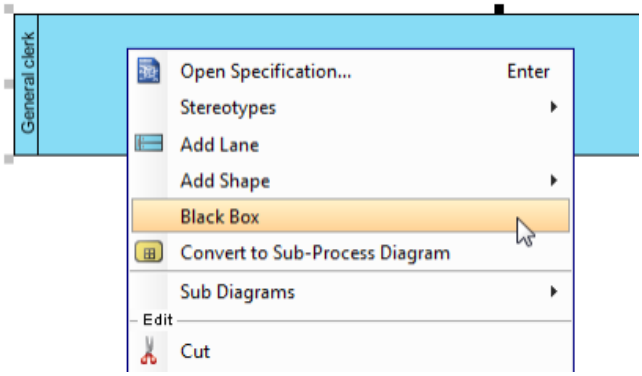


Changing the orientation of pool from vertical to horizontal

NOTE: You can only change the orientation of pool/lane when it contains no flow object

Defining black box pool

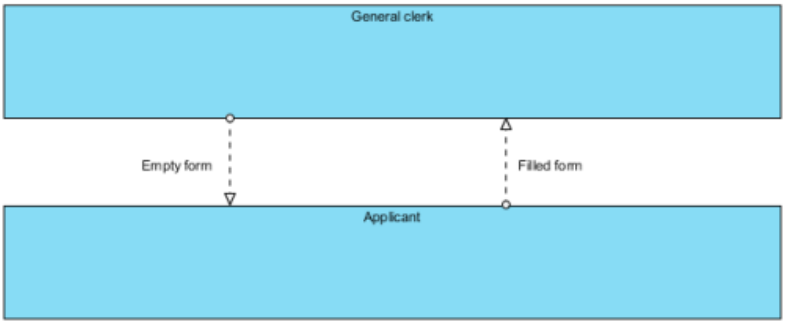
A pool can be shown as an empty box, called a black box. A black box represents a role solely, with all details hidden - You cannot create flow objects in it. To define a black box pool, right click on an empty pool and select **Black Box** from the popup menu.



Defining a black box

NOTE: You can only create a black box for an empty pool that has neither flow objects nor lanes in it.

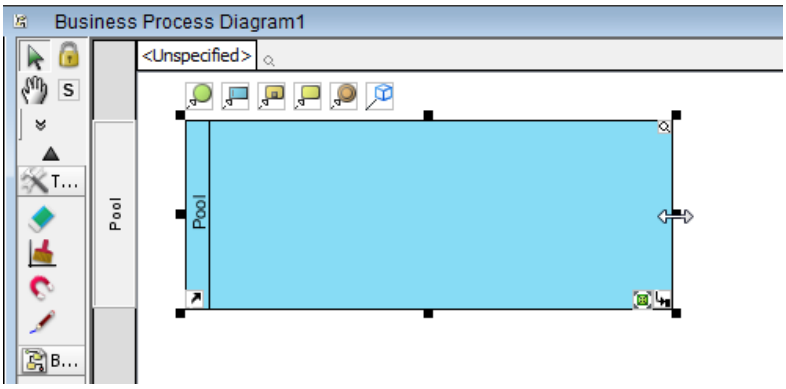
You can create message flows between black boxes to represent the collaboration between participants, or create message flows between flow objects in another pool/lane with and a black box.



Black boxes with message flows in between

Stretching of pool

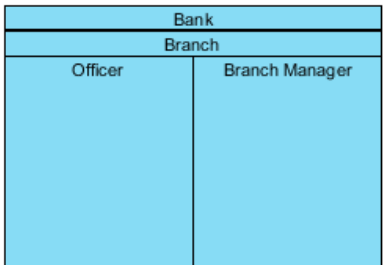
When you create a pool, it is automatically expanded to fit the width or length of diagram. We call this behavior *stretched*. When a pool is stretched, the pool and the contained lane(s) will expand or collapse following the size of diagram, and you cannot resize it manually. If you want to make a pool freely resize-able, you need to turn the stretch behavior off. To change the stretch option, right click on the pool involved and select **Presentation Options > Auto Stretch > Off** from the popup menu.



Pool can be resized freely when auto-stretch is turned off

Creating nested lanes

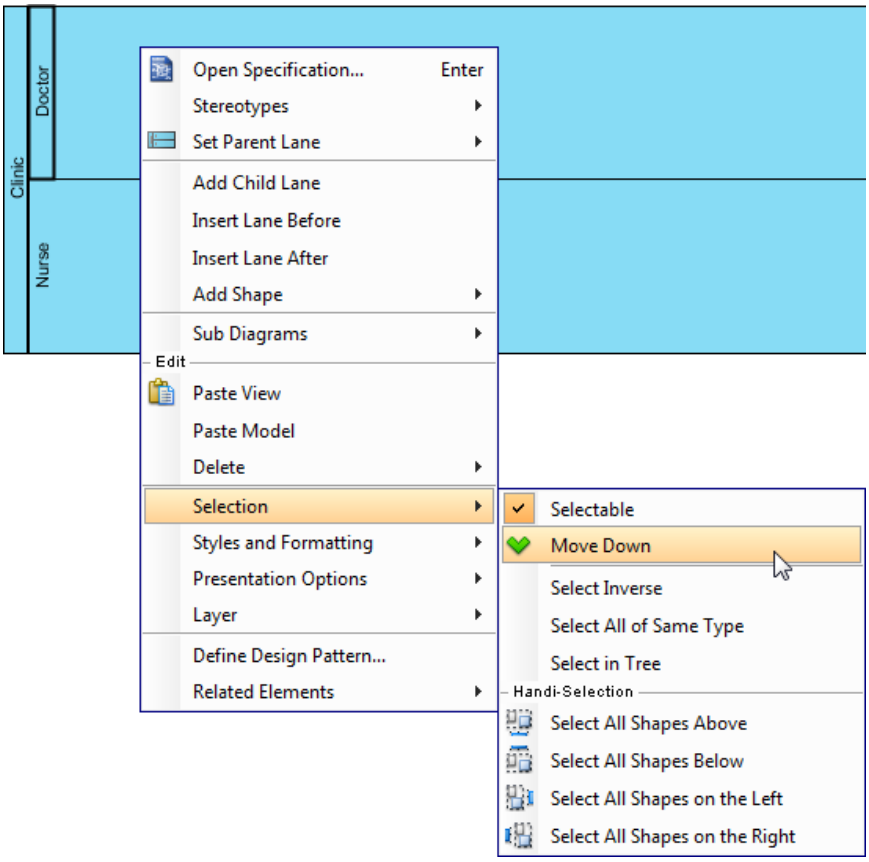
While a lane represents a sub-partition of a pool, a lane itself can contain lanes, to form a nested structure. To create a nested lane, right click on a lane and select **Add Child Lane** from the popup menu



Nested lanes

Reordering lane

You can change the position of lanes within a pool by moving them up and down. To reorder a lane, right click on the lane you want to reorder and select **Selection > Move Down** from the popup menu.



Moving a lane downward

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

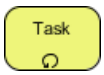
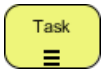
Task and sub-process

A business process is mainly formed by activities that need to be performed to complete the process. There are two kinds of activities - task and sub-process. A task is an atomic activity which represents work that cannot be broken down. On the contrary, sub-process represents work that can be broken down to a finer level of detail.

Task and sub-process

Task markers

You can assign markers to task. There are three markers: Loop, Multi-Instance, Compensation. A task can have one or two of these markers. Assignment of markers is done through the specification dialog box of task.

Name	Representation	Description
Loop		This marker indicates that the task will loop as long as the condition that defined in the loop is true. The condition is evaluated in each iteration, at either the beginning or the end of iteration. This marker can be used in combination with the compensation marker.
Multi-instance (parallel instances)		This marker indicates the execution of task in a desired number of instances, or in a data driven approach. The instances will be started at the same time.
Multi-instance (sequential instances)		This marker indicates the execution of task in a desired number of instances, or in a data driven approach. The instances will be executed one after the other.
Compensation		To undo (cancel) the result of another activity that were already successfully completed. The execution of compensation task is due to the undesired results and possibly side effects produced by antoher activity, that need to be reversed. A compensation task is performed by a compensation handler, which performs the steps necessary to reverse the effects of an activity.

Different markers of task

Adding a Loop marker

1. Right click on a task and select **Open Specification...** from the popup menu.
2. Select **Standard Loop** for **Loop type**. Click **OK** to confirm the changes.

> **NOTE:** You can click on the **...** button next to **Loop type** to set the loop condition, counter and the maximum number of iteration.

Adding a Multi-instance marker

1. Right click on a task and select **Open Specification...** from the popup menu.
2. Select **Multi-Instance Loop** for **Loop type**. Click **OK** to confirm the changes.

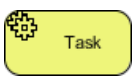
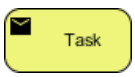
> **NOTE:** You can click on the **...** button next to **Loop type** to set the ordering of loop, either parallel or sequential.

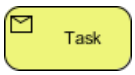
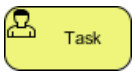
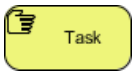
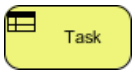
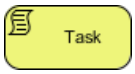
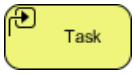
Adding a Compensation marker

1. Right click on a task and select **Open Specification...** from the popup menu.
2. Check **Compensation** at the bottom of specification and click **OK** to confirm the changes.

Task types

There are several types of task for you to separate the behavior of different tasks. You can set a type by right clicking on a task and selecting **Type**, then the type from the popup menu.

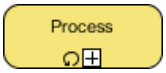
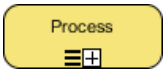
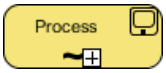
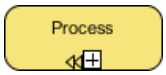
Name	Representation	Description
Service		A service task is a task that uses some sort of service, e.g. a Web service.
Send		A send task is a task that sends a message to an external participate. The task is said to be completed once the message has been sent.

Receive		A receive task is a task that waits for a message to arrive from an external participant. The task is said to be completed once the message has been received.
User		A user task is a task performed by a human with the assistance of a software application.
Manual		A manual task is a task that is performed without the aid of any business process execution engine.
Business Rule		A business rule task let the process to provide input to a business rules engine, and to get the output from engine.
Script		A script task involves a script defined by modeler or implementer in a language that a business process engine can understand, and is executed by a business process engine.
Reference		A reference task refers to another task for its content.

Types of tasks

Sub-process markers

You can assign markers to sub-process. There are four markers: Loop, Multi-Instance, Ad-hoc and Compensation. A sub-process can have up to three markers, excluding the marker for collapsed: A loop/multi-instance marker, an Ad-hoc marker, and a Compensation marker. Assignment of markers is done through the specification dialog box of sub-process.

Name	Representation	Description
Loop		This marker indicates that the sub-process will loop as long as the condition that defined in the loop is true. The condition is evaluated in each iteration, at either the beginning or the end of iteration. This marker can be used in combination with the ad-hoc and/or compensation marker.
Multi-instance (parallel instances)		This marker indicates the execution of sub-process in a desired number of instances, or in a data driven approach. The instances will be started at the same time.
Multi-instance (sequential instances)		This marker indicates the execution of sub-process in a desired number of instances, or in a data driven approach. The instances will be executed one after the other.
Ad-hoc		This marker indicates that a sub-process is a group of activities that have no required sequence relationships. The sequence and number of performances for activities are determined by the performers of the activities.
Compensation		To undo (cancel) the result of another activity that was already successfully completed. The execution of compensation sub-process is due to the undesired results and possibly side effects produced by another activity that need to be reversed. A compensation sub-process is performed by a compensation handler, which performs the steps necessary to reverse the effects of an activity.

Different markers of sub-process

Adding a Loop marker

1. Right click on a sub-process and select **Open Specification...** from the popup menu.
2. Select **Standard Loop** for **Loop type**. Click **OK** to confirm the changes.

> **NOTE:** You can click on the **...** button next to **Loop type** to set the loop condition, counter and the maximum number of iteration.

Adding a Multi-instance marker

1. Right click on a sub-process and select **Open Specification...** from the popup menu.
2. Select **Multi-Instance Loop** for **Loop type**. Click **OK** to confirm the changes.

> **NOTE:** You can click on the **...** button next to **Loop type** to set the ordering of loop, either parallel or sequential.

Adding an Ad-hoc marker

1. Right click on a sub-process and select **Open Specification...** from the popup menu.
2. Make sure the type of sub-process is set to be **Embedded Sub-Process**. Check **Ad-hoc** in the **Details** section and click **OK** to confirm the changes.

Adding a Compensation marker

1. Right click on a sub-process and select **Open Specification...** from the popup menu.
2. Check **Compensation** at the bottom of specification and click **OK** to confirm the changes.

Sub-process types

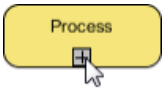
There are several types of sub-process for you to separate the behavior of different sub-processes. You can set a type by right clicking on a sub-process and selecting **Type**, then the type from the popup menu.

Name	Representation	Description
Embedded		An embedded sub-process is a sub-process that models its internal details in another process.
Reusable		A reusable sub-process calls a pre-defined process.
Reference		A reference sub-process refers to another sub-process.

Types of tasks

Breaking down a sub-process

A sub-process can be opened up to model the detail in a lower level. To open up a sub-process, click on the plus marker in sub-process and select **New Business Process Diagram**. This will create a new business process diagram that belongs to the sub-process.

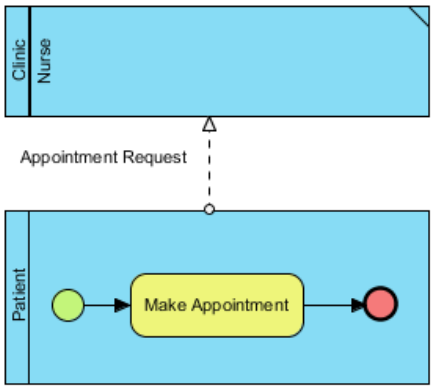


To break down a sub-process

NOTE: Once a sub-process diagram is created, its detail will be shown as the sub-process shape as a thumbnail of the diagram. To hide the thumbnail, click on the minus marker at the bottom of sub-process to turn it off.

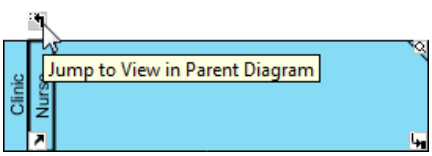
Re-using elements from parent diagram

In the sub-process diagram, you can re-use pools, lanes and flow objects that appear in the parent diagram. To do this, right click on the sub-process diagram and select Add Pools/Lanes/Sub-Processes/Gateways from Parent Diagram... from the popup menu, and choose the element to reuse. Elements being re-used will have dog ear appear at their corners.



A sub-process diagram with a lane reused from parent diagram

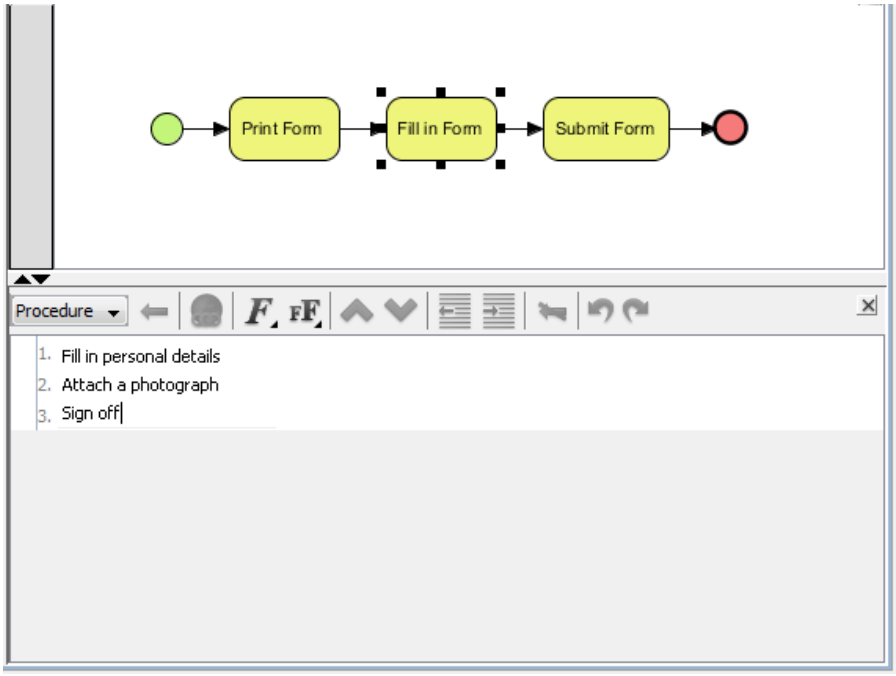
You can jump from a re-used element back to the parent diagram through the resource-centric interface.



Jump to parent diagram

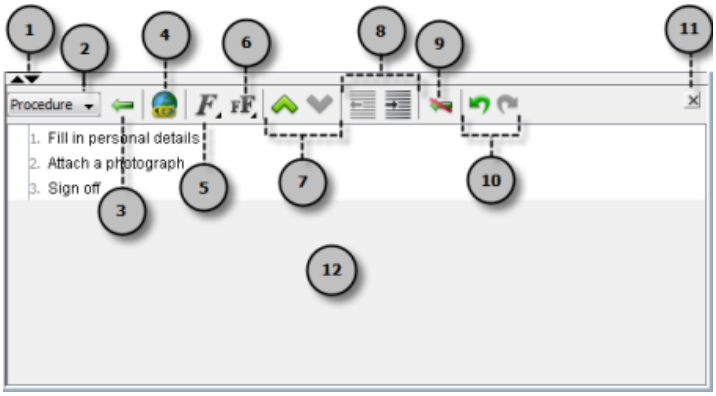
Defining procedure of activity

An activity within a process represents work need to be done. Each activity can be formed by a number of steps. For example, a task Process Application involves 2 steps - validate application, confirm application. To document the steps of an activity, you can make use of the procedure editor.



Procedure of a task

An overview of procedure editor



An overview of procedure editor

No.	Name	Description
1	Collapse/Expand	Click on the triangle on the left hand side to maximize the editor. On the contrary, click on the inverted triangle to minimize the editor.
2	Procedure selector	You can define multiple set of procedure per activity. Click on this drop down menu to select the one you want to read/edit.
3	Step	Click on this button to create a step under the step selected in editor.
4	Hyperlink...	Add a link in the selected step for reference.
5	Font formats selector	There are three buttons. The first one increases the font size for one level, while the second one decreases the font size for one level. The third button resets the font size setting to default.
6	Font size setting selector	Click on this drop-down menu to select the sizes of highlighted text. Press **Grow Font** button to increase the font size for one level, while press **Shrink Font** button to decrease the font size for one level. Press **Default Font** button reset the font size setting to default. Moreover, you can adjust the font size for the highlighted text manually by slider.
7	Reorder step	Click on **Move Up** button to move the selected step upward while **Move Down** button to move the selected step downward.
8	Decrease Indent/ Increase Indent	Click on **Decrease Indent** button to reduce the indentation of the selected step while click on **Increase Indent** button to indent the selected step.

9	Undo/ Redo	Click on **Undo** button to revert change while click on **Redo** button to redo reverted change.
10	Close editor	Click on this button to close the editor.
11	Steps editor	The place where you can read and edit steps.

Description of procedure editor

Showing/Hiding procedure editor

The procedure editor is opened in business process diagram by default. To hide it, right click on the background of business process diagram and deselect **Show Procedure Editor** from the popup menu. You can select the same menu to show it when hidden.

NOTE: Alternatively, you can close the editor by clicking on the cross button at the top right corner of editor panel.

Documenting the procedure

1. Select the task or sub-process that you want to document its procedure.
2. Click on the first row labelled 1 and enter the first step.
3. Press **Enter** to go to the next step. You can create a sub-step by pressing **Tab** on a step. Pressing **Shift-Tab** decreases the indentation of a sub-step. Repeat step 2 and 3 to enter the remaining steps the activities involve.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Event

An event in a business process refers to something that happens and affects the flow of process. There are three types of events: Start, intermediate and end.

Start, intermediate and end events with different kinds of triggers and results

Start event

A start event indicates the place where and possibly why a process start. Since start event is used for initiating a process, it does not have any incoming sequence flow.

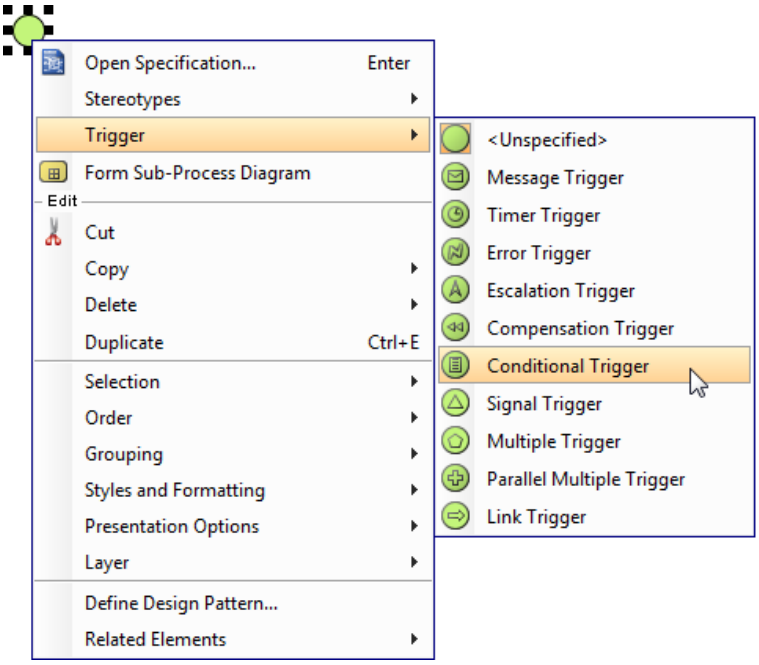
You can define a trigger for start event, to show the condition(s) that will cause a process to initiate.

Trigger name	Representation	Description
None		The none start event does not have a defined trigger.
Message		This trigger starts the process by receiving a message from a participant.
Timer		This trigger starts the process in a specific time-date or a specific cycle (e.g. every Friday).
Error		This trigger starts an in-line event sub-process when an error occurs. Note that this trigger can only be used with an event-sub-process.
Escalation		This trigger starts or not to start an in-line event sub-process when the constraint specified is not satisfied. Note that this trigger can only be used with an event-sub-process.
Compensation		This trigger starts an in-line event sub-process when an compensation occurs, which require undoing some steps. Note that this trigger can only be used with an event-sub-process.
Conditional		This trigger starts the process when a specific condition become true.
Signal		This trigger starts the process when a signal broadcasted from another process has arrived. Note that signal is different from message, which has a specific target for message.
Multiple		This means that there are multiple triggers of the process. Any one of them can cause the process to start.
Parallel Multiple		This means that there are multiple triggers of the process. All of the triggers must be triggered in order to start the process.
Link		This trigger provide a mean to connect the end result of one process to the start of another.

Different types of start event trigger

Defining a trigger

To define a trigger on an event, right click on the event and select **Trigger**, then the type of trigger from the popup menu.



To define a start event trigger

If you want to edit the properties of the trigger, such as the condition of a conditional trigger, right click on the event and select **Open Specification...** from the popup menu. Then, click on the **...** button next to the drop down menu of **Trigger** to edit its properties in the popup dialog box.

Interrupting or Non-interrupting event sub-process

Start event can be attached to the border of an event sub-process, to initiate the sub-process inline. You can define this kind of trigger as either interrupting or non-interrupting, which means to interrupt its containing process or not to interrupt its containing process respectively. To set a trigger to be Interrupting or Non-Interrupting, right click on the event and select/de-select **Triggers > Interrupting** from the popup menu.

Interrupting (left) and Non-Interrupting (right) events

NOTE: Only triggers that can be attached to event sub-process can set as interrupting/non-interrupting. The supported trigger types include: Message, Timer, Escalation, Error, Cancel, Compensation, Conditional, Signal, Multiple, and Parallel Multiple.

Intermediate event

An intermediate event indicates where something happens in between the start and end event of a process. You can use an intermediate event to show where messages are received or sent, show the necessary delay, perform exception handling and show the need of compensation. You can place an intermediate even in two places: Attaching the boundary of task/sub-process, Normal flow (i.e. connected from a flow without attaching to an activity).

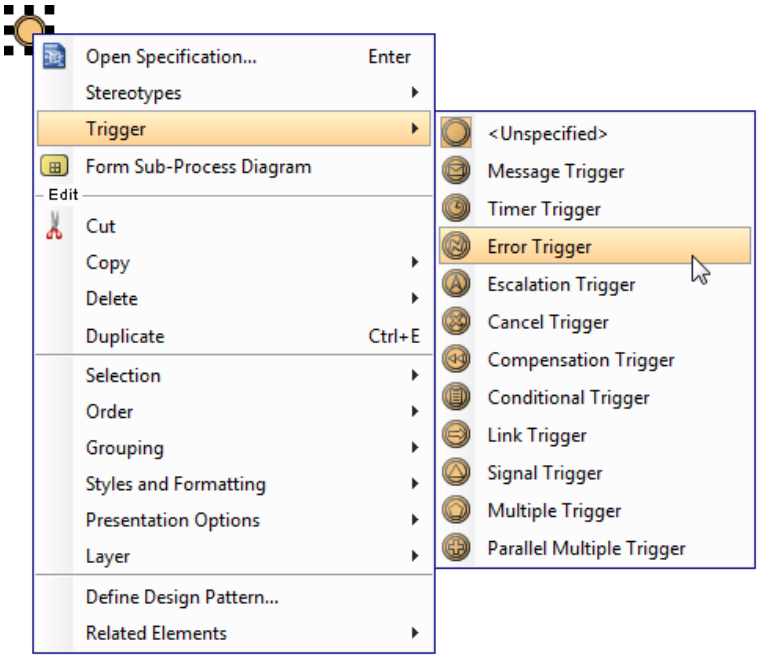
Trigger name	Representation	Description
None		The none intermediate event does not have a defined trigger. It is used to indicate change of state in the process. You can only use a none intermediate event in a normal flow.
Message		This trigger represents either a send or receive of message
Timer		This trigger acts as a delay mechanism on a specific date-time or cycle (e.g. every Friday). You can only use a timer intermediate event in a normal flow.
Error		This trigger reacts to a named error, or to any error if no name is specified.
Escalation		The trigger indicates where an escalation is raised. You can only use an escalation intermediate event in a normal flow.
Cancel		This trigger will be fired when a cancel end event is reached within the transaction sub-process. It also shall be triggered if a Transaction Protocol "Cancel" message has been received while the Transaction is being performed.
Compensation		The trigger indicates the need of compensation.
Conditional		The event will be triggered when the condition specified become true.

Link		This trigger is used for linking two sections of a process. You can use it to mode a looping of flow or to avoid having long sequence flow connectors appear on diagram. You can only use a link intermediate event in a normal flow.
Signal		This trigger indicates the sending or receiving of signals, which is for general communication within and across process levels, across pools, and between business process diagrams.
Multiple		This means that there are multiple triggers defined. Any one of them can cause the event to be triggered.
Parallel Multiple		This means that there are multiple triggers defined. All of the triggers must be triggered in order to trigger the multiple event.

Different types of start event trigger

Defining a trigger

To define a trigger on an event, right click on the event and select **Trigger**, then the type of trigger from the popup menu.



To define a intermediate event trigger

If you want to edit the properties of the trigger, such as the condition of a conditional trigger, right click on the event and select **Open Specification...** from the popup menu. Then, click on the **...** button next to the drop down menu of **Trigger** to edit its properties in the popup dialog box.

Throw and catch

You can set an event to be catch or throw. Catch means to react to a trigger, while throw means to create a trigger. To set, right click on an event and select **Trigger**, then either **Catching** or **Throwing** from the popup menu.

A catch event (left) and a throw event (right)

NOTE: The trigger types that can set as throw/catch include: Message, Escalation, Compensation, Link, Signal, and Multiple.

Interrupting or Non-interrupting event

Intermediate event can be attached to the border of an activity. You can set an event to interrupt or not to interrupt the activity to which it is attached. To set a trigger to be Interrupting or Non-Interrupting, right click on the event and select/de-select **Triggers > Interrupting** from the popup menu.

Interrupting (left) and Non-Interrupting (right) events

NOTE: Only triggers that can be attached to event sub-process can set as interrupting/non-interrupting. The supported trigger types include: Message, Timer, Escalation, Conditional, Signal, Multiple, and Parallel Multiple.

End event

As an opposite of start event, end event indicates where a process will end. Since end event is used for terminating a process, it does not have any outgoing sequence flow.

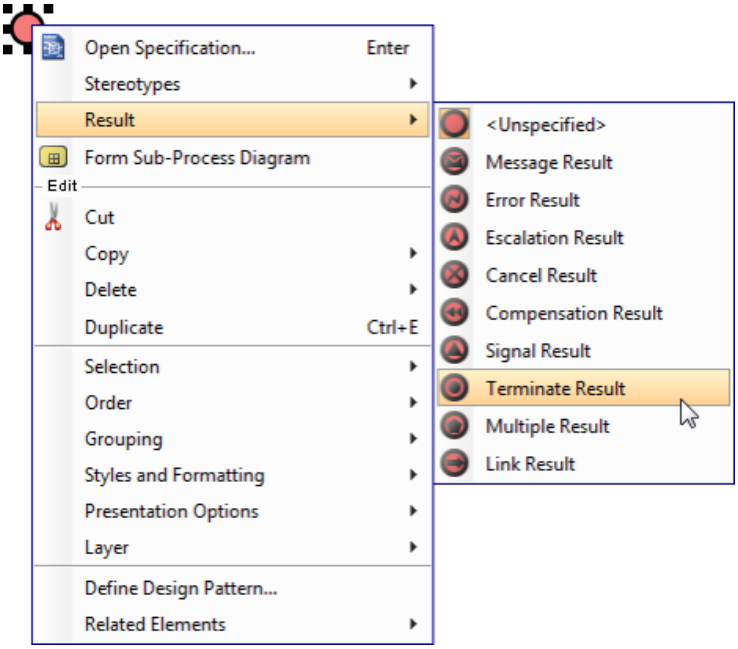
You can define a result for end event, to show what will happen when reaching the end.

Trigger name	Representation	Description
None		The none end event does not have a defined result.
Message		This result ends the process by sending a message to a participant.
Error		This result indicates the generation of a named error when the process ends.
Escalation		This result indicates the trigger of escalation when the process ends.
Cancel		This result indicates that the transaction should be cancelled.
Compensation		This result indicates the need of compensation, which require undoing some steps.
Signal		This result indicates that a signal will be broadcasted when the process ends. Note that signal is different from message, which has a specific target for message.
Terminal		This result indicates that all activities in the process should be immediately ended.
Multiple		This result indicates that there are multiple consequences of ending the process.
Link		This result provide a mean to connect the end result of one process to the start of another.

Different types of end event result

Defining a result

To define a result on an event, right click on the event and select **Result**, then the type of result from the popup menu.



To define an end event result

If you want to edit the properties of the result, such as the message produced by a message result, right click on the event and select **Open Specification...** from the popup menu. Then, click on the **...** button next to the drop down menu of **Result** to edit its properties in the popup dialog box.

Related Resources

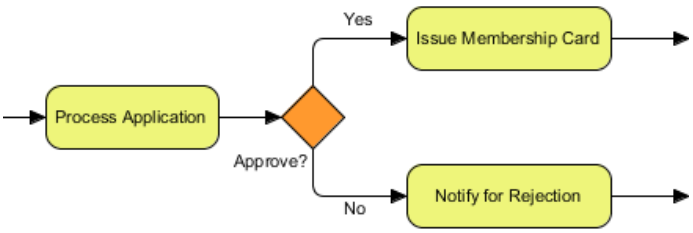
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Gateway

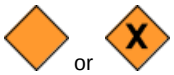
Gateway is a kind of flow objects which is used to direct sequence flows within a process, base on certain condition(s). It acts like a gate that either allow or disallow passage, and possibly to control the selection of outgoing flow that pass through the gateway.



A typical use of gateway - for modeling a situation of decision making

Gateway types

There are several kinds of gateway for different kinds of control behavior, such as making decision, branching, merging, forking and joining.

Gateway type	Representation	Description
Exclusive		An exclusive gateway can be used to model alternative paths within a flow. It is where the diversion take place.
Event-based		An event-based gateway can be used to model the alternative paths that follow the gateway are based on events that occur instead of the expression of flow.
Inclusive		An inclusive gateway can be used to model alternative and parallel paths within a process. Unlike exclusive gateway, all condition expressions are evaluated. All the outgoing paths that give a positive result of evaluation will be taken.
Complex		A complex gateway can be used to model complex synchronization behavior.
Parallel		A parallel gateway can be used to create and join parallel flows. It creates the paths without checking any conditions.

Different types of gateway

Showing internal indicator "X" for exclusive gateway

You can optionally show the letter "X" inside the shape of exclusive gateway to distinguish it from other gateways (see the table above). To do this, right click on the background of business process diagram and select **Presentation Options > Gateway Display Options > Data-Based gateway Markers Visible** from the popup menu.

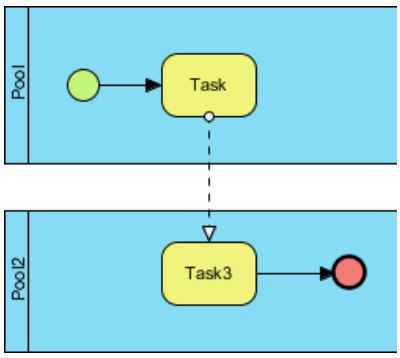
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Sequence and message flow

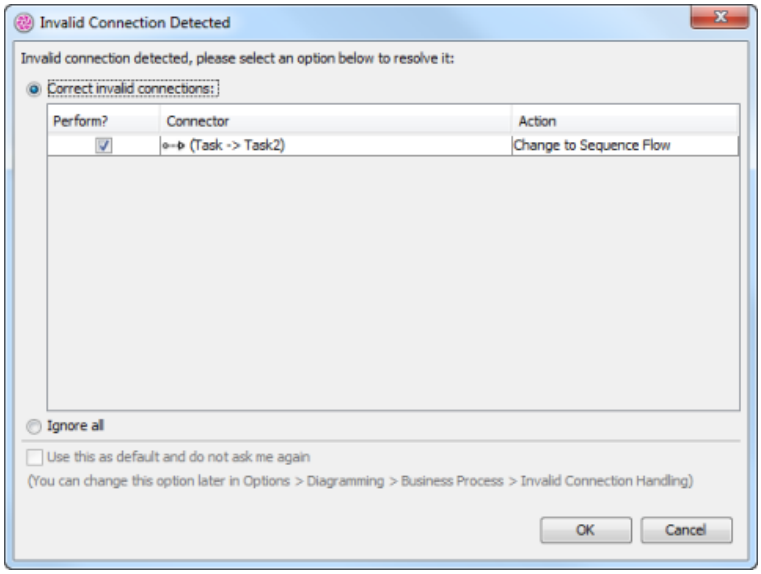
There are two types of connectors for modeling flows in a process - Sequence flow and Message flow . A sequence flow is used to connect flow objects in a process or a choreography, to show the flow. Message flow is used to show the flow of messages between separate pools/lanes. You cannot use message flow to connect flow objects within the same participant.



Sequence and message flows can be used to connect flow objects

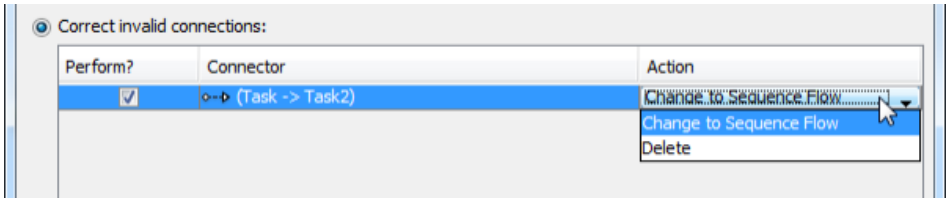
Correcting invalid flow

As mentioned before, you can use sequence flow to connect flow objects within a participant, and use message flow to connect flow objects in separate participants. If you attempt to use a type of flow incorrectly, like to connect flow objects within participant with message flow, you will be prompt to correct your flow.



Invalid connector is detected

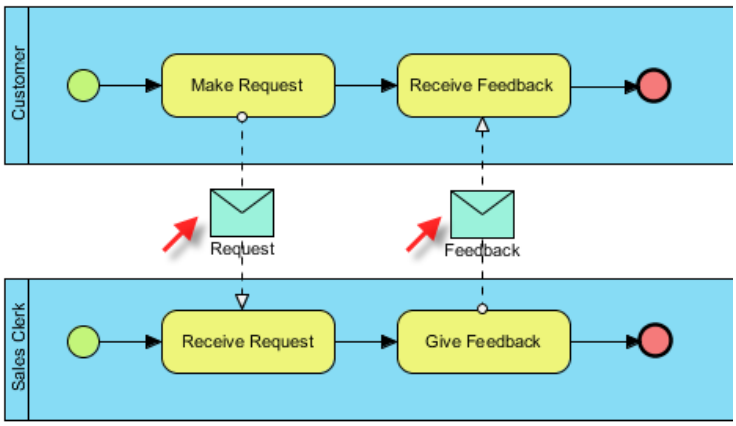
There are several actions you can perform. First, you can correct your invalid flow by changing its type, like to change from message to sequence flow. If the connector should not be there, you may select to delete it. If you really want to keep the invalid connector, choose **Ignore all** at the bottom of dialog box. Click **OK** to confirm.



Possible actions of handling invalid connector - Correct it or delete it

Modeling and visualizing message pass by message flow

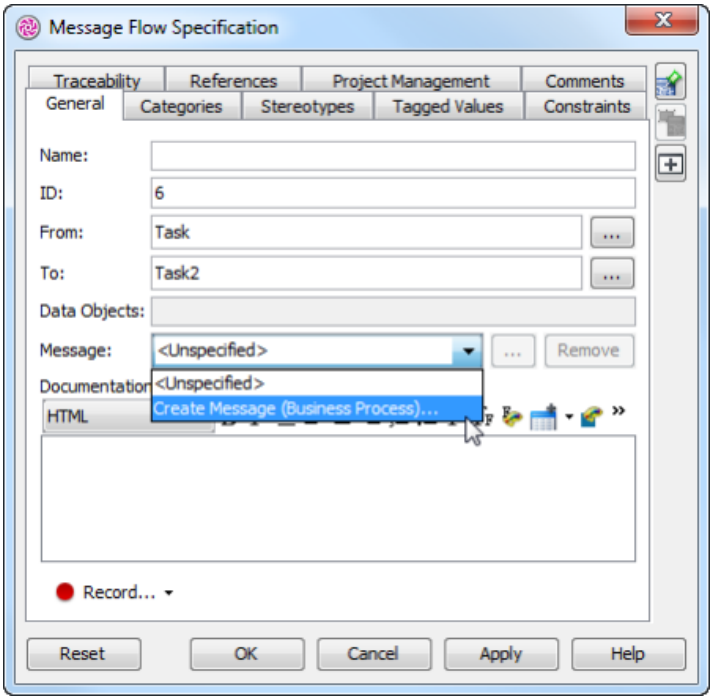
You can define a message that pass by message flow, and visualize it.



A sample BPD with messages

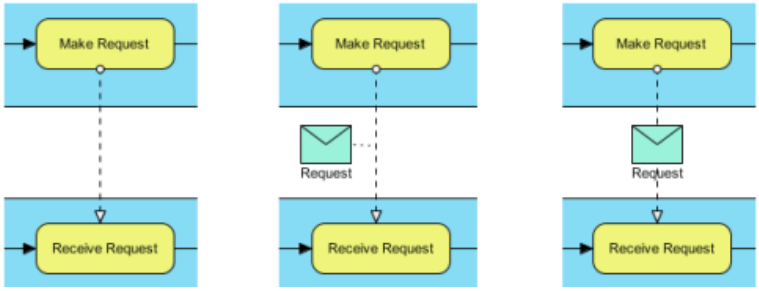
To do this:

1. Right click on the message flow that you want to model its message, and select **Open Specification...** from the pop-up menu.
2. Click on the drop down menu of Message and select **Create Message (Business Process)** from the pop-up menu.



Create a message

3. Name the message in the Message specification and click **OK** to confirm.
4. You should see the message added appear in the drop-down menu of **Mesasge**. Click **OK** to confirm the change and go back to diagram.
5. Although the message is defined, you still need to visualize it. To visualize the message, right click on the diagram's background, select **Presentation Options > Message Flow Display Option > Show Message of Message Flow** from the pop-up menu, then select the way of visualizing the message.



Ways of showing message (from left to right) - Do Not Show, Associated with Message Flow, Overlapping Message Flow

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian

- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Choreography task and sub-process

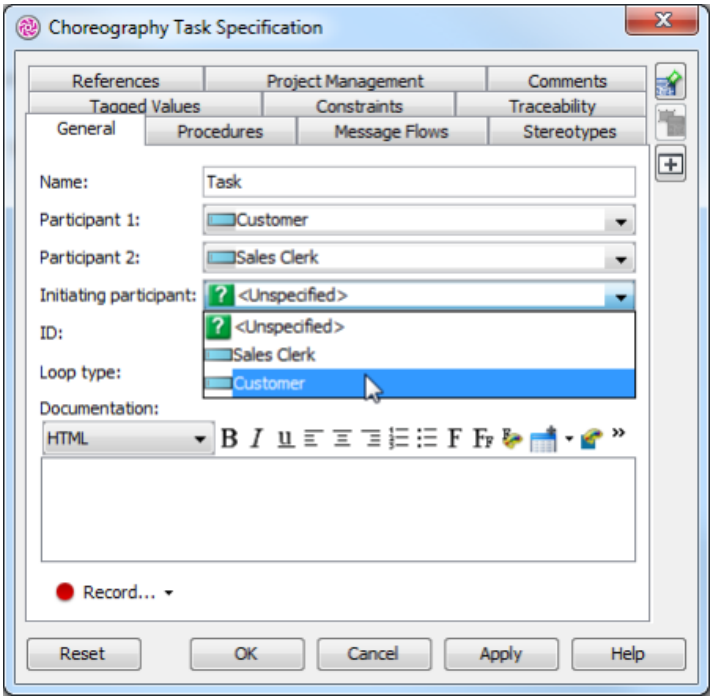
A choreography is a type of process which defines the sequence of interaction between participant. Unlike a standard BPMN process, which defines the flow of activities in a process. Choreography does not belong to any pool. It exist outside or in between pools, and shows the messages that pass between pools.

Choreography task

A choreography task is an atomic activity which represents an interaction among participants (pools), and consists of one or more messages that exchange between the pools. A choreography shape is formed by multiple parts. We call them bands. The name of choreography task and each of the participants are all displayed in different bands.

Setting participants and initiating participant

1. Right click on the choreography task and select **Open Specification...** from the popup menu.
2. In the specification dialog box, choose the pools for participant 1 and 2.
3. Select the pool which starts the interaction from the drop down menu of **Initiating participant**.



Selecting initiating pool

4. Click **OK** to confirm editing and go back to diagram.

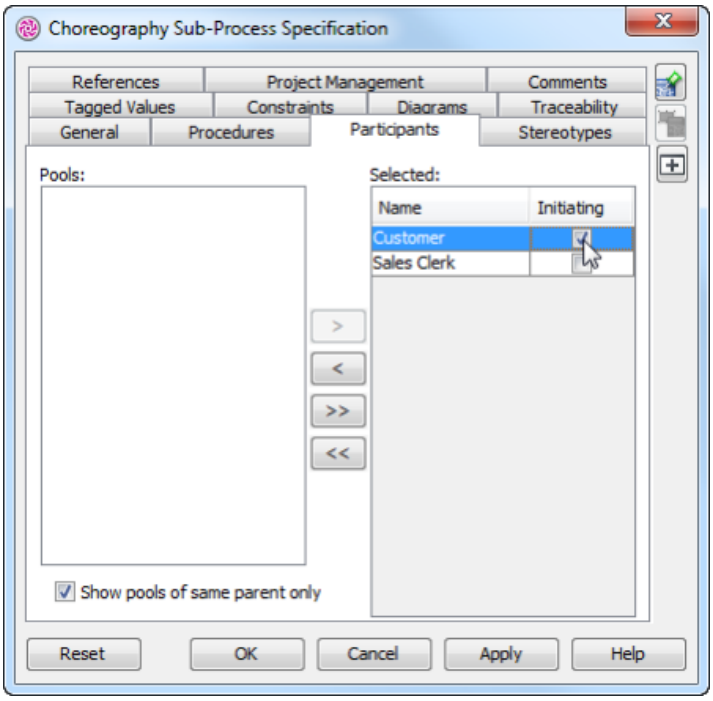
Choreography task

Choreography sub-process

A choreography sub-process is a compound activity in that it has detail that is defined as a flow of other activities.

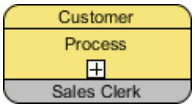
Setting participants and initiating participant

1. Right click on the choreography sub-process and select **Open Specification...** from the popup menu.
2. In the specification dialog box, open the **Participants** tab.
3. Select the pools the choreography sub-process involve and click **>** to assign them.
4. Check the initiating pool.



Select initiating participant

5. Click **OK** to confirm editing and go back to diagram.



Choreography sub-process

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Data object, Data Input, Data Output and Data Store

Data object
You can use data objects to model data within process flow. Typical examples of data object include purchase order, receipt, e-mail, delivery notice, etc.

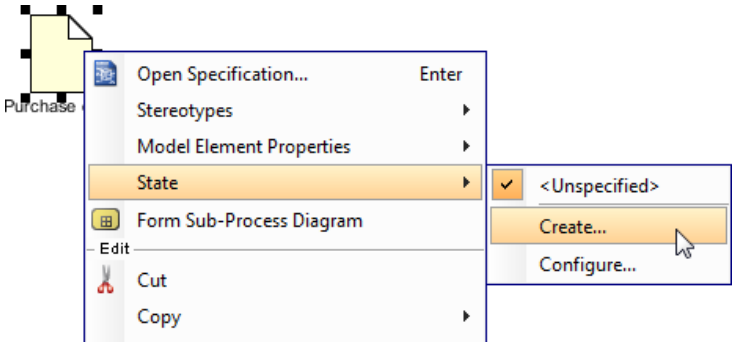
Data input
Data input is a special kind of data used as input of a process. You draw data input in a business process diagram to show the input of data to the top-level process or to show the input of a called process.

Data output
Data output is a special kind of data produced as output of a process. You draw data output in a top-level business process diagram to show the output of the process.

Data store
Data store enables activities to retrieve or updated stored information that will persist.

Defining state
You can optionally record the state of data object. For example, the data object *Order* has states created, submitted and processed. To define state:

1. Right click on data object and select **State > Create...** from the popup menu.



To create a state

2. In the **Create State** dialog box, enter the name of state and click **OK** to confirm.

NOTE: State is a view based option. You may copy a data object, paste as a new view, and set the state to the new view. This enables you to show the change of state of a data throughout a process flow.

Related Resources
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Conversation diagram

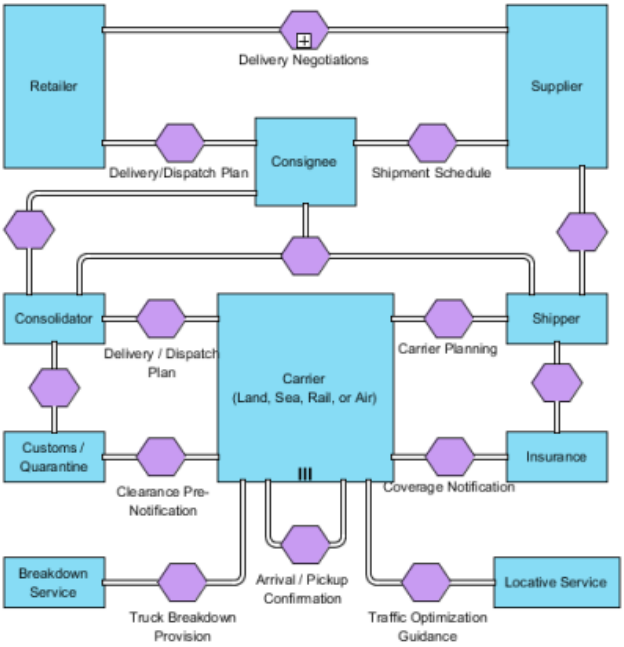
Conversation diagram gives you a high level understanding to the relationships between pools under the domain being modeled. This chapter teaches you how to create a conversation diagram.

Creating conversation diagram

This page teaches you how to create a conversation diagram through the diagram navigator.

Creating conversation diagram

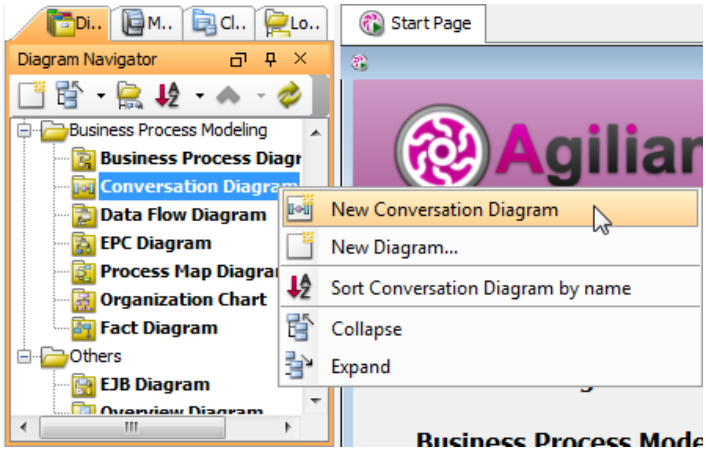
Conversation diagram gives you a high level understanding to the relationships between pools under the domain being modeled.



A sample conversation diagram

Creating conversation diagram

To create a conversation diagram, right click on **Conversation Diagram** in **Diagram Navigator** and select **New Conversation Diagram** from the popup menu.



*Create a conversation diagram through **Diagram Navigator***

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Data flow diagram

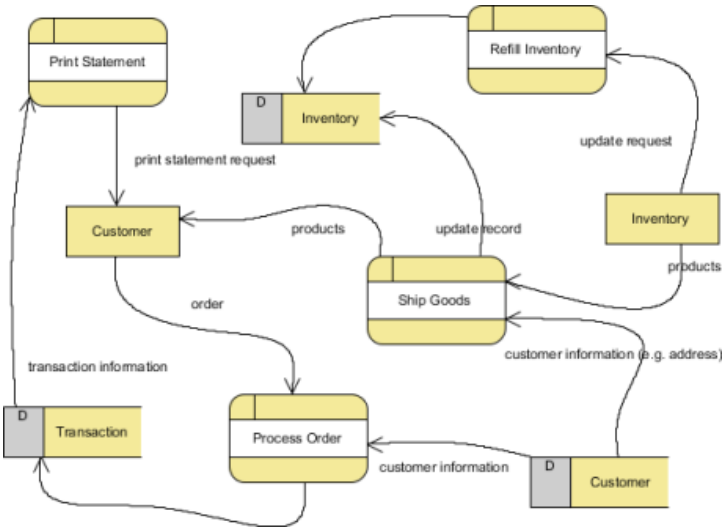
Data flow diagram is a well known approach to visualize the data processing in business analysis field. This chapters teaches you how to create a data flow diagram.

Creating data flow diagram

There is a list of supported notations in data flow diagram. You will also see how to decompose a process.

Creating data flow diagram

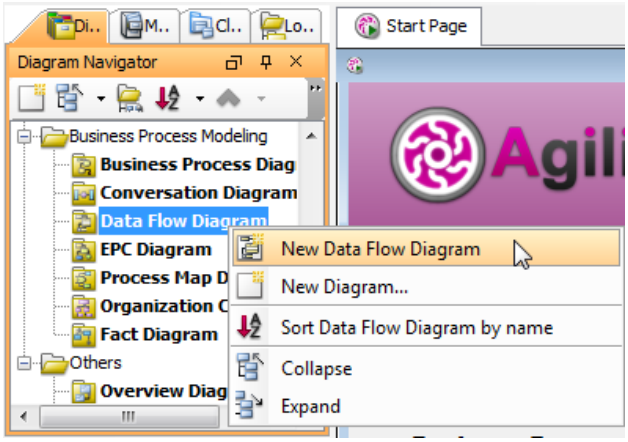
Data flow diagram is a well known approach to visualize the data processing in business analysis field. A data flow diagram is strong in illustrating the relationship of processes, data stores and external entities in business information system.



A sample data flow diagram

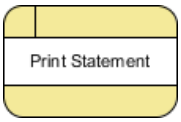
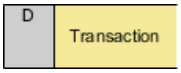
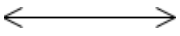
Creating data flow diagram

To create a data flow diagram, right click on **Data Flow Diagram** in **Diagram Navigator** and select **New Data Flow Diagram** from the popup menu.



*Create a data flow diagram through **Diagram Navigator***

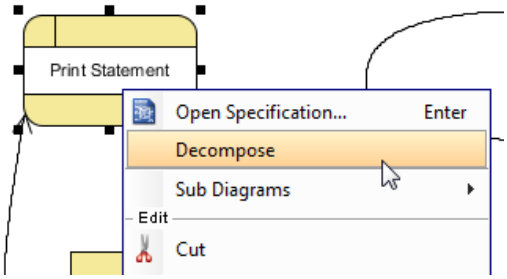
Notations

Name	Representation	Description
Process		A process takes data as input, execute some steps and produce data as output.
External Entity		Objects outside the system being modeled, and interact with processes in system.
Data Store		Files or storage of data that store data input and output from process.
Data Flow		The flow of data from process to process.
Bidirectional Data Flow		The flow of data that flow both from and to process.

A list of supported notations in data flow diagram

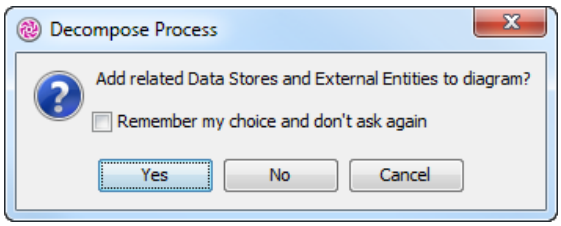
Decomposing a process

You can create multiple data flow diagrams for different levels of detail. A new level can be decomposed from a process in diagram. To decompose a process, right click on the process and select Decompose from the popup menu.



Decompose process

The **Decompose Process** dialog box will prompt to ask you whether to add related data stores and external entities to the new data flow diagram. If you choose **Yes**, those connected data stores and external entities will be copied to the new diagram.



Choose whether to add related model elements to diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Event-driven process chain diagram

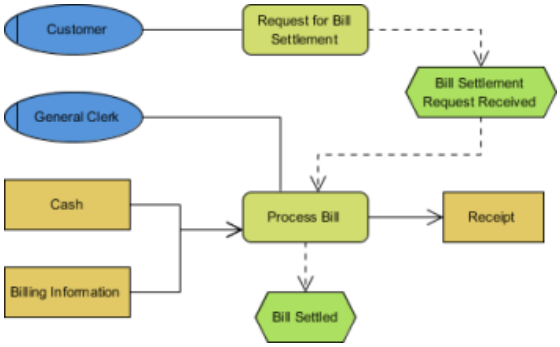
EPC diagram, short for event-driven process chain diagram, is a flowchart based diagram that can be used for resource planning and identifying possible improvements of a business process. This chapters teaches you how to create a EPC diagram.

Creating event-driven process chain (EPC) diagram

This page teaches you how to create a EPC diagram through the diagram navigator. There is a list of supported notations in EPC diagram.

Creating event-driven process chain (EPC) diagram

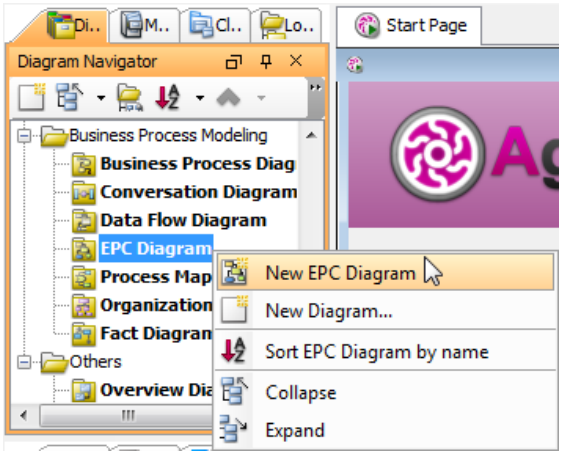
EPC diagram , short for event-driven process chain diagram, is a flowchart based diagram that can be used for resource planning and identifying possible improvements of a business process.



A sample EPC diagram

Creating EPC diagram

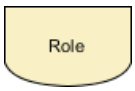
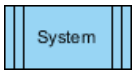
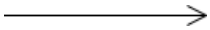
To create an EPC diagram, right click on **EPC Diagram** in **Diagram Navigator** and select **New EPC Diagram** from the pop-up menu.



*Create a EPC diagram through **Diagram Navigator***

Notations

Name	Representation	Description
Event	Bill Settled	An event describes what circumstances a function or a process works or which state a function or process results in.
Function	Process Bill	A function describes the transformations from an initial state to a resulting state.
Operator	or or	And - An and operation corresponds to activating all paths in the control flow concurrently. Or - An or operator corresponds to activating one or more paths among control flows. XOR - An XOR operator corresponds to making decision of which path to choose among several control flows.
Organization unit	General Clerk	An organization unit determines which person or organization within the structure of an enterprise is responsible for a specific function.
Control flow		A control flow connects events with function, process paths, or operators creating chronological sequence and logical interdependencies between them.
Process path	Record Overdue Payment	A process path shows the connection from or to other processes.
Organization unit assignment		An organization unit assignment shows the connection between an organization unit and the function it is responsible for.

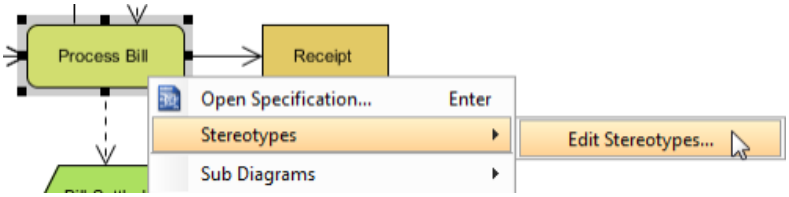
Information resource		An information resource portrays objects in the real world that can be input data serving as the basic of a function, or output data produced by a function.
Role		A role represents a unit, organization or a party that performs a function in a process.
System		A system is the provider of functions in process.
Information flow		Information flows show the connection between functions and input or output data, upon which the function reads changes or writes.

A list of supported notations in EPC diagram

Applying stereotype to EPC elements

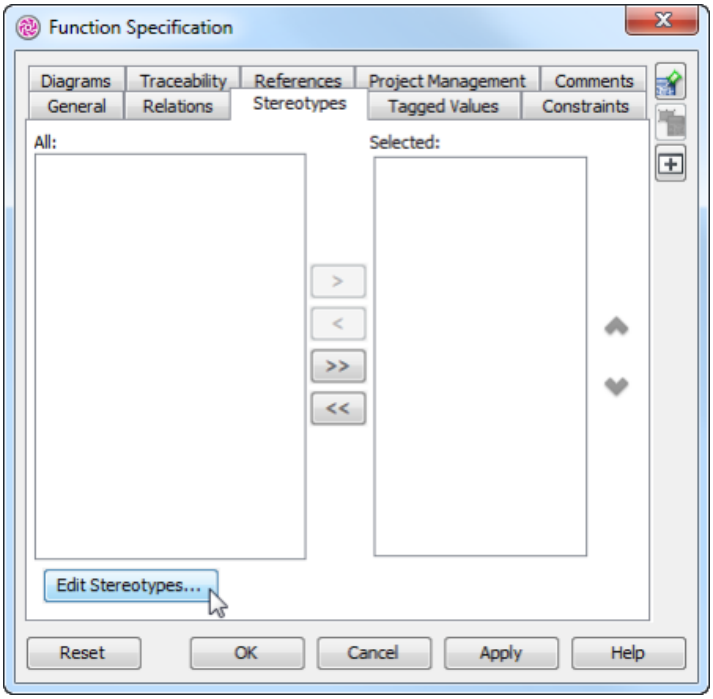
A stereotype defines how a model element may be extended, and enables the use of platform or domain specific terminology or notation in place of, or in addition to, the ones used for the extended metaclass. You can apply one or more stereotypes to model elements, and decide whether or not to visualize the stereotype or tagged values in views. To apply stereotype to model element:

1. Right click on the model element, or the view of the model element that you want to apply stereotype to. Select **Stereotypes > Stereotypes...** from the pop-up menu. As a side note for you, once you have ever applied a stereotype on the selected kind of element, you can re-select the same stereotype in this popup menu.



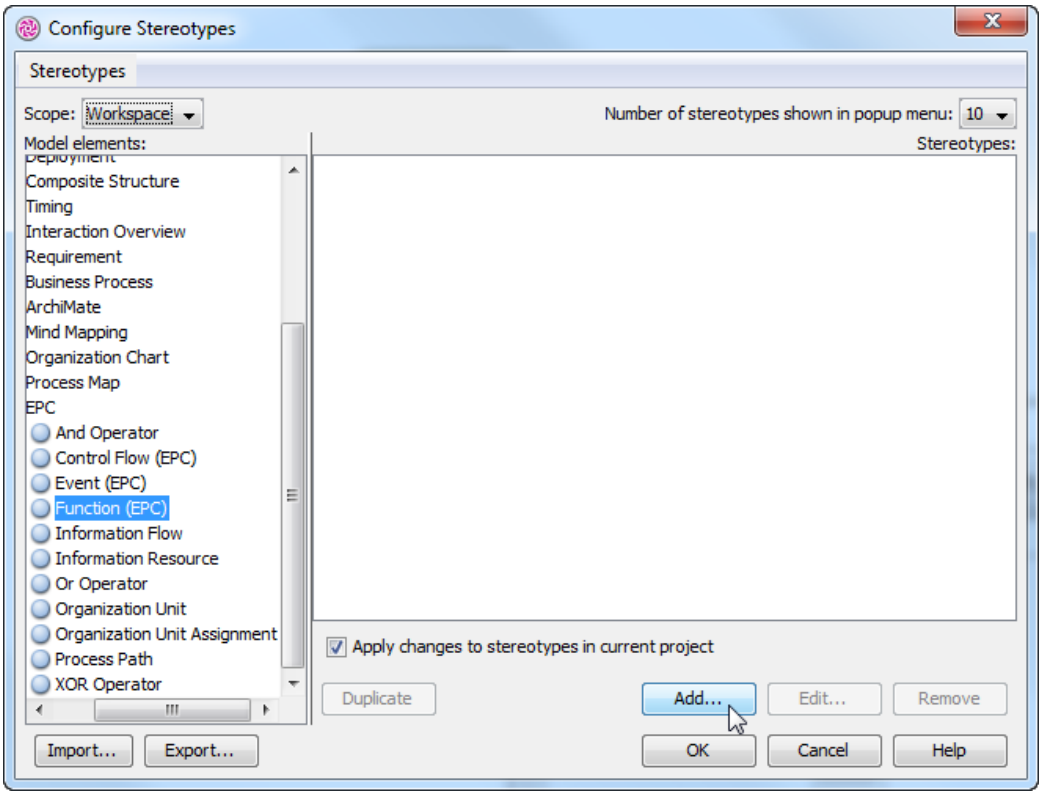
Open Stereotypes page

2. In the model element specification, open the **Stereotypes** tab and then click on **Edit Stereotypes...**.



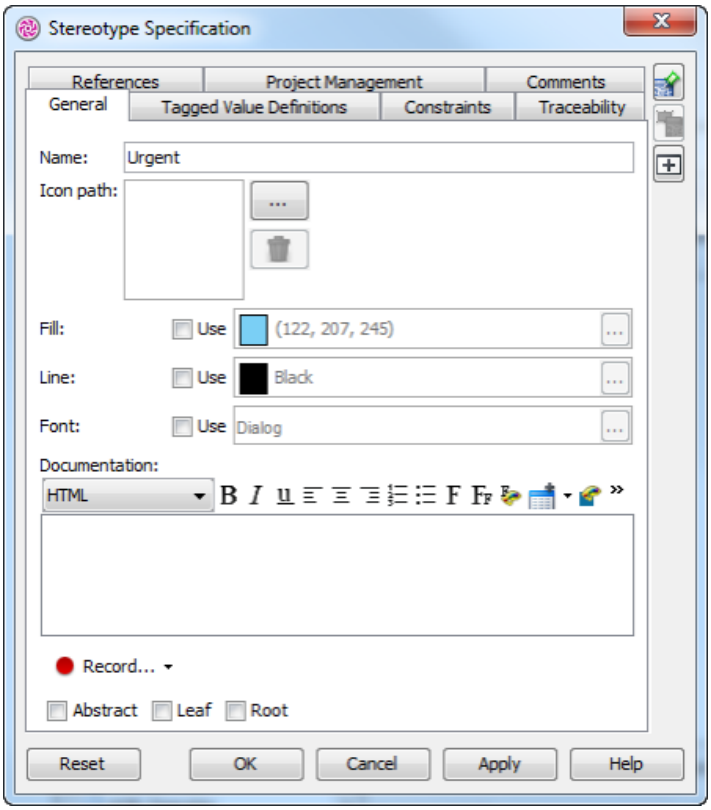
Editing stereotypes

3. In the **Configure Stereotypes** dialog box, click **Add**....



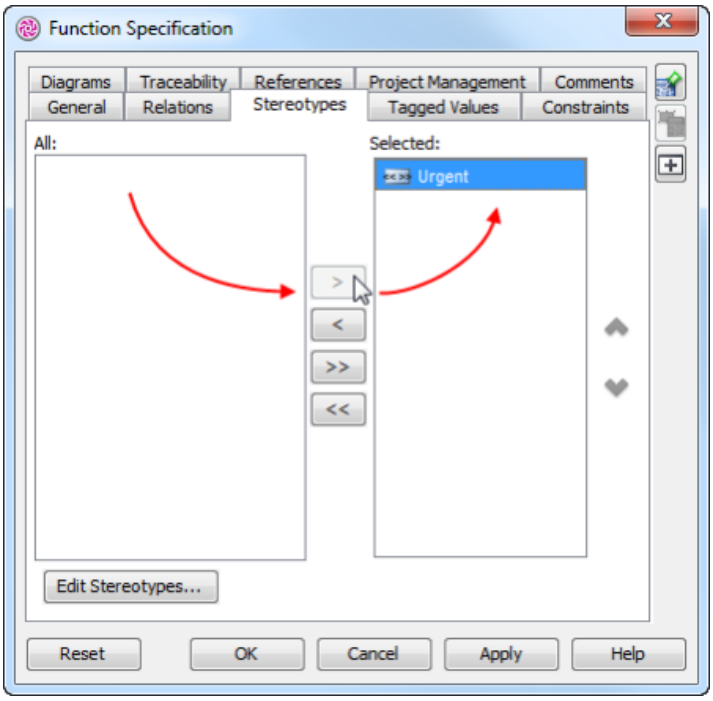
Adding stereotype

4. In the **Stereotype** specification dialog box, enter the name of stereotype and click **OK**.



Naming stereotype

5. This goes back to the specification dialog box. Select the stereotype you want to apply, then click **>** to assign it to the **Selected** list.



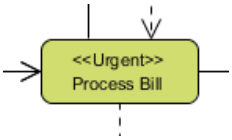
Selecting stereotype

NOTE: You can also double click on a stereotype to apply it.

NOTE: While clicking on **>** applies the selected stereotype to model element, **<** removes a stereotype selected in **Selected** list.

To apply all available stereotypes to model element, click **>>**, and likewise, clicking on **<<** removes all the applied stereotypes.

6. Click **OK** to confirm. The stereotype will then be shown within a pair of guillemets above the name of the model element. If multiple stereotypes are applied, the names of the applied stereotypes are shown as a comma-separated list with a pair of guillemets.



Stereotype is applied to element

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Process map diagram

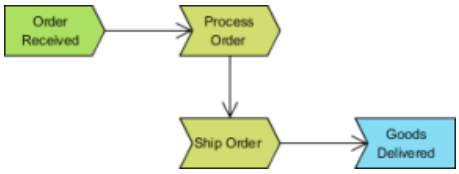
Process map diagram gives an overview that shows the processes needed to approach a business goal. This chapters teaches you how to create a process map diagram.

Creating process map diagram

This page teaches you how to create a process map diagram through the diagram navigator. There is a list of supported notations in process map diagram.

Creating process map diagram

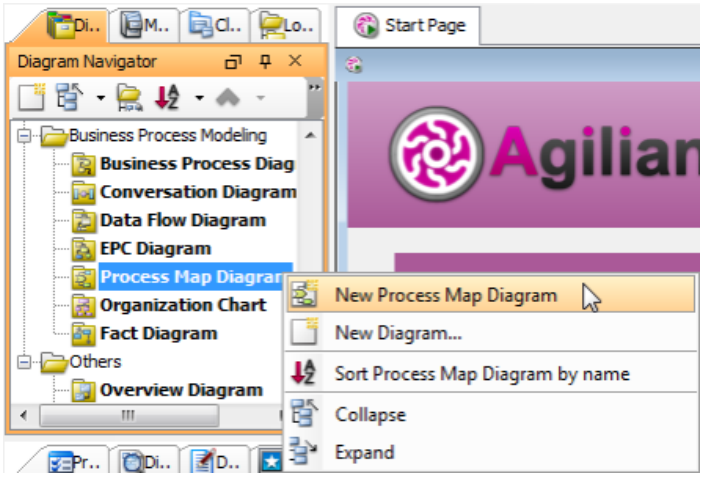
Process map diagram gives an overview that shows the processes needed to approach a business goal. It is rather in an upper level of analyzing and understanding a business process.



A sample process map diagram

Creating process map diagram

To create a process map diagram, right click on **Process Map Diagram** in **Diagram Navigator** and select **New Process Map Diagram** from the popup menu.



*Create a process map diagram through **Diagram Navigator***

Notations

Name	Representation	Description
Process	Process Order	A process is a part of process flow in achieving a goal.
Send	Order Received	An event that initiate the process chain.
Receive	Goods Delivered	The result of process chain.
Process Link	⟶	The flow of process.

A list of supported notations in process map diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Organization chart

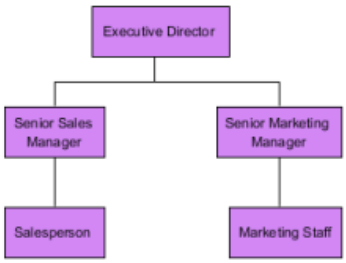
An organization chart is a diagram that visualizes the formal structure of an organization as well as the relationships and relative ranks of its positions. This chapters teaches you how to create an organization chart.

Creating organization chart

This page teaches you how to create an organization chart through the diagram navigator, as well as the steps to create subordinate, coworker and relocate branch.

Creating organization chart

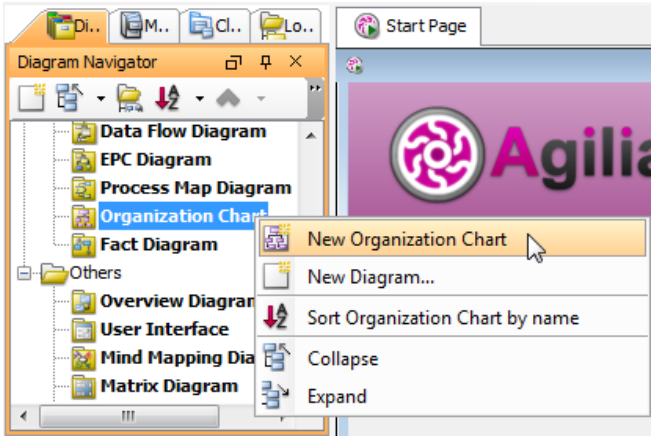
An organization chart is a diagram that visualizes the formal structure of an organization as well as the relationships and relative ranks of its positions. It is usually drawn and read from the top to the bottom. The default unit will pop out when a new organization chart is created.
In Agilian, organization chart is not only a diagram, but also a reference used for other parts of your model. For example, you may use an organization chart to depict the company hierarchy involved in a business process model. Its prime function is to help a business analyst to visualize efficiently the company structure as well as the division of works when performing business analysis.
The organization chart sample is shown below:



Organization chart sample

Creating an organization chart

To create a conversation diagram right click on **Organization Chart** in **Diagram Navigator** and select **New Organization Chart** from the popup menu.

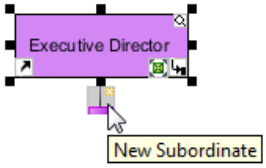


Create a new organization chart

Creating a subordinate

Subordinate, which is subject to the superior, is belonging to a lower rank. To create a subordinate under a superior unit:

1. Move mouse pointer on a unit and press its resource icon **New Subordinate**.



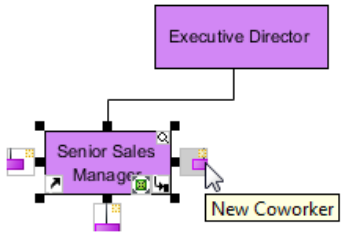
Create subordinate

2. Name the newly created subordinate unit and press **Enter** to confirm editing.

Creating a coworker

The coworker is a fellow worker of the same rank to the branch next to it. To create a coworker next to an existing unit:

1. Move the mouse pointer a unit and click its resource icon **New Coworker**, either on its left or right hand side. Clicking left resource icon will create coworker on the left of the unit, while clicking right resource icon will create coworker on its right.



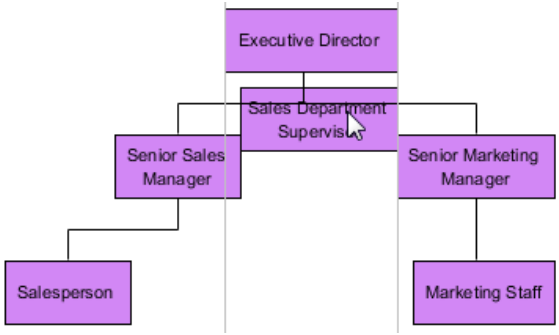
Create coworker

2. Name the newly created coworker unit and press **Enter** to confirm editing.

Relocating a branch

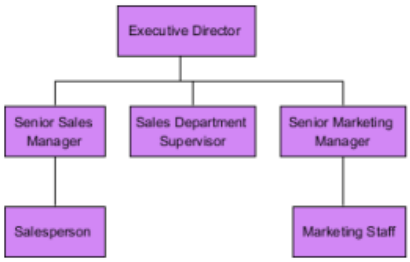
A unit can be relocated even when it has been placed under the subordination of another unit.

1. Press on a branch you want to relocate and drag it to the preferred branch.



Moving a unit

2. Release the mouse to confirm the position.



Completed relocating a branch

> **NOTE:** If you are not satisfied the relocation, press **Esc** to cancel the

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

RACI chart

A RACI chart is a matrix capable of showing how people or roles are related to business activities in a business process. You may form a RACI chart from a BPD to record the responsibility roles among swimlanes (i.e. Pool and lane) and activities (i.e. Task and sub-process.

Creating RACI chart

This page teaches you how to create a RACI chart through generating one from BPD.

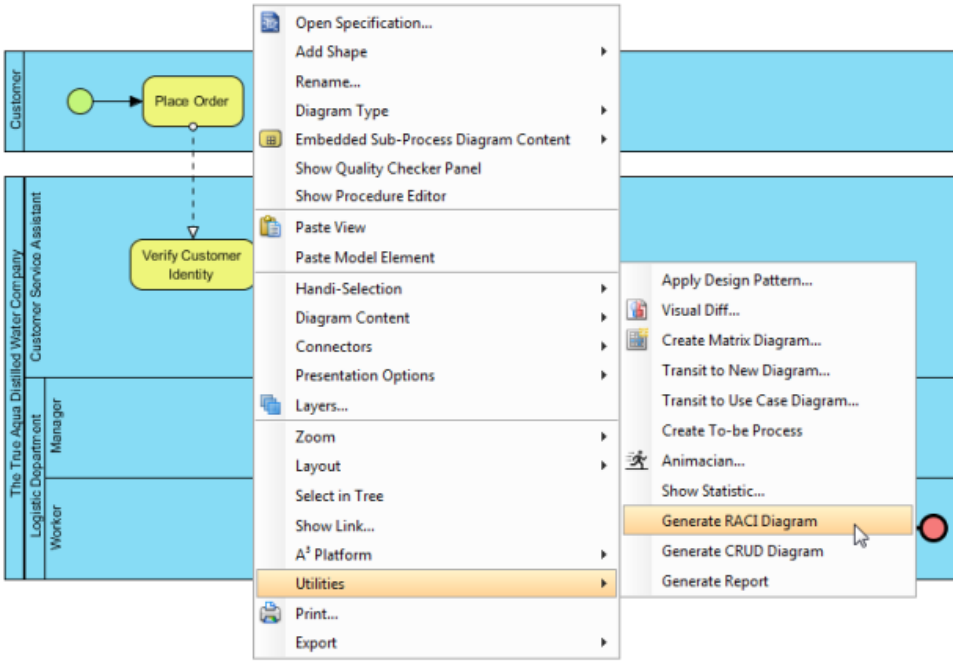
Creating RACI chart

A RACI chart is a matrix capable of showing how people or roles are related to business activities in a business process. You may form a RACI chart from a BPD to record the responsibility roles among swimlanes (i.e. Pool and lane) and activities (i.e. Task and sub-process. At any intersection of the participant and activity you can assign participant any of the four available responsible roles for a particular activity:

- Responsible - The participant who do the activity.
- Approver - The participant who approves or disapprove against an activity.
- Consulted - The participant who need to comment on the activity.
- Informed - The participant who need to be informed for any update about the activity.

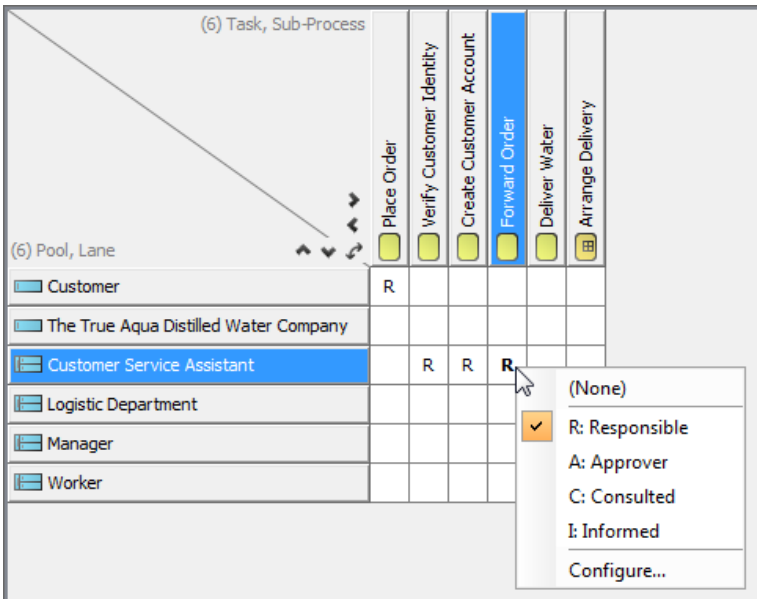
To create a RACI chart from BPD:

1. Right click on the background of a BPD and select **Utilities > Generate RACI Diagram** from the popup menu.



Generate RACI chart from BPD

2. The role R, abbreviation for responsible, is automatically assigned to the intersection of participant and activitiy whenever the activity is put inside an participant. You may add or remove roles by clicking on a cell in the chart and update the role selection.



Update the responsibility roles

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube

- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Business rule

A business rule defines guideline with necessary constraint(s) needed for executing certain business operations. You will learn how to manage business rules.

Managing business rules

Shows you how to use the rule editor to edit business rule.

Business rule grid

Shows you how to use the rule grid to manage business rules.

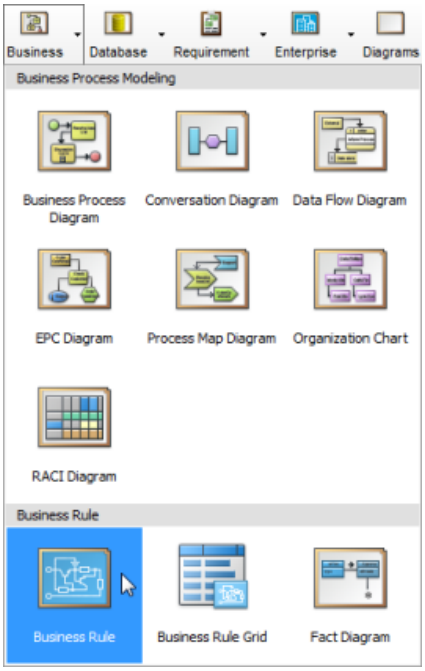
Managing business rules

A business rule defines guideline with necessary constraint(s) needed for executing certain business operations. You can record and describe business rules with rule editor, as well as to identify the term (vocabulary) involved in the rule, which helps tracing fact concepts around rules.

Defining a business rule

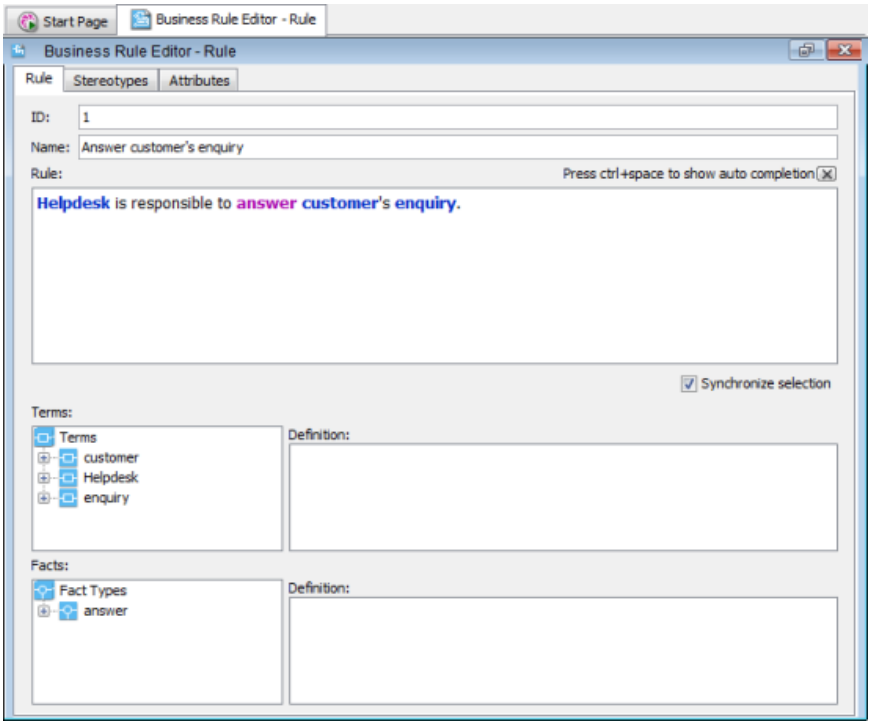
If you want to define a business rule, you need to create the rule and define it in the rule editor. Here are the steps:

1. Click on **Business** in toolbar, and select **Business Rule** from the drop down menu.



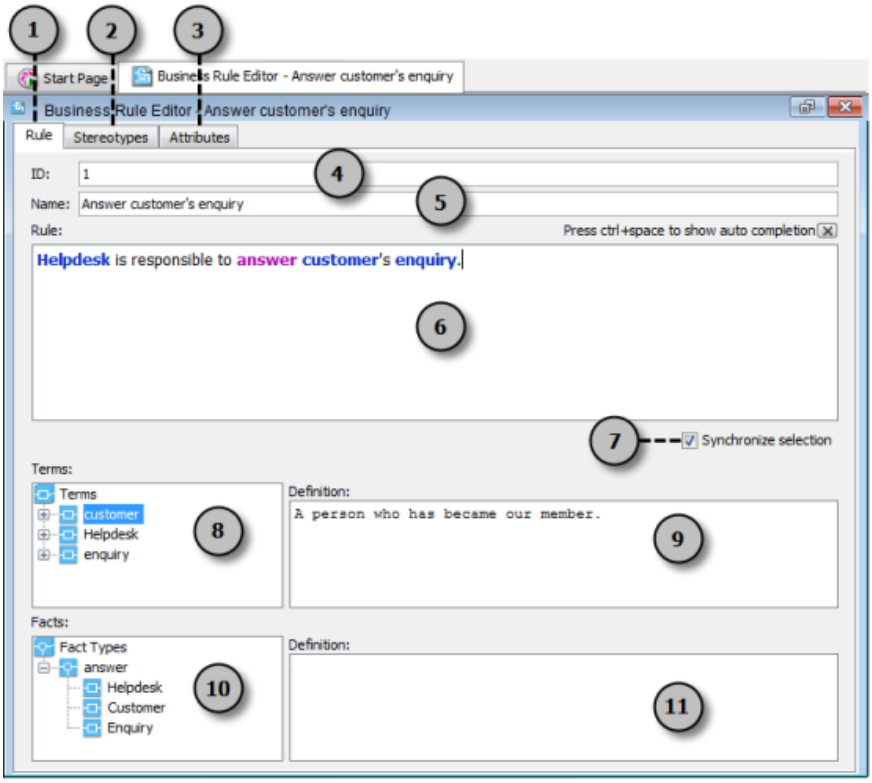
To create a business rule

2. This shows the rule editor. An ID, which is a number indicating the order of rule creation, is assigned automatically. You may change it if you like. Name the rule with a short and descriptive phrase. Fill in the rule content in the **Rule** field.



Rule editor

An overview of rule editor



Overview of rule editor

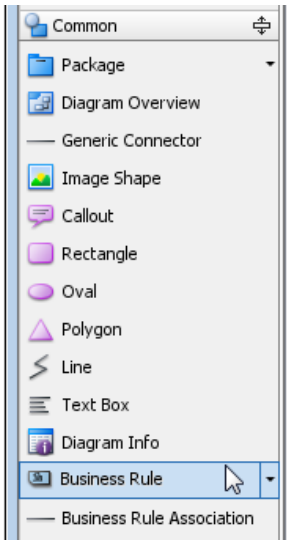
No.	Name	Description
1	Rule	A tab that shows the primary rule information such as its name and definition
2	Stereotypes	A list of stereotypes applied to the rule. You can extend a rule from a stereotype to add domain specific meaning to it. For example, you can extend a rule from stereotype critical to represent an important rule.
3	Attributes	Read, add and remove attributes from the rule. Attributes can be added to denote extra properties to a rule.
4	ID	A value that makes each rule unique. When you create a rule, an ID will be assigned automatically. The assigned ID indicates the order of rule creation.
5	Name	A short phrase that describes the rule.
6	Rule definition	A longer and more detailed description of rule.
7	Synchronize selection	When the checkbox is checked, the **Terms** and **Facts**' active node selection will follow the selection as pointed by the mouse pointer in the **Rule** definition field.
8	Terms	A list of terms that involve in the rule definition.
9	Term definition	By selecting a term in Terms list, its definition will appear in the Definition field.
10	Facts	A list of facts that involve in the rule definition.
11	Term definition	By selecting a term in Facts list, its definition will appear in the Definition field.

Description of rule editor

Visualizing business rule on diagram

Instead of creating a business rule as described above, you can also create a business rule shape on a diagram. Although there is no specific type of diagram made for presenting business rule, you can draw business rule on any type of diagram. To draw a business rule on a diagram:

1. Scroll to the **Common** category in diagram toolbar and select **Business Rule**.

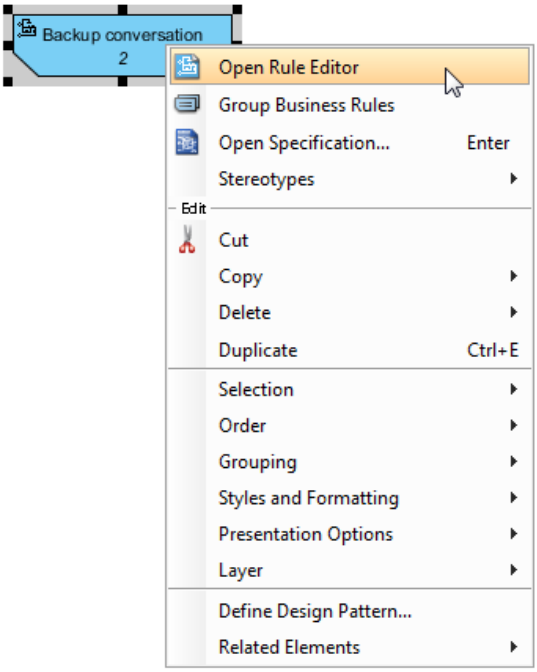


Selecting Business Rule in diagram toolbar

2. Click on the diagram to create a business rule. Name the rule with a short and descriptive phrase and press **Enter** to confirm.

A business rule is created

If you need to describe the rule in detail, right click on the rule shape and select **Open Rule Editor** from the popup menu. After that, fill in the rule definition in rule editor as mentioned above.

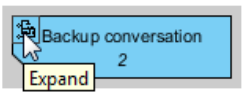


Open the rule editor to define the rule

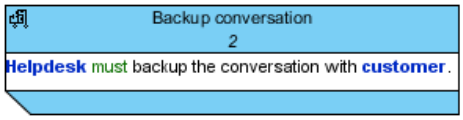
Expanding and collapsing rule

A business rule shape has two visual states - collapsed and expanded. In collapsed state, business rule hideaway the definition of rule, while in expanded state, a new compartment will appear in the middle rule shape for showing the rule definition.

To expand or collapse a rule shape, click on the top left corner of rule shape.



Expand a collapsed rule shape

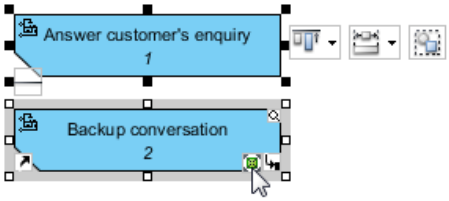


An expanded rule shape

Business rule group

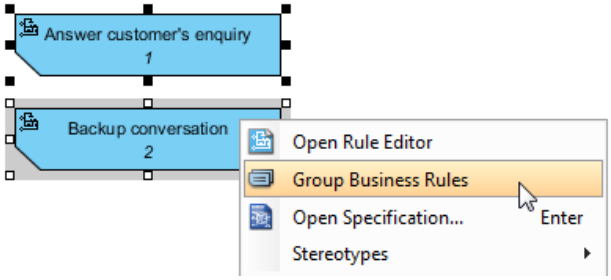
If you find a number of rules are of the same category, you can group them.

1. Select the rule shapes.

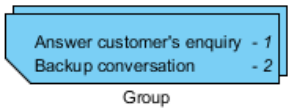


Select rules

2. Right click on the selection and select **Group Business Rules** from the popup menu.



To group rules



Rules are grouped

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

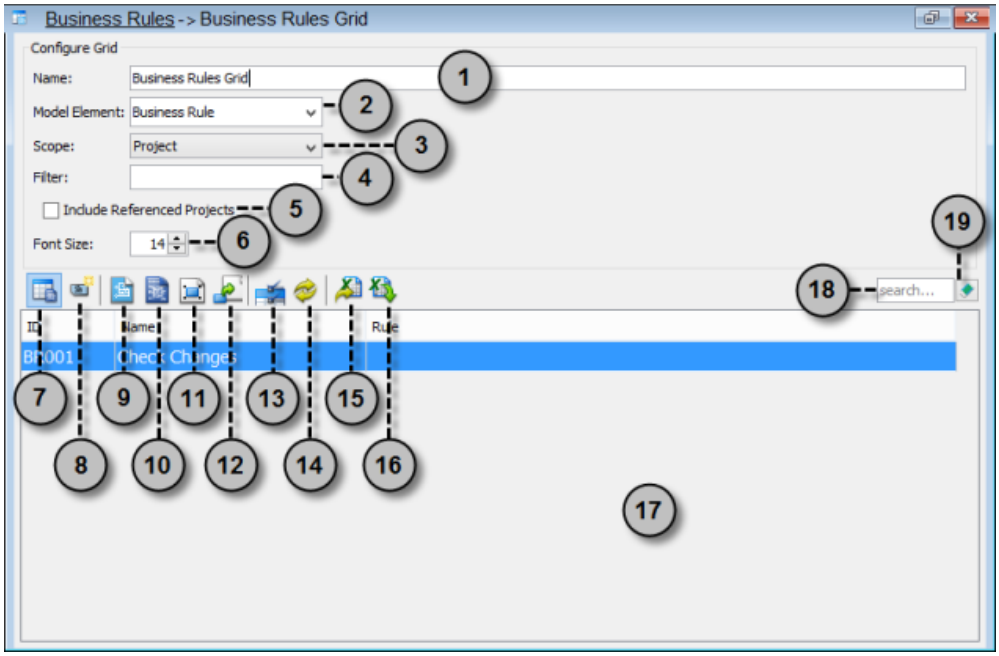
Business rule grid

Business rule grid is a table with business rules listed in it. It lets you to access all business rules in a project or diagram, lookup and create business rule.

Creating business rule grid

- Click on **Business** on toolbar and select **Business Rule Grid** from the drop down menu **.**
- Select **File > New Diagram > Business Modeling > Business Rule Grid** from the main menu.

The overview of Business Rule Grid



The Business Rule Grid

No.	Name	Description
1	Name	The name of this **Business Rule Grid**.
2	Model Element	Change the type of model element to be listed in grid. Although **Business Rule** is selected by default, you may change it to other types.
3	Scope	The location to look for the business rules to list in grid. By default, business rules are found from the whole project. You can change to find use cases from specific model or package, or to find only business rules right at the root level. You can also restrict the scope to all diagrams, to within a specific diagram or to all business rules that has not been visualized in any diagram.
4	Filter	Apply filter to grid content. Text entered here is matched against the **Name** property of business rules listed in grid. Business rules that do not contain the entered text in their name are hidden.
5	Include Referenced Projects	Check it to list also business rules in referenced projects, in **Business Rule Grid**.
6	Font Size	Click to adjust the font size of text in **Business Rule Grid**.
7	Configure Grid	Click to show/hide the grid configuration panel, which allows you to enter the name of grid, the model element to be listed in grid, the scope and to apply filter to grid content.
8	New Business Rule	Click to create a business rule.
9	Open Rule Editor	Select a business rule in **Business Rule Grid** and click this button to open the rule editor.
10	Open Specification...	Select a business rule in **Business Rule Grid** and click this button to open its specification.
11	Open Business Rule Details	Select a business rule in **Business Rule Grid** and click this button to acces its details in **Business Rule Details** editor.
12	Show View...	Select a business rule in **Business Rule Grid** and click this button to list the diagrams that contains the view of the selected business rule.
13	Visualize...	Select a business rule in **Business Rule Grid** and click this button to show it in a new or existing diagram.
14	Configure Columns...	Click to select the property(ies) of business rules to be listed in the grid, as columns.
15	Refresh	Click to refresh the grid content by showing the most updated information of business rules listed.

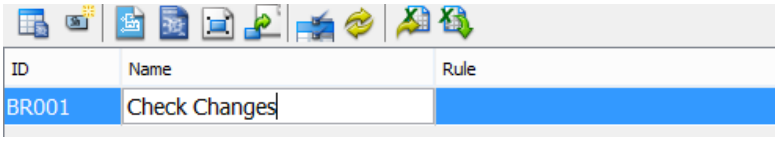
16	Export to Excel	Click to export grid content to Excel file.
17	Import from Excel	Click to import grid content from exported Excel file.
18	List of business rules	Business rules are listed here.
19	Search	Find business rule(s) by entering search criteria.
20	Clear	Click to clear the text entered in **Search** box.

The fields in Business Rule Grid

Creating business rule in Business Rule Grid

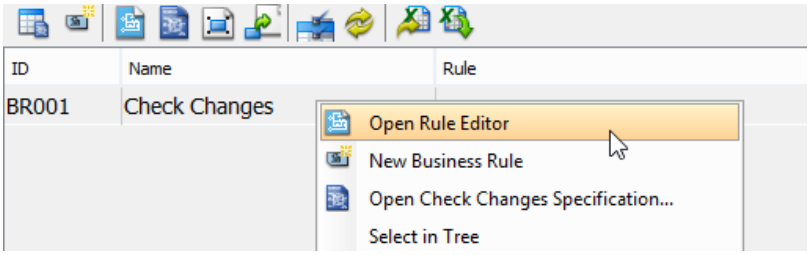
To create a business rule in **Business Rule Grid**:

1. Click on **New Business Rule** above the **Business Rule Grid**.
2. Enter the name of business rule.



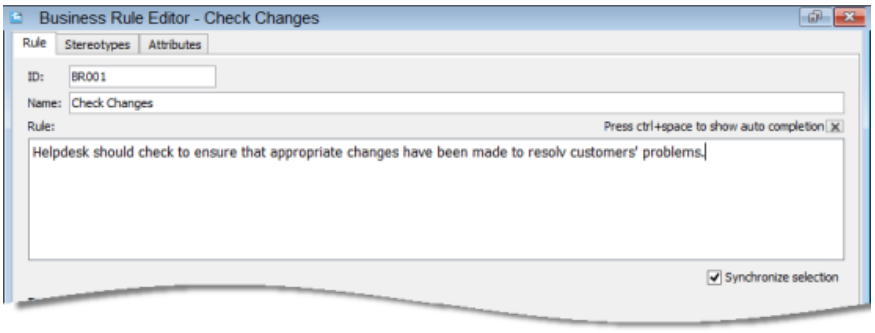
Creating business rule in Business Rule Grid

3. Press **Enter** to confirm editing.
4. To enter the details of business rule, right click on the new business rule and select **Open Rule Editor** from the pop-up menu.



Open Rule Editor

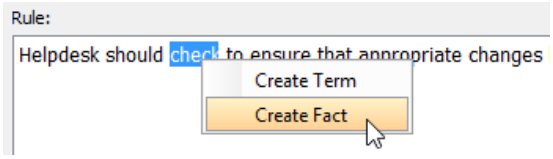
5. In **Business Rule Editor**, enter name and rule for the newly created business rule.



Business Rule Editor

Creating fact

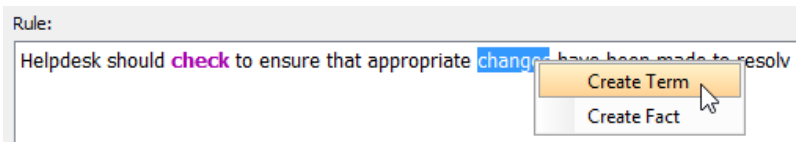
Highlight a word which you want to create it to be a fact and select **Create Fact** from the pop-up menu.



Create check as fact

Creating term

Highlight a word which you want to create it to be a term and select **Create Term** from the pop-up menu.



Create changes as term

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Fact diagram

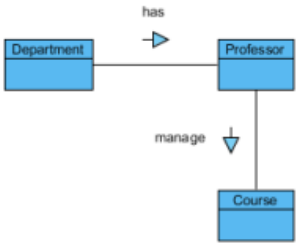
Record and related terminologies under specific business domain by using fact diagram.

Creating fact diagram

Shows you how to use the fact diagram to model the relationships between terms in a business domain.

Creating fact diagram

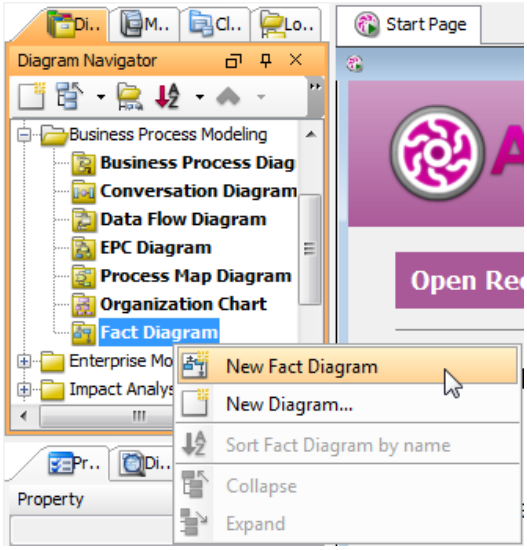
Record and related terminologies under specific business domain by using fact diagram. In a fact diagram, terms are visualized as rectangular blocks. You can link terms up with connectors, and specify the kind of relationship in between. A fact diagram is closely related to business rules. It helps you identify the rules by studying the relationship between terms.



A sample fact diagram

Creating fact diagram

To create a fact diagram, right click on **Fact Diagram** in **Diagram Navigator** and select **New Fact Diagram** from the popup menu.



*To create a fact diagram through **Diagram Navigator***

Notations

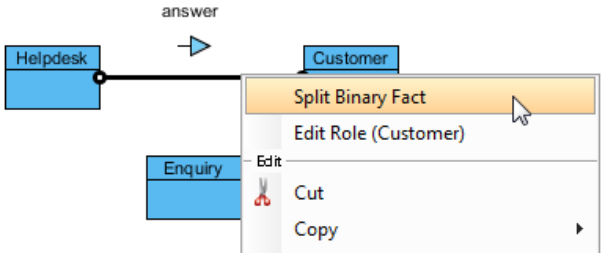
Name	Representation	Description
Term	Department	A meaningful noun or noun phase that business participants recognize and use in communication.
Fact Type	Fact	A fact type represents the relationship between terms.
Fact Association	has	A fact association connects terms for visualizing the relationship in between.
Term-Fact Type Association	Fact	By connecting a term to a fact type, it represents a u-nary fact-type that provides a simple yes or no answer to business question.
Generalization		Represents a parent and child relationship between terms.

A list of supported notations in fact diagram

Spliting and joining binary fact

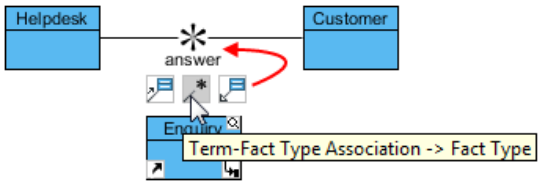
When a fact type involves more than two terms, we call it a n-ary fact type. For example, fact type worded "helpdesk answer customer's enquiry" involves three terms helpdesk, answer and enquiry. If you have already created a fact model that involves two terms, and now want to add an additional one, you need to split a fact type, and connect the split fact type with the new term.

1. Split a fact type by right clicking on the fact association and selecting **Split Binary Fact** from the popup menu.



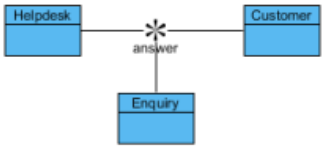
Split a binary fact

2. Move the mouse pointer over the added term. Press on the resource **Term-Fact Type Association -> Fact Type** and drag to the split fact type.



To connect term with fact type

3. Release the mouse button. This connects the new term with split fact type.



Term is connected with fact type

On the contrary, you can join a split fact type by right clicking on an association and selecting **Join Fact Type** from the popup menu.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Decision diagram

Represent complex decision situations and business rules in the most simplest, organized manner.

Creating decision table

Shows you how to create and understand decision table.

Documenting rules, conditions and actions in decision table

Shows you how to edit the documentation of rules, conditions and actions in decision table

Creating decision table

In order to develop a truly functional information system, features must be designed and developed base on users’ business needs, with behaviors following the business rules strictly. Decision table provides a compact way to represent complicated business rules. Thanks to the easy to comprehend layout, decision table can be understood by developers and end-users easily.

Decision table involves three sections - conditions, actions and rules. From developer’s point of view, decision table is pretty much like the tabular form of an if-then-else statement. Business users use decision table to document business rules, while system developers study a decision table to think about the right way to implement those rules.

	Rules					
Conditions	1	2	3	4	5	6
C1. Infant passengers (< 2)	Y	Y				
C2. Youth passengers (>2 and <16)			Y		Y	
C3. Domestic flights	Y					
C4. International flights		Y				Y
C5. Early reservation				Y	Y	
C6. Off-season traveling						Y
Actions	1	2	3	4	5	6
A1. Offer 10% discounts			Y	Y		
A2. Offer 15% discounts						Y
A3. Offer 20% discounts					Y	
A4. Offer 70% discounts		Y				
A5. Offer 80% discounts	Y					

Decision table sample

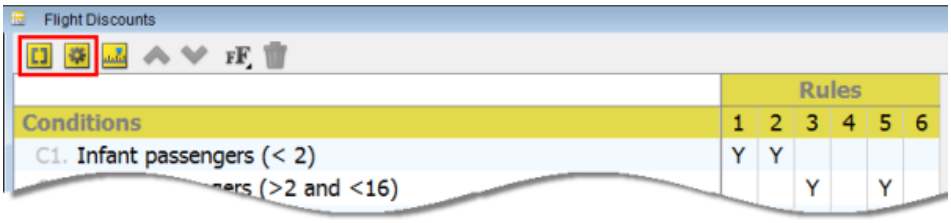
The condition rows in a decision table list out the factors that can influence the final decision. The action rows list out the possible operations to perform. Each of the rule columns represents a combination of condition(s) and action(s), meaning that when one or more conditions are met, action or multiple actions will be performed accordingly. Decision table does not enforce any rule regarding the way cells are filled. People usually use simple true/false (or simply T/F, Y/N) values to represent the matching of conditions and actions. Some prefer using ticks. There is real limitation though.

Creating decision table

- Click on **Business** on toolbar and select **Decision Table** from the drop down menu **.**
- Right click on **Decision Table** in **Diagram Navigator**, under the Business Modeling category, and select **New Decision Table** from the popup menu.
- Select **File > New Diagram > Business Modeling > Decision Table** from the main menu.

Creating conditions and actions

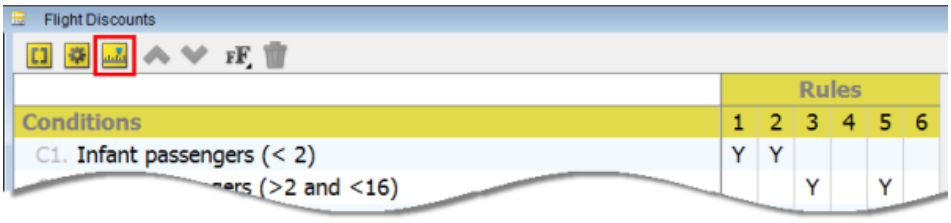
The first two buttons in the toolbar at the top of the decision table allows you to create conditions and actions respectively. By selecting an existing condition or action and click on the create button, a new condition/action will be inserted after the selected condition/action. Then, enter the description of condition/action. Alternatively, you can right click on a condition/action and add a new one after it via the popup menu.



Buttons to create conditions and actions

Creating rules

The third button in the toolbar at the top of the decision table allows you to create rules. By selecting an existing rule and click on the create button, a new rule will be inserted after the selected rule. Find the conditions and actions that match your rule. Double click on the corresponding cell and place a mark on it. For example, enter "Y" to indicate "Yes". Alternatively, you can right click on a rule and add a new one after it via the popup menu.



Buttons to create conditions and rules

Reorder conditions, actions and rules

To reorder conditions and actions, select them in the table and click on the **Move Up/Down** button in toolbar. Or right click on them and select **Move Up/Down** from the popup menu. To reoder rules, select and right click on them, select **Move Left/Right** from the popup menu.

Delete conditions, actions and rules

To delete conditions, actions and rules, select the rows or columns to delete, click on the **Delete** button in toolbar.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

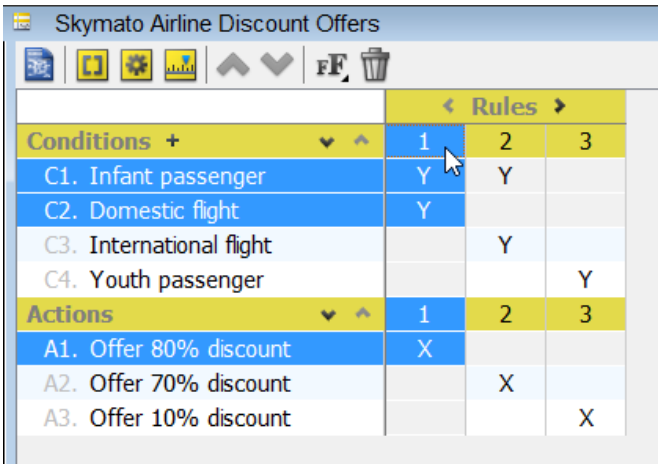
- What is decision table?
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Contact us if you need any help or have any suggestion

Documenting rules, conditions and actions in decision table

You may want to describe the reason why the rules are setup in the way they are in a decision table. You can do this by editing the documentation of rules added in a decision table. Besides rules, you can also edit the documentation of conditions and actions.

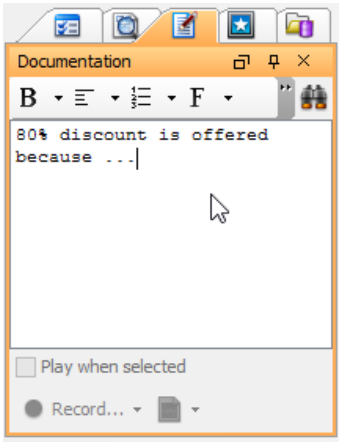
Documenting rules, conditions and actions in decision table

1. Select the desired rule in the decision table.



Selecting a rule

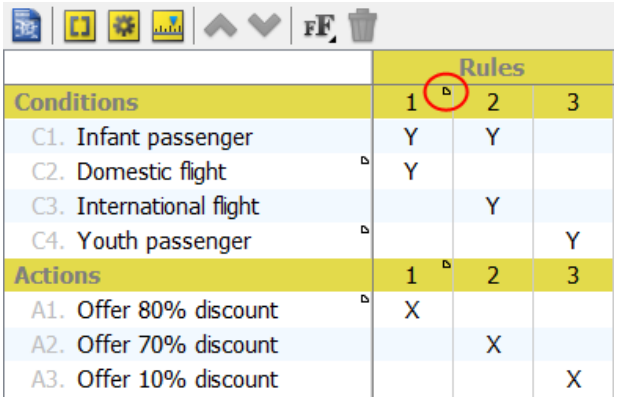
2. Edit the documentation in the **Documentation** pane (**View > Panes > Documentation**).



Documentation entered to a rule

Identifying rules, conditions and actions that have documentation entered

Dog ear icons are shown in rules, conditions and actions that have documentation entered. You can read the documentation by selecting a rule/ condition/action and viewing the **Documentation** pane.



Dog ear icon

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- What is decision table?
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian

- Visual Paradigm on YouTube
- Contact us if you need any help or have any suggestion

Visual simulation

With simulacian, you can simulate the execution of business process for studying the resource consumption (e.g. human resources, devices, etc.) throughout a process, identifying bottlenecks, quantifying the differences between improvement options which help study and execute process improvements. This chapter provides you with detailed information about simulacian.

What is simulacian?

Introduce simulacian, and shows you some of the key features like simulation and resource chart.

Simulacian control panel

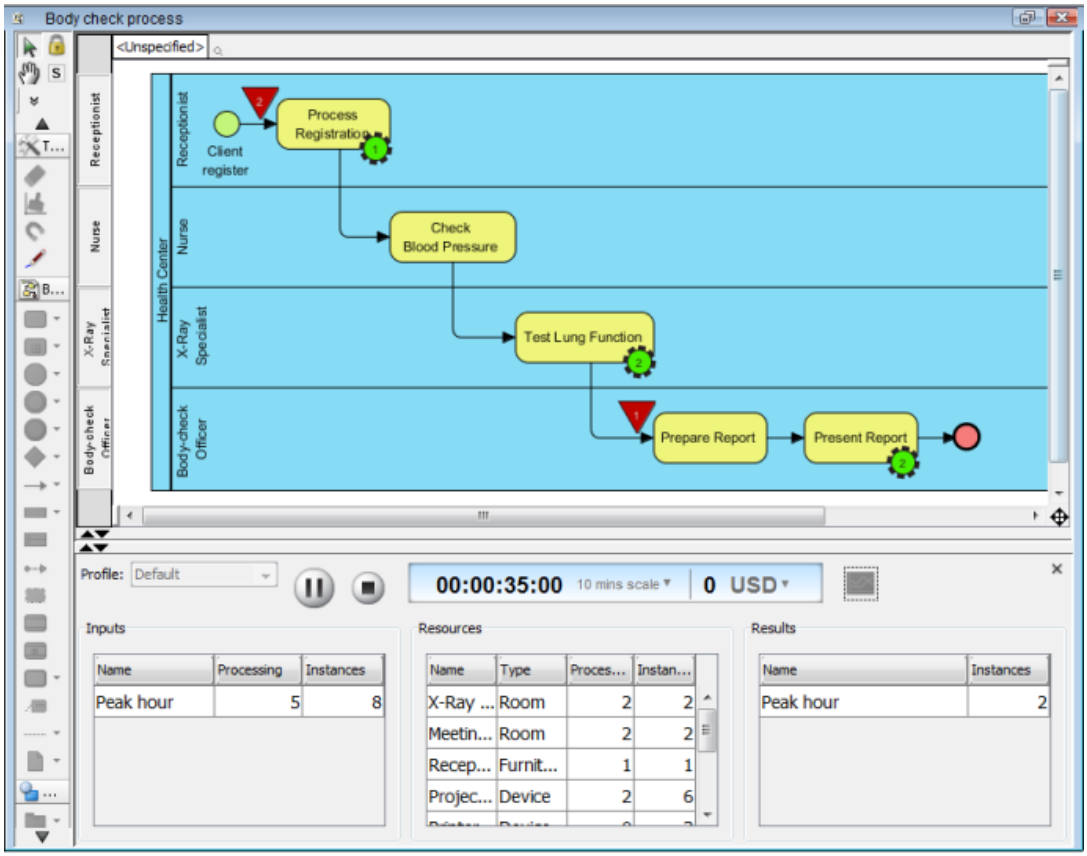
The simulacian control panel is where you can configure and run a simulation. It consists of several parts, and will be described in detail.

Simulating business process

This page is aimed to give you an example to show you how to simulate a business process diagram.

What is simulacian?

The objective of performing business process modeling is to facilitate the communication with stakeholders, to perform cost and benefit analysis and to perform process improvement, etc. Simulacian is a set of value-added tools designed to aid business process modeling. With simulacian, you can simulate the execution of business process for studying the resource consumption (e.g. human resources, devices, etc.) throughout a process, identifying bottlenecks, quantifying the differences between improvement options which help study and execute process improvements.

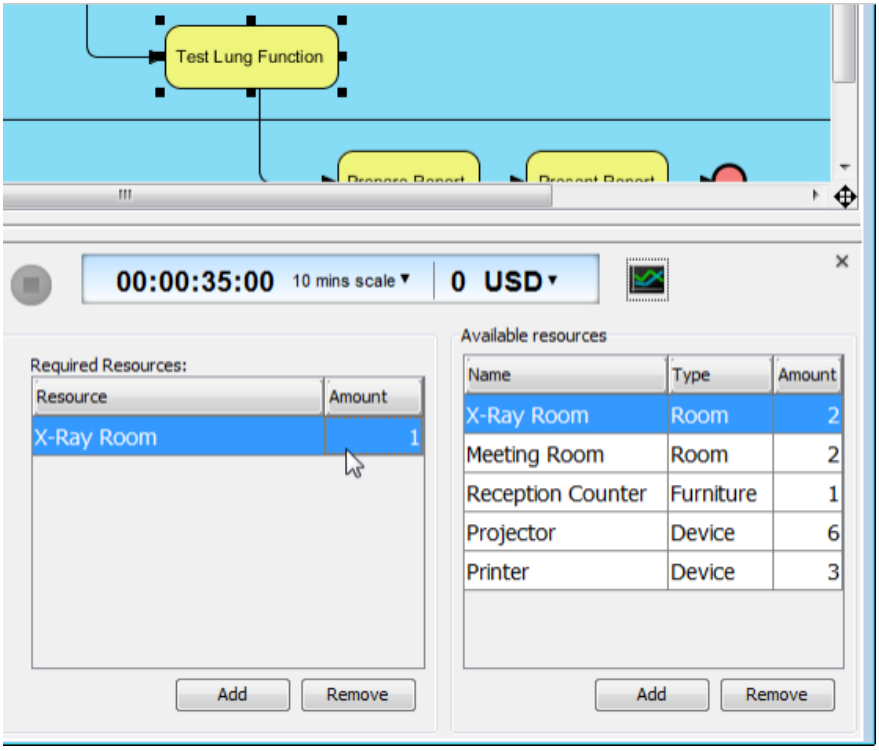


*Simulating a business process with **Simulacian***

Key concepts

Resources

Resources refer to any kind and form of input essential for the execution of a process. Each resource has three properties - name, type and amount. There are two types of resources - available resources and required resources. Available resources refer to the resources that can be used by business process, but may not be fully used. For example, a post office has 10 counters as resources, but only 3 are in used at peak hours. Required resources are a flow object wide option. You can set the resource and the amount of resource required by completing a flow object. For example, task *Answering Enquiry* requires 1 counter as resource.



Required and available resources

Very often, the allocation of resource is critical to the efficiency of a business process. For example, if there are more available staffs and counters, this helps increase the efficiency of customer service. But of course, if the available staffs and counters are more than enough, those non-used resources are wasted. With simulacian, you can determine the optimal resource allocation by evaluating the resource consumption of current process.

Duration

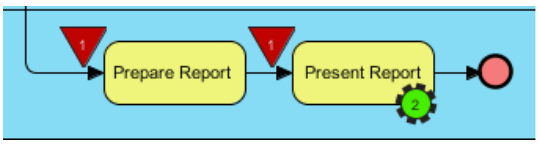
Duration is the time elapsed from the entering of flow object until the leaving of that flow object. It is understandably that the duration of flow object has significant effect to the efficiency of a business process. Imagine if it takes 5 minutes to complete the process of just one payment in a supermarket, there will accumulate a long queue waiting for paying.

Input

Input is a way of simulating a given business process. It has a name that describe the input, and an instance, which is a number that represent the number of time the input will happen at a particular instant. If you have modeled a general order processing system, you can add an input public holiday, with instance 100 to represent the case that in public holiday there will be 100 customers that need to undergo payment. In order to help you improve your process, you must set input that reflects the reality. If you set 10000 as instance of input *public holiday* which will never happen, you will not obtain useful information to aid in process improvement.

Simulation

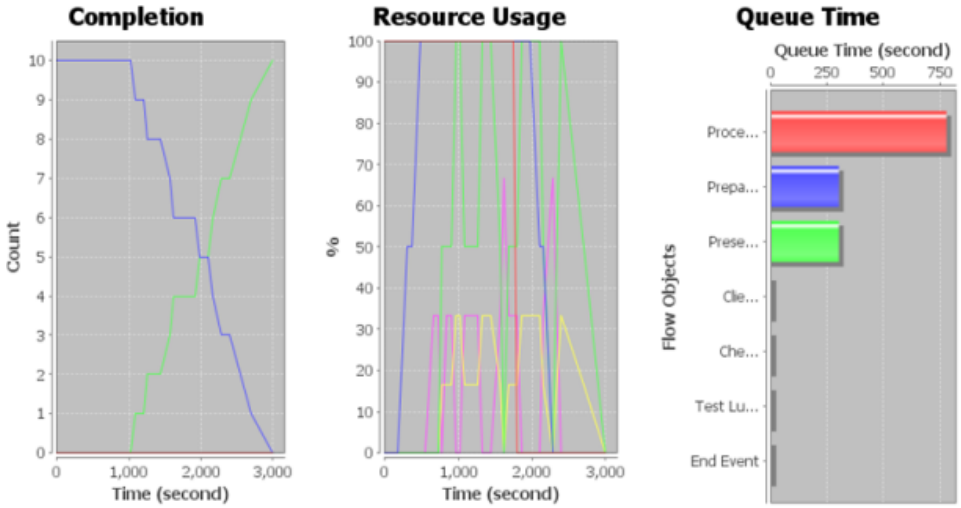
Once you have specified the available and required resources, the duration of flow objects and added input(s), you can run simulation. During simulation, diagram will be locked to avoid collision between your edit action and the simulation operation. Executing jobs are represented by a running green gear shape, with a number indicating the number of running job, and are attached to the task where the job is being processed. Pending jobs are represented by inverted triangles, with a number indicating the number of pending jobs.



Executing and pending jobs

Performance analysis through charts

During simulation of business process, you can identify the bottleneck(s) by observing the occurrence of pending jobs (i.e. the inverted triangle). This works well in a relatively small process. However, if your business process diagram is large you may not be able to study the simulacian outcome just by observation because the simulacian can be lengthy and involve several bottlenecks. Furthermore, you may want to know the exact figure of resource consumption for conducting a more accurate resource re-allocation plan. In those cases, you can produce simulacian charts that report the completion of inputs, resource usage and queue time of flow objects against time.



Different kinds of chart

Related Resources

The following resources may help you learn more about the topic discussed in this page.

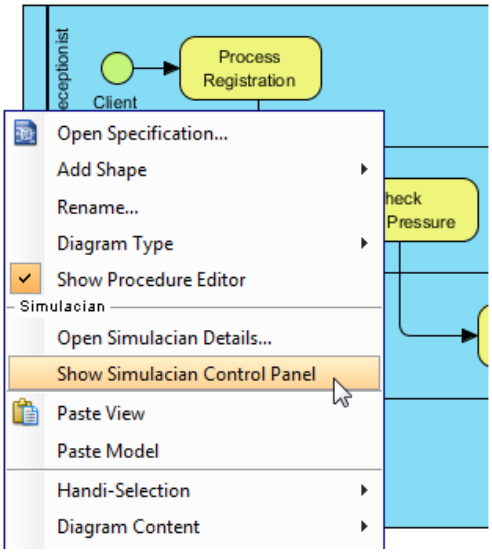
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Simulacian control panel

In order to simulate a business process you need to define simulation details like available and required resources, duration of tasks/sub-processes, instances of pools/lanes and inputs. All these information can be defined in simulacian control pane, which is a pane that display at the bottom of diagram, important for adjusting any settings related to simulacian. The panel will be updated base on your selection in active diagram. Besides setting simulacian details, start/pause of simulation can also be done in the panel.

Opening simulacian control panel

To open the simulacian control panel, right click on thebackground of business process diagram that you want to simulate and select **Show Simulacian Control Panel** from the pop-up menu.

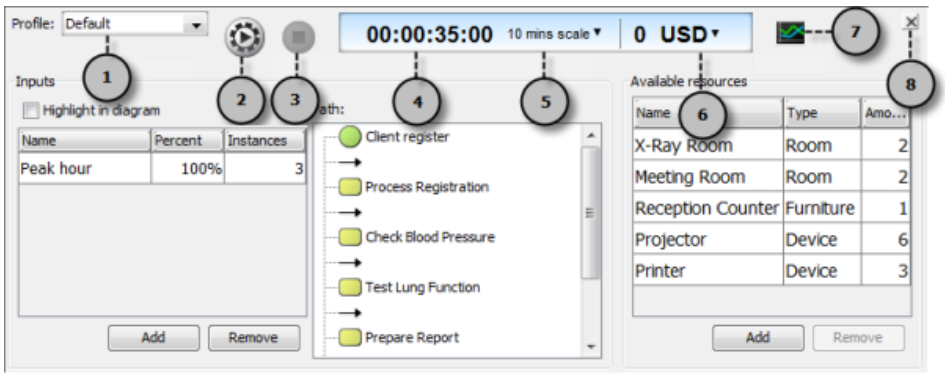


Open simulacian control panel

NOTE: You can open simulacian control panel only for diagram that has selected **Simulacian** as diagram type. To check/edit diagram type, right click on the background of business process diagram and select **Diagram Type** from the popup menu.

Overview of simulacian control panel

Common



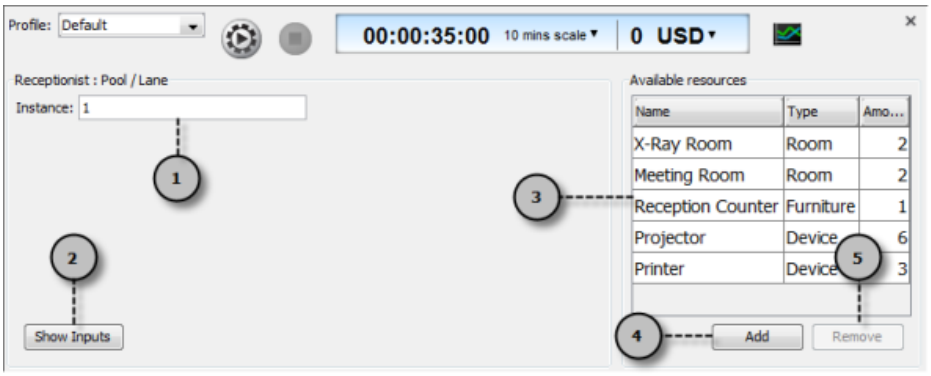
An overview of simulacian control panel

No.	Name	Description
1	Profile	To create a new profile for simulacian, select **Configure Profile...** from the drop-down menu and name the newly created profile in the **Configure Profile** dialog box.
2	Start / Pause	Click to start simulation when paused or stopped, base on the resource, duration and input settings, or to pause a simulation when it is playing.
3	Stop	Stop a simulating business process.
4	Clock	Displays the time elapsed from the beginning of simulation until the current moment. It is for reflecting the duration of the execution of business process, and is based on the selection of time scale.
5	Time scale	Control the speed of simulation. For example, a selection of *10 mins scale* simulates the business process in speed 10 min per second.
6	Current unit	Click to select the current unit, such as US Dollar, Hong Kong Dollar and Yen, for the consumption during simulating.

7	Simulacian charts	Click to display a new window that display the completion, resource usage and queue time charts base on the settings of resource, duration and input. You can treat it as a chart form of simulation outcome.
8	Close	Click to close the simulacian control panel. You can open it again by right clicking on the business process diagram and selecting **Show Simulacian Control Panel** from the popup menu.

Description of simulacian control panel

When selected pool/lane

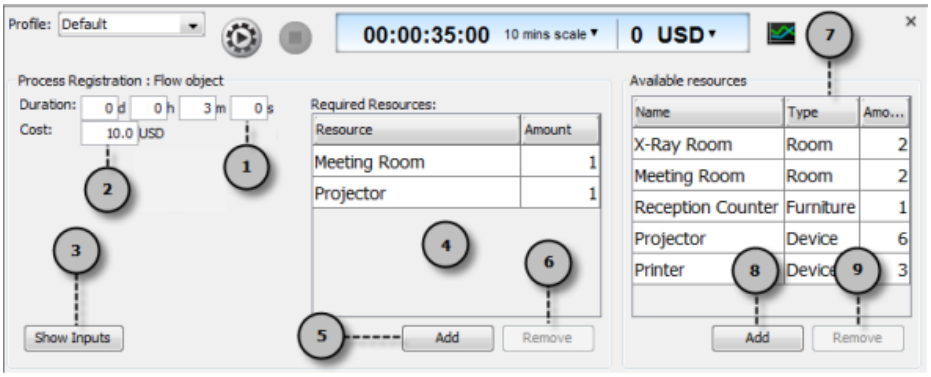


An overview of simulacian control panel while selected pool/lane

No.	Name	Description
1	Instance	When you model a business operation in business process diagram, you use pools and lanes to represent participants and sub-participants of the process, such as *Client* and *Receptionist*. No matter how many actual participants are there, you will still use a single pool (or lane) to represent all of them. For example, you will draw a pool *Receptionist* to represent all receptionists instead of drawing 5 pools for representing the fact that there are 5 receptionists. Here the **Instance** field let you set the number of instances of selected pools or lanes. If there are 5 receptionists, enter 5 for instance of pool/lane *Receptionist*. Provided that there are sufficient resources for performing jobs, the number of instances affects the performance of process - The more instances, the more efficient. During process improvement, you can adjust the instance to evaluate the impact of increasing or decreasing the number of staff to handle certain job.
2	Show inputs	Inputs are the expected way of executing a business process during simulacian. Click **Show Inputs** to list the inputs, and add/remove inputs in further.
3	Available resources	The table of available resources list the resources needed by the business process. For example, a process of body checking has resources X-Ray room and reception counter. Each resource has a name, a type and its amount.
4	Add available resource	Click to add an available resource by giving its name, selecting/entering its type and setting its amount.
5	Remove available resource	Select an available resource and click this button to remove it.

Description of simulacian control panel while selected pool/lane

When selected flow objects



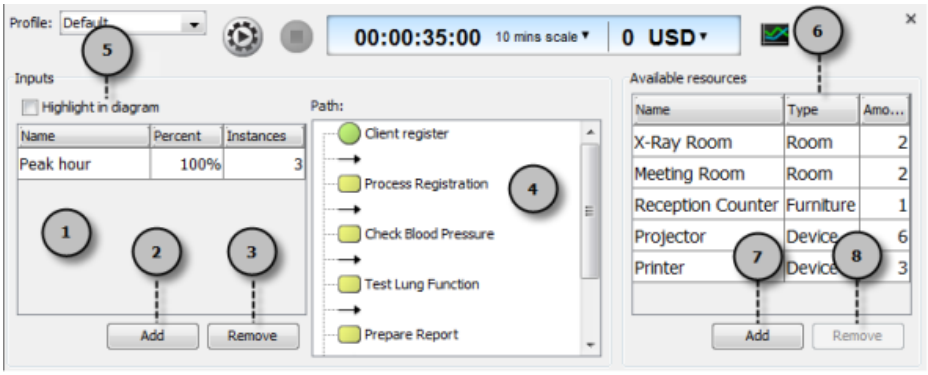
An overview of simulacian control panel while selected flow object

No.	Name	Description
1	Duration	The time the selected flow object needs to take to process and complete a job. The 4 boxes d, h, m and s mean day, hour, minute and second, respectively. For example, it takes five 5 minutes to present a report

		to client in a body check operation, set the business task *Present Report*'s duration to be 5 m, meaning 5 minutes. The duration setting affects the performance of process - The less time needed, the more efficient. However, do not forget that the less time it takes to complete a task may affect the quality of work. During process improvement, you need to keep the balance between efficiency and quality of work, and adjust the duration accordingly.
2	Cost	The cost the selected flow object needs to consume to process and complete a job. The current unit you selected in common control pane will be shown in here as the unit of cost. For example, you select US dollar in current unit, it will be shown USD in here.
3	Show inputs	Inputs are the expected way of executing a business process during simulacian. Click **Show Inputs** to list the inputs, and add/remove inputs in further.
4	Required resources	The resources the participant needed in order to complete a job when processing the selected flow object. The **Resource** column shows the names of resources. The **Amount** column shows the number of resource needed to use in completing the task per participant. For example, to present report to client, you need one resource *Meeting Room*, and one resource *Projector*.
5	Add required resource	Click to add the resource the participant needed in order to complete a job when processing the selected flow object. Make sure you have available resources defined in order to select a required resource. Also note that you cannot set an amount that exceed the amount set in available resource. For example, if you have 6 available projectors (resource), the maximum of projector a flow object can require is from 0 to 6.
6	Remove required resource	Select a required resource and click this button to remove it.
7	Available resources	The table of available resources list the resources needed by the business process. For example, a process of body checking has resources X-Ray room and reception counter. Each resource has a name, a type and its amount.
8	Add available resource	Click to add an available resource by giving its name, selecting/entering its type and setting its amount.
9	Remove available resource	Select an available resource and click this button to remove it.

Description of simulacian control panel while selected flow objects

When selected diagram background / showing inputs



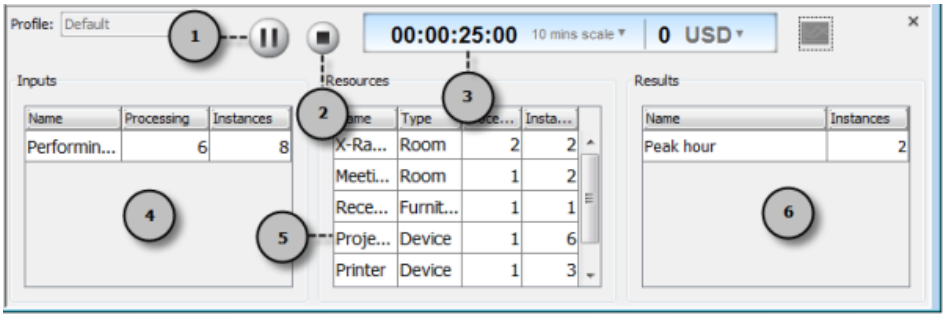
An overview of simulacian control panel while selected diagram background or showing inputs

No.	Name	Description
1	Inputs	Inputs are the definitions of how to execute a business process during simulation. An input consists of a selection of possible execution path formed by flow objects in diagram, with the number of instances, which represents the number of time the path will be executed at a specific instant. For example, if you want to simulate the case that there are 10 clients needed to perform body checking in a business process of body checking, to see if the process can handle this situation well, you will add an input *Performing body check*, with 10 as number of instances.
2	Add input	Click this button to add an input with name and number of instances.
3	Remove input	Select an input and click this button to remove it.
4	Path	The flow objects involved in an input. If there is a gateway in your diagram, you need to make a decision to the outgoing path of the gateway object.
5	Highlight in diagram	Check this button to make the diagram highlight the path involved in the input selected in **Inputs** table.
6	Available resources	The table of available resources list the resources needed by the business process. For example, a process of body checking has resources X-Ray room and reception counter. Each resource has a name, a type and its amount.

7	Add available resource	Click to add an available resource by giving its name, selecting/entering its type and setting its amount.
8	Remove available resource	Select an available resource and click this button to remove it.

Description of simulacian control panel while selected diagram background or showing inputs

When simulating



An overview of simulacian control panel while simulating

No.	Name	Description
1	Pause	To pause a simulation temporarily.
2	Stop	To stop simulating the business process.
3	Clock	To display the time elapsed from the beginning of simulation until the current moment. It is for reflecting the duration of the execution of business process, and is based on the selection of time scale.
4	Inputs	A list of inputs with their completeness throughout the simulation. The **Processing** column shows the jobs under processing by the simulating process. The **Instances** column shows the amount of non-completed jobs. It should keep decreasing, and become 0 at the end of simulation.
5	Resources	A list of resources with the status of consumption throughout the simulation. The **Processing** column shows the amount of resources be consumed by the simulating process. The **Instances** column shows the total amount of resources. You can observe this table to study the current resource allocation.
6	Results	A list of completed inputs. The **Instances** column shows the amount of completed inputs.

Description of simulacian control panel while simulating

Related Resources

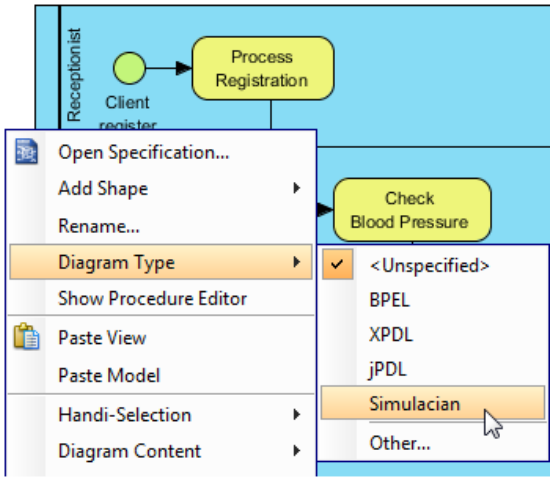
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Simulating business process

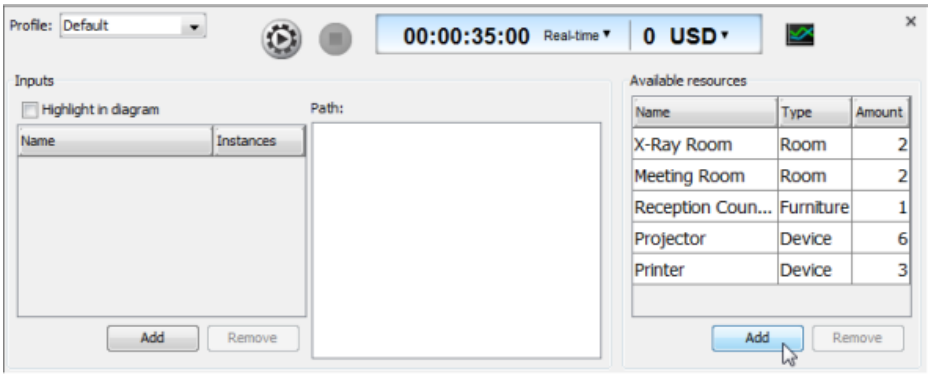
In order to simulate a business process , you must provide performance-related information to the business process diagram , such as resource consumption and duration of flow objects, so that simulacian can analyze the information and conduct a simulation. Below are what you have to do to start simulation.

1. Right click on the background of the business process diagram that you want simulate and select **Diagram Type > Simulacian** from the pop-up menu.



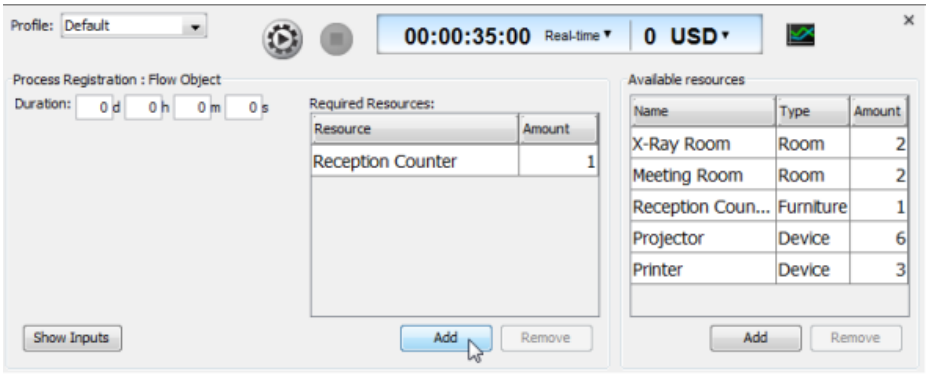
*Set diagram's type to **Simulacian***

2. In the **Simulacian Control Panel**, click **Add** under **Available resources** to define the resources that can be used by the flow objects in the business process diagram in order to complete the tasks. If you want to know more about resources, please refer to the chapter *What is simulacian?*.



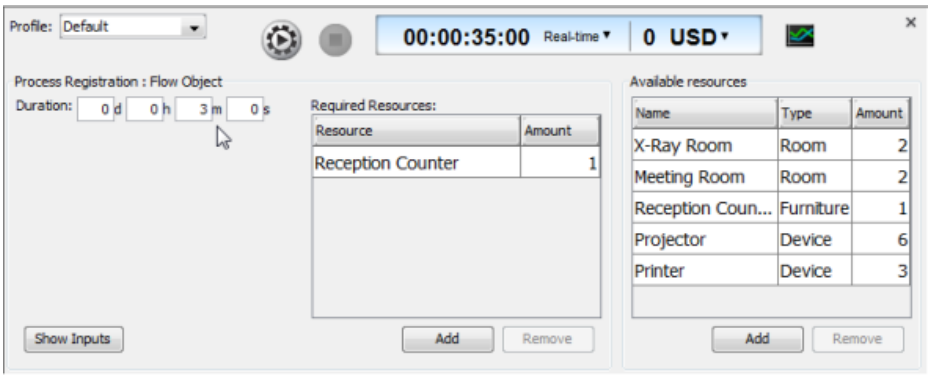
Add available resources

3. For each of the flow objects, select it in diagram, and click **Add** under **Required Resources** in **Simulacian Control Panel** to add the resource(s) the participant needed to complete the job when reaching the selected flow object.



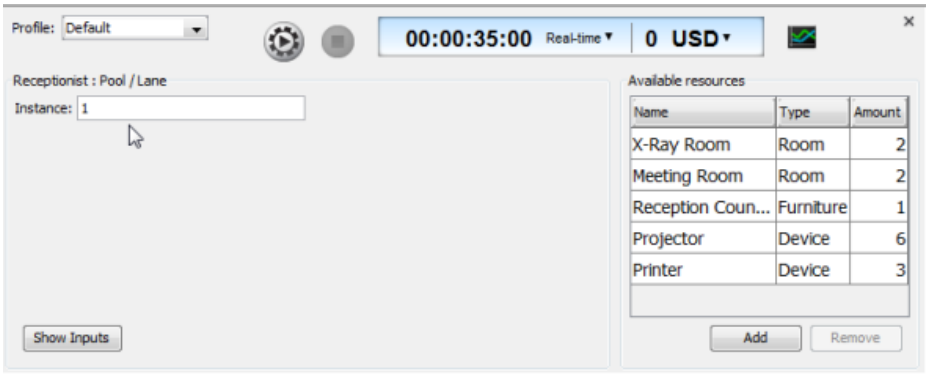
Add required resources

4. For each of the flow objects, select it in diagram, and set in **Simulacian Control Panel** its duration, which is the time it takes to get completed. If you want to know more about duration, please refer to the chapter *What is simulacian?*.



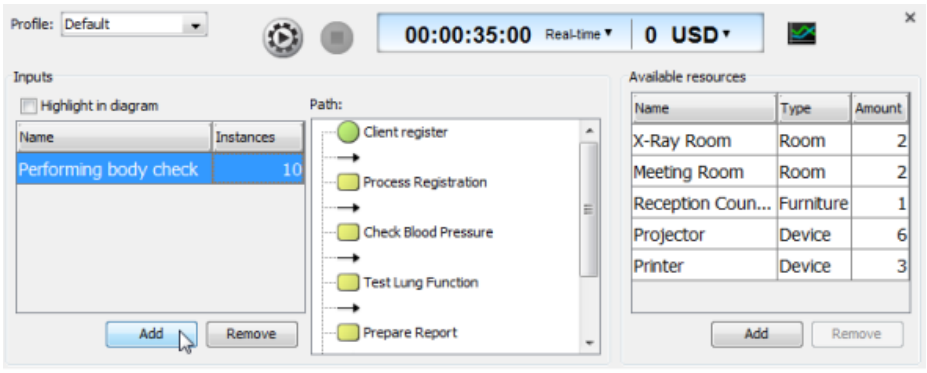
Set the duration of flow object

5. For each lane and pool (without lane), select it in diagram, and set its instance in **Simulacian Control Panel**, which represents the number of participants that take part in the process.



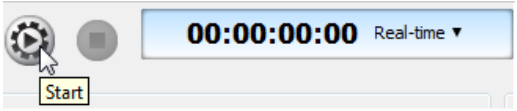
Set instance of pools or lanes

6. Click on the background of diagram or click **Show Inputs** in **Simulacian Control Panel**.
7. Click **Add** under Inputs in **Simulacian Control Panel** to add the paths to be executed during simulation. For each input, set the name that describes the path, and the number of instances for the number of copies of the path to execute.



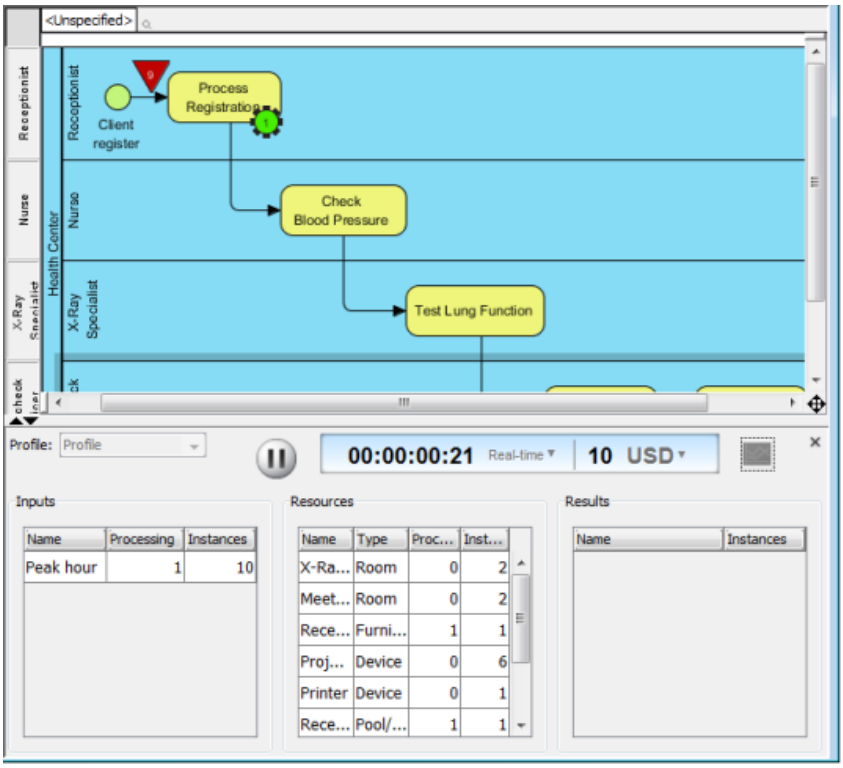
Add input

8. Click Start in **Simulacian Control Panel** to start simulation.



Start simulation

Now, you can watch the simulation in diagram, to see the executions of inputs and identify bottlenecks.



Simulation is running

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Simulacian charts

In addition to process simulation, you can produce charts to aid in the analysis of process performance. This chapter describes the six charts in details.

Completion

The completion chart primarily shows the status of inputs completion against time.

Resource usage

The resource usage chart shows the percentage of resource consumption throughout simulation.

Queue time

The queue time chart shows the time the flow objects spent on waiting, which corresponds to the time an inverted triangle appear during simulacian.

Cost per flow object

The cost per flow object chart shows the cost each flow object consumed throughout simulation.

Cost per input

The cost per input chart shows the cost each input consumed throughout simulation.

Time cost

The time cost chart shows the cost consumed against time spent throughout simulation.

Completion

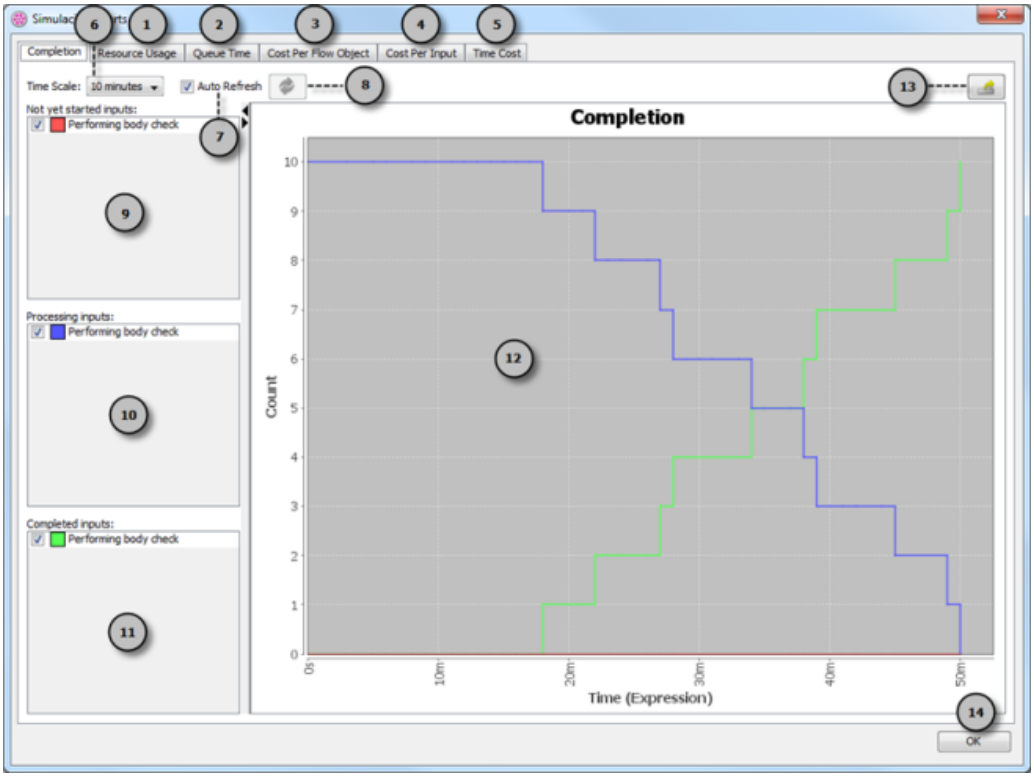
Sometimes, just by watching the simulacian outcome is not enough in finding out the bottleneck, especially when the diagram is large, and has many, many bottlenecks. In such cases, you can produce charts for simulacian outcome, which helps quantify resource consumption and queuing time for each flow object.

To read the charts, click **Simulacian Charts** in simulacian control panel.

Open simulacian charts window

Completion chart

The completion chart primarily shows the status of inputs completion against time.



An overview of completion chart

No.	Name	Description
1	Resource usage	Click to show resource usage chart.
2	Queue time	Click to show queue time chart.
3	Cost Per Flow Object	Click to show cost per flow object chart.
4	Cost Per Input	Click to show cost per input chart.
5	Time Cost	Click to show time cost chart.
6	Time scale	Control the length of chart by selecting a time scale.
7	Auto refresh	Check to let the chart body auto update against chart settings such as time scale and selection of inputs. Uncheck to update manually by clicking **Refresh**.
8	Refresh	Click to update the chart body against the chart settings such as time scale and selection of inputs. This button is available only when **Auto Refresh** is unchecked.
9	Not yet started inputs	Check or uncheck the inputs to show or hide in chart the change of the amount of not-yet-started inputs throughout simulation.
10	Processing inputs	Check or uncheck the inputs to show or hide in chart the change of the amount of processing inputs throughout simulation.
11	Completed inputs	Check or uncheck the inputs to show or hide in chart the change of the amount of completed inputs throughout simulation.

12	Chart body	The chart body.
13	Export	Click to export the opening chart to Microsoft Excel or image file.
14	OK	Click to close the simulacian charts window and go back to the diagram.

Description of completion chart

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Resource usage

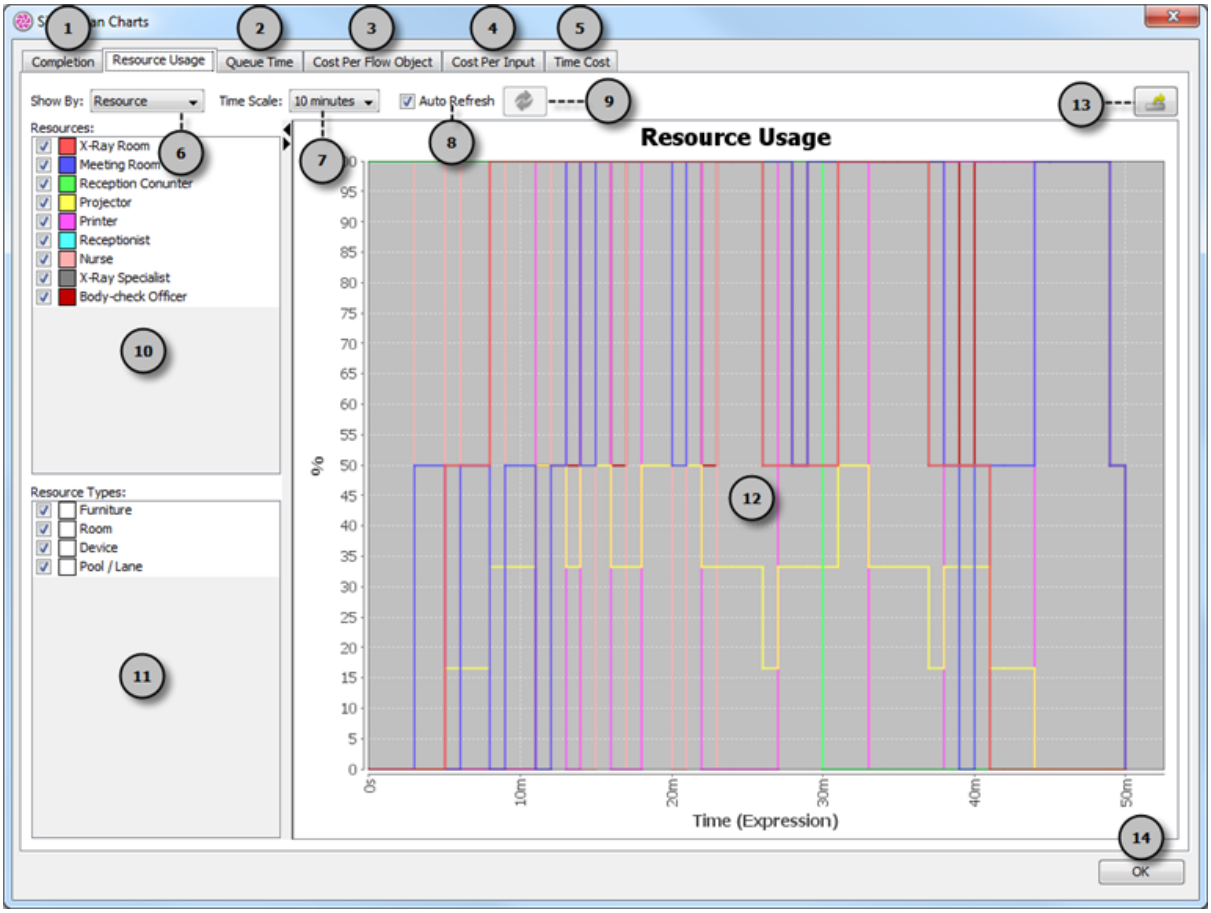
Sometimes, just by watching the simulacian outcome is not enough in finding out the bottleneck, especially when the diagram is large, and has many, many bottlenecks. In such cases, you can produce charts for simulacian outcome, which helps quantify resource consumption and queuing time for each flow object.

To read the charts, click **Simulacian Charts** in simulacian control panel.

Open simulacian charts window

Resource usage chart

The resource usage chart shows the percentage of resource consumption throughout simulation. If a resource has its peak reaching 100%, it means that the allocation of that resource is in optimum state. If the peak is below 100%, it means that some of the resources are not used throughout the simulation, which usually signifies that they are wasted, and you should consider adjusting the amount of available resource to optimize the resource consumption.



An overview of resource usage chart

No.	Name	Description
1	Complete	Click to show completion chart.
2	Queue time	Click to show queue time chart.
3	Cost Per Flow Object	Click to show cost per flow object chart.
4	Cost Per Input	Click to show cost per input chart.
5	Time Cost	Click to show time cost chart.
6	Show by	Select what to present in the chart. Resource - Cause the chart to presents the resource consumption of each available resource. Resource Type - Cause the chart to presents the resource consumption of resource types.
7	Time scale	Control the length of chart by selecting a time scale.

8	Auto refresh	Check to let the chart body auto update against chart settings such as whether to show by resource or resource type, time scale and selection of resources and resources type. Uncheck to update manually by clicking **Refresh**.
9	Refresh	Click to update the chart body against the chart settings such as whether to show by resource or resource type, time scale and selection of resources and resources type. This button is available only when **Auto Refresh** is unchecked.
10	Resources	Check or uncheck the resources to show or hide their usage in chart. This pane is active only when **Resource** is selected in the drop down menu **Show By**.
11	Resource types	Check or uncheck the resource types to show or hide their usage in chart. This pane is active only when **Resource Type** is selected in the drop down menu **Show By**.
12	Chart body	The chart body.
13	Export	Click to export the opening chart to Microsoft Excel or image file.
14	OK	Click to close the simulacian charts window and go back to the diagram.

Description of resource usage chart

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Queue time

Sometimes, just by watching the simulacian outcome is not enough in finding out the bottleneck, especially when the diagram is large, and has many, many bottlenecks. In such cases, you can produce charts for simulacian outcome, which helps quantify resource consumption and queuing time for each flow object.

To read the charts, click **Simulacian Charts** in simulacian control panel.

Open simulacian charts window

Queue time chart

The queue time chart shows the time the flow objects spent on waiting, which corresponds to the time an inverted triangle appear during simulacian. You may study whether the queue time of certain flow object is reasonable or not, and think of the improvement.

An overview of queue time chart

No.	Name	Description
1	Completion	Click to show completion chart.
2	Resource usage	Click to show resource usage chart.
3	Cost Per Flow Object	Click to show cost per flow object chart.
4	Cost Per Input	Click to show cost per input chart.
5	Time Cost	Click to show time cost chart.
6	Time scale	Control the length of chart by selecting a time scale.
7	Auto refresh	Check to let the chart body auto update against chart settings such as time scale and selection of nodes. Uncheck to update manually by clicking **Refresh**.
8	Refresh	Click to update the chart body against the chart settings such as time scale and selection of nodes. This button is available only when **Auto Refresh** is unchecked.

9	Nodes	Check or uncheck flow objects nodes to show or hide their queue time in chart.
10	Chart body	The chart body.
11	Export	Click to export the opening chart to Microsoft Excel or image file.
12	OK	Click to close the simulacian charts window and go back to the diagram.

Description of queue time chart

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Cost per flow object

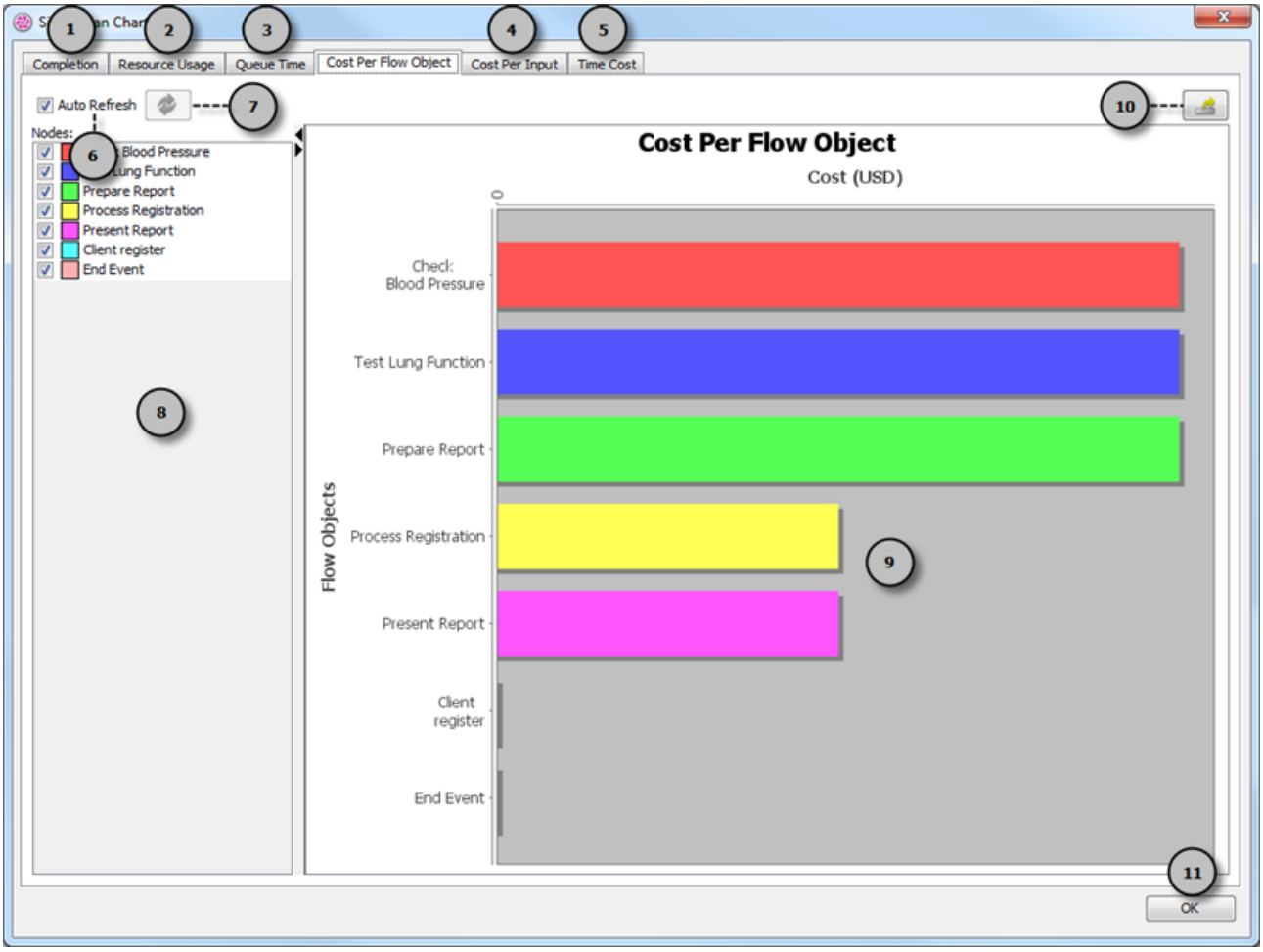
Sometimes, just by watching the simulacian outcome is not enough in finding out the bottleneck, especially when the diagram is large, and has many, many bottlenecks. In such cases, you can produce charts for simulacian outcome, which helps quantify resource consumption and queuing time for each flow object.

To read the charts, click **Simulacian Charts** in simulacian control panel.

Open simulacian charts window

Cost per flow object chart

The cost per flow object chart shows the cost each flow object consumed throughout simulation.



An overview of cost per flow object chart

No.	Name	Description
1	Completion	Click to show completion chart.
2	Resource Usage	Click to show resource usage chart.
3	Queue Time	Click to show queen time chart.
4	Cost Per Input	Click to show cost per input chart.
5	Time Cost	Click to show time cost chart.
6	Auto refresh	Check to let the chart body auto update against chart settings such as time scale and selection of inputs. Uncheck to update manually by clicking **Refresh**.
7	Refresh	Click to update the chart body against the chart settings such as time scale and selection of inputs. This button is available only when **Auto Refresh** is unchecked.
8	Nodes	Check or uncheck flow objects nodes to show or hide their cost per flow object in chart.
9	Chart body	The chart body.

10	Export	Click to export the opening chart to Microsoft Excel or image file.
11	OK	Click to close the simulacian charts window and go back to the diagram.

Description of cost per flow object chart

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Cost per input

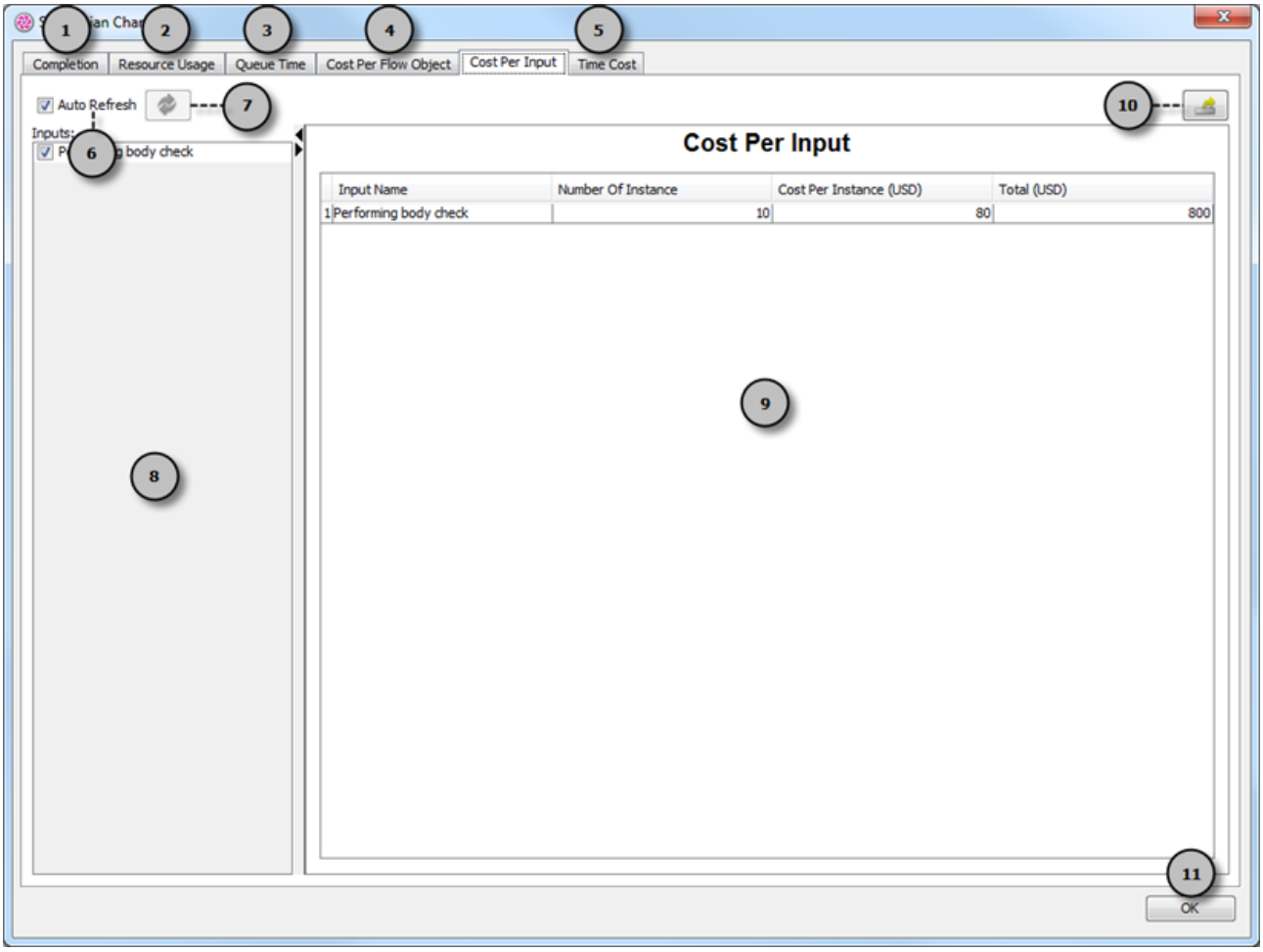
Sometimes, just by watching the simulacian outcome is not enough in finding out the bottleneck, especially when the diagram is large, and has many, many bottlenecks. In such cases, you can produce charts for simulacian outcome, which helps quantify resource consumption and queuing time for each flow object.

To read the charts, click **Simulacian Charts** in simulacian control panel.

Open simulacian charts window

Cost per input chart

The cost per input chart shows the cost each input consumed throughout simulation.



An overview of cost per input chart

No.	Name	Description
1	Completion	Click to show completion chart.
2	Resource Usage	Click to show resource usage chart.
3	Queue Time	Click to show queen time chart.
4	Cost Per Flow Object	Click to show cost per flow object chart.
5	Time Cost	Click to show time cost chart.
6	Auto refresh	Check to let the chart body auto update against chart settings such as time scale and selection of inputs. Uncheck to update manually by clicking **Refresh**.
7	Refresh	Click to update the chart body against the chart settings such as time scale and selection of inputs. This button is available only when **Auto Refresh** is unchecked.
8	Nodes	Check or uncheck flow objects nodes to show or hide their cost per input in chart.
9	Chart body	The chart body.

10	Export	Click to export the opening chart to Microsoft Excel or image file.
11	OK	Click to close the simulacian charts window and go back to the diagram.

Description of cost per flow object chart

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Time cost

Sometimes, just by watching the simulacian outcome is not enough in finding out the bottleneck, especially when the diagram is large, and has many, many bottlenecks. In such cases, you can produce charts for simulacian outcome, which helps quantify resource consumption and queuing time for each flow object.

To read the charts, click **Simulacian Charts** in simulacian control panel.

Open simulacian charts window

Time cost chart

The time cost chart shows the cost consumed against time spent throughout simulation.

An overview of time cost chart

No.	Name	Description
1	Completion	Click to show completion chart.
2	Resource Usage	Click to show resource usage chart.
3	Queue Time	Click to show queue time chart.
4	Cost Per Flow Object	Click to show cost per flow object chart.
5	Cost Per Input	Click to show cost per input chart.
6	Time Scale	Control the length of chart by selecting a time scale.
7	Auto refresh	Check to let the chart body auto update against chart settings such as time scale and selection of inputs. Uncheck to update manually by clicking **Refresh**.
8	Refresh	Click to update the chart body against the chart settings such as time scale and selection of inputs. This button is available only when **Auto Refresh** is unchecked.
9	Nodes	Check or uncheck flow objects nodes to show or hide their cost per input in chart.

10	Chart body	The chart body.
11	Export	Click to export the opening chart to Microsoft Excel or image file.
12	OK	Click to close the simulacian charts window and go back to the diagram.

Description of cost per flow object chart

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Entity relationship diagram

ERD, short for entity relationship diagram, is a kind of diagram for presenting the properties as well as the relationships between data or participants. This chapters teaches you how to create an ERD.

Creating entity relationship diagram

This page teaches you how to create an ERD through the diagram navigator, as well as to describe the differences between conceptual, logical and physical model.

Database configuration

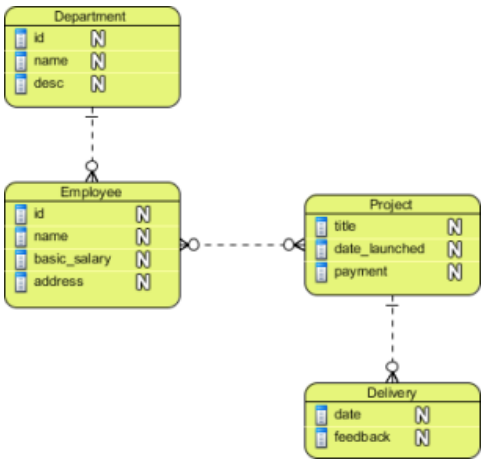
This page teaches you how specify the type of database (e.g. Oracle, MS SQL, MySQL...) ERD to conform to.

Using Auto Column

This page teaches you how to draw ERD quicily with the Auto Column feature.

Creating entity relationship diagram

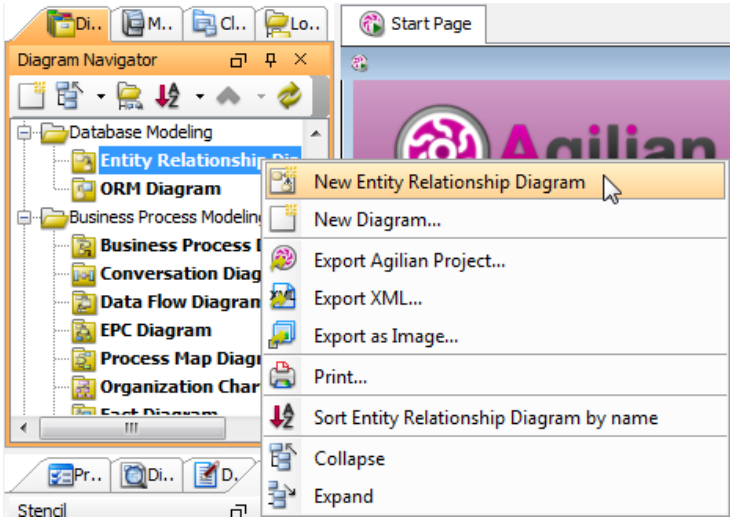
ERD , short for entity relationship diagram, is a kind of diagram for presenting the properties as well as the relationships between data or participants. Database designer use of ERD to model physical structure of a relationship database, while business analyst uses ERD to model the data that is logically required or produced by processes.



A sample ERD

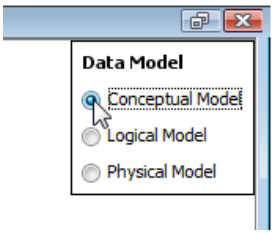
Creating entity relationship diagram

To create an entity relationship diagram right click on **Entity Relationship Diagram** in **Diagram Navigator** and select **New Entity Relationship Diagram** from the popup menu.



*Create an ERD through **Diagram Navigator***

When a new diagram is created, you can name it at the top left corner of diagram. At the top right of diagram you can select a type of model of which the diagram will present. The meaning of model types will be described in the next section.



Selecting a type of model

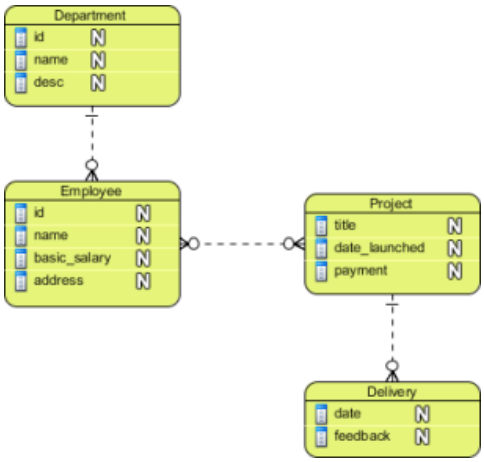
Creating entity and adding columns

To create an entity, select **Entity** in diagram toolbar and click on diagram. To add a column in entity, right click on the entity and select **New Column** from the popup menu.

Conceptual, logical and physical model

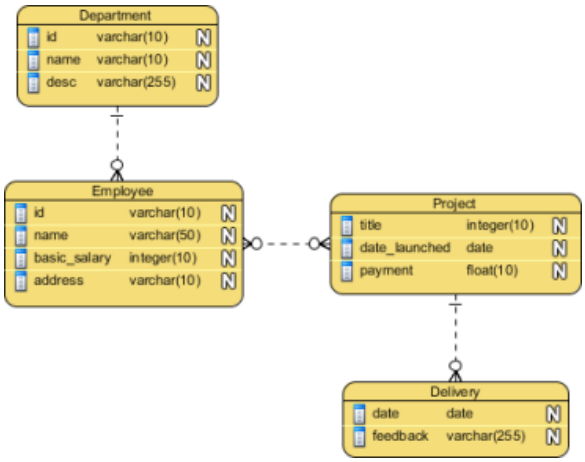
Conceptual, logical and physical model are three different ways of modeling data in a domain. The all contain entities and relationships, but differs in complexity and purposes. A general understanding to the three models is that, business analyst use conceptual and logical model for modeling the data required and produced by system from a business angle, while database designer refine the early design to produce the physical model for presenting physical database structure ready for database construction.

Conceptual model is the simplest model among all. It consists of entities and relationships. You can add columns to entities to represent the properties of entities.



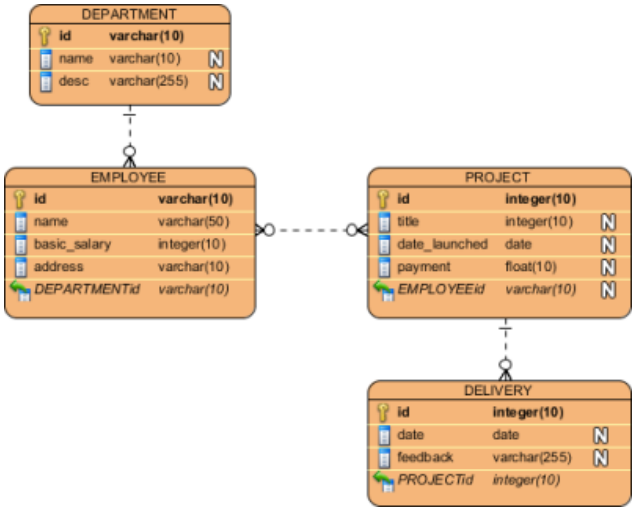
A conceptual model

Logical model is more complex than conceptual model. You can include more details for columns by specifying their type (e.g. text, integer, date).



A logical model

Physical model is the most complex one. Database designers add primary or foreign keys to entities, and rename entities and columns to satisfy the table or column naming convention. The physical model represents how to implement your data model in database.

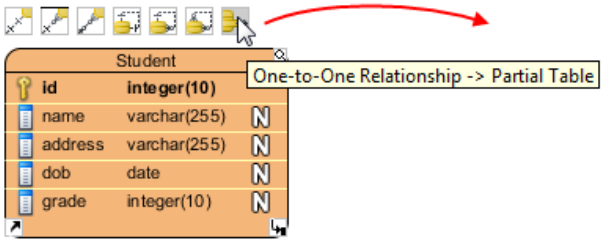


A physical model

Creating partial table

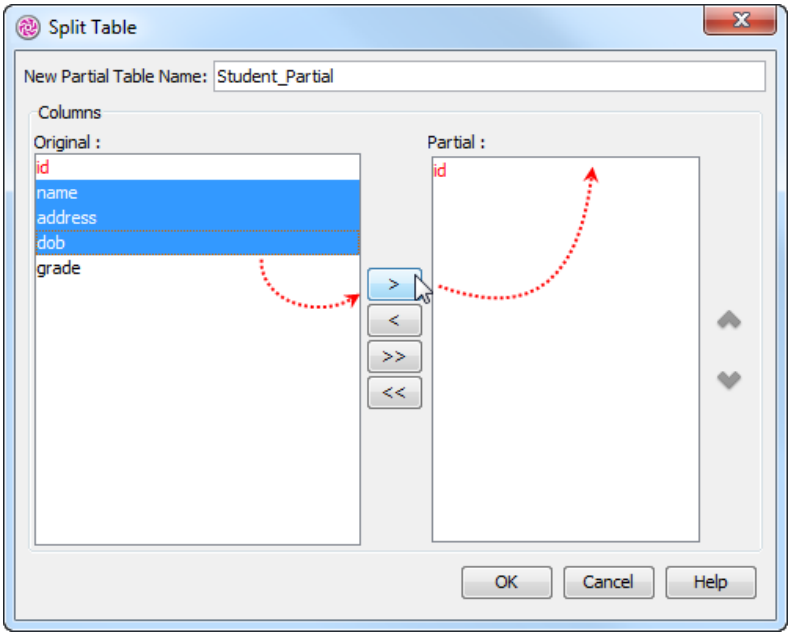
Sometimes, you want to separate an entity into two, and create a one-to-one relationship in between. For example, to extract information on a Student entity into another entity StudentInfo. You can split a table by creating a partial table through the resource centric-interface.

1. To split a table, move the mouse pointer over the entity you want to split and drag out the **One-to-One Relationship -> Partial Table**.



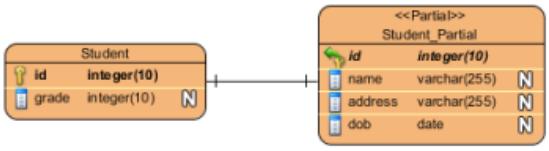
Create a partial table

2. Release the mouse button to create the split entity.
3. In the **Split Table** dialog box, select the columns you want to move to the split table and click **>**.



Move columns to partial table

4. Click **OK** button. A split table is created.



Split table is created

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

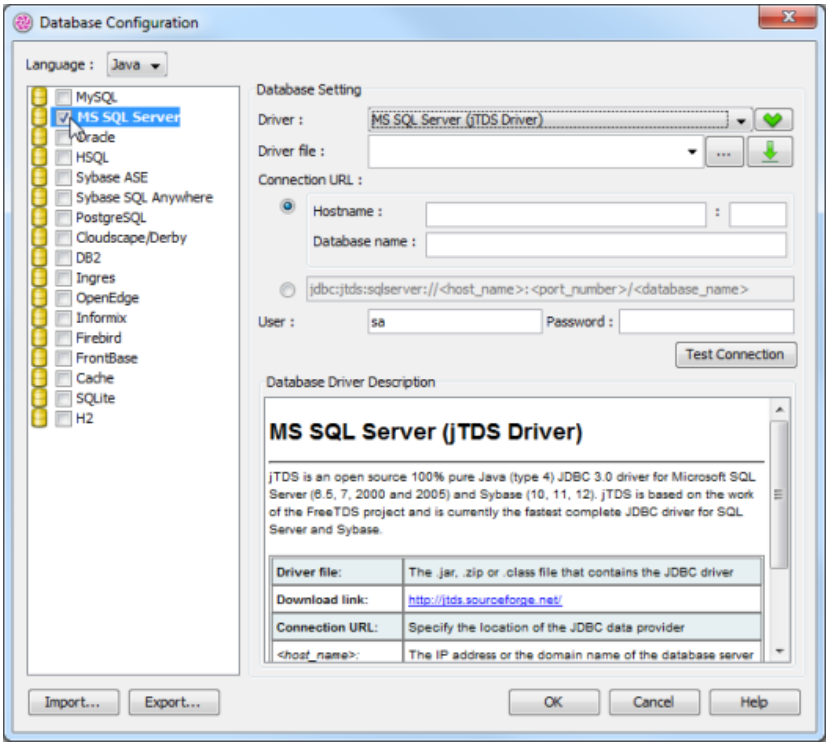
Database configuration

Opening database configuration dialog

Select **Tools > Database > Database Configuration...** from the main menu.

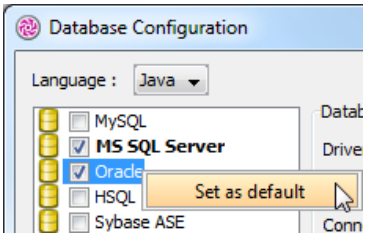
Selecting database

In the **Database Configuration** dialog box, check the target database.



Select database

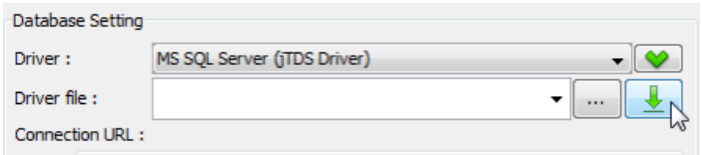
You can select multiple databases, and set one of them as default database. The default database is used for rendering the column type and generating SQL. To set a database as default, right click on the database and select **Set as default** from the pop-up menu.



Set as default database

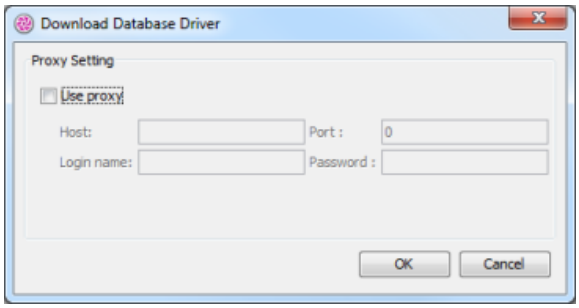
Downloading JDBC driver

If the JDBC driver is free and available to public, Agilian can help you to download it automatically. Click the **Download or Update** button under **Database Setting** section.



Download JDBC driver

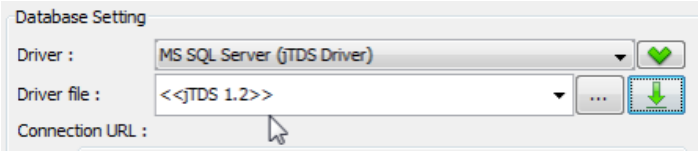
Configure proxy if required, click **OK** button to continue download.



Download JDBC proxy setting

The **Download** dialog shows the URL, file size, speed and progress information. Click **Close** button after download is completed.

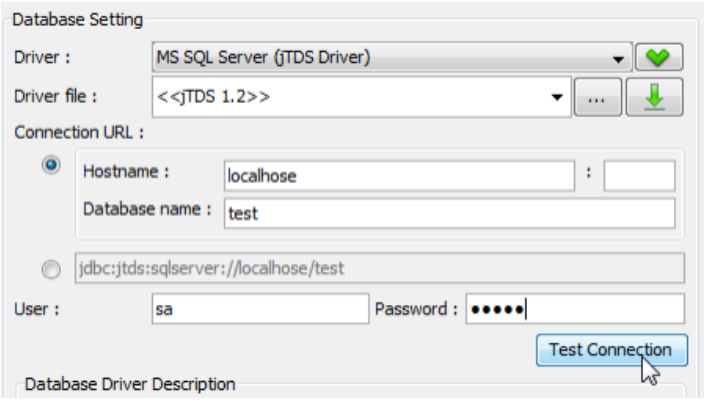
The **Driver file** field shows the driver downloaded by Agilian.



Downloaded JDBC driver

Testing connection

After configure the database setting, click the **Test Connection** button to confirm the setting is valid.



Test connection

If it's successful, a dialog will show connection successful.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

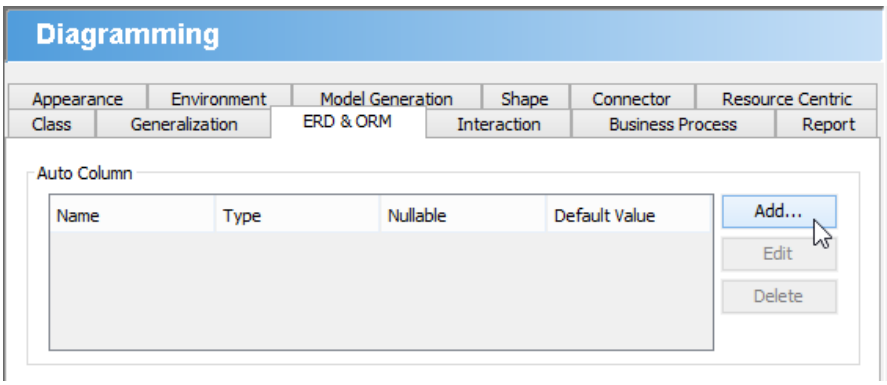
Using Auto Column

Auto Column provides you with a quick way in creating entities in ERD. Just by entering the name of a column when you create it, its type, nullable and default value will be specified, based on the pre-defined column schema.

Configuring Auto Column

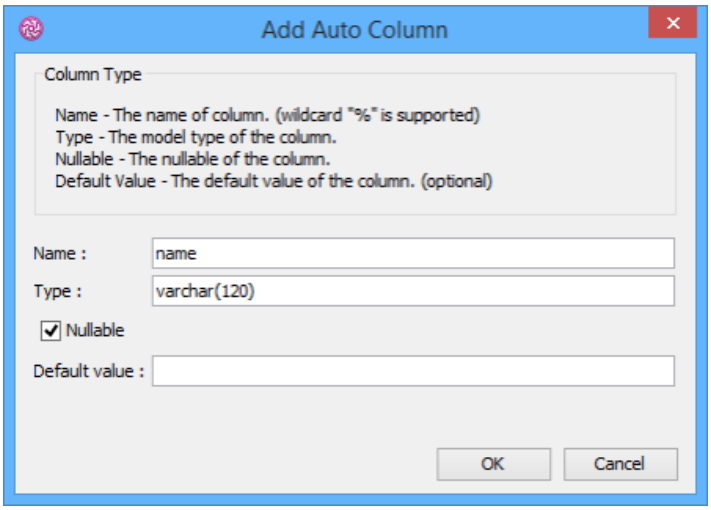
You need to configure the auto columns in order to use them in ERD. To configure auto columns:

1. Select **Tools > Application Options** from the main menu.
2. In the **Application Options** window, open **Diagramming > ERD & ORM**.
3. Click **Add...**.



Add an auto column

4. Define the auto column by entering its name, type, nullable and default value. When you create columns by using the name specified, the type, nullable and default value will follow the definition here. Note that you can specify the length of datatype when specifying **Type**.



Defining an auto column

5. Click **OK** to confirm.
6. Click **OK** again to close the **Application Options** window.

Pattern matching using wildcard

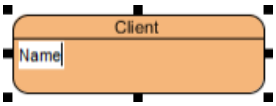
Sometimes, columns are named based on the entities they are in. For example, for entity *Teacher* there may be a column named *TeacherID* and for entity *Student* there may be another column named *StudentID*. Although these columns are named differently, they may share the same type, nullable and default values. In such case you can make use of pattern matching in defining an auto column.

When naming an auto column, you can use the '**%**' character to indicate the occurrence of ANY character. For example, naming an auto column *%Name* will result in causing *TeacherName* and *StudentName* apply the definition specified.

Using pattern matching in defining an auto column

Adding a column to entity with Auto Column

When you create a column in entity, give it a name that follows an Auto Column defined. When you confirm editing, the type, nullable and default values will be automatically set for you. Note that if you enter a type yourself, Auto Column will NOT function.



Naming a column

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Object relational mapping diagram

ORM diagram can be used to show the mapping between the ORM-Persistable class and its corresponding entity (table).

Creating object relational mapping diagram

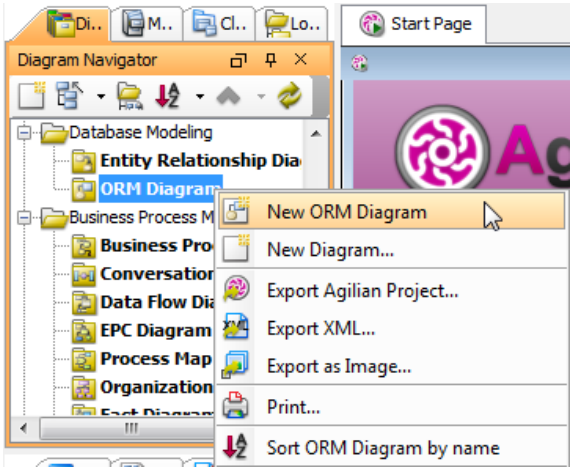
This page teaches you how to create an ORM diagram, how to map between entity and class, and how to synchronize between them.

Creating object relational mapping diagram

When you need to develop applications that involve storing data in database, you need to take care of the communication between the application and the database. Some people prefer to communicate with database through JDBC, while some prefer object relational mapping (ORM). ORM diagram can be used to show the mapping between the ORM-Persistable class and its corresponding entity (table). With ORM diagram, mappings between classes and entities, and between attributes and columns can be clearly identified.

Creating object relational mapping diagram

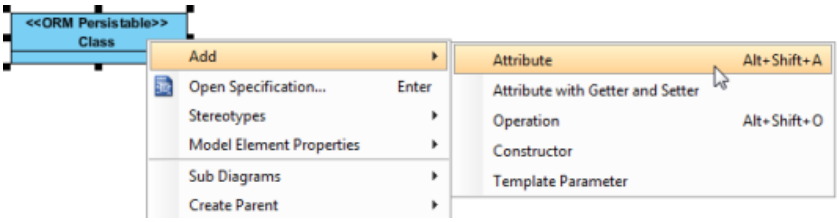
To create an object relational mapping diagram right click on **ORM Diagram** in **Diagram Navigator** and select **New ORM Diagram** from the pop-up menu.



*Create an ORM diagram through **Diagram Navigator***

Creating ORM-Persistable class and adding attributes

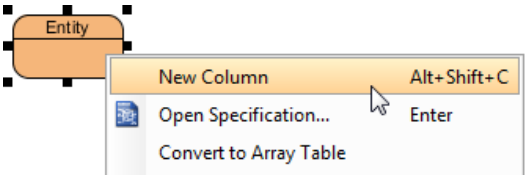
To create an entity, select **ORM-Persistable Class** in diagram toolbar and click on diagram. To add an attribute in class, right click on the class and select **New Attribute** from the popup menu.



To create an attribute in class

Creating entity and adding columns

To create an entity, select **Entity** in diagram toolbar and click on diagram. To add a column in entity, right click on the entity and select **New Column** from the popup menu.



To create column in entity

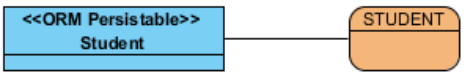
Mapping class and entity

To map between a class and an entity:

1. Move the mouse pointer over the class (or the entity). Press on the resource icon **Class-Entity Mapping -> Entity** (**Class-Entity Mapping -> Class** for entity).

To map from class to entity

2. Drag to the entity to map and release the mouse button. A connector will be created between the class and entity, indicating a mapping relationship.

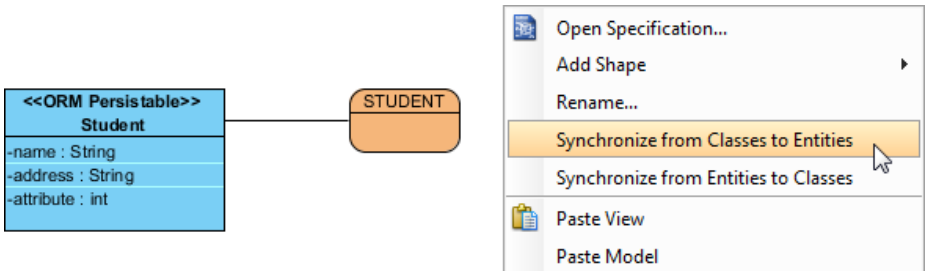


Class and entity is mapped

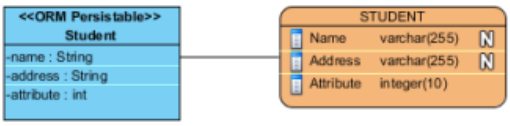
NOTE: One class can be mapped with at most one entity. Once a mapping exist, you will not see the resource icon for creating mapping anymore.

Synchronizing from classes to entities

When you have made change on a class say to add attributes or to rename class, you can synchronize the change from class to entity. To do this, right click on the background of diagram and select **Synchronize from Classes to Entities** from the popup menu.



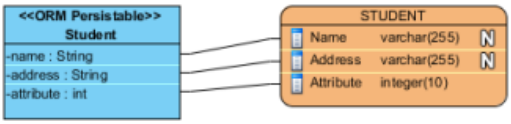
To synchronize changes from class to entity



Columns are created in entity resulted by synchronization

Changing the view of mapping

Instead of viewing the mapping in class and entity level, you can view the mapping in attribute and column level, too. To change the view, right click on the background of diagram and select **Presentation Options > Mapping View**, then **Class Mapping** or **Attribute Mapping** from the popup menu.



Attribute mapping is applied

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

UeXceler

What is UeXceler?

Learn what UeXceler is.

Why UeXceler?

Know the benefits of following UeXceler in requirements gathering.

Overview of UeXceler - The Five Phases

Familiar yourself with the 5 phases of UeXceler.

Responsibilities of UeXceler Roles

Check out the roles involved in UeXceler.

What is UeXceler?

One of the keys to a successful software project is to build the system according to users’ needs, but unfortunately, many software companies fail to gather the right requirements from stakeholders, mostly due to stakeholders fail to express their needs and concerns clearly and thoroughly.

While it is easier for stakeholders to tell you what they need when they can experience a fully functional system or at least, prototypes, it is still time consuming in developing any product or semi-product. Consequently, most of the development teams end up with two extremes: gain real customer feedback at the end of project, or spent a lot of time and effort at the beginning of project until any real feedback is gathered. UeXceler strikes a balance between the two extremes by letting user see and feel the final product early in requirements gathering phases, but without spending too much resources of the development team.

UeXceler is a guideline for software development teams to identify real system requirements at the very beginning of development projects. UeXceler emphasizes the involvement and contribution of stakeholders. Throughout the requirement gathering process, stakeholders are encouraged to express their concerns and needs, which are then converted into use cases, user scenarios and requirements of the solution. Wireframes are created for the illustration of the end product in quick and cost-effective manner. At the end of the requirements gathering process, stakeholders not just see a list of requirements but a set of business stories (scenarios) and a visualized system through wireframes. Both stakeholders and development teams can comfortably consent to the planned requirements specification.

UeXceler stresses the importance of collaboration with customer rather than any kind of negotiation. By adopting UeXceler in requirements gathering, stakeholders can easily preview what the system will look like without any functional features or prototype developed. Software development lifecycle is highly optimized by having clear direction, resulted by clear requirements. Solid partnership can be built between business stakeholders and technology team, which facilitates the accurate identification of system requirements and resulting in building software system that meets or even excels customer’s expectation.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - Requirements Gathering with UeXceler
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Why UeXceler?

As a contemporary way for requirement gathering, UeXceler addresses key challenges of historical ways of identifying system requirements. UeXceler focuses on identifying stakeholders and users' real needs and the result is both a better software product and a higher customer satisfaction. The following lists out the benefits that different roles of people, both in the stakeholders and people in project team, can gain by adapting UeXceler.

To Stakeholder and User

- UeXceler values the expertise and opinions contributed by stakeholders. Instead of guessing what the stakeholders needs, stakeholders are encouraged to share their knowledge, concerns and suggestions, which are then converted into requirements of system. Because stakeholders do have say in the project, they feel empowered, respected and trusted. Customer satisfaction is, as a result, significantly increased.
- Because the solution is mostly made from stakeholders' input, it is more likely for them to buy into the final solution.
- Stakeholders are presented both the system interactions and wireframe early in the requirement gathering phases. They know how the end product will look like in early, and feel more comfortable.
- With the early inspection of product design, modification and new ideas can be added into the project without significant cost.
- Because the project team develops feature according to the priority agreed with stakeholders, stakeholders can be more confident that the business values and business priorities are aligned with development activities. Critical features can therefore be done, tested and even used in early iterations.

To Project Team

- UeXceler values the participation of stakeholders in developing a good piece of system. The active involvement of stakeholders allows for clear project direction throughout the product development, and avoids spending valuable time on re-work due to any late feedback.
- Development activities are determined based on the use cases identified. By evaluating the nature of use cases, parallel development can be planned and conducted. This saves a great deal of time on unnecessary waiting.
- The ability to prioritize use case and requirements allows for delivering highest priority features earlier, avoiding risk resulted by late change.
- The seamless connection between flow of events and wireframe provides an effective way for stakeholder to feel the user experience without any real product get done - not even any prototype.
- Save time by avoiding all sorts of non-productive work - no unnecessary documentation, no ineffective meetings, no complex screen design and no prototyping.
- Use case based stakeholders meeting makes the meeting fewer, shorter and more focused.
- Definition of success is very clear at the beginning of development – you have the flow of events, requirements, wireframes as targets.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - Requirements Gathering with UeXceler
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Overview of UeXceler - The Five Phases

Identify

The purpose of the Identify phase is to outline the objectives that the client want to achieve by using the end product, and these objectives, in UeXceler, are called use cases. Use cases provide insight into the overall needs that must be addressed by the system from end-user perspective. By the end of this phase, you should have a prioritized list of use cases, a list of actors (i.e. users) who will interact with the system and a list of front-line stakeholders who have subject matter knowledge to contribute for each use case.

Discuss

The purpose of the discuss phase is to gain detailed understanding of use case-related ideas such as the preferred way of accomplishing the use case, related concerns and specific requirements. Through the discussions with stakeholders, basic system flows and user concerns are captured and are drafted as simple use case notes, which may be converted into use case flow of events and system requirements in the Elaborate phase.

Elaborate

The purpose of the Elaborate phase is to convert the project-related ideas gathered in the Discuss phase into description of product behaviors and a consolidated list of system requirements. Standard flow of events scenarios are created base on the use case notes gathered, to describe the way how user can interact with the system to achieve the use cases. Concrete requirements are also created from some of the use case notes, which are likely from the concerns raised by the stakeholders.

Design

The purpose of the Design phase is to produce wireframe of the system going to build, for visualizing user interface and representing screen flows. Wireframing is a technique to illustrate and modify the user interaction experience in very quick manner without big investment. In UeXceler, wireframes are suggested to be created in accordance with use case flow of events to visualize the user experience at particular states throughout the system interactions.

Consent

The purpose of the Consent phase is to seek approval from stakeholders prior to initiating any actual development activity. Stakeholder is shown both the wireframe flow (like a screen flow) so that he/she can verify easily the behaviors of the system, and to request modifications, if necessary.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - Requirements Gathering with UeXceler
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Responsibilities of UeXceler Roles

Here are the most common roles that appear in the UeXceler gudeline. Notice that different teams and companies may have different titles for those roles (or no name at all). And sometimes, a team member has to play several roles, due to the scale of project team, the type of project, the expertise and experience of that member, etc.

System analyst

A system analyst (sometimes simply referred to as analyst) is responsible for identifying project stakeholders, identifying actors and their use cases, documenting use case flow of events, managing the gathering and prioritizing of system requirements and seeking the approval of system design from stakeholders. System analyst has to work hand-in-hand with the business stakeholders to complete these tasks.

In some development teams or projects, the role system analyst may not exist, and the developers can take the role of system analyst.

Business stakeholder

A business stakeholder is someone who has the business vision and business authority in defining “what” the end product need to accomplish, that is, the use cases. Business stakeholder works with system analyst in identifying use cases, noting business problems and concerns.

Business stakeholder may or may not be involved in day-to-day operations, but is knowledgeable enough to know the current state of the business and the problems that need to be addressed.

Front-line stakeholder

A front-line stakeholder is anyone who have a stake in the success of the project or have subject matter knowledge to contribute. He or she isn’t necessarily be the user of the system to develop, but must be capable of delivering project-related ideas that allows for the development of system that can achieve the end users' expectations.

A front-line stakeholder is responsible for defining preferred the way to operate with the system. When use cases are identified, different groups of front-line stakeholders will be interviewed. Ideas are gathered from the front-line stakeholders as use case notes, and those notes will become flow of events and concrete requirements of use cases.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - Requirements Gathering with UeXceler
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Actor List

Record and manage actors in Actor List

Learn how to record actors in Actor List.

Record and manage user goals of an actor

Learn how to record the user goals for actors created in Actor List.

Derive use case from user goal

Learn how to derive use cases from user goals.

Record and manage actors in Actor List

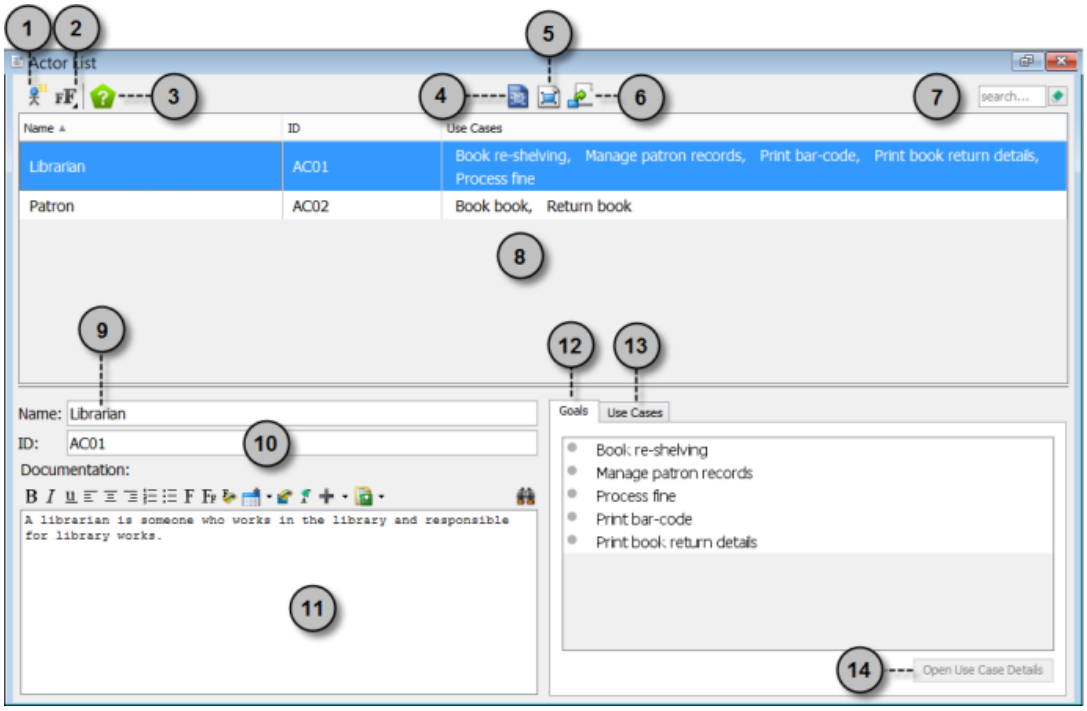
Actors are entities that interact with a system. They can be people (e.g. Teacher, Student, User, Administrator, etc), hardware, devices (e.g. printer) and external systems.

Actor List is a place where you can record, manage and organize the actors of your project.

Opening the Actor List

To open **Actor List**, select **Modeling > Actor List** from the main menu. If you see the panes like **Diagram Navigator** and **Property Pane** minimized, with help contents showing in the middle of screen, this means that this is the first time you open the **Actor List**. The panes are minimized to maximize the viewable list area. Click **Keep Change** if you want to keep the panes minimized. If you want to roll back to the original screen layout, click **Revert**. And if you keep the change now, you can still customize the screen layout by applying a perspective later on (**View > Open Perspective**).

Overview of Actor List



Actor List

No	Name	Description
1	New Actor	Click to create an actor.
2	Font Size	Click to adjust the font size of text in **Actor List**.
3	How to	Click to hide/show the help contents of **Actor List**.
4	Open Specification...	Select an actor in **Actor List** and click this button to open its specification.
5	Show View...	Select an actor in **Actor List** and click this button to list the diagrams that contains the view of the selected actor.
6	Visualize	Select an actor in **Actor List** and click this button to show it in a new or existing diagram.
7	Search	Find actor(s) by entering search criteria.
8	List of actors	Actors are listed here.
9	Actor name	Name of selected actor.
10	Actor ID	ID of selected actor. ID are automatically generated when you create actor. You may customize the pattern of ID in the **Project Options** window (**Tools > Project Options > Diagramming > Model Generation**).
11	Actor documentation editor	Documentation of selected actor. The tools above the editor enables you to enter documentation in rich text format.
12	User goals	User goals are the things that user want to do. Here you can enter the user goals of the selected actor.
13	Use cases	Use cases are the things that can be achieved by the system to be developed. By opening the Use cases tab you can see the use cases associated with the selected actor.

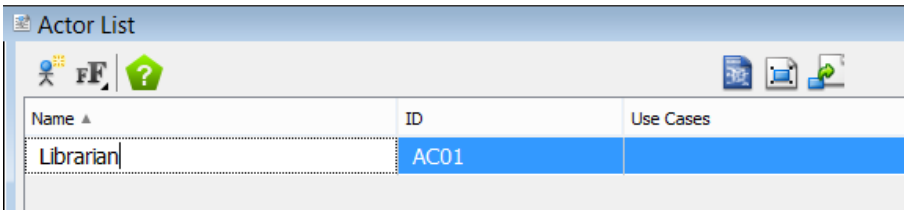
14	Derive/Open Use Case Details	You can derive a use case from user goal by first selecting a user goal and clicking **Derive Use Case**. Once a use case is derived, the button caption will be changed to " **Open Use Case Details**", which allows you the see and edit the details of the use case drived from the selected user goal.

Description of Actor List

Creating actor in Actor List

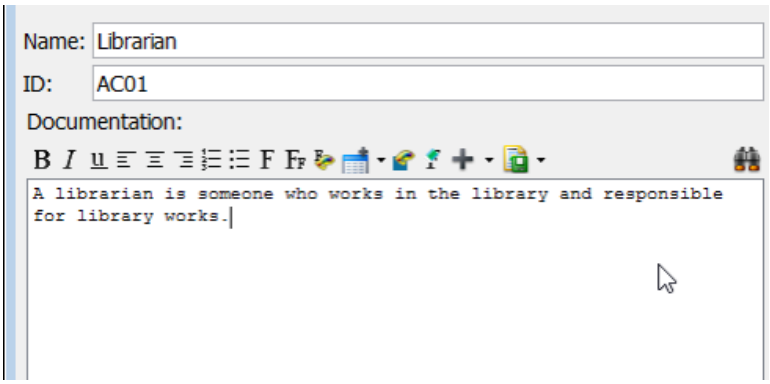
To create an actor in **Actor List**:

1. Click on **New Actor** above the **Actor List**.
2. Enter the name of actor.



Creating actor in Actor List

3. Press **Enter** to confirm editing.
4. You can optionally edit the ID and documentation of actor.



Edit actor documentation

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Actors with Actor List?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

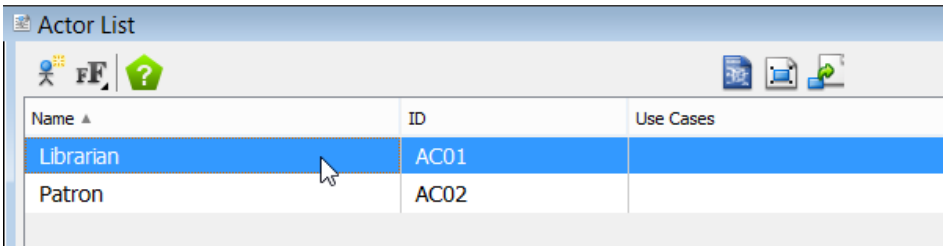
Record and manage user goals of an actor

A user goal is something that the end user want to achieve. Recording the user goals aids the analysis of problem and the creation of solution that can address the user's problems and concerns.

Recording user goals of an actor

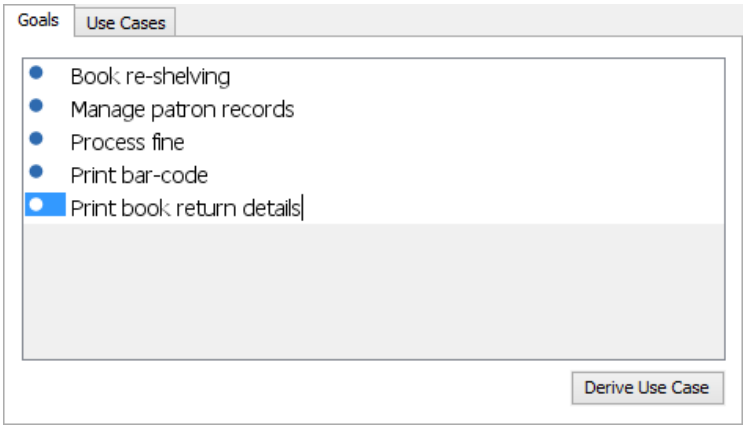
To record the user goals of an actor:

1. Open the **Actor List** by selecting **Modeling > Actor List** from the main menu.
2. Select the desired actor in the **Actor List**.



Select actor in Actor List

3. At the bottom right corner, in the **Goals** tab, enter the user goals. One goal per row. Press **Enter** when you want to add a new user goal.



Enter user goals of an actor

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Actors with Actor List?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

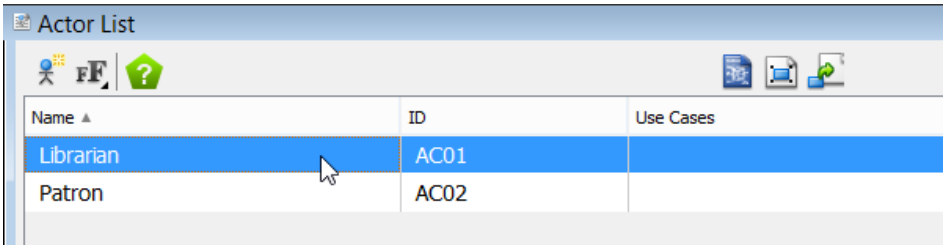
Derive use case from user goal

While a user goal is something that the actor want to achieve, a use case describes what a system must do to provide value to actor. You can derive a use case from a particular user goal when they are aligned, which means, when the user goal can be achieved by using the system to be developed. For user goals that cannot be satisfied, leave them as user goals without deriving use cases from them.

Deriving use case from user goal

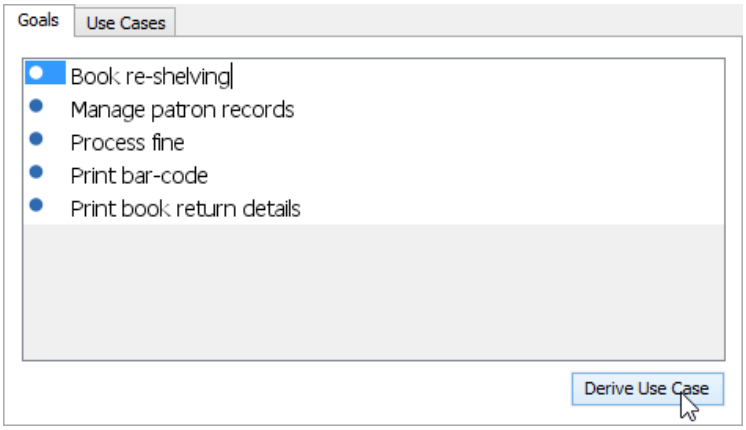
To derive a use case from a user goal:

1. Open the **Actor List** by selecting **Modeling > Actor List** from the main menu.
2. Select the desired actor in the **Actor List**.



Select actor in Actor List

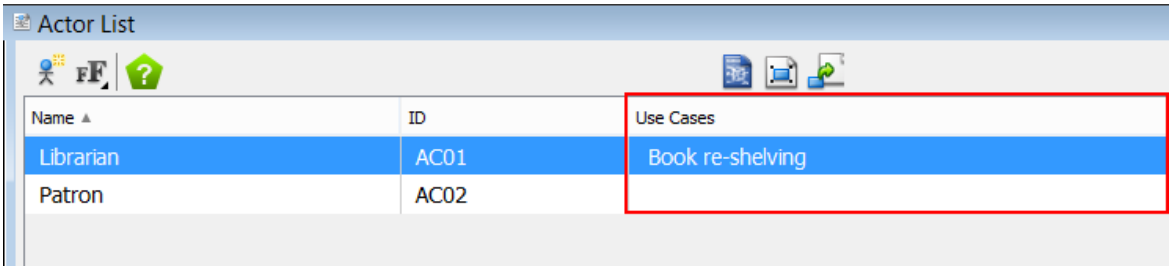
3. Click on the desired user goal in the **Goals** tab at the bottom right of the **Actor List**.
4. Click **Derive Use Case**.



Derive use case from user goal

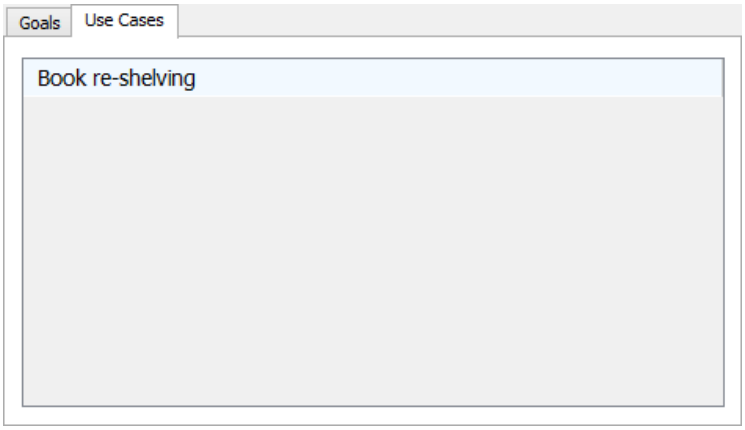
Viewing the derived use cases

You can find the use cases derived from user goals in two places. The first place is the **Use Cases** column in the **Actor List**.



The Use Cases column

Second, you can find the use cases in the Use Cases tab, next to the Goals tab.

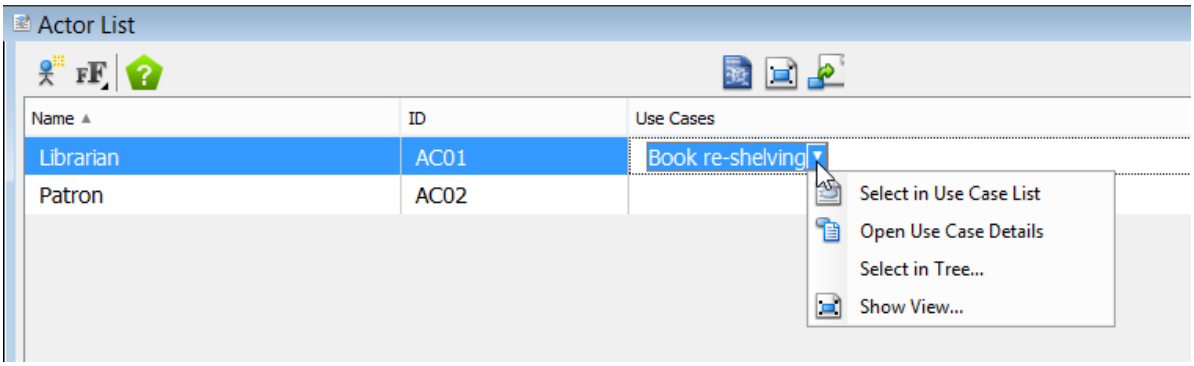


The Use Cases tab

Accessing a derived use case

Sometimes, you may want to rename a use case derived from a user goal to make it reflect the real content or objective of the use case. You can do so by accessing the use case.

By clicking on a use case in the **Use Cases** column of the **Actor List**, you can see a down arrow appear next to the use case.



Access a use case

The popup menu allows you to:

- **Select in Use Case List**: Open the **Use Case List** with that use case selected. The **Use Case List** is a place where all use cases are listed. You can view the selected use case and the other use cases in list form.
- **Open Use Case Details**: Open the **Use Case Details** editor where you can edit the details of a use case. You can rename the use case, edit/view its scenario, create requirements from a use case etc. You can also open the use case details by double clicking on a use case.
- **Select in Tree...**: Popup the **Model Explorer** with the use case highlighted.
- **Show View...**: Popup a new window that allows you to open the diagram that has the use case visualized in it.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Actors with Actor List?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Requirement List

Record and document software requirements in Requirement List

Learn how to record and document software requirements using Requirement List

Record and document software requirements in Requirement List

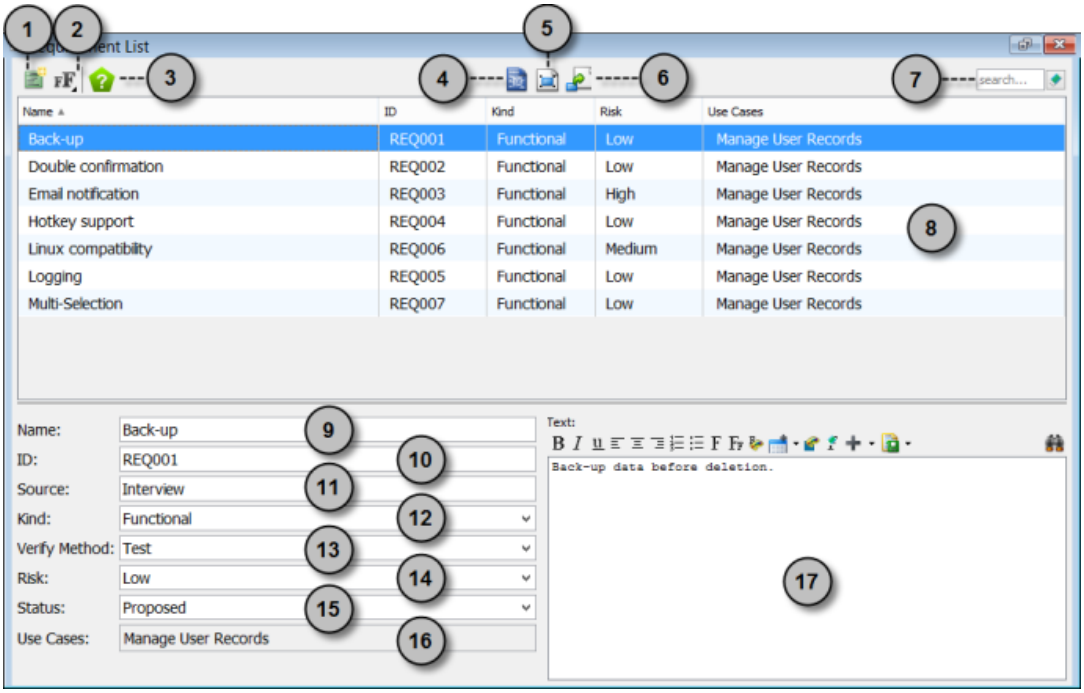
Every use case can be achieved by implementing a set of relevant requirements. Requirement states what the system needs to deliver. We will identify a set of requirements under use cases. While use case focuses on what user wants to do with our system, requirement focuses on what the system needs to deliver to fulfill the use cases.

The **Requirement List** is a place where you can store and manage requirements. You can also gain an overview of requirements involved in the entire system.

Opening the Requirement List

To open **Requirement List**, select **Modeling > Requirement List** from the main menu. If you see the panes like **Diagram Navigator** and **Property Pane** minimized, with help contents showing in the middle of screen, this means that this is the first time you open the **Requirement List** . The panes are minimized to maximize the viewable list area. Click **Keep Change** if you want to keep the panes minimized. If you want to roll back to the original screen layout, click **Revert**. And if you keep the change now, you can still customize the screen layout by applying a perspective later on (**View > Open Perspective**).

Overview of Requirement List



Requirement List

No	Name	Description
1	New Requirement	Click to create a requirement.
2	Font Size	Click to adjust the font size of text in **Requirement List**.
3	How to	Click to hide/show the help contents of **Requirement List**.
4	Open Specification...	Select a requirement in **Requirement List** and click this button to open its specification.
5	Show View...	Select a requirement in **Requirement List** and click this button to list the diagrams that contains the view of the selected requirement.
6	Visualize	Select a requirement in **Requirement List** and click this button to show it in a new or existing diagram.
7	Search	Find requirement(s) by entering search criteria.
8	List of requirements	Requirements are listed here.
9	Requirement name	Name of selected requirement.
10	Requirement ID	ID of selected requirement. ID are automatically generated when you create requirement. You may customize the pattern of ID in the **Project Options** window (**Tools > Project Options > Diagramming > Model Generation**).
11	Source	The way how the requirement was created.
12	Kind	The type of requirement.
13	Verify Method	The way how the requirement can be verified.

14	Risk	The level of risk in supporting the requirement.
15	Status	The current status of requirement.
16	Use Cases	Use cases can be achieved by implementing requirements. If the selected requirement was created from a use case, or added as a requirement of a use case, you can see the use cases here.
17	Requirement documentation editor	Documentation of selected requirement. The tools above the editor enables you to enter documentation in rich text format.

Description of Requirement List

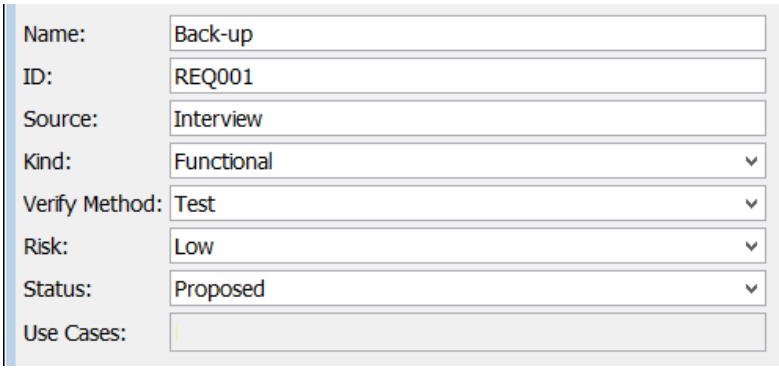
Creating requirement in Requirement List

To create a requirement in **Requirement List**:

1. Click on **New Requirement** above the **Requirement List**.
2. Enter the name of requirement.

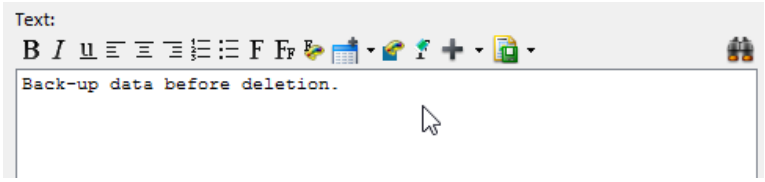
Creating requirement in Requirement List

3. Press **Enter** to confirm editing.
4. You can optionally edit the properties of the requirement



Edit requirement properties

5. You can optionally edit the properties of the requirement.



Edit requirement documentation

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Requirements with Requirement List?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Use Case List

Record and manage use cases in Use Case List

Learn how to record use cases in Use Case List

Rank use cases in Use Case List

Learn how to rank use cases base on their business value.

Prioritize use cases in Use Case List

Learn how to prioritize use cases in Use Case List.

Setting use cases' status in Use Case List

Learn how to set use cases into different statuses.

Record and manage use cases in Use Case List

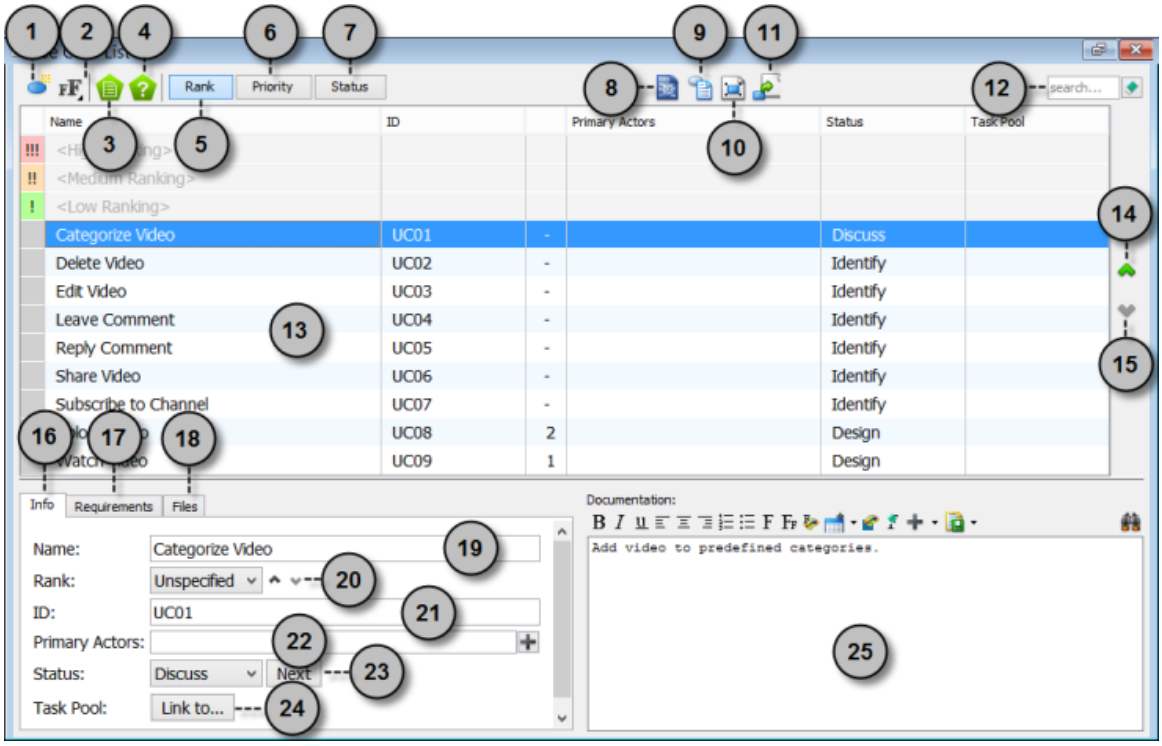
Use cases represent what the end user want to achieve by using the system.

The **Use Case List** is a place where you can store and manage use cases.

Opening the Use Case List

To open **Use Case List**, select **Modeling > Use Case List** from the main menu. If you see the panes like **Diagram Navigator** and **Property Pane** minimized, with help contents showing in the middle of screen, this means that this is the first time you open the **Use Case List**. The panes are minimized to maximize the viewable list area. Click **Keep Change** if you want to keep the panes minimized. If you want to roll back to the original screen layout, click **Revert**. And if you keep the change now, you can still customize the screen layout by applying a perspective later on (**View > Open Perspective**).

Overview of Use Case List



Use Case List

No	Name	Description
1	New Use Case	Click to create a use case.
2	Font Size	Click to adjust the font size of text in **Use Case List**.
3	Open Use Case Note	Select a use case in **Use Case List** and click this button to open the **Use Case Notes** editor to enter the use case -related notes about the selected use case.
4	How to	Click to hide/show the help contents of **Use Case List**.
5	Rank	Click to view the rank of use cases. Click here to learn more about ranking use cases.
6	Priority	Click to view the priority of use cases. Click here to learn more about prioritizing use cases.
7	Status	Click to view the status of use cases. Click here to learn more about setting use case status.
8	Open Specification...	Select a use case in **Use Case List** and click this button to open its specification.
9	Open Use Case Details	Select a use case in **Use Case List** and click this button to acces its details in **Use Case Details** editor.
10	Show View...	Select a use case in **Use Case List** and click this button to list the diagrams that contains the view of the selected use case.
11	Visualize	Select a use case in **Use Case List** and click this button to show it in a new or existing diagram.
12	Search	Find use case(s) by entering search criteria.
13	List of use cases	Use cases are listed here.
14	Move up	Move the selected use case(s) upwards.
15	Move down	Move the selected use case(s) downwards.

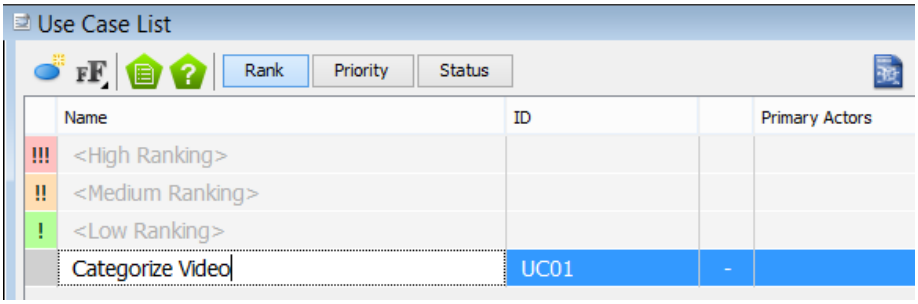
16	Info	Informatin of the selected use case is show in this tab.
17	Requirements	A use case can be achieved by implementing a set of relevant requirements. Click this tab to see or add requirements into this use case.
18	Files	You can add reference to use case -related files. Referenced files can be opened directly under the **Files** tab.
19	Use case name	Name of selected use case.
20	Rank	Rank of use case. The rank of a use case is determined by its business values. You can select a rank from the drop down menu, or by pressing the up and down button.
21	ID	ID of selected use case. ID are automatically generated when you create use case. You may customize the pattern of ID in the **Project Options** window (**Tools > Project Options > Diagramming > Model Generation**).
22	Primary Actors	The primary actors that interact with the selected use case.
23	Status	The current status of use case.
24	Task Pool	Development tasks can be created in Tasifier. Here you can map the selected use case with a task pool in Tasifier for task management purpose.
25	Use case documentation editor	Documentation of selected use case. The tools above the editor enables you to enter documentation in rich text format.

Description of Use Case List

Creating use case in Use Case List

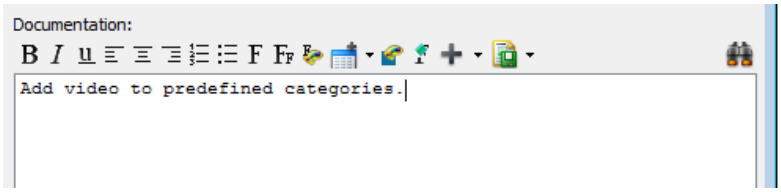
To create a use case in **Use Case List**:

1. Click on **New Use Case** above the **Use Case List**.
2. Enter the name of use case.



Creating use case in Use Case List

3. Press **Enter** to confirm editing.
4. You can optionally edit the properties of the use case.



Edit use case documentation

Related Resources

The following resources may help you learn more about the topic discussed in this page.

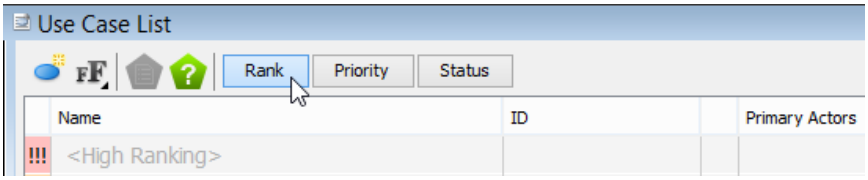
- YouTube Video - How to Manage Use Cases with Use Case List?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Rank use cases in Use Case List

The rank of a use case is determined by looking at its business value. Generally, you pay more attention to use cases that represent the most business value.

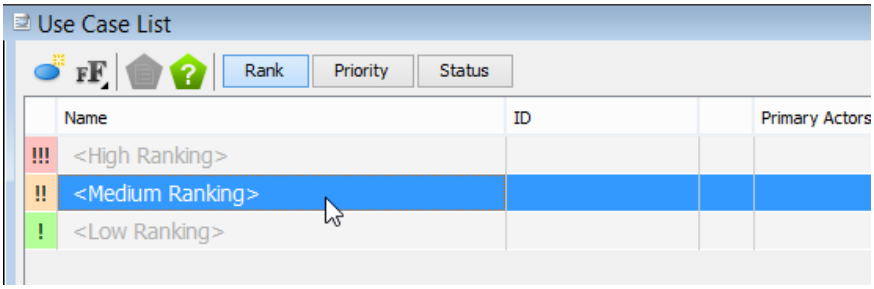
Creating use case in specific rank

1. Open the **Use Case List** by selecting **Modeling > Use Case List** from the main menu.
2. Make sure you are opening the **Rank** page.



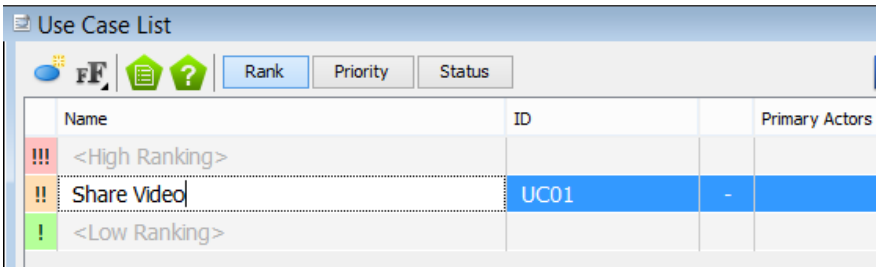
Opening the Rank page

3. There are three ranks - **High**, **Medium** and **Low**. Click on the row that represent the rank of the use case to be created. If there is no use case in the rank, click the row < **High/ Medium/ Low Ranking**> directly. If there is already use cases in the rank, click on any use cases in that rank.



Select the rank to create use case in it

4. Click on **New Use Case** above the **Use Case List**.
5. Enter the name of use case.



Creating use case in Use Case List

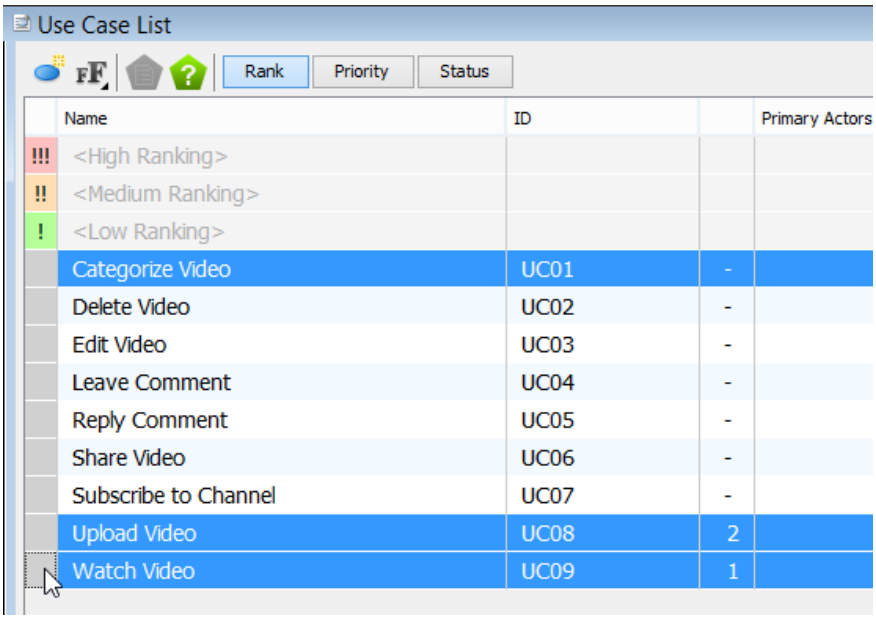
6. Press **Enter** to confirm editing.

Ranking existing use cases

To move existing use cases to a rank:

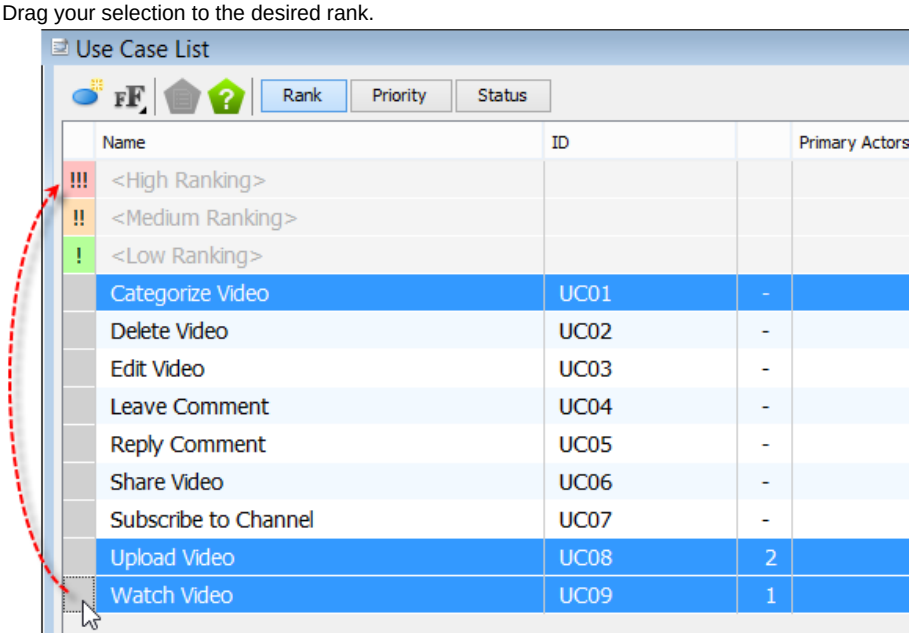
Method 1 - Drag and Drop

1. Open the **Use Case List** by selecting **Modeling > Use Case List** from the main menu.
2. Make sure you are opening the **Rank** page.
3. In the **Use Case List**, select the use cases that belong to the same rank. You can perform multiple selection by holding the **Ctrl** or **Shift** key.



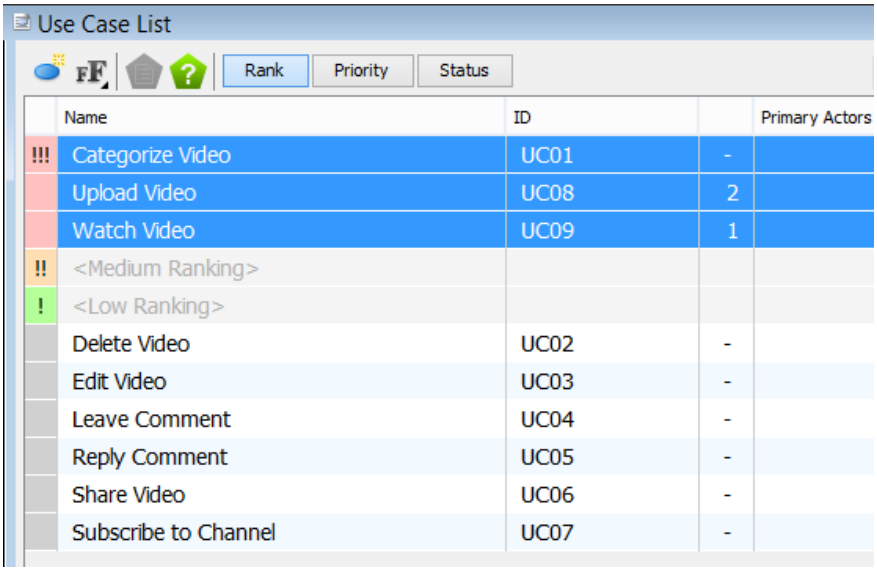
Select use cases

4. Drag your selection to the desired rank.



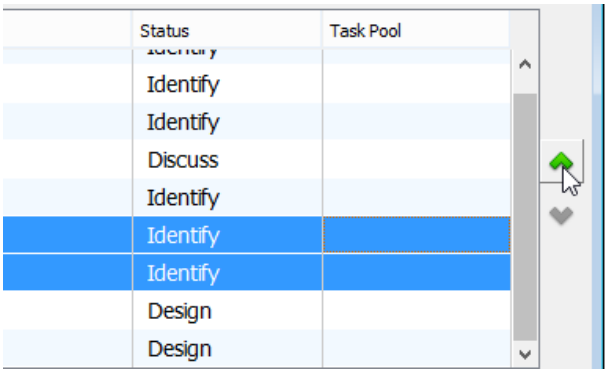
Move use cases to high rank

5. Release the mouse button to"drop" the use cases to the rank.



Method 2 - Move Up/Down

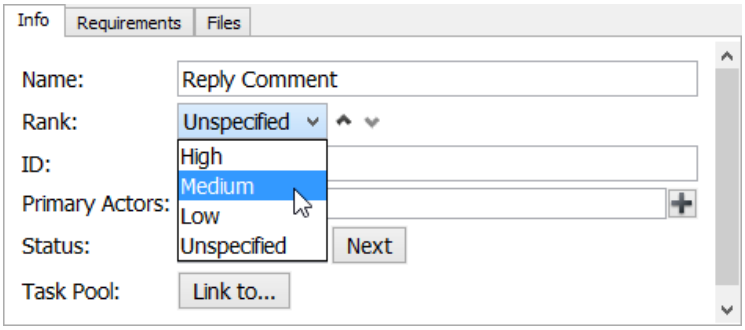
You can also move multiple use cases to a higher or lower rank by selecting the use cases first and clicking on the **Move Up** or **Move Down** button on the right hand side of the **Use Case List**. By clicking once, the selected use cases are moved one rank upwards or downwards. This method works only when the selected use cases are all in the same status. If not, the **Move Up** and **Move Down** button will be disabled.



Move use cases to higher rank

Method 3 - Edit in Info tab

You can also edit the status of use case under the **Info** tab, either by selecting the status from the drop down menu or by clicking the **Next** button to move one status forward. This method works only with single use case selection.



Edit use case's rank in Info tab

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Use Cases with Use Case List?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

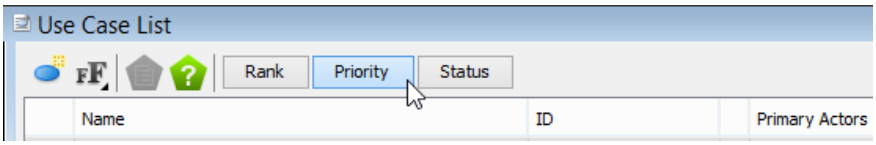
Prioritize use cases in Use Case List

To prioritize use case is to determine which use case(s) to work on first. The priority of a use case is generally determined by factors like the business value of use case, the availability of developers, the dependency between use cases and the level of risk.

Prioritizing use cases

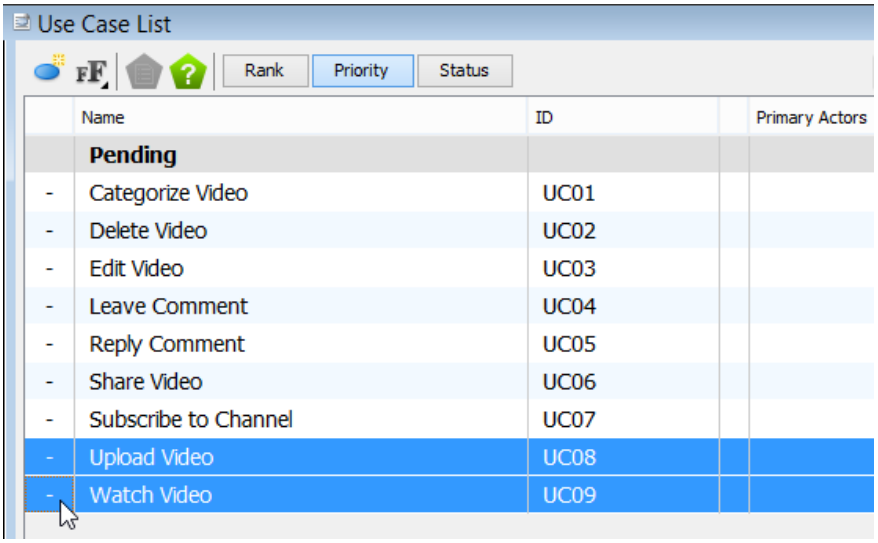
To prioritize use cases:

1. Open the **Use Case List** by selecting **Modeling > Use Case List** from the main menu.
2. Open the **Priority** page.



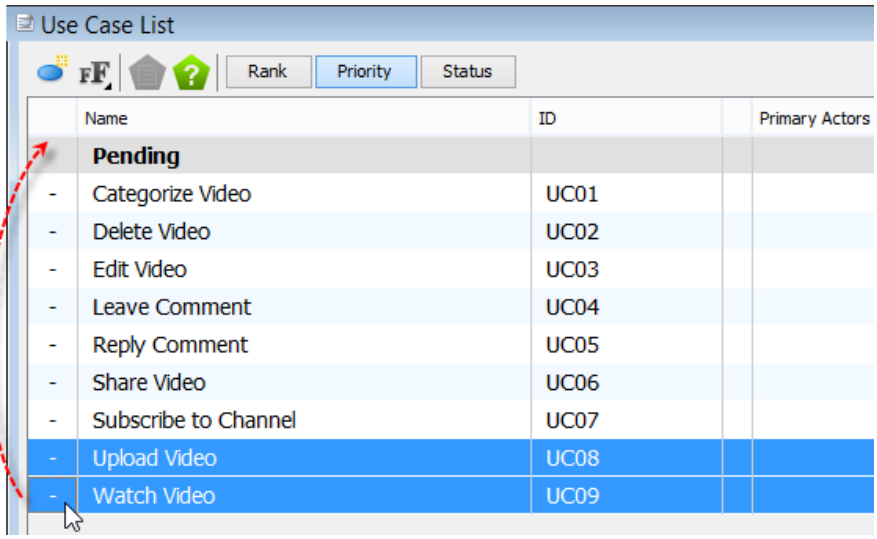
Open the Priority page

3. Use cases without being priortized are in **Pending** state. Select the use cases with high priority. You can perform multiple selection by holding the **Ctrl** or **Shift** key.



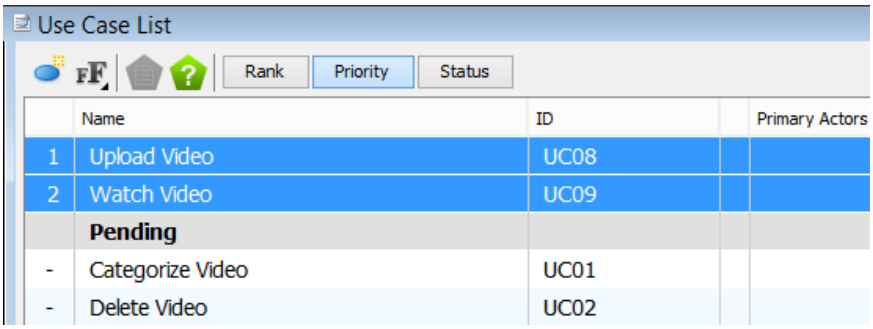
Select use cases

4. Drag your selection up to the **Pending** row.



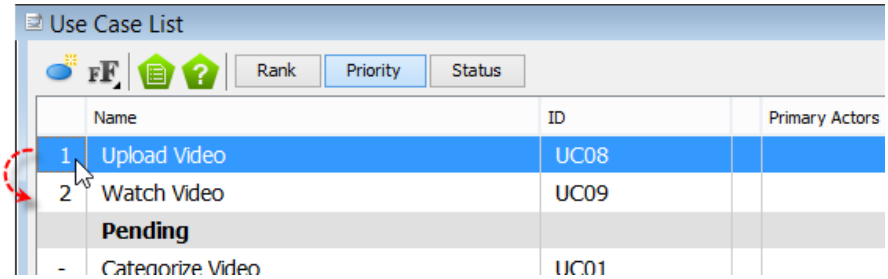
Move use cases to high priority

5. Release the mouse button. The selected use cases will appear above Pending. This means that they are no longer pending, but prioritized.



Use cases in high priority

6. Customize the order of prioritized use cases with drag-and-drop. Alternatively, you can click the Move Up and Move Down button on the right hand side of the Use Case List for re-ordering. Note that you can only re-order prioritized use cases. In other words, pending use cases cannot be ordered.



Re-order use cases

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Use Cases with Use Case List?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Setting use cases' status in Use Case List

The status of a use case is about how much the use case is completed. There are seven status in total:

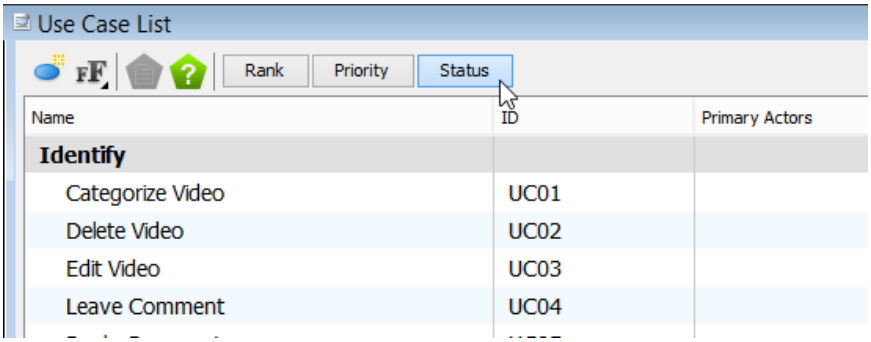
- Identify - Use case is identified by discussing with or interviewing stakeholder
- Discuss - Discussion is made with stakeholder in clarifying the details of use case
- Elaborate - Use Case scenario(s) is created to describe the interaction between actor and system in achieving the use case. Requirements are created, too.
- Design - Wireframes are created to illustrate screen layout
- Consent - The scenario, requirements and wireframes are all approved by stakeholder, ready for the actual development.
- Develop - Actual development activities are launched for the use case
- Complete - The development of use case has been done.

Setting the status of use cases

To set the status of use cases:

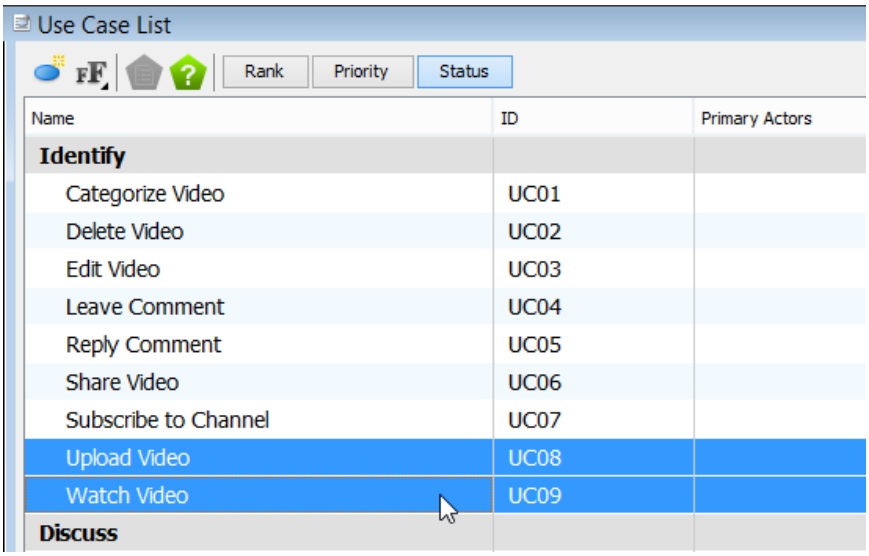
Method 1 - Drag and drop

1. Open the **Use Case List** by selecting **Modeling > Use Case List** from the main menu.
2. Open the **Status** page.



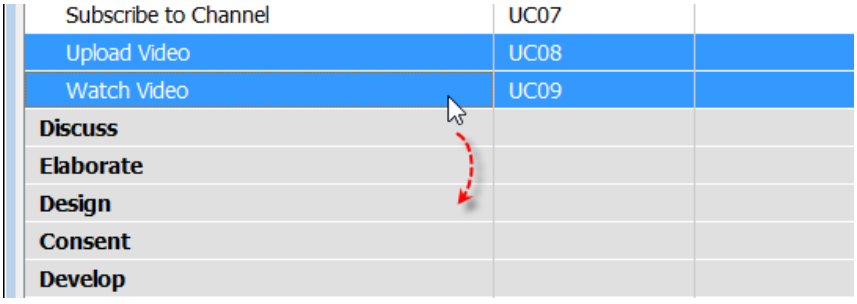
Open the Status page

3. Use cases are all in **Identify** state by default. Select the use cases to have their status updated. You can perform multiple selection by holding the **Ctrl** or **Shift** key.



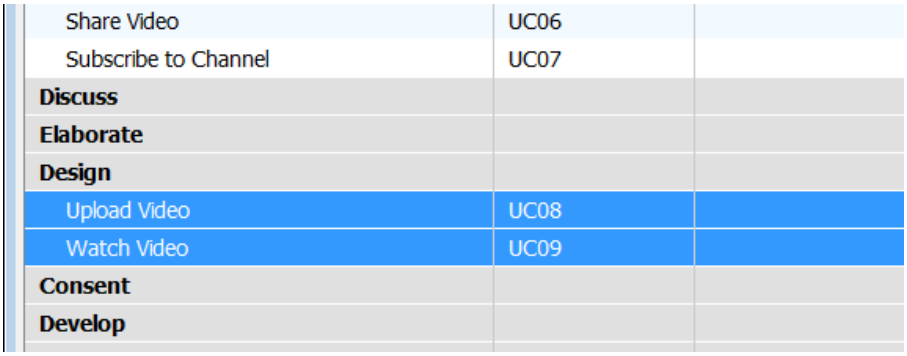
Select use cases

4. Drag your selection to the appropriate row of status.



Move use cases to design state

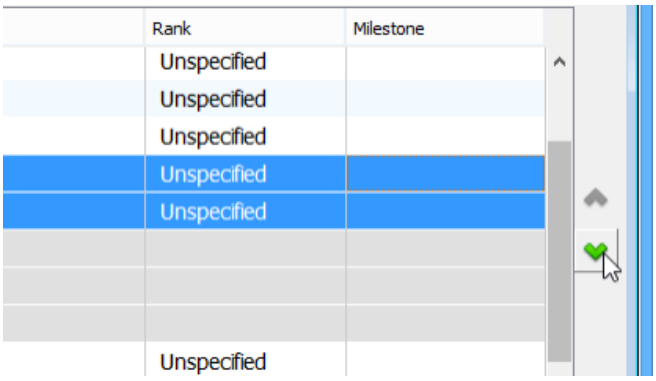
5. Release the mouse button. The selected use cases will appear under the target status.



Use cases in design state

Method 2 - Move Up/Down

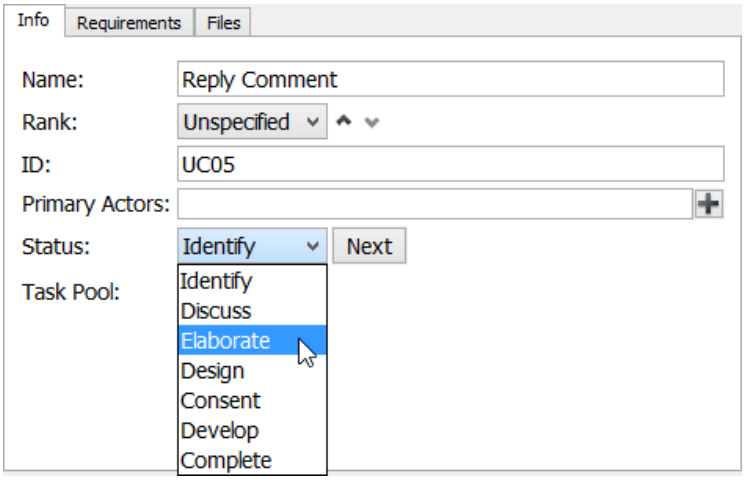
You can also move multiple use cases to the next or previous status by selecting the use cases first and clicking on the **Move Up** or **Move Down** button on the right hand side of the **Use Case List**. By clicking once, the selected use cases are moved one status backward or forward. This method works only when the selected use cases are all in the same status. If not, the **Move Up** and **Move Down** button will be disabled.



Move use cases to next status

Method 3 - Edit in Info tab

You can also edit the status of use case under the **Info** tab, either by selecting the status from the drop down menu or by clicking the **Next** button to move one status forward. This method works only with single use case selection.



Edit use case's status in Info tab

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Use Cases with Use Case List?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Use Case Notes

Note down use case related ideas in Use Case Note

Learn how to take notes for use case related ideas.

Produce use case scenario from notes

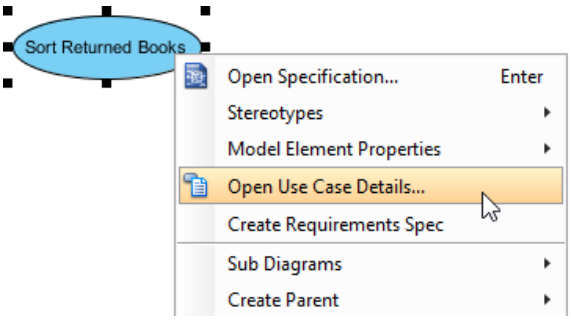
Learn how to instantly generate scenario from Use Case Notes.

Note down use case related ideas in Use Case Note

While meeting with users is an important part of requirements capturing, multiple meetings are essential for clarifying what user really wants, for clarifying usage of the system, and for obtaining users' consent to the requirements. Use Case Notes is designed for you to note down the discussion during requirements capturing meetings. You can outline the workflow, write down users' concerns and conclusion in a Use Case Note. You can also turn your note item into use case scenario or requirements by just one click. Please watch below video tutorial for details. All Use Case Notes will be kept under the use case. You don’t need to spend extra effort to manage your meeting notes any more.

Accessing use case notes of a use case

The **Use Case Notes** (editor) is grouped as part of the **Use Case Details**. There are mainly two ways you can take to access the use case notes of a use case. When you are reading a use case diagram, you can access use case notes of a use case by right clicking on that use case and selecting **Open Use Case Details...** from the popup menu.

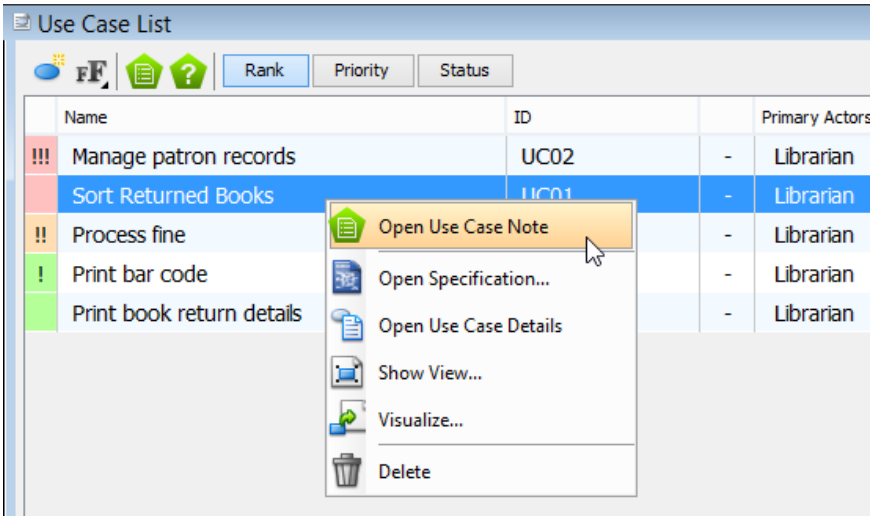


Open Use Case Details

Then, open the **Use Case Notes** tab in **Use Case Details**.

Open Use Case Notes

When you are working in the **Use Case List**, you can access use case notes by right clicking on the desired use case and selecting **Open Use Case Note**. Then, open the **Use Case Notes** tab in **Use Case Details**.

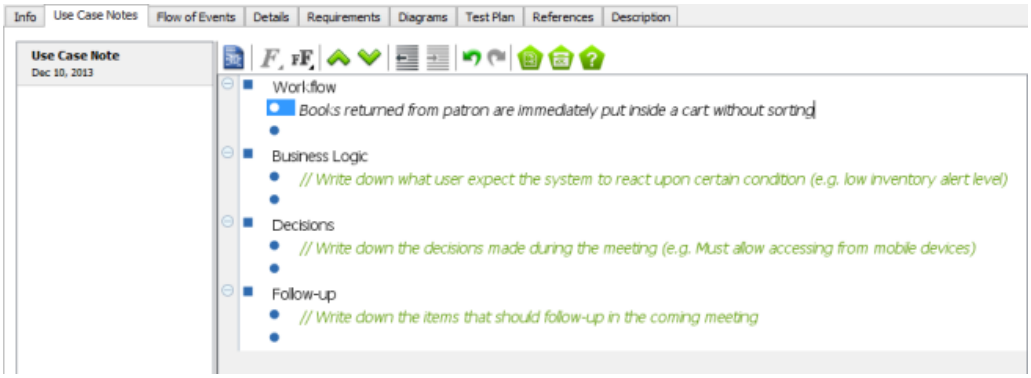


Open Use Case Note from Use Case List

If you see the panes like **Diagram Navigator** and **Property Pane** minimized, with help contents showing in the middle of screen, this means that this is the first time you open the **Use Case Note**. The panes are minimized to maximize the viewable list area. Click **Keep Change** if you want to keep the panes minimized. If you want to roll back to the original screen layout, click **Revert**. And if you keep the change now, you can still customize the screen layout by applying a perspective later on (**View > Open Perspective**).

Entering use case notes

Once you have opened the **Use Case Notes** of a use case, you can start entering notes. There is a pre-defined template, with four points in it - **Workflow**, **Businses Logic**, **Decisions** and **Follow-up**. You can follow these points in note taking. All you need to do is to click on the green text and replace it by entering your note content.



Entering a note by following the template

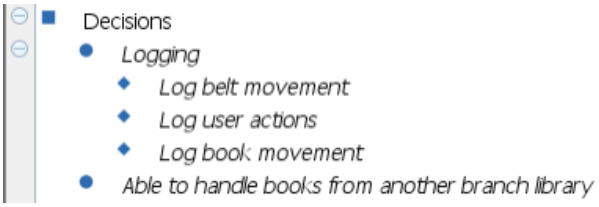
If you do not want to follow the points suggested by the templates, you can delete it by highlighting the rows and pressing the **Delete** key.

To enter a note, type the note content in the editor.

To create a new note, press **Enter**.

Working with nested notes

Different kinds of use case -related ideas can be recorded by creating multiple nested notes. You can press **Tab** to indent, and press **Shift-Tab** to reduce indentation.



Nested notes

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Take Use Case Notes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

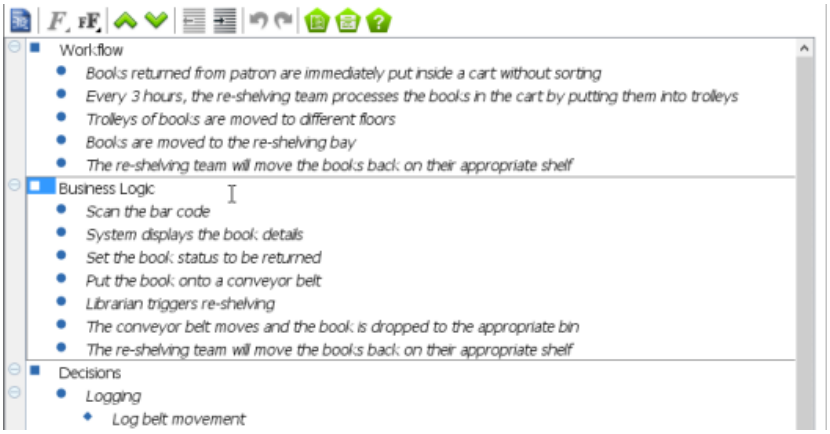
Produce use case scenario from notes

When you meeting with the stakeholders, they will let you know their expectation regarding the features of the system to be developed. Very often, it involves the system behaviors they preferred. By noting down the preferred system behaviors as use case notes, you can easily produce an initial use case scenario, and to make further changes in it.

Producing a new use case scenario from use case notes

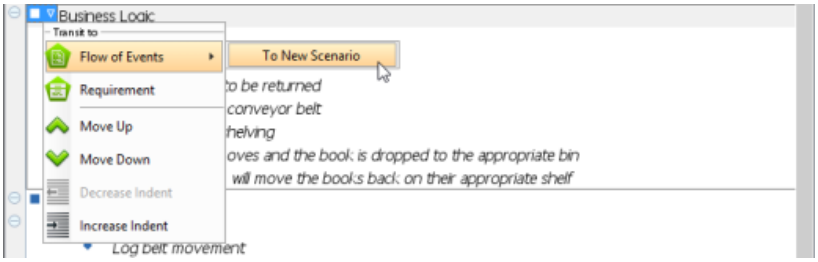
To produce a new use case scenario from use case notes:

1. Open the **Use Case Notes** of the desired use case.
2. Move the mouse pointer over the parent note item where the suggested system behaviors are recorded.



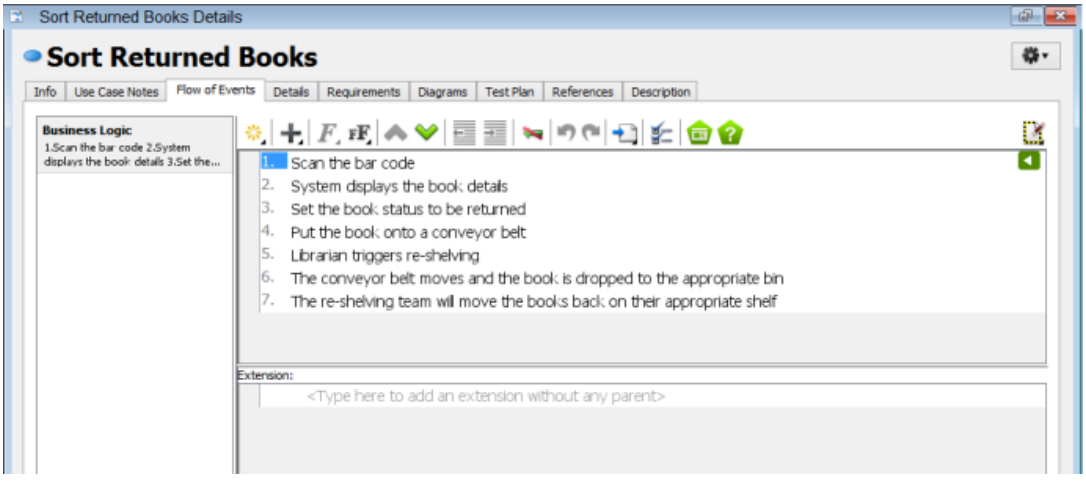
Moving mouse pointer over a note item

3. Move the mouse pointer to the beginning of the note item. Click on the down arrow next to the bullet point and select Flow of Events > To New Scenario from the popup menu.



Creating a new scenario

This produces a new scenario, with the text of the chosen note item becomes the name of scenario, and the sub-note items becoming the steps of the scenario.



Scenario produced

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Take Use Case Notes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Use Case Flow of Events

Develop use case scenario in Flow of Events editor

Learn how to write use case scenario.

Working with multiple use case scenarios

Learn how to represent multiple stories by creating multiple use case scenarios

Perform scenario-based wireframing

Learn how to perform scenario-based wireframing.

Wireframe playback

Learn how to play your wireframes, which is important in a presentation.

Producing requirement specification

Learn how to generate requirement specification from use case flow of events.

Elaborating use case

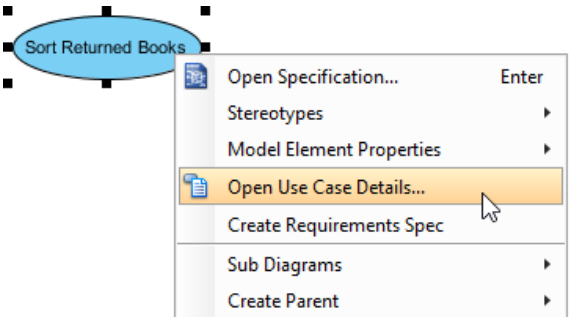
Learn how to elaborate use case with sequence diagrams.

Develop use case scenario in Flow of Events editor

A use case scenario refers to the steps required to go through and then achieve a user goal. When you want to specify the process of how to achieve a use case clearly as a guideline, you can define it step by step in flow of events editor. This page will introduce the flow of events editor by describing features, such as model element link presentation options and use case extension.

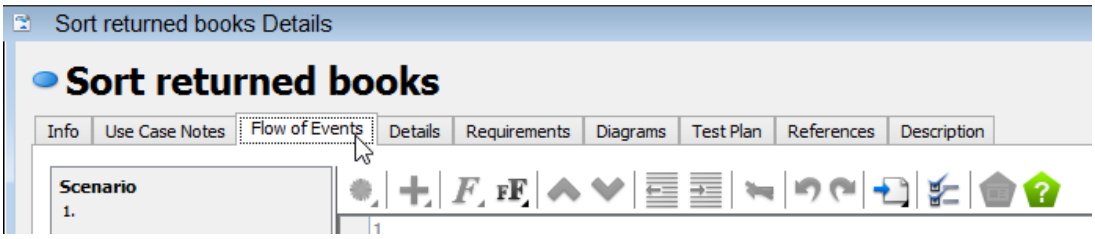
Accessing use case scenarios of a use case

The **Flow of Events** (editor) is grouped as part of the **Use Case Details**. There are mainly two ways you can take to access the scenarios of a use case. When you are reading a use case diagram, you can access use case notes of a use case by right clicking on that use case and selecting **Open Use Case Details...** from the popup menu.



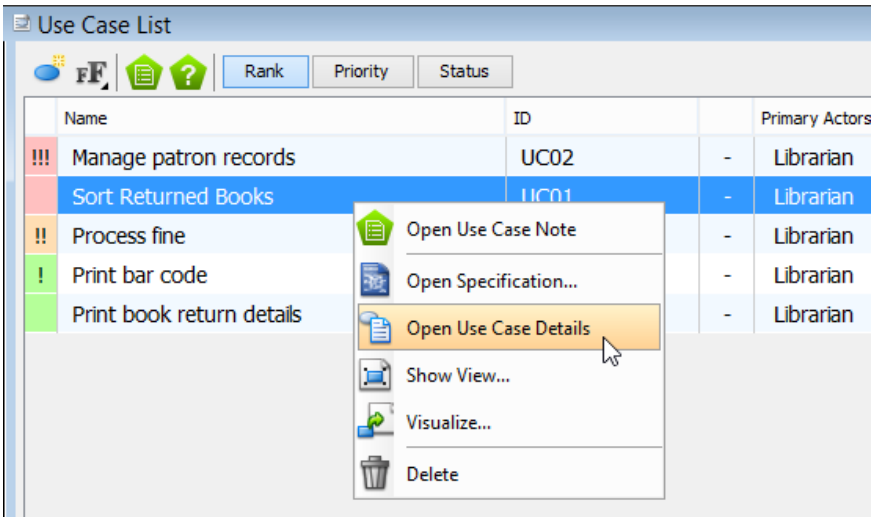
Open Use Case Details

Then, open the Flow of Events tab in **Use Case Details**.



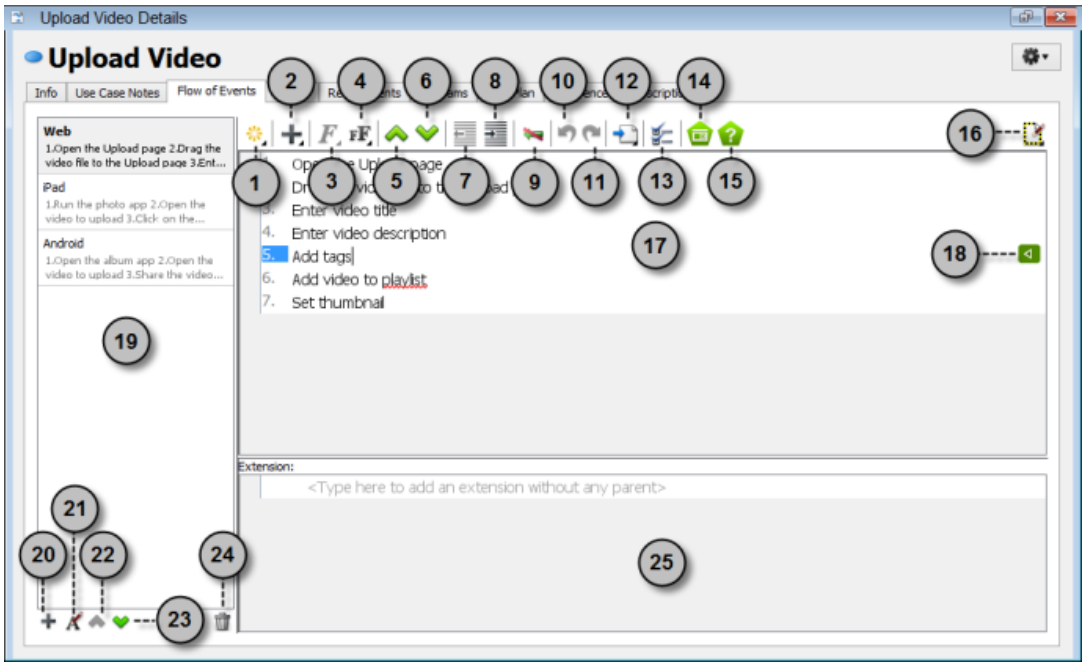
Open Flow of Events

When you are working in the **Use Case List**, you can access use case notes by right clicking on the desired use case and selecting **Open Use Case Details**. Then, open the **Flow of Events** tab in **Use Case Details**.



Open Use Case Note from Use Case List

Overview of the Flow of Events editor



Flow of Events editor

No.	Name	Description
1	Add Step	You may select various functions in the menu: **Step (Enter)**: Enter next step. **Extension (Shift+Enter)**: Enter extension for the selected step. **If**: Enter conditional situation for the selected step. **Else if**: Insert another situation under If. **Else**: Insert it to control **If** and **Else if**. **Clear control**, **Go to previous condition** and **Go to end**. **While**: Perform some actions as long as the condition stated in the while clause is valid. **For each**: Perform some actions by walking through each item as stated by the for each clause. **Loop until**: Perform some actions until condition stated in loop remaining valid. **Exit**: Exit in the middle of loop. **Jump**: Insert **Jump** to manipulate the sub-step after the variable situation happened. You can go back to the previous step by clicking the small yellow arrow.
2	Select model element	Click to select an existing use case/actor/requirement/business rule/class/diagram in the current project and insert it into the selected step as referenced link.
3	Font	Select font effect for the highlighted text: **Bold**, **Italic** and **Font Color**.
4	Font size	Click to adjust the font size of text in **Requirement List**.
5	Move Up	Click to move the selected step (in the scenario) upwards.
6	Move Down	Click to move the selected step (in the scenario) downwards.
7	Decrease Indent	Click to decrease the indentation of the selected step by one level.
8	Increase Indent	Click to increase the indentation of the selected step by one level.
9	Delete Step	Click to delete the selected step(s).
10	Undo	Undo the last editing action made in the **Flow of Events** editor.
11	Redo	Redo the last editing action reverted by undo.
12	Synchronize to diagram	Synchronize the flow of events to activity diagram or sequence diagram. If no activity diagram has created previously, a new activity diagram will be generated.
13	Model Element Link Presentation Options	To select the presentation for use case link/actor link/requirement link/business rule link. **Name** : To display the name of model element only. **ID** : To display the ID of model element only. **ID: Name**: To display both the name and ID of model element.
14	Show Wireframe	Click to hide/show the wireframe pane that displays the wireframe(s) associated with the selected step in the scenario.
15	How to	Click to hide/show the help contents of **Flow of Events**.

16	Testing Procedure	In order to ensure that a use case is able to achieve as your expectation, you can test each step of procedure defined in flow of events to view whether they will produce prospective result or not.
17	Scenario	Content of the scenario.
18	Define wireframe	Add or view the wireframe(s) associated with the selected step.
19	Scenario list	The scenarios of the use case is listed here.
20	Add Scenario	Click to create a new scenario. Mutliple scenarios can be created for describing the different ways of achieving the use case.
21	Edit Scenario	Click to rename the active scenario.
22	Move Up	Click to move the selected scenario upward.
23	Move Down	Click to move the selected scenario downward.
24	Delete Scenario	Click to delete the active scenario.
25	Extension	Use case extension pane that list the steps extended from the main flow.

Description of Flow of Events editor

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Use Case Scenario with Flow of Events?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Working with multiple use case scenarios

Very often, a use case can be achieved by multiple ways. Take the cinema ticket selling system as an example. "Sell ticket" is one of its use cases. It can be achieved by two ways - selling tickets both online and by staff. Sometimes, different results may be produced under different conditions during the course of use case. Take ATM as an example. To withdraw cash from ATM may result in a success or a failure when there is an insufficient funds. All these are variations of a use case, and can be described by creating multiple use case scenarios.

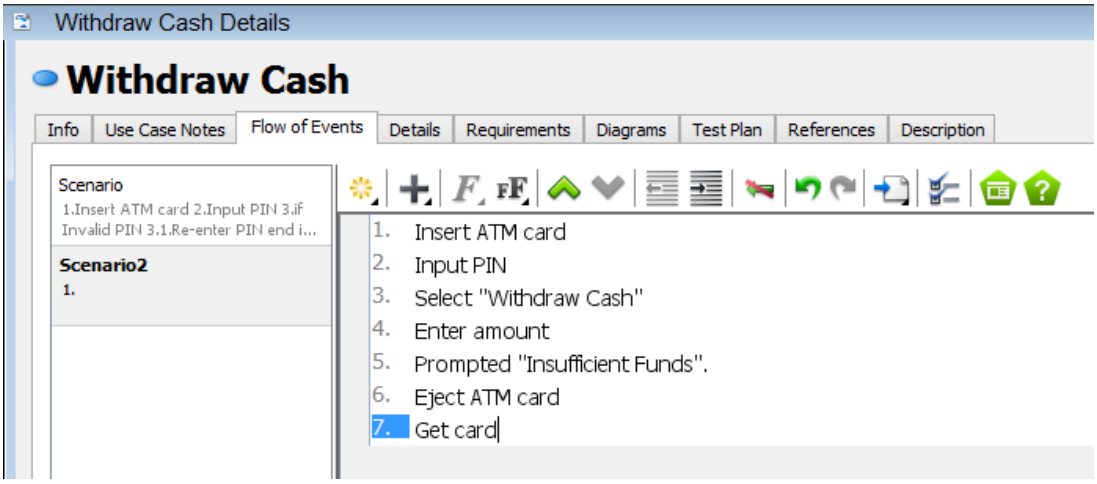
Creating multiple use case scenarios

To create a use case scenario:

1. Open the Flow of Events of the desired use case.
2. Click **Add Scenario** at the bottom left corner.

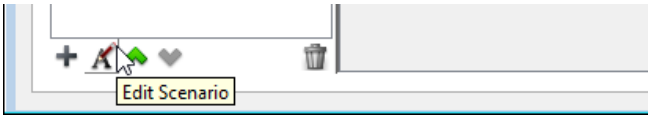
Add a scenario

3. Start entering the steps of the new scenario in the **Flow of Events** editor.



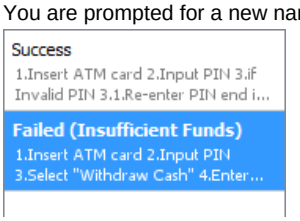
Enter the steps of the new scenario

4. Rename the scenarios. You can rename scenario by clicking **Edit Scenario** at bottom left.



Edit Scenario

5. You are prompted for a new name. Enter the name and confirm editing.



Scenarios are renamed

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Use Case Scenario with Flow of Events?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Perform scenario-based wireframing

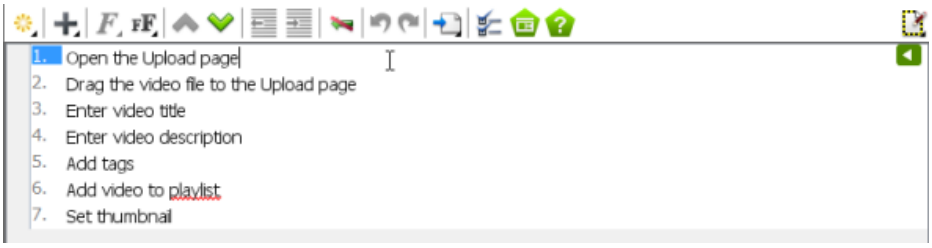
Showing a screen flow of the system to your customer guarantees your customer knows what will be delivered by the end of the project. This also saves us a lot of efforts in modifying the system in later stage of the development process because customer is involved and informed early. Instead of doing heavy system prototyping, you can "sketch" the user interface by wireframes. You can either sketch a new wireframe or reuse existing wireframes in each step of your scenario. The wireframe shows "just enough" information of the screen instead of the full details. The actual screen design will be produced at a later stage by referencing the wireframe.

By performing scenario-based wireframing, you can present your scenarios to your customer visually to obtain consent to the requirements easier.

Creating a wireframe for a step in scenario

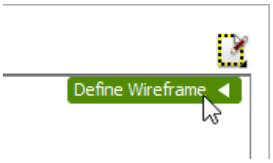
To create a wireframe for a step in scenario:

1. Open the Flow of Events of the desired use case.
2. Click on the step that you want to create a wireframe.



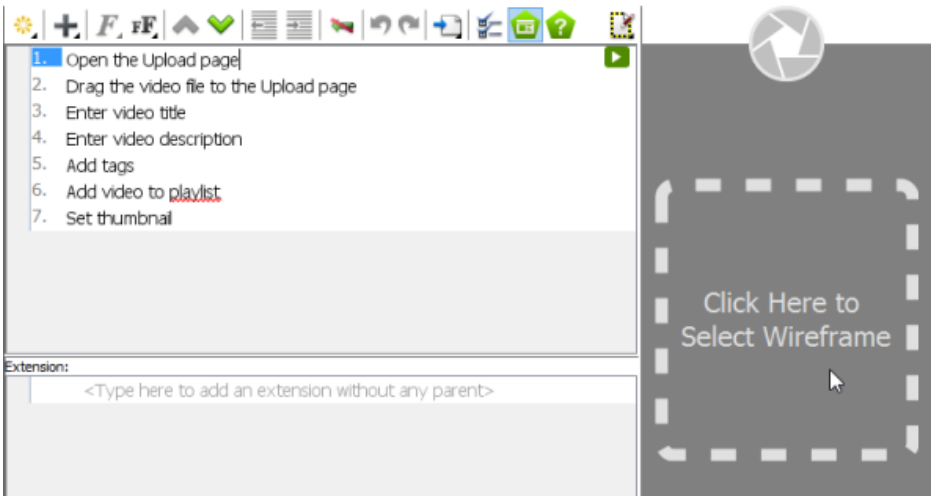
Selected a step in use case scenario

3. Move the mouse pointer over the **Define Wireframe** button (i.e. the green button) on the right hand side of the step. Click on it.



Define a wireframe for a scenario step

4. This shows a gray pane on the right hand side. Click on it to select a kind of wireframe to create.

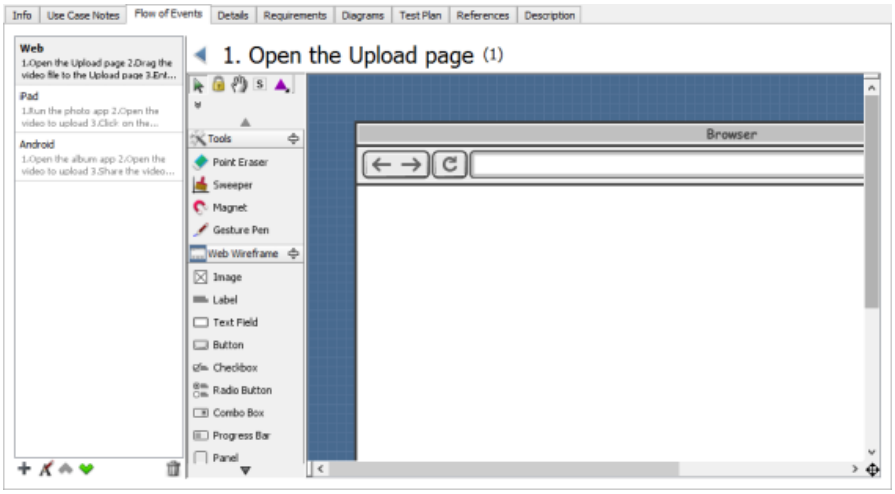


Select a wireframe

5. In the popup window, select the suitable type of device/platform for your application/system. If your system will run on multiple devices/platforms, please consider creating multiple scenarios.

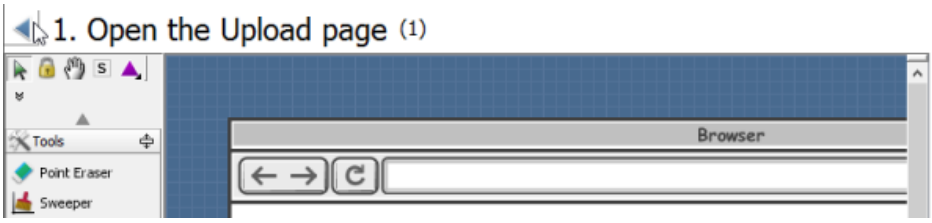
Select a type of wireframe to create

6. Click **New %TYPE% Wireframe** where **%TYPE% is** the type of device/platform you selected.
7. A blank, new wireframe appear and you can now begin editing.



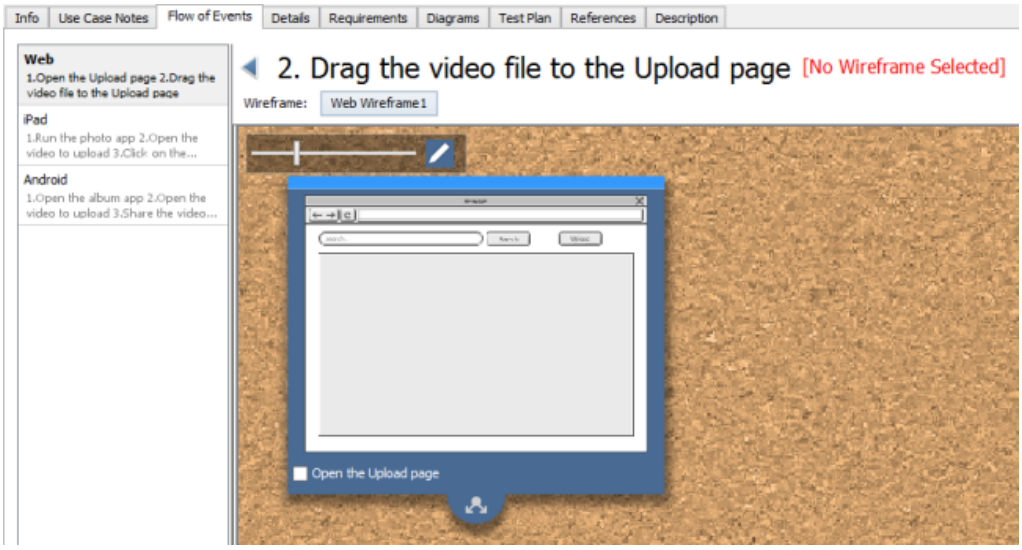
New wireframe created

8. When you finish editing, you can go back to the scenario by clicking on the back button on top of the wireframe.



Go back to the flow of events

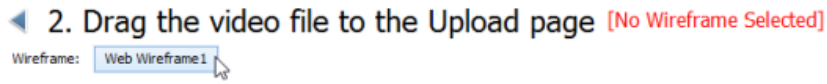
The above are the steps that involve in creating a wireframe from a scenario step when there is no wireframe in your project. Once you have created a wireframe, you will see something different after step 4, when you attempt to create a wireframe for another scenario step. Here is what you will see:



State overview

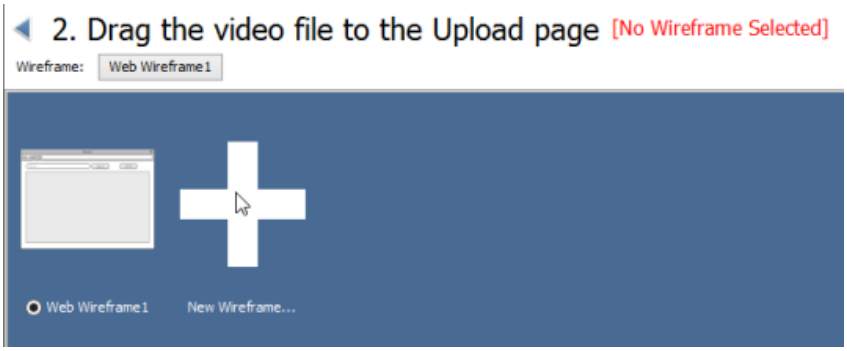
If you want to create an entirely new wirefram:

1. Click on the button next to **Wireframe:**.



Choose another wireframe

2. Click **New Wireframe...**.



Create a new wireframe

3. The remaining steps are same as those mentioned above, starting from step 5.

If you want to re-use an existing wireframe but make a bit of change, you should create a child state instead:

1. Click **Create Child State** below the thumbnail of wireframe to create a child state under it.

Create a child state

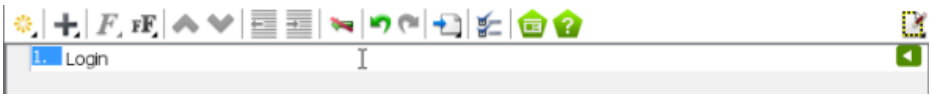
2. Once clicked, a new wireframe state will be created. You can start editing it.

Selecting an existing wireframe for a step in scenario

Sometimes, you may want to re-use a wireframe created earlier. For example, to reuse a wireframe about account login in scenarios that require user to login to do something.

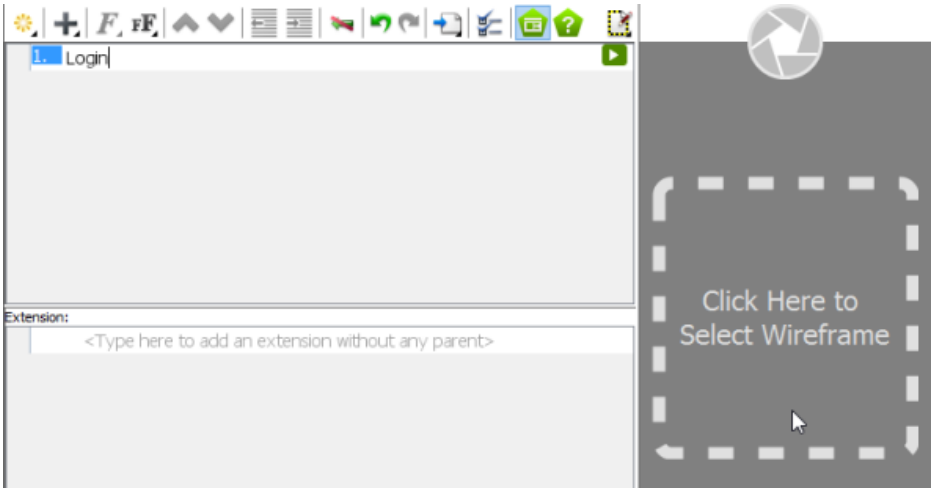
To select an existing wireframe for a step in scenario:

1. Open the Flow of Events of the desired use case.
2. Click on the step that you want to associate a wireframe with it.



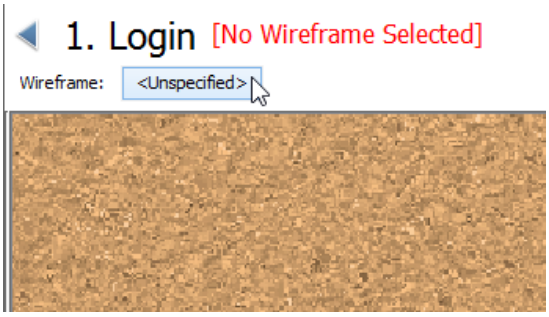
Selected a scenario step

3. Move the mouse pointer over the **Define Wireframe** button (i.e. the green button) on the right hand side of the step. Click on it.
4. This shows a gray pane on the right hand side. Click on it.



Select a wireframe for step

5. Click on the **<Unspecified>** button next to **Wireframe:**.



Clicking on Unspecified

6. Choose the wireframe for your step.

Choosing a wireframe

7. This shows the available states of the wireframe. Select the right one by checking the checkbox at bottom left corner. Make sure you did selected a state in this step. Without doing so, the wireframe won't be associated with the scenario step.

Selecting a wireframe state

8. Go back to the scenario by clicking on the back button on top of the state selection page.

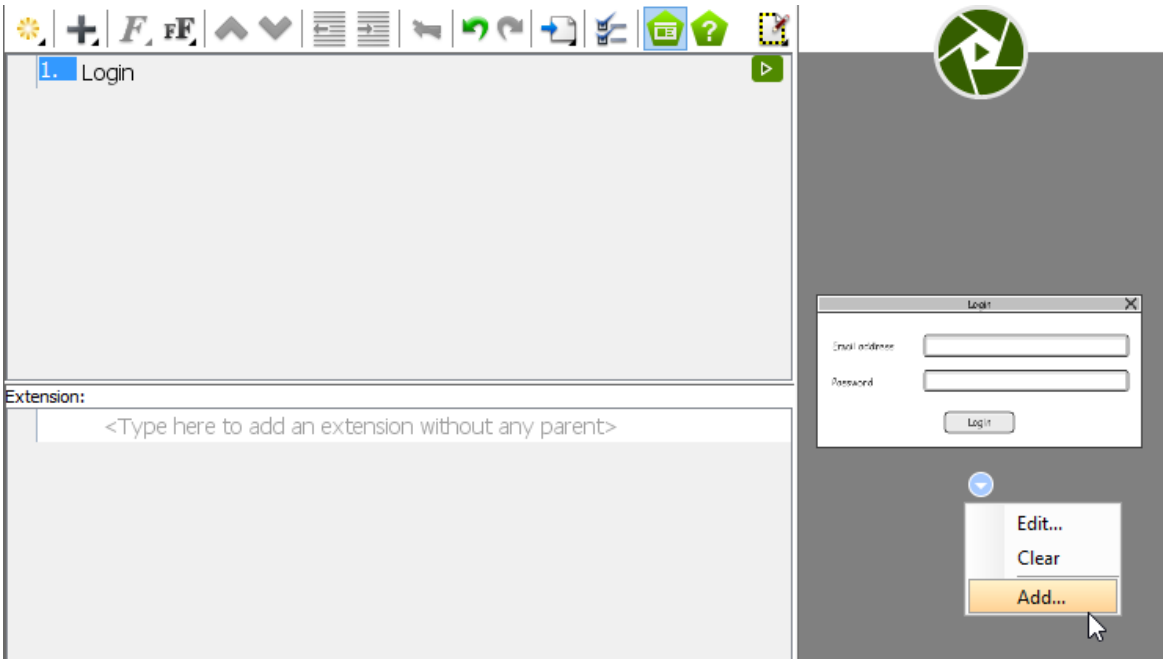
Go back to the flow of events

Adding extra wireframes to a step

If what you have written as a step of a scenario involves more than one screen change, you may need to add multiple wireframes to that step.

To add extra wireframes to a step:

1. Open the Flow of Events of the desired use case.
2. Click on the step that you want to add an extra wireframe to it.
3. Move the mouse pointer over the **Show Wireframe** button (i.e. the green button) on the right hand side of the step. Click on it.
4. There is a button under the thumbnail of the existing wireframe. Click on it and select **Add...** from the popup menu.

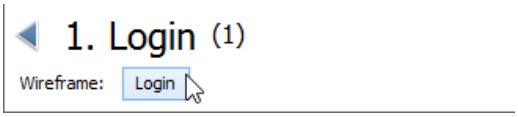


Add a wireframe to a scenario step

5. If you want to select another state of the selected wireframe, just check the checkbox of the wireframe state:

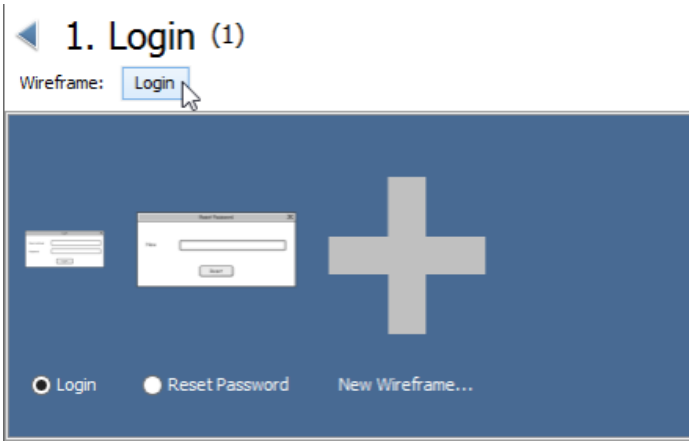
Selecting a wireframe state

If you want to select another wireframe, click on the button next to **Wireframe:**.



Select a wireframe

Then, select the wireframe or click **New Wireframe...** to create a new one.

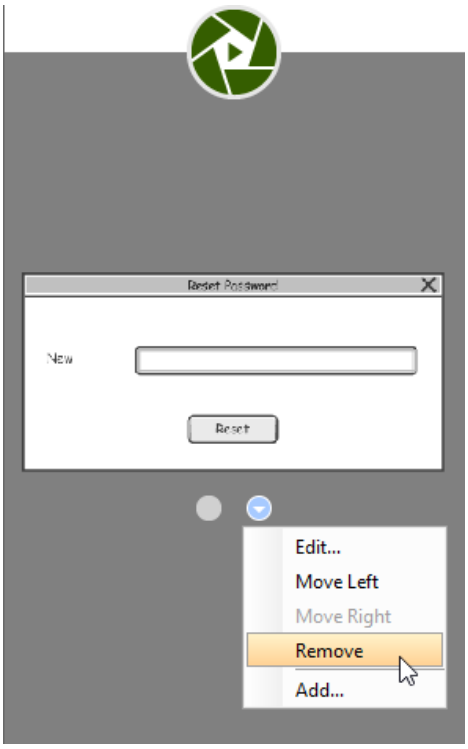


Select a wireframe

Removing a wireframe from a step

To remove a wireframe from a step:

1. Open the Flow of Events of the desired use case.
2. Click on the step that you want to remove a wireframe from it.
3. Move the mouse pointer over the **Show Wireframe** button (i.e. the green button) on the right hand side of the step. Click on it.
4. There is a button under the thumbnail of the existing wireframe. Click on it and select **Remove** from the popup menu.



Remove a wireframe

Related Resources

The following resources may help you learn more about the topic discussed in this page.

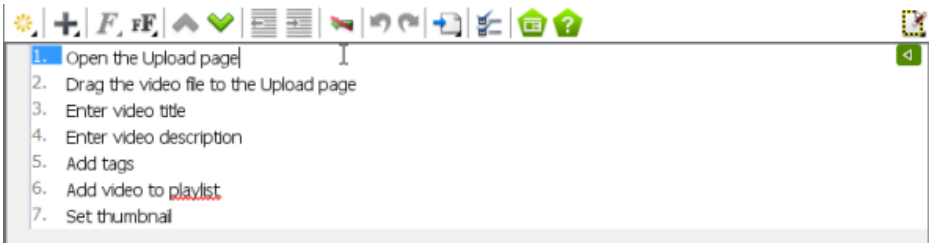
- YouTube Video - How to Create Scenario-Based Wireframe?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Wireframe playback

Showing a screen flow of the system to your customer guarantees your customer knows what will be delivered by the end of the project. Agilian not only allow you to associate use case scenario with wireframes in illustrating system interations, but also supports playing the wireframes associated with use case scenario. This can be very useful when you need to present the system design ideas to your customers, and to look for their consent.

Playing wireframes

1. Open the Flow of Events of the desired use case.
2. Click on the first step of the scenario.



Use case scenario

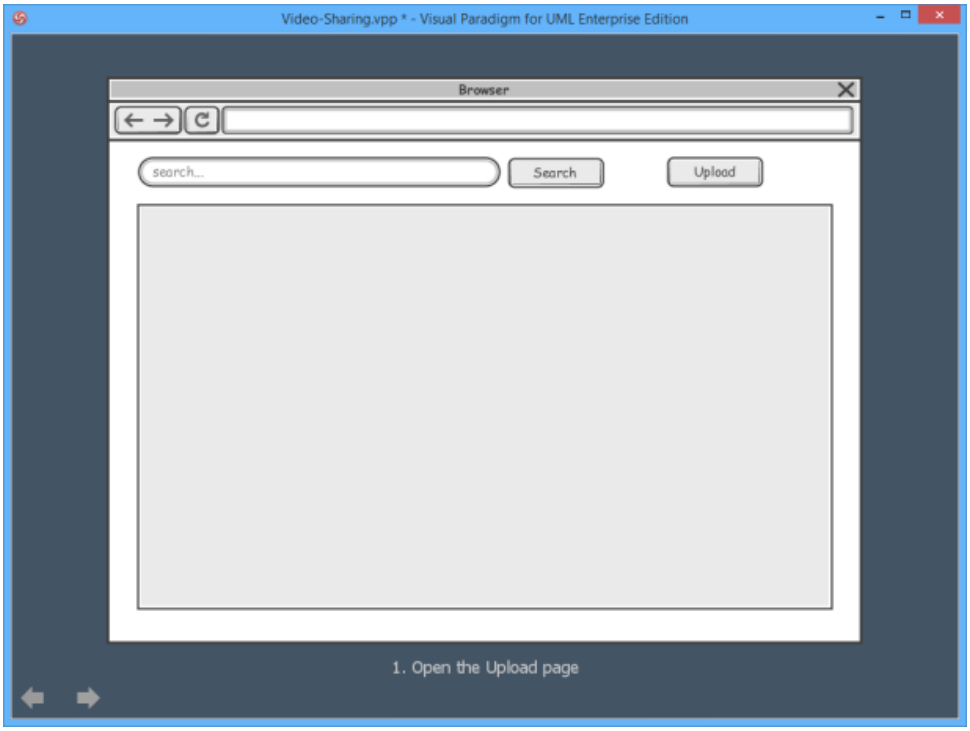
3. Move the mouse pointer over the **Show Wireframe** button (i.e. the green button) on the right hand side of the step. Click on it.

Show wireframe of a step in use case scenario

4. The preview of wireframe is now shown on the right hand side. On top of it there is a play button. Click on it.

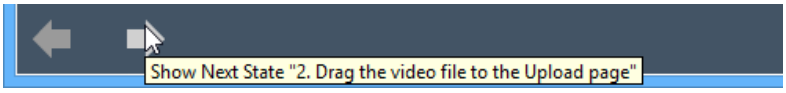
Play wireframes

5. This opens the wireframe player. The wireframe of the first step is shown in the player.



Playing a wireframe

You can move on to the next wireframe by pressing the **Right** key, or clicking on the arrow button at the bottom left of the player.



Move to the next wireframe

Similarly, you can press the Left key or click on the back button at the bottom left of the player to move to the previous wireframe.

Showing Annotations

To keep the wireframe clear and readable, annotations are hidden by default. You may click on the Show Annotations button t the bottom left of the player to have them visible. Let’s show the annotations.

Show annotations

Ending the Show

You can end the show anytime by pressing the Esc key. When it arrives the final wireframe, you can also exit by clicking on the Exit button at the bottom left of the player.

Exit playback

Related Resources

The following resources may help you learn more about the topic discussed in this page.

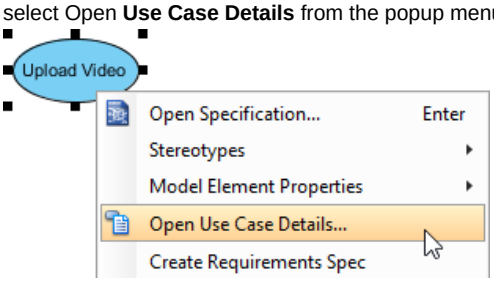
- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Producing requirement specification

Requirement specification is a document that contains all the details of use cases, requirements and wireframes. In Agilian you can produce such requirement specification from the **Use Case Details** of use cases.

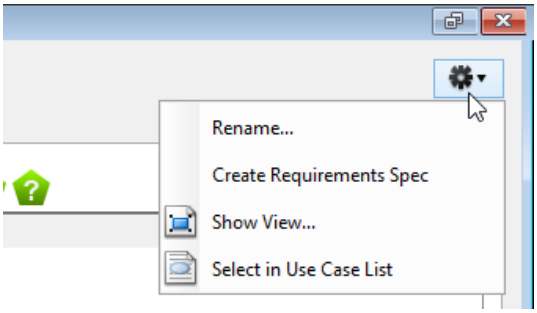
Producing requirement specification

1. Open the **Use Case Details** of the desired use case. When you are reading a use case diagram, right click on the desired use case and select **Open Use Case Details...** from the popup menu. If you are working with the **Use Case List**, right click on the desired use case in the list and select Open **Use Case Details** from the popup menu.



Open use case details

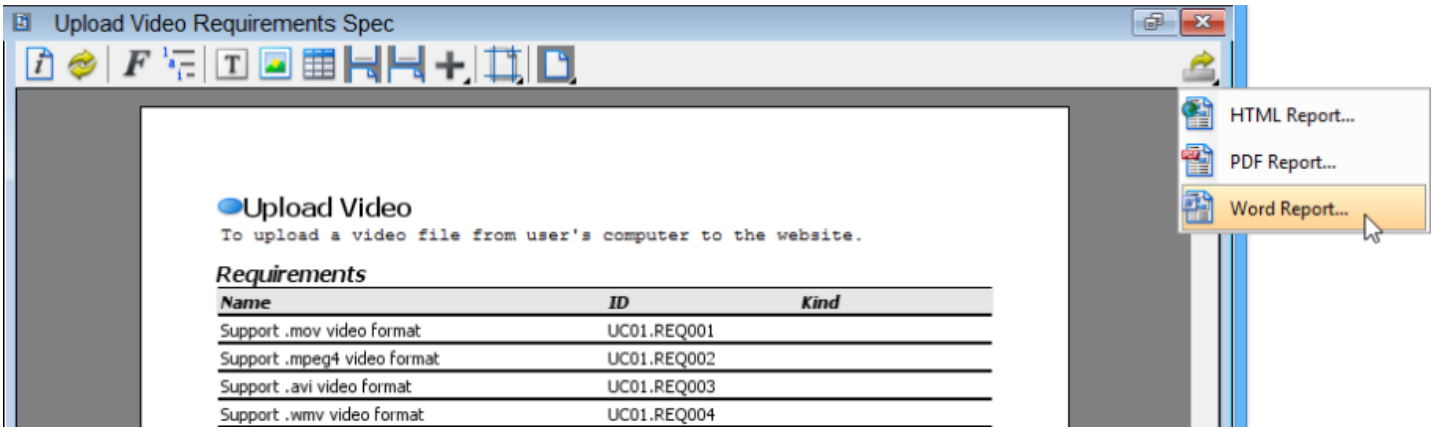
2. This opens the **Use Case Details**. On the right hand side of the screen, click on the gear button.



Create requirement specification

3. Select **Create Requirement Spec** from the popup menu.

4. This creates a requirement specification in the Report Composer. You can export the specification to Word/HTML/PDF by clicking on the **Export** button at the top right of the Report Composer. You may also edit the content of the report before exporting. To learn more about the use of Report Composer, click here.



Export requirement specification from Report Composer

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Produce to Requirement Specification?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Elaborating use case

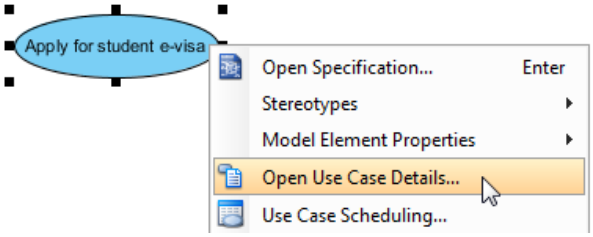
A use case can be elaborated by using sequence diagrams and activity diagrams to model its interactions and activity flows respectively.

Elaborating use case by sequence diagram

You can document the communication between user and system through the use of flow of events. You can then visualize the communication in sequence diagram through synchronizing flow of events to sequence diagram.

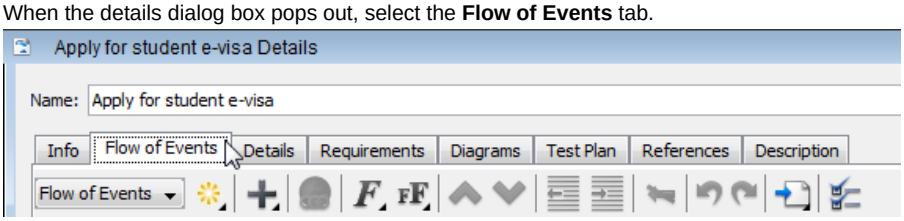
Synchronizing flow of events to sequence diagram

1. Right click on the target use case and select **Open Use Case Details...** from the pop-up menu.



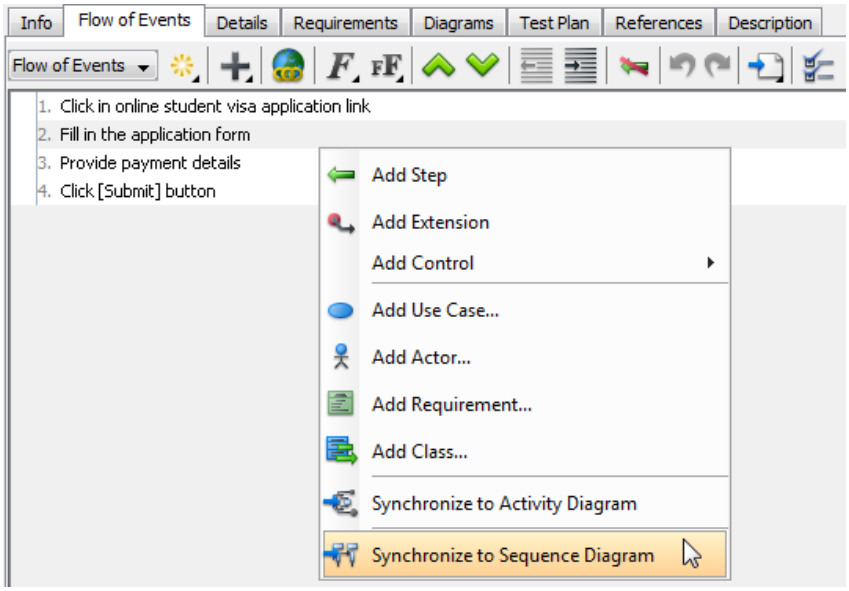
Open use case details

2. When the details dialog box pops out, select the **Flow of Events** tab.



Open flow of events editor

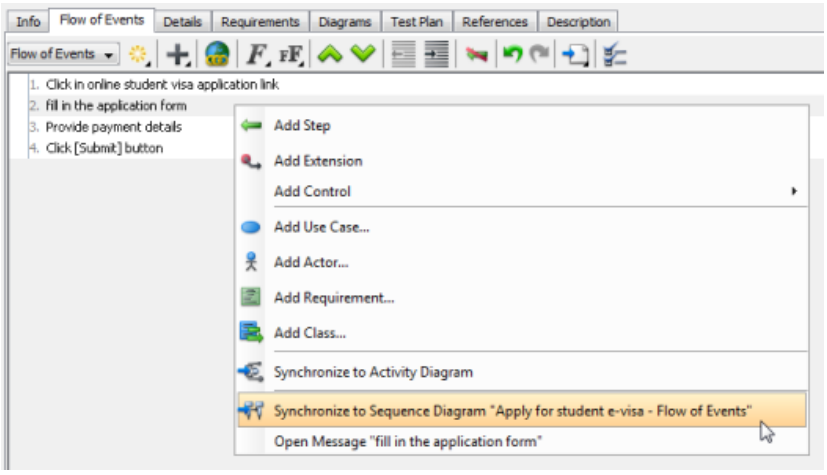
3. Edit the steps in the flow of events editor. When finish editing, right click on the editor and select **Synchronize to Sequence Diagram** from the pop-up menu.



Generate sequence diagram from flow of events

Updating sequence diagram from flow of events

When you have made some changes on flow of events, you can update the changes to sequence diagram accordingly. Right click on the flow of events editor and select synchronize to the name of sequence diagram from the pop-up menu.



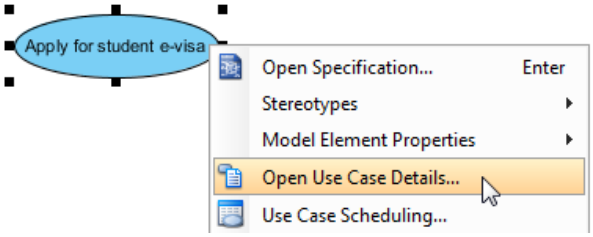
Update changes from flow of events to sequence diagram

Elaborating use case by activity diagram

You can document the events of a use case through the use of flow of events. You can then visualize the events in activity diagram through synchronizing flow of events to activity diagram.

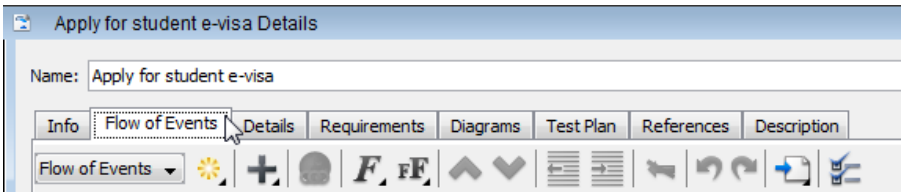
Synchronizing flow of events to activity diagram

1. Right click on the target use case and select **Open Use Case Details...** from the pop-up menu.



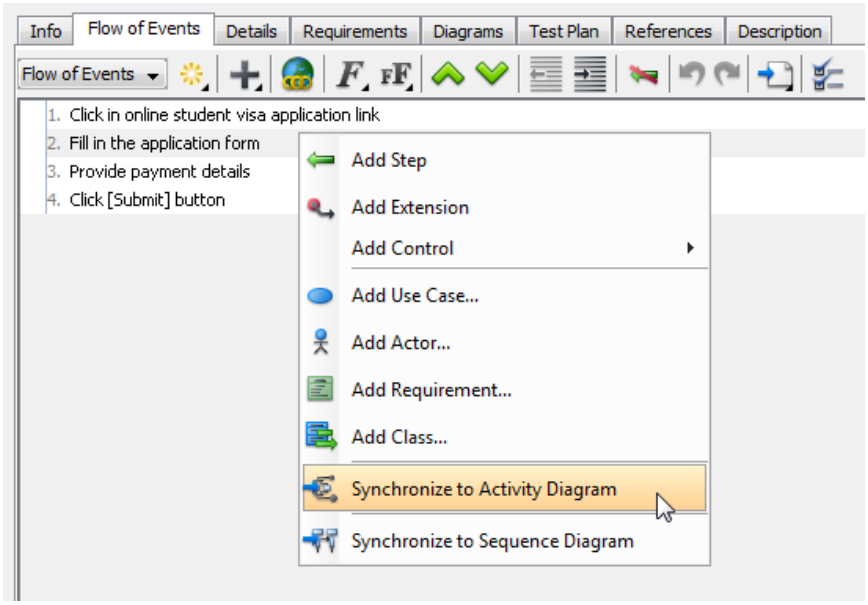
Open use case details

2. When the details dialog box pops out, select the **Flow of Events** tab.



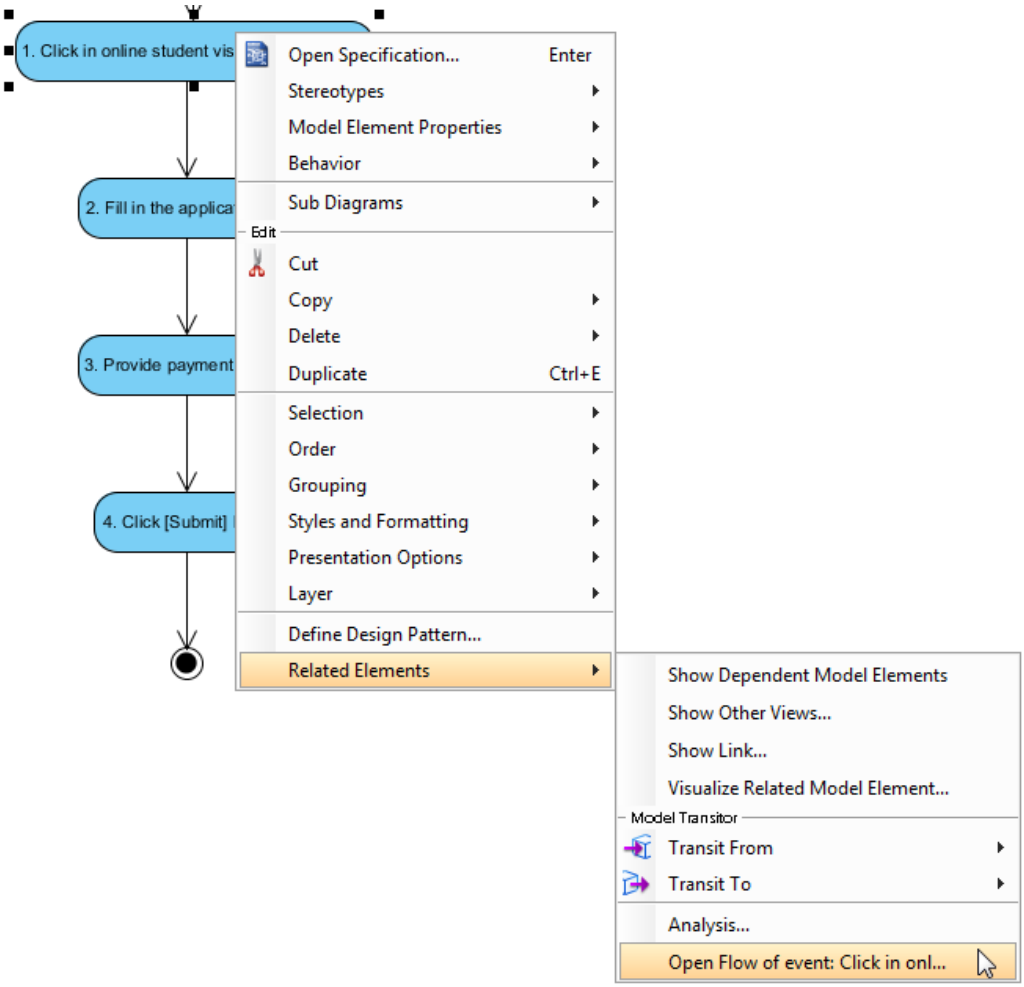
Open flow of events editor

3. Edit the steps in the flow of events editor. When finish editing, right click on the editor and select **Synchronize to Activity Diagram** from the pop-up menu.



Navigating to flow of events

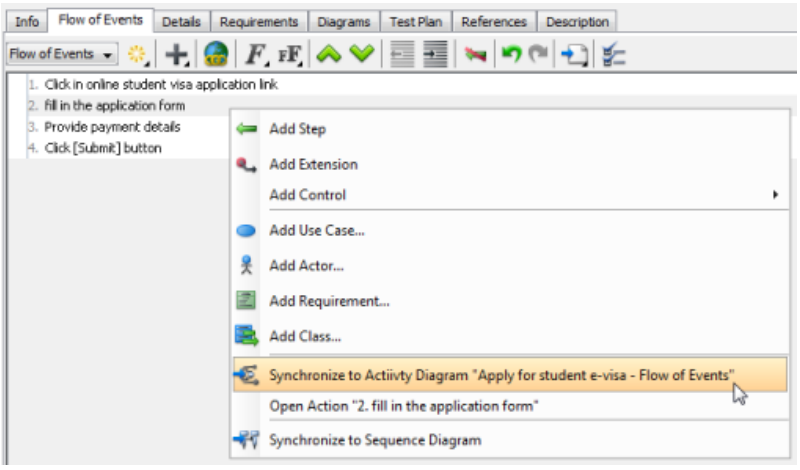
If you want to return to the flow of events, right click on a target action in activity diagram and select **Related Elements > Open Flow of event** from the popup menu.



Open flow of events from activity diagram

Updating activity diagram from flow of events

When you have made some changes on flow of events, you can update the changes to activity diagram accordingly. Right click on the flow of events editor and select synchronize to the name of activity diagram from the pop-up menu.



Update changes from flow of events to activity diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Manage Use Case Scenario with Flow of Events?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Wireframe

What is wireframe?

Learn what wireframe is.

What is a wireframe state?

Learn how to re-use existing wireframe and make small modification by creating wireframe state.

Android phone wireframing skills

Learn how to draw wireframe for Android phones.

Android tablet wireframing skills

Learn how to draw wireframe for Android tablet.

Desktop wireframing skills

Learn how to draw wireframe for desktop applications.

iPad wireframing skills

Learn how to draw wireframe for iPad apps.

iPhone wireframing skills

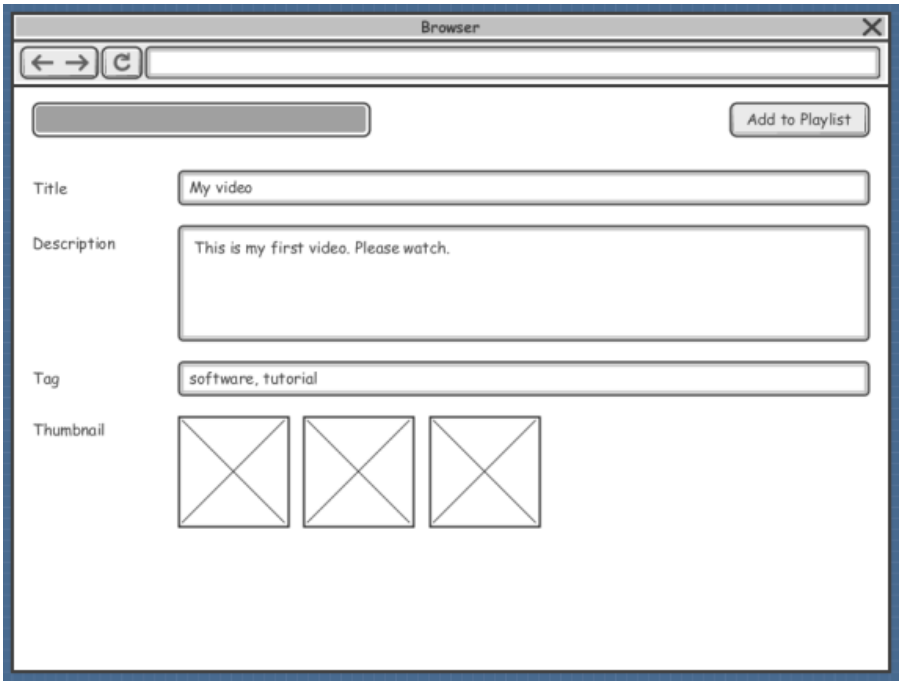
Learn how to draw wireframe for iPhone apps.

Web wireframing skills

Learn how to draw wireframe for web site.

What is wireframe?

A wireframe is a sketch of the system to be built. It’s simple, clear and allows everyone to read and understand easily. Wireframe shows "just enough" information of the screen instead of the full details. The actual screen design will be produced at a later stage by referencing the wireframe. You can show the scenario to your customer visually to obtain consent to the requirements.



Sample wireframe

Benefits of wireframing

Comparing to prototyping or any kind of detailed screen designs, wireframe features the following advantages:

- Easy to draw: Wireframe has a simple and clean layout. It is formed by simple screen elements without any detailed styling and formattings.
- Easy to understand: Wireframe is welcomed by both the development team and business people. It is so simple that everyone can understand without learning required.
- Easy to modify: You don't need any programming to visualize new design ideas.
- No coding required: No heavy prototyping, no coding. You just need to draw the wireframe as if you are using a drawing tool.
- In-line annotations: Annotate design ideas in-place with the help of annotation shapes. These annotations can be shown in requirement specification, too.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

What is a wireframe state?

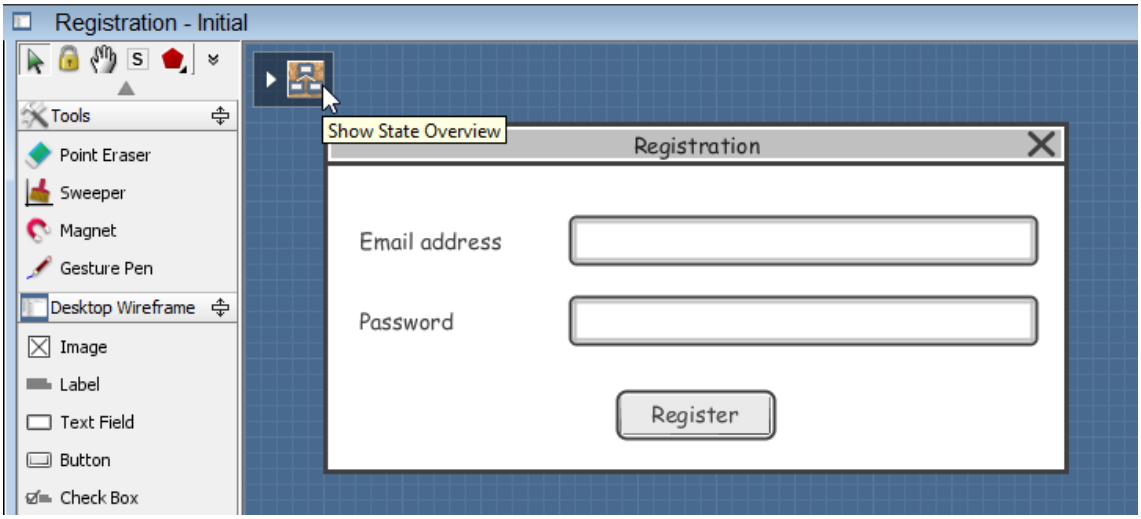
A wireframe state is a snapshot of wireframe throughout an interaction. Generally speaking, a state changes when an event occurs. For example, an entry of text in text field, a click of a button, etc.

Agilian supports wireframing. You can create wireframe, and represent the modifications of screen over time by defining states.

Adding a child wireframe state

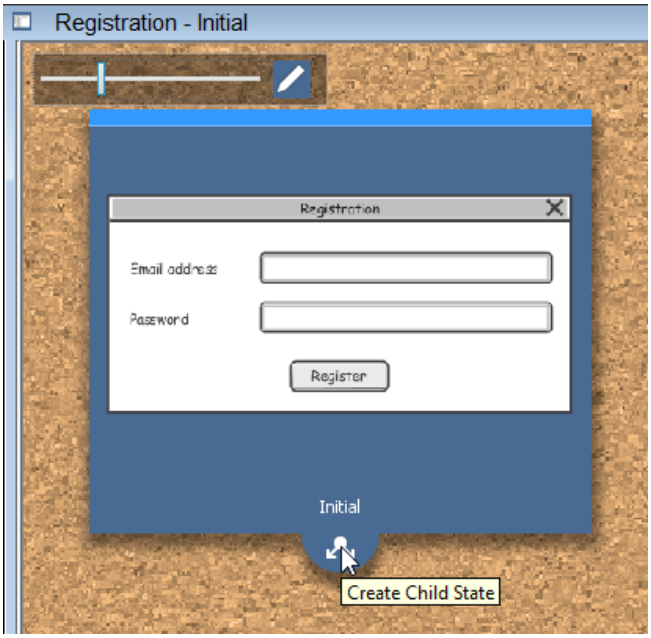
To create a wireframe state:

1. Open the wireframe where you want to add a state.
2. Click on **Show State Overview** at the top left of wireframe.



Show State Overview

3. This opens the state overview, with states placed on a corkboard-like background. Click **Create Child State** below the Initial state to create a child state under it.

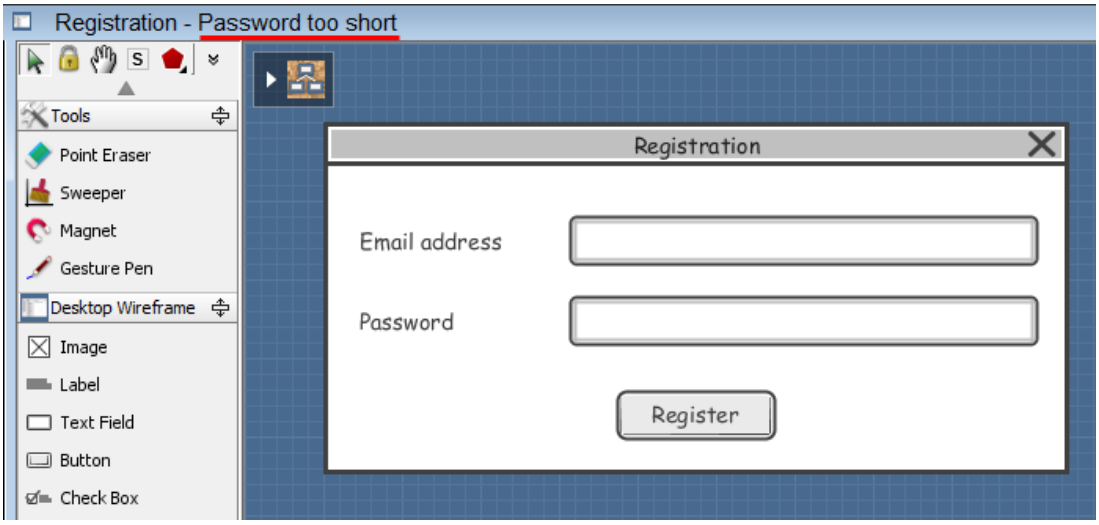


Create child state

4. Name the wireframe state meaningfully to reflect the change of the wireframe at this particular state.

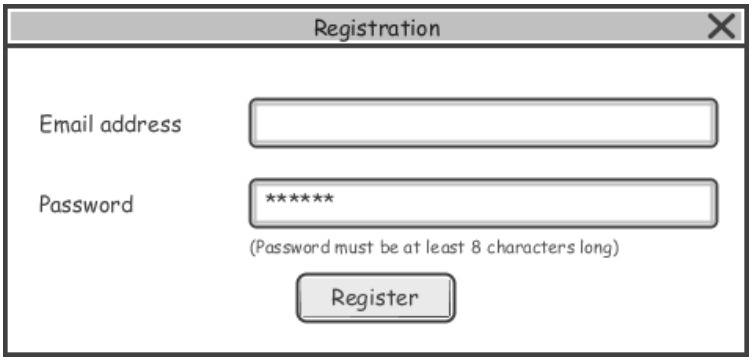
Naming state

5. Press **Enter**. This opens the wireframe state for editing. You can check the editing state by referring to the diagram title.



Check the name of editing state

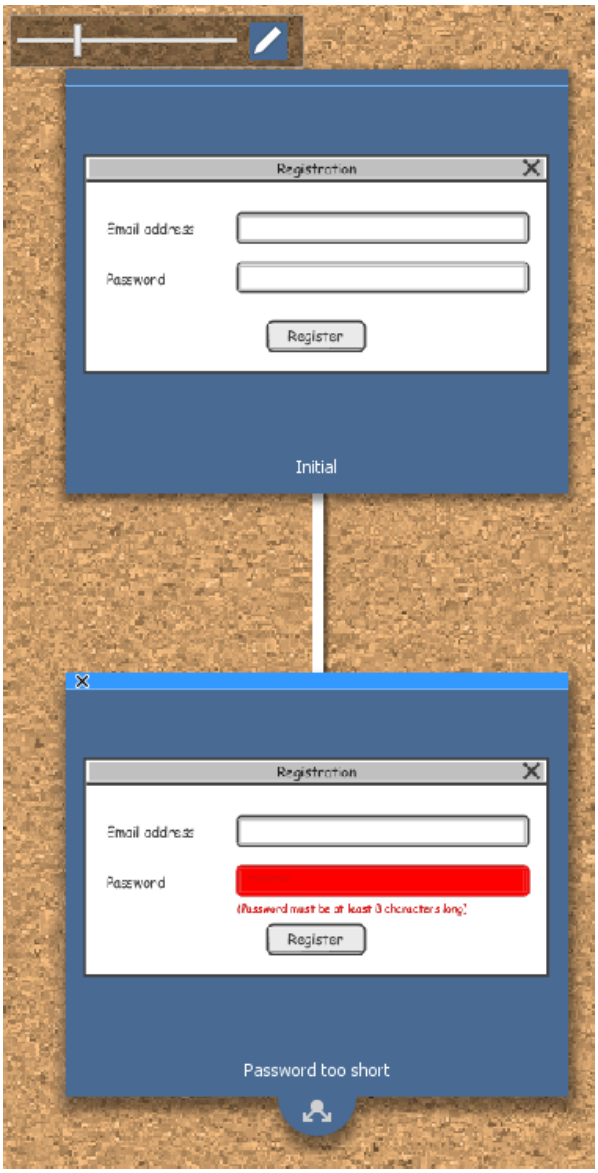
6. You can now start editing.



Wireframe edited

Checking the differences between states

The state overview shows you the thumbnails of wireframe states, plus the differences between parent and child state. The modified regions are paint in red. You can check the red regions to identify the changes made in a particular wireframe state. If you are failed to read the thumbnail clearly, you may double click on a thumbnail to open it, or adjust the zoom ratio by dragging on the slider at the top of the overview.



Differences are highlighted

Synchronization of change between states

Generally speaking, when you make a change in a parent state, like adding a new wireframe element, modifying an element (e.g. enter a text in text box) or removing element, the change apply to all its child state, provided that the modified element has not been modified in the child states. Here are some cases that can help in explaining the idea:

Case 1
If I add a new button in parent state, the same button with appear in its child state.

Case 2
If I removed a button in parent state, and that button has not been modified in its child state, the button will be removed.

Case 3
If I renamed the caption of a button in parent state, and the same button has not been renamed in child state, the caption of the button in child state will be renamed. Note that if the button in child state was resized, renaming will still take place.

Case 4
If I renamed the caption of a button in parent state, and the same button has been renamed in child state, the caption of the button in child state will remain unchanged.

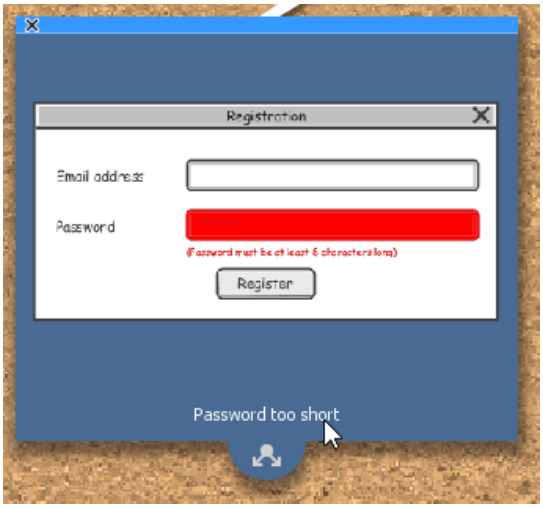
Editing a child wireframe state

To edit a child wireframe state:

1. Open the wireframe where you want to edit a state.
2. Click on **Show State Overview** at the top left of wireframe.
3. Double click on the thumbnail of the state that you want to edit.
4. The wireframe state is opened and you can start editing.

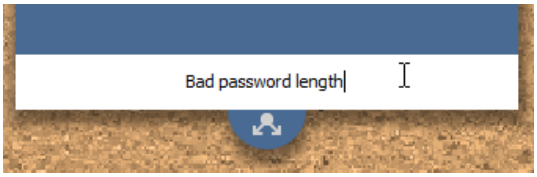
Renameing a wireframe state
To rename a child wireframe state:

1. Open the wireframe where you want to rename a state.
2. Click on **Show State Overview** at the top left of wireframe.
3. Double click on the name under the thumbnail of the state that you want to rename.



Double click on the name of a wireframe state

4. Enter a new name and press **Enter** to confirm editing.

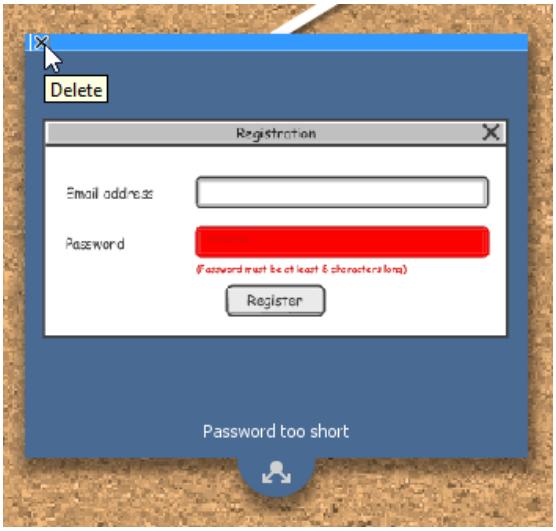


Renaming a wireframe state

Deleting a child wireframe state

By deleting a child wireframe state, you delete not only the chosen state, but also all of its children states. The deletion is not undo-able so think twice before you delete a wireframe state. To delete a child wireframe state

1. Open the wireframe where you want to delete a state.
2. Click on **Show State Overview** at the top left of wireframe.
3. Click on the thumbnail of the state that you want to delete.
4. Click on the tiny cross button at the top left corner.



Delete a wireframe state

Related Resources

The following resources may help you learn more about the topic discussed in this page.

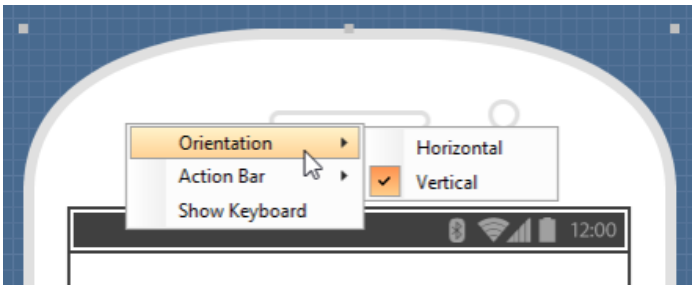
- YouTube Video - How to Create Scenario-Based Wireframe?

- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Android phone wireframing skills

Changing the orientation of Android phone

Initially, the Android phone is shown vertically in the wireframe. If you apps works under a horizontal layout, you can change its orientation. To adjust the orientation, right click on the phone border and select **Orientation > Horizontal/Vertical** from the popup menu.

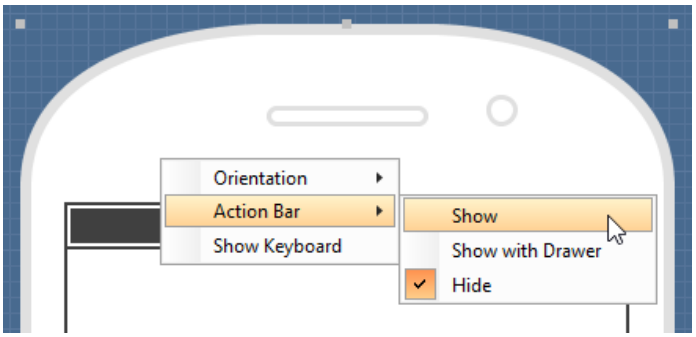


Changing orientation of Android phone

NOTE: You can only change orientation when there is no wireframe element created inside the phone body

Show/Hide action bar

To hide the action bar and toolbar, right click on the phone border and select **Action Bar > Show** or **Action Bar > Show with Drawer** from the popup menu.



Show action bar

Show/Hide keyboard

To show the keyboard, right click on the phone border and select **Show Keyboard** from the popup menu.

Show keyboard

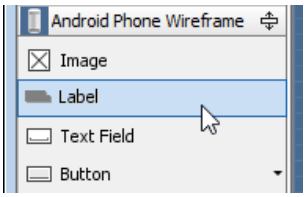
This shows the keybaord at the bottom of the phone:

Keyboard shown

Creating a wireframe element

Method 1 - Diagram toolbar: Select-and-Click

1. Select the desired wireframe element from the diagram toolbar (e.g. Label).

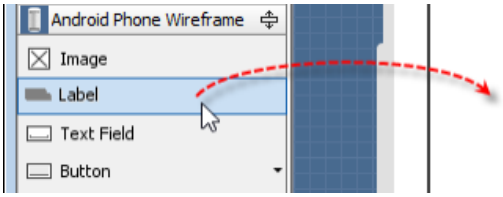


Create a label by selecting it from the diagram toolbar

2. Click on the wireframe, at the position where you want the wireframe element to be created.

Method 2 - Diagram toolbar: Drag and drop

1. Press on the desired wireframe element in the diagram toolbar
2. Hold the mouse button.
3. Drag to the wireframe.

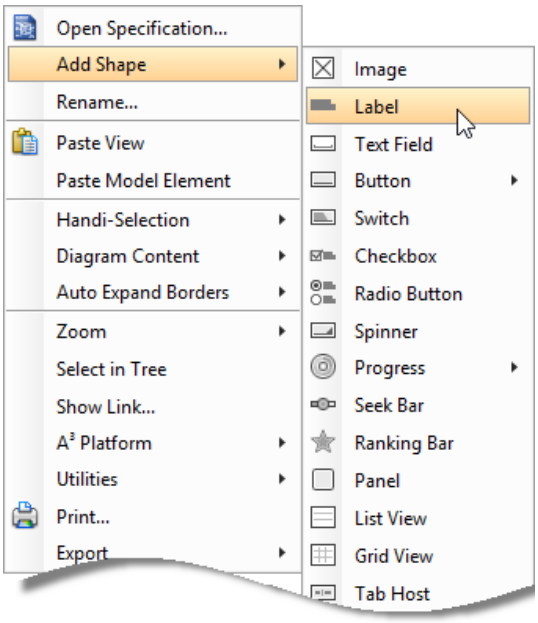


Create a Label with drag-and-drop

4. Release the mouse button in the wireframe, at the position where you want the wireframe element to be created.

Method 3 - Popup menu

1. Right click on the wireframe, at the position where.
2. Select **Add Shape > %SHAPE_TYPE%** from the popup menu, where **%SHAPE_TYPE%** is the kind of wireframe element you want to create.



Creating a wireframe element via the popup menu

Method 4 - Through smart create resource

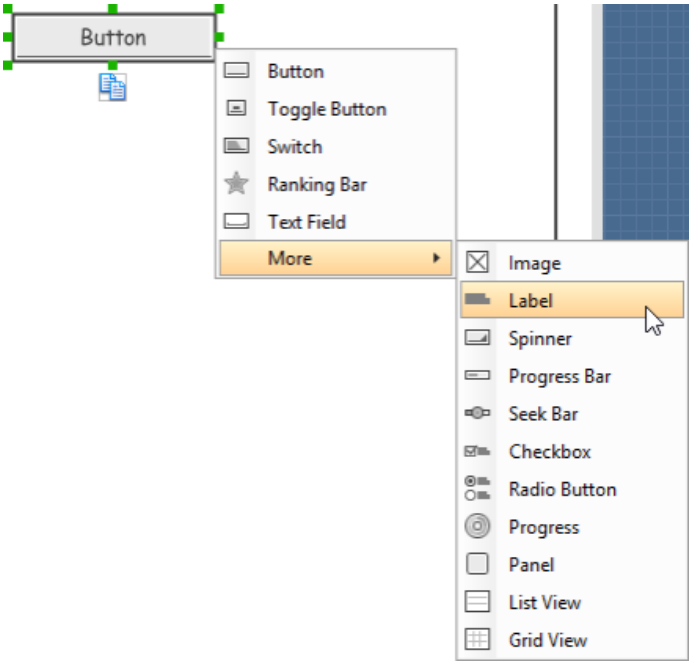
1. Click directly on the wireframe, at the position where you want the wireframe element to be created. You should see a green icon appear, known as the **Smart Create** resource.

Creating a wireframe element using Smart Resource

2. Press on the **Smart Create** resource and hold the mouse button.
3. Drag to outline the size of the wireframe element to be created.

Creating a label in specific size

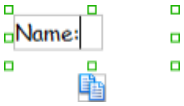
4. Release the mouse button. In the popup menu, choose the type of wireframe element to create.



Choosing the wireframe element to be created

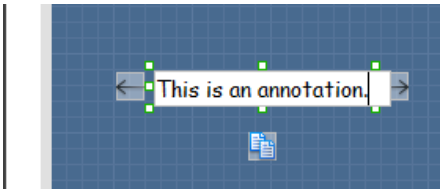
Method 5 - Double-clicking (Label and annotation only)

To create a label, double-click on the wireframe and enter the label caption.



Creating a label

To create an annotation, double-click outside the phone border and enter the annotation text.



Entering annotation text

Accurate positioning of wireframe element using the alignment guide

Alignment guide is a dotted line that appears when you move wireframe elements in a wireframe. It helps you align elements perfectly with others. Simple select element(s) and drag it around. When the selection approaches another element in the wireframe, you can adjusting the positioning of selection with the help of the guide.

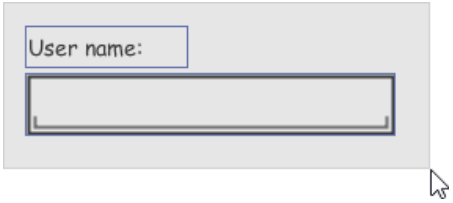
Using the alignment guide

Duplicating wireframe elements

Duplicate wireframe elements enable you to create new elements base on existing ones. This saves you a lot of time in creating elements with same/ similar style, size and content to the existine ones.

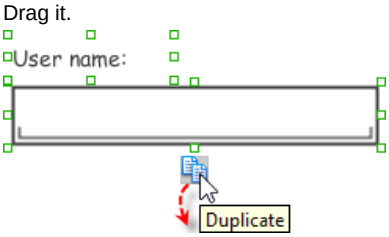
To duplicate wireframe elements:

1. Drag in the wireframe to select a range of elements to duplicate.



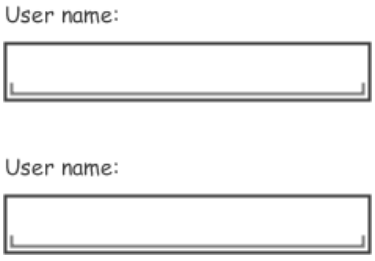
Selecting wireframe elements to duplicate

2. At the bottom of your selection, press on the **Duplicate** resource icon and hold the mouse button.
3. Drag it.



To duplicate wireframe elements

4. Release the mouse button at the position where you want the wireframe element to be created.



Wireframe elements duplicate

5. Touch-up the duplicate elements.

Renamed a label

NOTE: Instead of dragging the **Duplicate** resource, you may click on it, too..

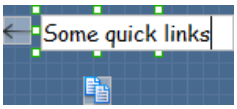
Annotating wireframe with Annotation shape

The use of annotations in wireframe allows you to detail the elements on the wireframe. With annotation, you can describe or explain the existence of certain wireframe element, as well as to describe the calls to action and the expected results.

In order to keep the wireframe content readable, annotations are forced to put outside the phone border. In other words, you cannot create or move an annotation to inside the phone body.

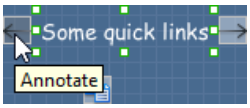
To create an annotation:

1. Double click on the background (i.e. the blue region) of the wireframe.
2. Enter the annotation text.



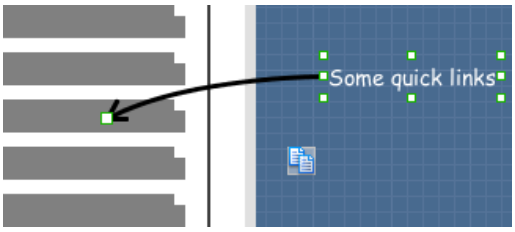
Entering annotation text

3. Press on the arrow resource and hold the mouse button.



To annotate a wireframe element

4. Drag to the wireframe element to annotate it.



Label annotated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

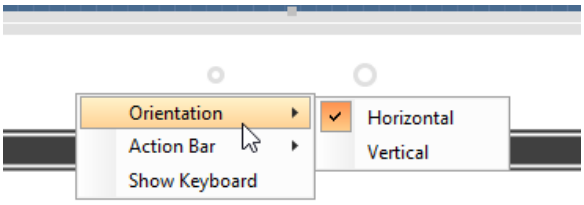
Trademark Disclaimer

- Android is a trademark of Google Inc.

Android tablet wireframing skills

Changing the orientation of Android tablet

Initially, the Android tablet is shown horizontally in the wireframe. If you apps works under a vertical layout, you can change its orientation. To adjust the orientation, right click on the tablet border and select **Orientation > Horizontal/Vertical** from the popup menu.

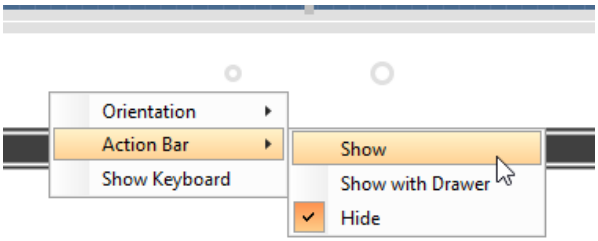


Changing orientation of Android tablet

NOTE: You can only change orientation when there is no wireframe element created inside the tablet

Show/Hide action bar

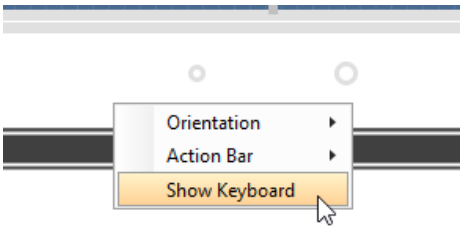
To hide the action bar and toolbar, right click on the tablet border and select **Action Bar > Show** or **Action Bar > Show with Drawer** from the popup menu.



Show action bar

Show/Hide keyboard

To show the keyboard, right click on the tablet border and select **Show Keyboard** from the popup menu.



Show keyboard

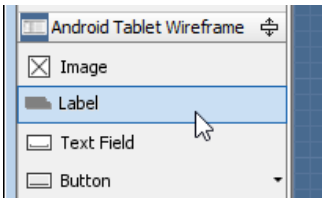
This shows the keybaord at the bottom of the phone:

Keyboard shown

Creating a wireframe element

Method 1 - Diagram toolbar: Select-and-Click

1. Select the desired wireframe element from the diagram toolbar (e.g. Label).

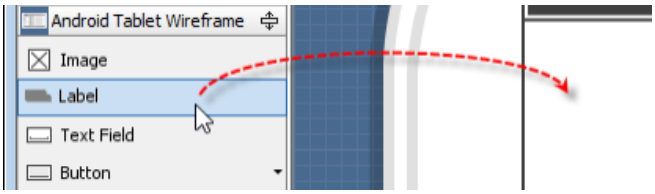


Create a label by selecting it from the diagram toolbar

2. Click on the wireframe, at the position where you want the wireframe element to be created.

Method 2 - Diagram toolbar: Drag and drop

1. Press on the desired wireframe element in the diagram toolbar
2. Hold the mouse button.
3. Drag to the wireframe.

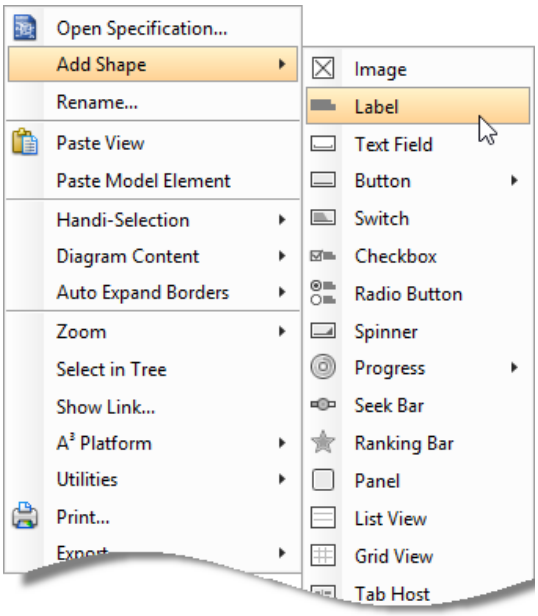


Create a Label with drag-and-drop

4. Release the mouse button in the wireframe, at the position where you want the wireframe element to be created.

Method 3 - Popup menu

1. Right click on the wireframe, at the position where.
2. Select **Add Shape > %SHAPE_TYPE%** from the popup menu, where **%SHAPE_TYPE%** is the kind of wireframe element you want to create.



Creating a wireframe element via the popup menu

Method 4 - Through smart create resource

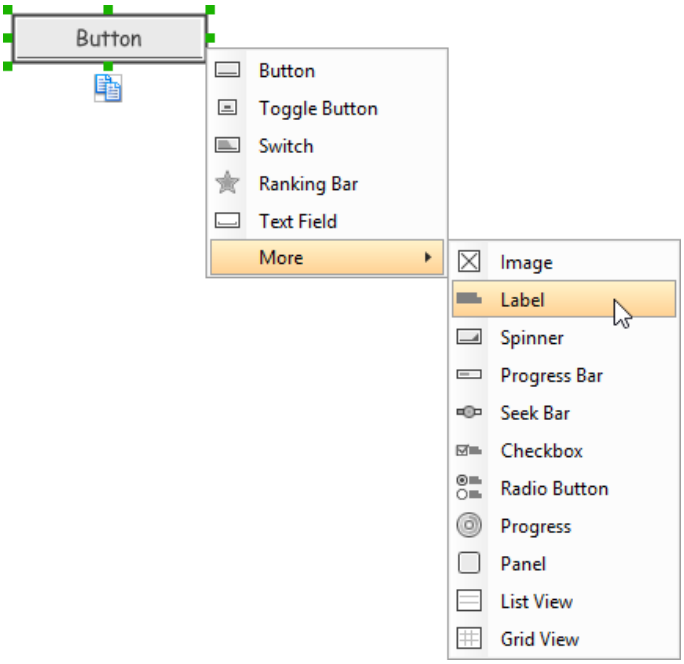
1. Click directly on the wireframe, at the position where you want the wireframe element to be created. You should see a green icon appear, known as the **Smart Create** resource.

Creating a wireframe element using Smart Resource

2. Press on the **Smart Create** resource and hold the mouse button.
3. Drag to outline the size of the wireframe element to be created.

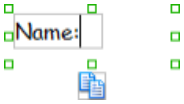
Creating a label in specific size

4. Release the mouse button. In the popup menu, choose the type of wireframe element to create.



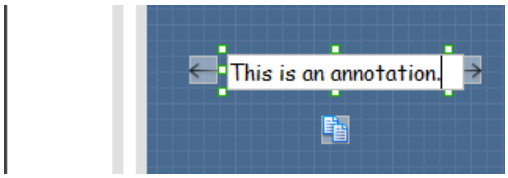
Choosing the wireframe element to be created

Method 5 - Double-clicking (Label and annotation only)
To create a label, double-click on the wireframe and enter the label caption.



Creating a label

To create an annotation, double-click outside the tablet and enter the annotation text.



Entering annotation text

Accurate positioning of wireframe element using the alignment guide

Alignment guide is a dotted line that appears when you move wireframe elements in a wireframe. It helps you align elements perfectly with others. Simple select element(s) and drag it around. When the selection approaches another element in the wireframe, you can adjusting the positioning of selection with the help of the guide.

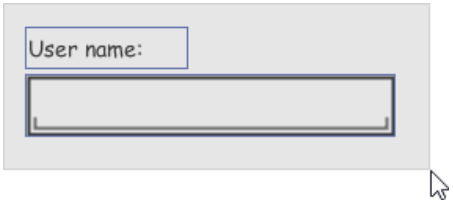
Using the alignment guide

Duplicating wireframe elements

Duplicate wireframe elements enable you to create new elements base on existing ones. This saves you a lot of time in creating elements with same/ similar style, size and content to the existine ones.

To duplicate wireframe elements:

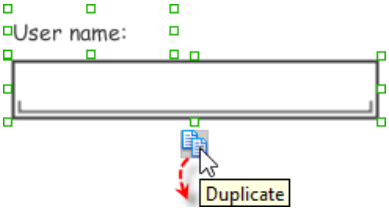
1. Drag in the wireframe to select a range of elements to duplicate.



Selecting wireframe elements to duplicate

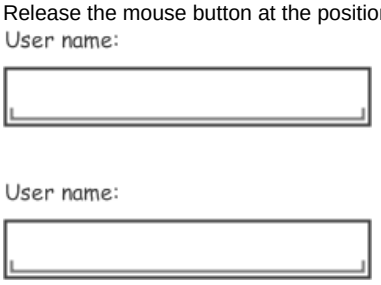
2. At the bottom of your selection, press on the **Duplicate** resource icon and hold the mouse button.

3. Drag it.



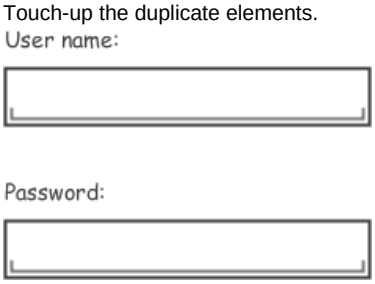
To duplicate wireframe elements

4. Release the mouse button at the position where you want the wireframe element to be created.



Wireframe elements duplicate

5. Touch-up the duplicate elements.



Renamed a label

NOTE: Instead of dragging the **Duplicate** resource, you may click on it, too..

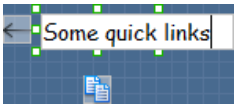
Annotating wireframe with Annotation shape

The use of annotations in wireframe allows you to detail the elements on the wireframe. With annotation, you can describe or explain the existence of certain wireframe element, as well as to describe the calls to action and the expected results.

In order to keep the wireframe content readable, annotations are forced to put outside the tablet. In other words, you cannot create or move an annotation to inside the tablet.

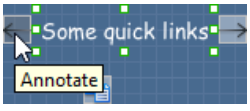
To create an annotation:

1. Double click on the background (i.e. the blue region) of the wireframe.
2. Enter the annotation text.



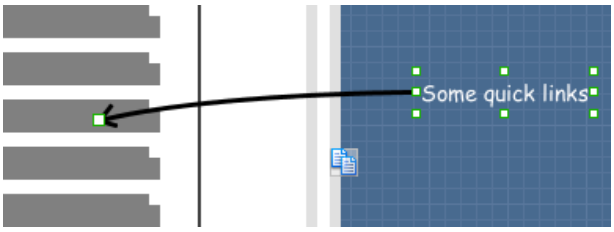
Entering annotation text

3. Press on the arrow resource and hold the mouse button.



To annotate a wireframe element

4. Drag to the wireframe element to annotate it.



Label annotated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Trademark Disclaimer

- Android is a trademark of Google Inc.

Desktop wireframing skills

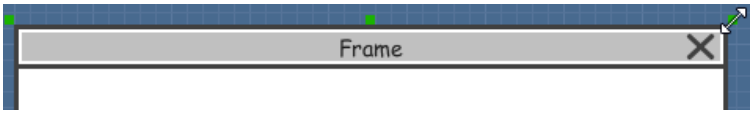
Adjust the size of frame

If you find the frame too large or too small, you can resize it to your preferred size. To adjust the size of frame:

1. Click on the frame's title bar.

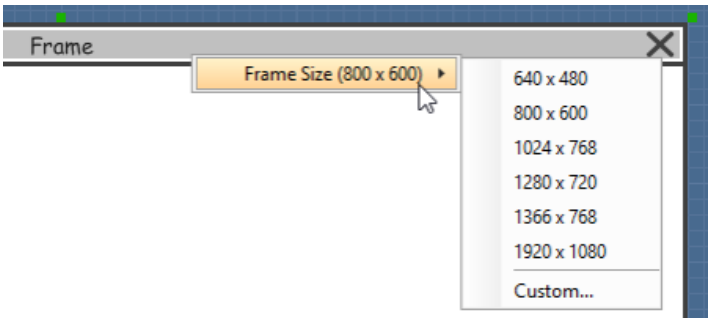
Clicking on frame's title bar

2. This shows the resize handlers at the corners and edges of the frame. You can drag on them to resize the frame.



Resizing frame

Alternatively, you can right click on the frame's title bar and select **Browser Size > %PREFERRED_SIZE%** from the popup menu.

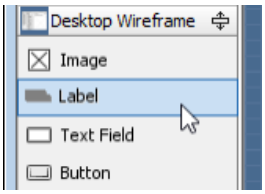


Adjusting browser window size

Creating a wireframe element

Method 1 - Diagram toolbar: Select-and-Click

1. Select the desired wireframe element from the diagram toolbar (e.g. Label).

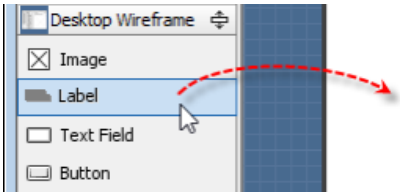


Create a label by selecting it from the diagram toolbar

2. Click on the wireframe, at the position where you want the wireframe element to be created.

Method 2 - Diagram toolbar: Drag and drop

1. Press on the desired wireframe element in the diagram toolbar
2. Hold the mouse button.
3. Drag to the wireframe.

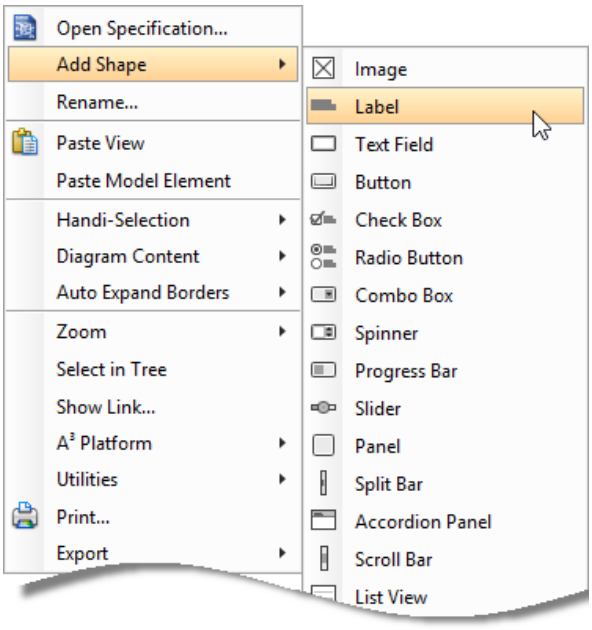


Create a Label with drag-and-drop

4. Release the mouse button in the wireframe, at the position where you want the wireframe element to be created.

Method 3 - Popup menu

1. Right click on the wireframe, at the position where.
2. Select **Add Shape > %SHAPE_TYPE%** from the popup menu, where **%SHAPE_TYPE%** is the kind of wireframe element you want to create.



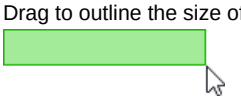
Creating a wireframe element via the popup menu

Method 4 - Through smart create resource

1. Click directly on the wireframe, at the position where you want the wireframe element to be created. You should see a green icon appear, known as the **Smart Create** resource.

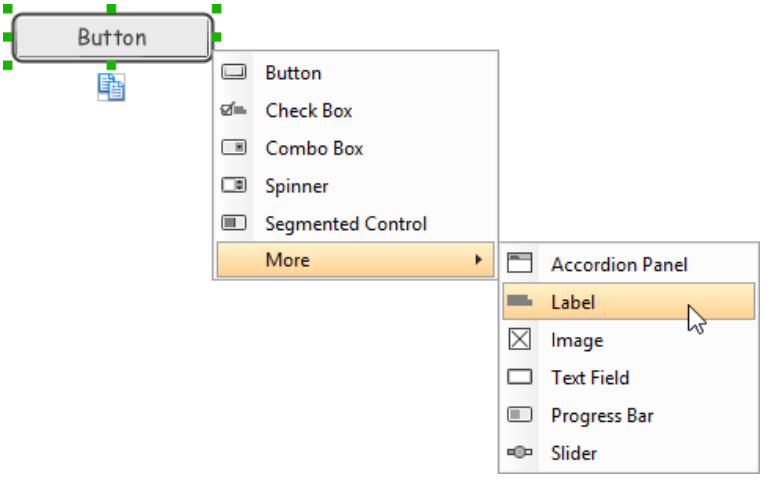
Creating a wireframe element using Smart Resource

2. Press on the **Smart Create** resource and hold the mouse button.
3. Drag to outline the size of the wireframe element to be created.



Creating a label in specific size

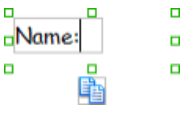
4. Release the mouse button. In the popup menu, choose the type of wireframe element to create.



Choosing the wireframe element to be created

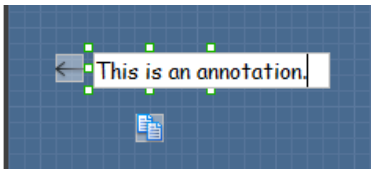
Method 5 - Double-clicking (Label and annotation only)

To create a label, double-click on the wireframe and enter the label caption.



Creating a label

To create an annotation, double-click outside the frame and enter the annotation text.



Entering annotation text

Accurate positioning of wireframe element using the alignment guide

Alignment guide is a dotted line that appears when you move wireframe elements in a wireframe. It helps you align elements perfectly with others. Simple select element(s) and drag it around. When the selection approaches another element in the wireframe, you can adjusting the positioning of selection with the help of the guide.

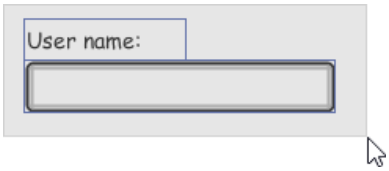
Using the alignment guide

Duplicating wireframe elements

Duplicate wireframe elements enable you to create new elements base on existing ones. This saves you a lot of time in creating elements with same/ similar style, size and content to the existine ones.

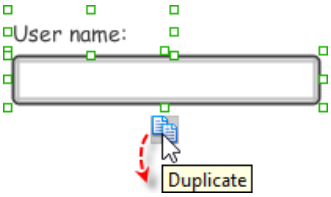
To duplicate wireframe elements:

1. Drag in the wireframe to select a range of elements to duplicate.



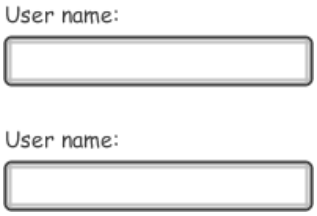
Selecting wireframe elements to duplicate

2. At the bottom of your selection, press on the **Duplicate** resource icon and hold the mouse button.
3. Drag it.



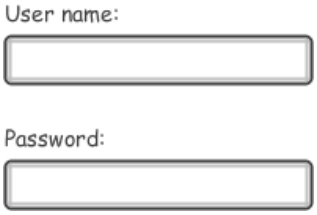
To duplicate wireframe elements

4. Release the mouse button at the position where you want the wireframe element to be created.



Wireframe elements duplicate

5. Touch-up the duplicate elements.



Renamed a label

NOTE: Instead of dragging the **Duplicate** resource, you may click on it, too..

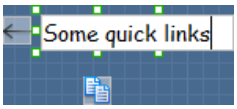
Annotating wireframe with Annotation shape

The use of annotations in wireframe allows you to detail the elements on the wireframe. With annotation, you can describe or explain the existence of certain wireframe element, as well as to describe the calls to action and the expected results.

In order to keep the wireframe content readable, annotations are forced to put outside the frame. In other words, you cannot create or move an annotation to inside the frame area.

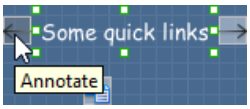
To create an annotation:

1. Double click on the background (i.e. the blue region) of the wireframe.
2. Enter the annotation text.



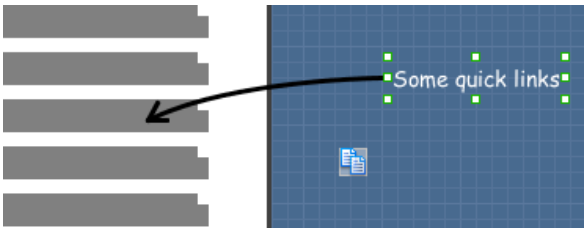
Entering annotation text

3. Press on the arrow resource and hold the mouse button.



To annotate a wireframe element

4. Drag to the wireframe element to annotate it.



Label annotated

Related Resources

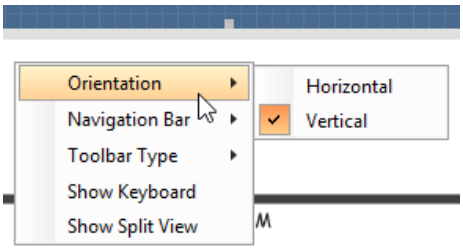
The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

iPad wireframing skills

Changing the orientation of iPad

Initially, iPad is shown horizontally in the wireframe. If you apps works under a vertical layout, you can change its orientation. To adjust the orientation, right click on the iPad border and select **Orientation > Horizontal/Vertical** from the popup menu.

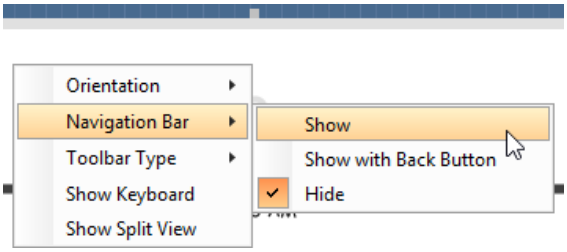


Changing orientation of iPad

NOTE: You can only change orientation when there is no wireframe element created inside iPad

Show/Hide navigation bar

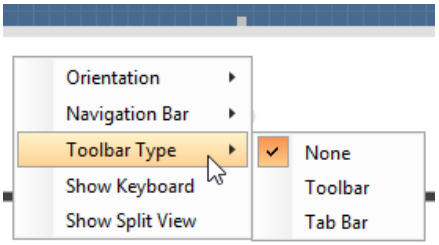
To hide the navigation bar, right click on the iPad border and select **Navigation Bar > Show** or **Navigation Bar > Show with Back Button** from the popup menu.



Show navigation bar

Show/Hide toolbar and tab bar

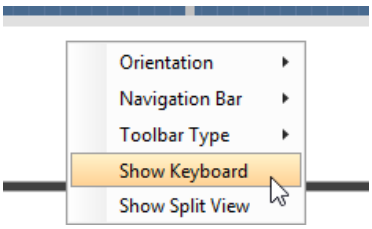
To hide the toolbar and tab toolbar, right click on the iPad border and select **Toolbar Type > Toolbar** or **Toolbar Type > Tab Bar** from the popup menu.



Show toolbar bar

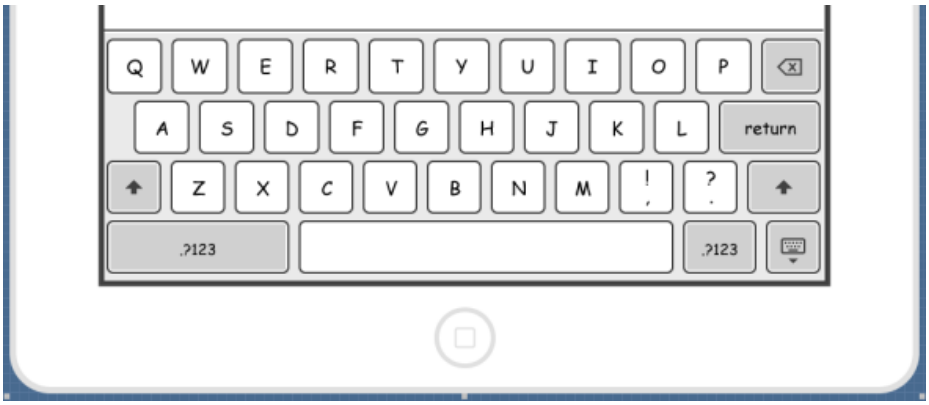
Show/Hide keyboard

To show the keyboard, right click on the iPad border and select **Show Keyboard** from the popup menu.



Show keyboard

This shows the keybaord at the bottom of iPad:

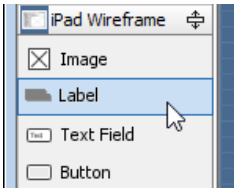


Keyboard shown

Creating a wireframe element

Method 1 - Diagram toolbar: Select-and-Click

1. Select the desired wireframe element from the diagram toolbar (e.g. Label).

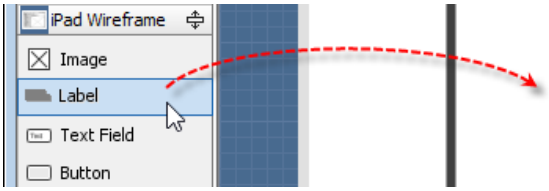


Create a label by selecting it from the diagram toolbar

2. Click on the wireframe, at the position where you want the wireframe element to be created.

Method 2 - Diagram toolbar: Drag and drop

1. Press on the desired wireframe element in the diagram toolbar
2. Hold the mouse button.
3. Drag to the wireframe.

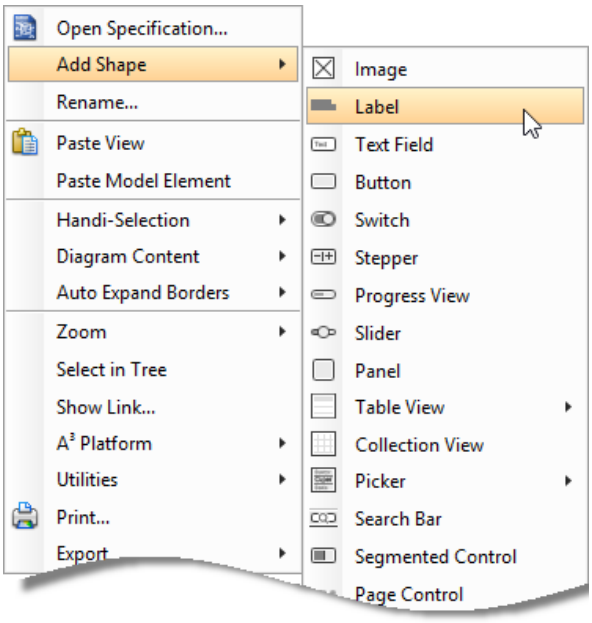


Create a Label with drag-and-drop

4. Release the mouse button in the wireframe, at the position where you want the wireframe element to be created.

Method 3 - Popup menu

1. Right click on the wireframe, at the position where.
2. Select **Add Shape > %SHAPE_TYPE%** from the popup menu, where **%SHAPE_TYPE%** is the kind of wireframe element you want to create.



Creating a wireframe element via the popup menu

Method 4 - Through smart create resource

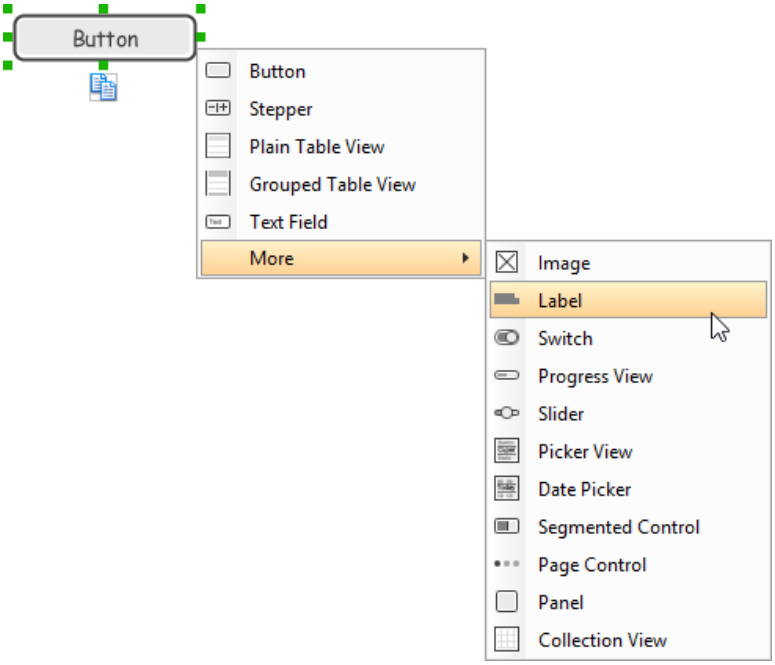
1. Click directly on the wireframe, at the position where you want the wireframe element to be created. You should see a green icon appear, known as the **Smart Create** resource.

Creating a wireframe element using Smart Resource

2. Press on the **Smart Create** resource and hold the mouse button.
3. Drag to outline the size of the wireframe element to be created.

Creating a label in specific size

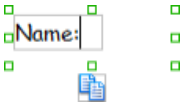
4. Release the mouse button. In the popup menu, choose the type of wireframe element to create.



Choosing the wireframe element to be created

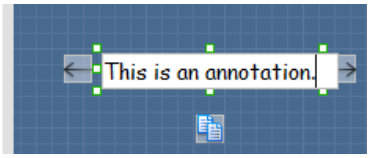
Method 5 - Double-clicking (Label and annotation only)

To create a label, double-click on the wireframe and enter the label caption.



Creating a label

To create an annotation, double-click outside the iPad and enter the annotation text.



Entering annotation text

Accurate positioning of wireframe element using the alignment guide

Alignment guide is a dotted line that appears when you move wireframe elements in a wireframe. It helps you align elements perfectly with others. Simple select element(s) and drag it around. When the selection approaches another element in the wireframe, you can adjusting the positioning of selection with the help of the guide.

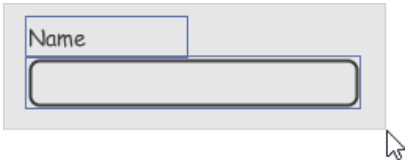
Using the alignment guide

Duplicating wireframe elements

Duplicate wireframe elements enable you to create new elements base on existing ones. This saves you a lot of time in creating elements with same/ similar style, size and content to the existine ones.

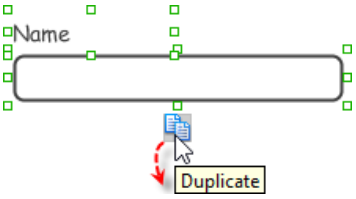
To duplicate wireframe elements:

1. Drag in the wireframe to select a range of elements to duplicate.



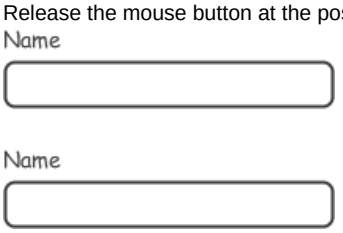
Selecting wireframe elements to duplicate

2. At the bottom of your selection, press on the **Duplicate** resource icon and hold the mouse button.
3. Drag it.



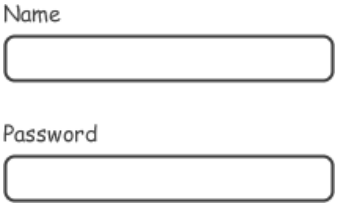
To duplicate wireframe elements

4. Release the mouse button at the position where you want the wireframe element to be created.



Wireframe elements duplicate

5. Touch-up the duplicate elements.



Renamed a label

NOTE: Instead of dragging the **Duplicate** resource, you may click on it, too..

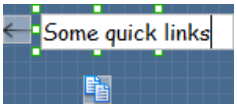
Annotating wireframe with Annotation shape

The use of annotations in wireframe allows you to detail the elements on the wireframe. With annotation, you can describe or explain the existence of certain wireframe element, as well as to describe the calls to action and the expected results.

In order to keep the wireframe content readable, annotations are forced to put outside the iPad. In other words, you cannot create or move an annotation to inside the iPad.

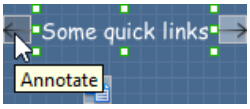
To create an annotation:

1. Double click on the background (i.e. the blue region) of the wireframe.
2. Enter the annotation text.



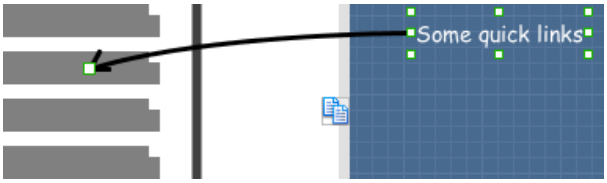
Entering annotation text

3. Press on the arrow resource and hold the mouse button.



To annotate a wireframe element

4. Drag to the wireframe element to annotate it.



Label annotated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

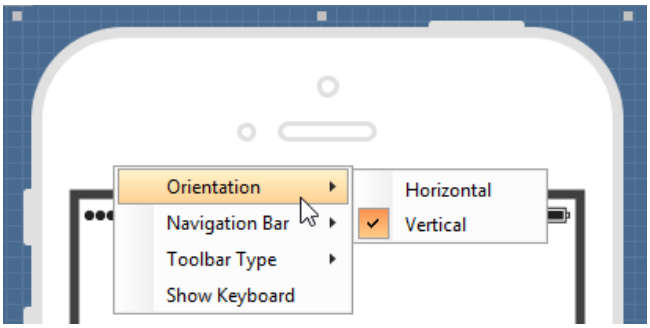
Trademark Disclaimer

- iPad is a trademark of Apple Inc.

iPhone wireframing skills

Changing the orientation of iPhone

Initially, iPhone is shown vertically in the wireframe. If you apps works under a horizontal layout, you can change its orientation. To adjust the orientation, right click on the phone border and select **Orientation > Horizontal/Vertical** from the popup menu.

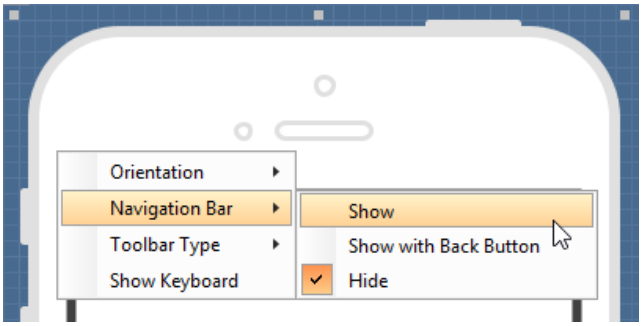


Changing orientation of iPhone

NOTE: You can only change orientation when there is no wireframe element created inside the phone body

Show/Hide navigation bar

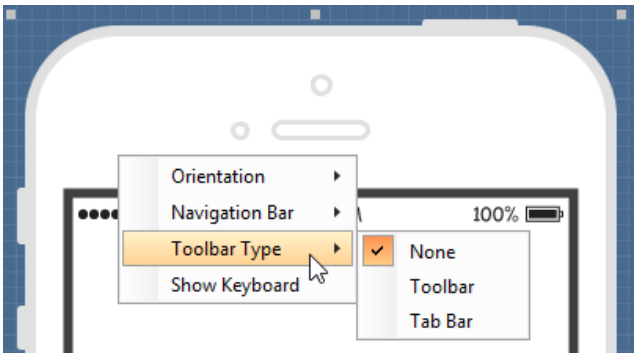
To hide the navigation bar, right click on the phone border and select **Navigation Bar > Show** or **Navigation Bar > Show with Back Button** from the popup menu.



Show navigation bar

Show/Hide toolbar and tab bar

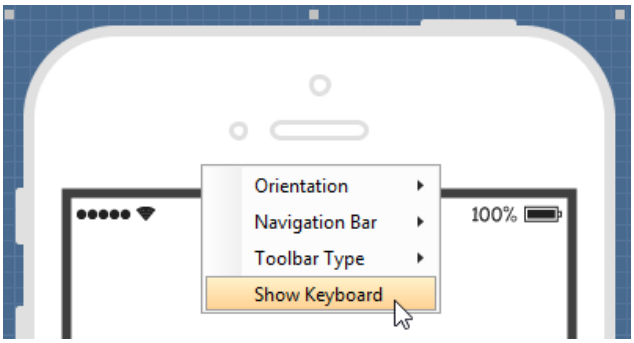
To hide the toolbar and tab toolbar, right click on the phone border and select **Toolbar Type > Toolbar** or **Toolbar Type > Tab Bar** from the popup menu.



Show toolbar bar

Show/Hide keyboard

To show the keyboard, right click on the phone border and select **Show Keyboard** from the popup menu.



Show keyboard

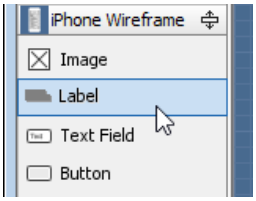
This shows the keybaord at the bottom of the phone:

Keyboard shown

Creating a wireframe element

Method 1 - Diagram toolbar: Select-and-Click

1. Select the desired wireframe element from the diagram toolbar (e.g. Label).

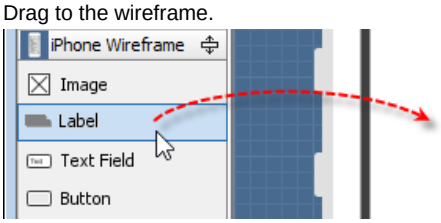


Create a label by selecting it from the diagram toolbar

2. Click on the wireframe, at the position where you want the wireframe element to be created.

Method 2 - Diagram toolbar: Drag and drop

1. Press on the desired wireframe element in the diagram toolbar
2. Hold the mouse button.
3. Drag to the wireframe.



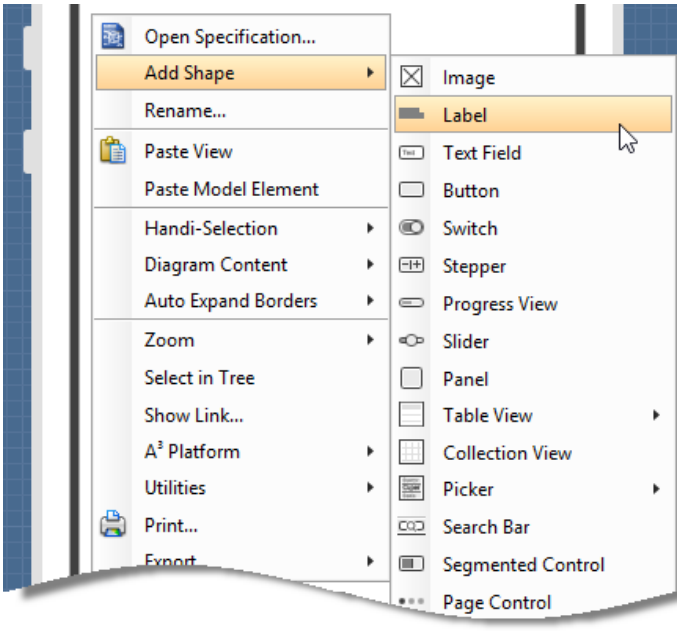
Create a Label with drag-and-drop

4. Release the mouse button in the wireframe, at the position where you want the wireframe element to be created.

Method 3 - Popup menu

1. Right click on the wireframe, at the position where.

2. Select **Add Shape > %SHAPE_TYPE%** from the popup menu, where **%SHAPE_TYPE%** is the kind of wireframe element you want to create.



Creating a wireframe element via the popup menu

Method 4 - Through smart create resource

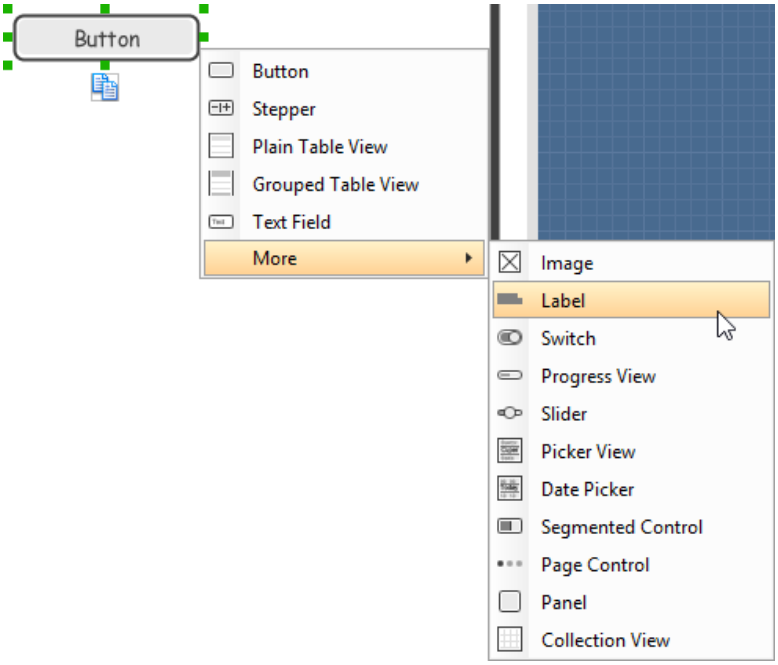
1. Click directly on the wireframe, at the position where you want the wireframe element to be created. You should see a green icon appear, known as the **Smart Create** resource.

Creating a wireframe element using Smart Resource

2. Press on the **Smart Create** resource and hold the mouse button.
3. Drag to outline the size of the wireframe element to be created.

Creating a label in specific size

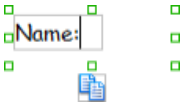
4. Release the mouse button. In the popup menu, choose the type of wireframe element to create.



Choosing the wireframe element to be created

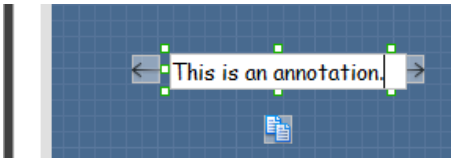
Method 5 - Double-clicking (Label and annotation only)

To create a label, double-click on the wireframe and enter the label caption.



Creating a label

To create an annotation, double-click outside the phone border and enter the annotation text.



Entering annotation text

Accurate positioning of wireframe element using the alignment guide

Alignment guide is a dotted line that appears when you move wireframe elements in a wireframe. It helps you align elements perfectly with others. Simple select element(s) and drag it around. When the selection approaches another element in the wireframe, you can adjusting the positioning of selection with the help of the guide.

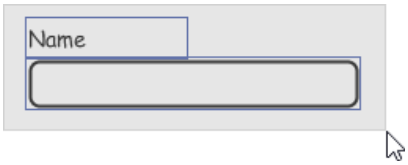
Using the alignment guide

Duplicating wireframe elements

Duplicate wireframe elements enable you to create new elements base on existing ones. This saves you a lot of time in creating elements with same/ similar style, size and content to the existine ones.

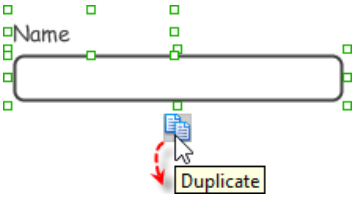
To duplicate wireframe elements:

1. Drag in the wireframe to select a range of elements to duplicate.



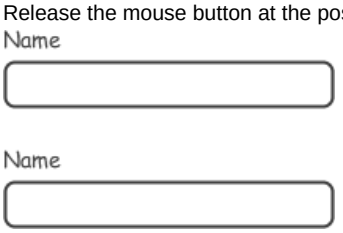
Selecting wireframe elements to duplicate

2. At the bottom of your selection, press on the **Duplicate** resource icon and hold the mouse button.
3. Drag it.



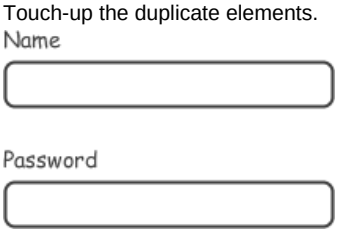
To duplicate wireframe elements

4. Release the mouse button at the position where you want the wireframe element to be created.



Wireframe elements duplicate

5. Touch-up the duplicate elements.



Renamed a label

NOTE: Instead of dragging the **Duplicate** resource, you may click on it, too..

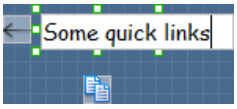
Annotating wireframe with Annotation shape

The use of annotations in wireframe allows you to detail the elements on the wireframe. With annotation, you can describe or explain the existence of certain wireframe element, as well as to describe the calls to action and the expected results.

In order to keep the wireframe content readable, annotations are forced to put outside the phone border. In other words, you cannot create or move an annotation to inside the phone body.

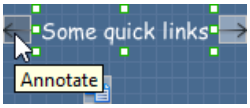
To create an annotation:

1. Double click on the background (i.e. the blue region) of the wireframe.
2. Enter the annotation text.



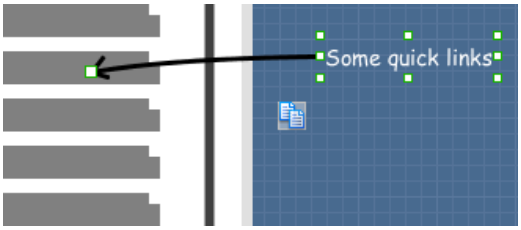
Entering annotation text

3. Press on the arrow resource and hold the mouse button.



To annotate a wireframe element

4. Drag to the wireframe element to annotate it.



Label annotated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Trademark Disclaimer

- iPhone is a trademark of Apple Inc.

Web wireframing skills

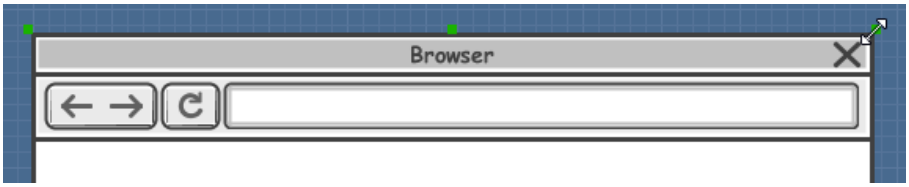
Adjust the size of browser window

If you find the browser window too large or too small, you can resize it to your preferred size. To adjust the size of browser window:

1. Click on the browser window's title bar.

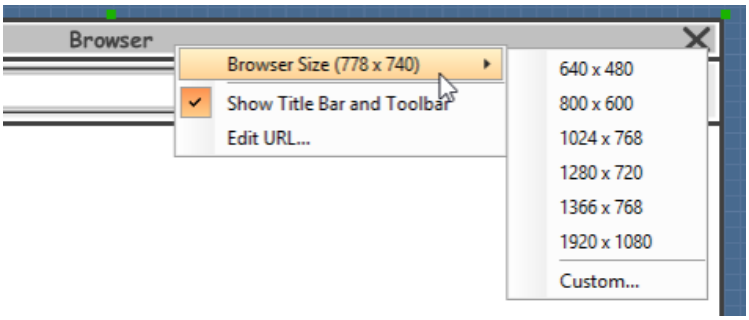
Clicking on browser window's title bar

2. This shows the resize handlers at the corners and edges of the browser window. You can drag on them to resize the browser window.



Resizing browser window

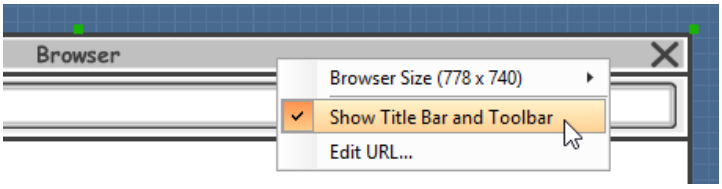
Alternatively, you can right click on the browser window's title bar and select **Browser Size > %PREFERRED_SIZE%** from the popup menu.



Adjusting browser window size

Show/Hide title bar and toolbar

To hide the title bar and toolbar, right click on the browser window's title bar and de-select **Show Title Bar and Toolbar** from the popup menu.



Hide Show Title Bar and Toolbar

To show it again, right click on the wireframe and select Edit Browser Frame. Move the mouse pointer near the top most of the browser window. Right click and select **Show Title Bar and Toolbar** from the popup menu.

Editing the displaying URL

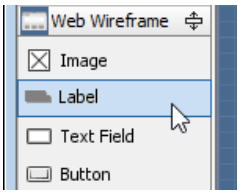
To edit the displaying URL, just double click on the URL field and enter the URL.

Editing the displaying URL

Creating a wireframe element

Method 1 - Diagram toolbar: Select-and-Click

1. Select the desired wireframe element from the diagram toolbar (e.g. Label).

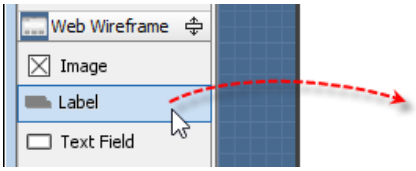


Create a label by selecting it from the diagram toolbar

2. Click on the wireframe, at the position where you want the wireframe element to be created.

Method 2 - Diagram toolbar: Drag and drop

1. Press on the desired wireframe element in the diagram toolbar
2. Hold the mouse button.
3. Drag to the wireframe.

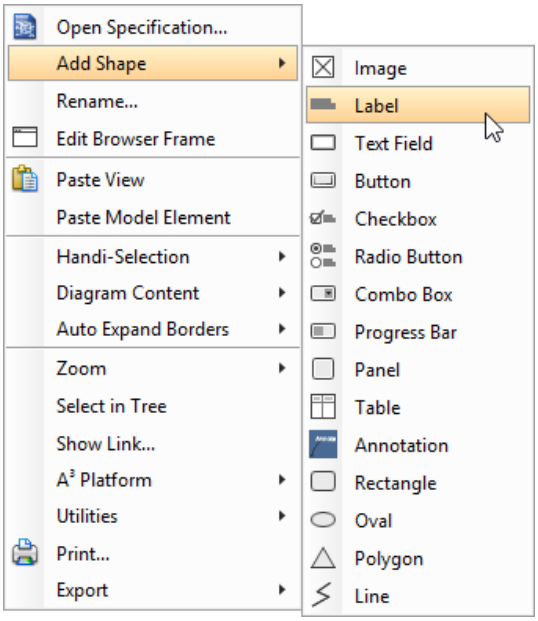


Create a Label with drag-and-drop

4. Release the mouse button in the wireframe, at the position where you want the wireframe element to be created.

Method 3 - Popup menu

1. Right click on the wireframe, at the position where.
2. Select **Add Shape > %SHAPE_TYPE%** from the popup menu, where **%SHAPE_TYPE%** is the kind of wireframe element you want to create.



Creating a wireframe element via the popup menu

Method 4 - Through smart create resource

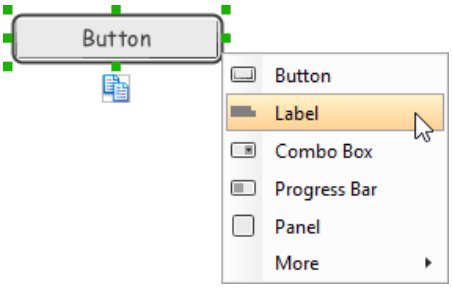
1. Click directly on the wireframe, at the position where you want the wireframe element to be created. You should see a green icon appear, known as the **Smart Create** resource.

Creating a wireframe element using Smart Resource

2. Press on the **Smart Create** resource and hold the mouse button.
3. Drag to outline the size of the wireframe element to be created.

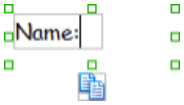
Creating a label in specific size

4. Release the mouse button. In the popup menu, choose the type of wireframe element to create.



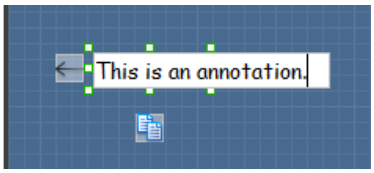
Choosing the wireframe element to be created

Method 5 - Double-clicking (Label and annotation only)
To create a label, double-click on the wireframe and enter the label caption.



Creating a label

To create an annotation, double-click outside the browser window and enter the annotation text.



Entering annotation text

Accurate positioning of wireframe element using the alignment guide

Alignment guide is a dotted line that appears when you move wireframe elements in a wireframe. It helps you align elements perfectly with others. Simple select element(s) and drag it around. When the selection approaches another element in the wireframe, you can adjusting the positioning of selection with the help of the guide.

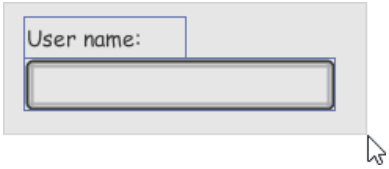
Using the alignment guide

Duplicating wireframe elements

Duplicate wireframe elements enable you to create new elements base on existing ones. This saves you a lot of time in creating elements with same/ similar style, size and content to the existine ones.

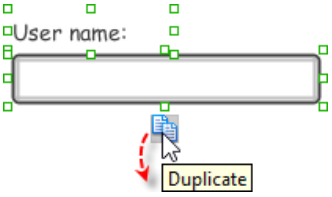
To duplicate wireframe elements:

1. Drag in the wireframe to select a range of elements to duplicate.



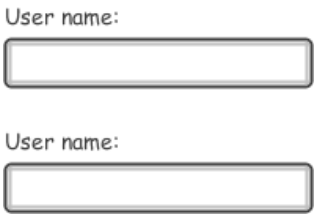
Selecting wireframe elements to duplicate

2. At the bottom of your selection, press on the **Duplicate** resource icon and hold the mouse button.
3. Drag it.



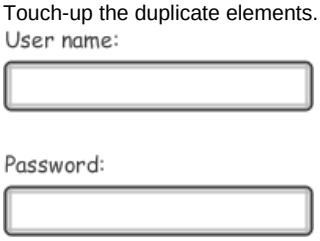
To duplicate wireframe elements

4. Release the mouse button at the position where you want the wireframe element to be created.



Wireframe elements duplicate

5. Touch-up the duplicate elements.



Renamed a label

NOTE: Instead of dragging the **Duplicate** resource, you may click on it, too..

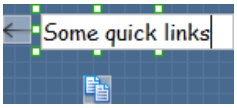
Annotating wireframe with Annotation shape

The use of annotations in wireframe allows you to detail the elements on the wireframe. With annotation, you can describe or explain the existence of certain wireframe element, as well as to describe the calls to action and the expected results.

In order to keep the wireframe content readable, annotations are forced to put outside the browser window. In other words, you cannot create or move an annotation to inside the browser window area.

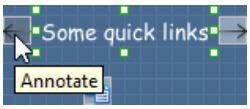
To create an annotation:

1. Double click on the background (i.e. the blue region) of the wireframe.
2. Enter the annotation text.



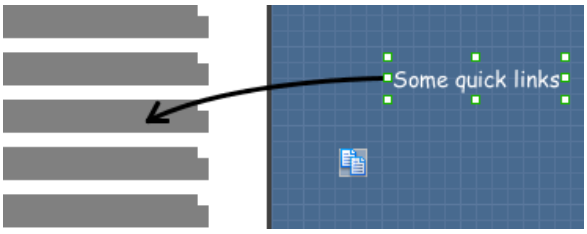
Entering annotation text

3. Press on the arrow resource and hold the mouse button.



To annotate a wireframe element

4. Drag to the wireframe element to annotate it.



Label annotated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Use Case Note List

Browse use case notes in Use Case Note List

Learn how to browse use case notes in Use Case Note List

Browse use case notes in Use Case Note List

Use case -related ideas such as the stakeholder's concern, problem and suggested solution can be recorded as notes, inside the Use Case Notes editor. In addition to browsing the notes use case by use case, the **Use Case Note List** provides you with a way to browse the notes of all use cases in the project.

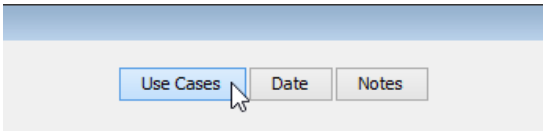
Opening the Use Case Note List

To open **Use Case Note List**, select **Modeling > Use Case Note List** from the main menu.

Browsing use case notes of all use cases

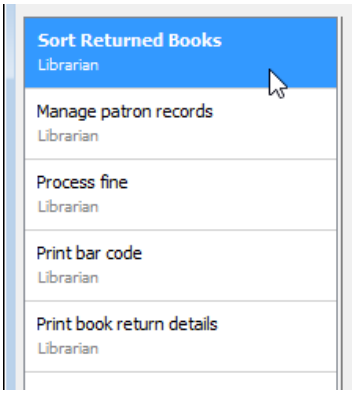
To browse use cases notes of all use cases:

1. At the top of the **Use Case Note List**, open the **Use Cases** page.



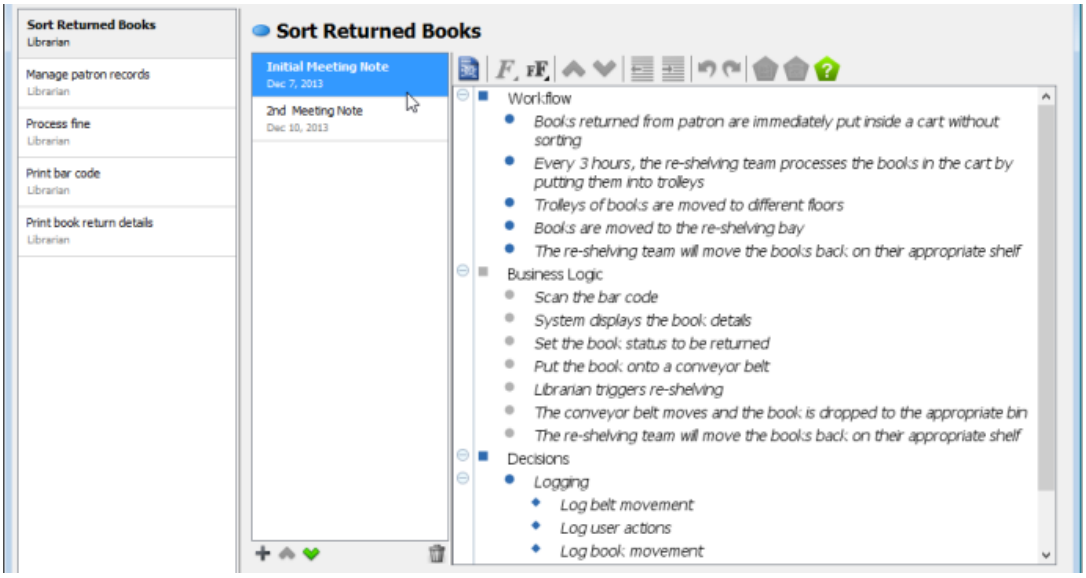
View the Use Cases page

2. All the use cases in the project are listed on the left hand side. Click on a use case to browse its use case notes.



Open a use case

3. Click on a note to open it.

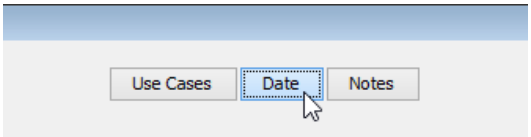


Open a use case note

Browsing use case notes taken on specific date

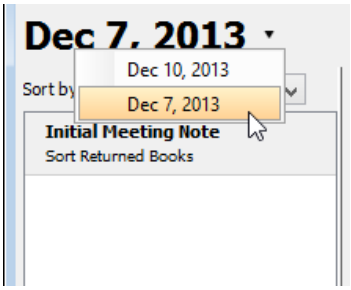
To browse use cases notes taken on specific date:

1. At the top of the **Use Case Note List**, open the **Date** page.



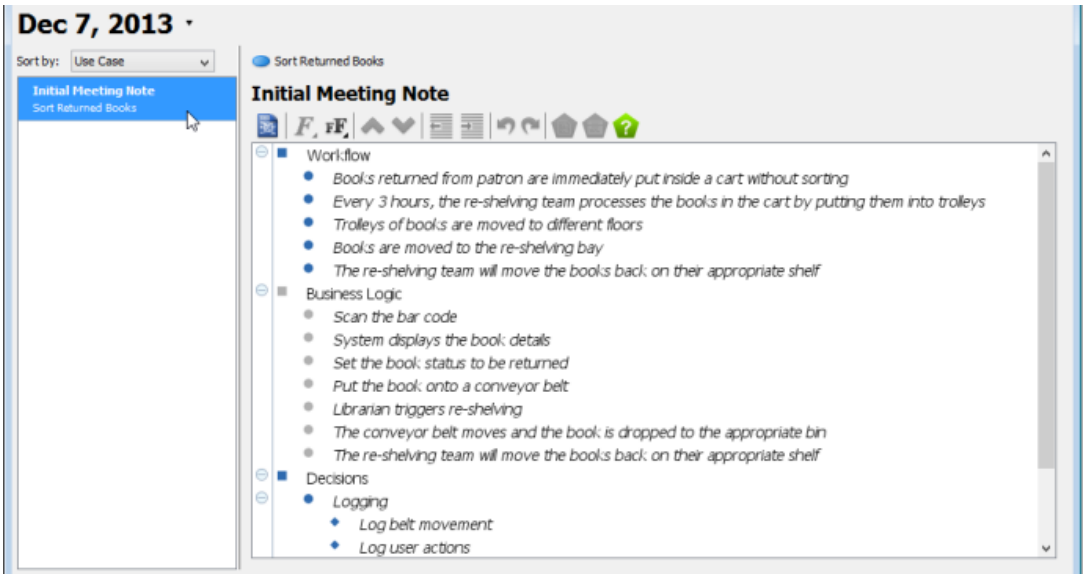
View the Date page

2. On the left hand side, choose the create date of notes to be listed.



To list use case notes created on specific date

3. All the use case notes created at the chosen date are listed on the left hand side. You can sort them by the use case they belong to or by the name of note by updating the drop down menu on top of the list.
4. Click on a note to open it.

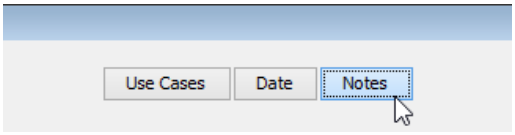


Open a use case note

Browsing use case notes in note base

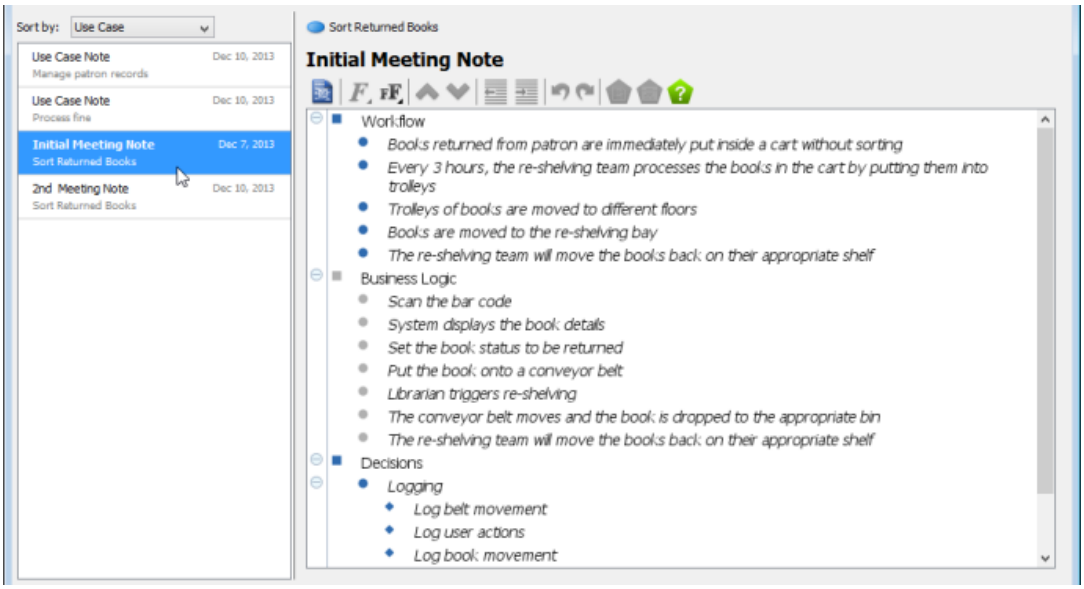
To browse all the use cases notes in note base:

1. At the top of the **Use Case Note List**, open the **Notes** page.



View the Notes page

2. All the use case notes are listed on the left hand side. You can sort them by the use case they belong to, by the name of note or by create date by updating the drop down menu on top of the list.
3. Click on a note to open it.



Open a use case note

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube - How to Take Use Case Notes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Scenario List

Browse use case scenario in Scenario List

Learn how to browse use case scenarios in Scenario List

Browse use case scenario in Scenario List

A use case scenario defines the way how user will interact with the system in achieving a use case. In addition to browsing the scenarios use case by use case, the **Scenario List** provides you with a way to browse the scenarios of all use cases in the project.

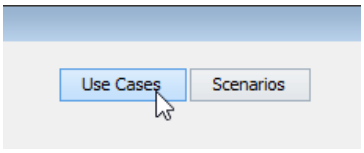
Opening the Scenario List

To open **Scenario List**, select **Modeling > Scenario List** from the main menu.

Browsing use case scenarios of all use cases

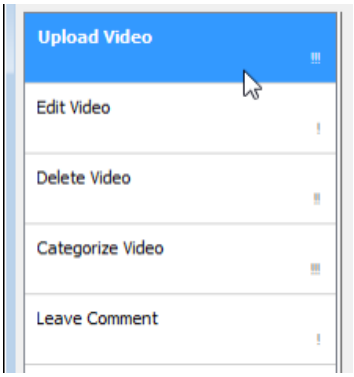
To browse use cases scenarios of all use cases:

1. At the top of the **Scenario List**, open the **Use Cases** page.



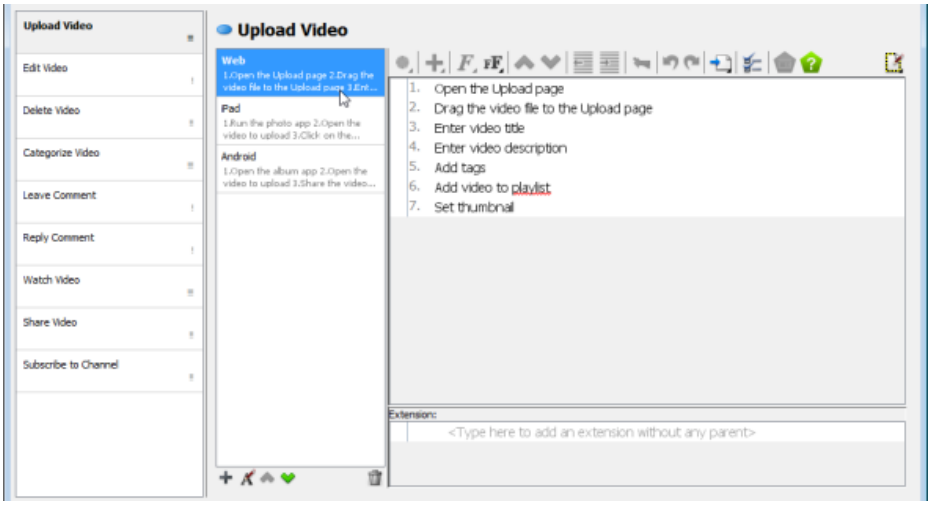
View the Use Cases page

2. All the use cases in the project are listed on the left hand side. Note that the tiny exclaimation marks that appear next to the name of use cases indicate their rankings. Click on a use case to browse its use case scenarios.



Open a use case

3. Click on a scenario to open it.

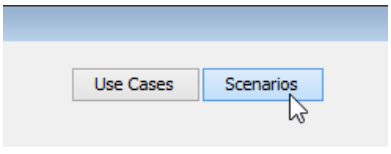


Open a use case scenario

Browsing use case scenarios in scenario base

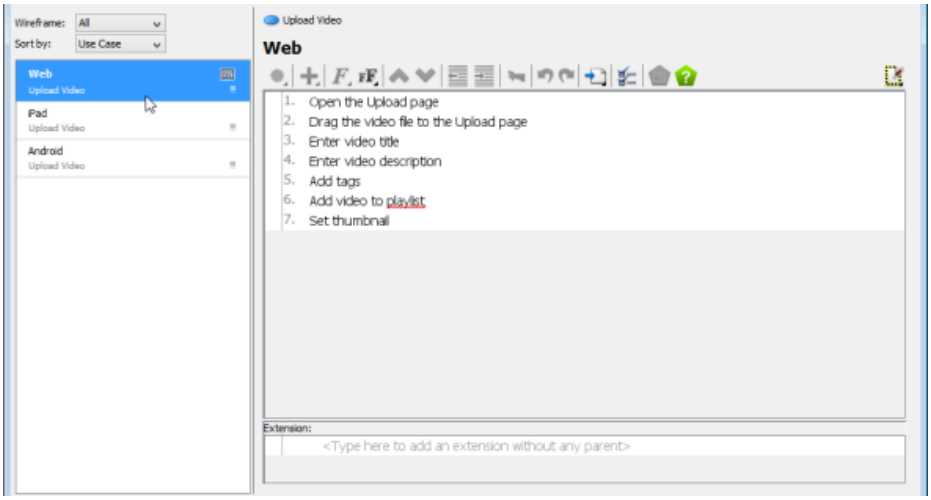
To browse all the use cases scenario in scenario base:

1. At the top of the **Scenario List**, open the **Scenarios** page.



View the Scenarios page

2. All the use case scenarios are listed on the left hand side. You can display scenarios that have or have no wireframe defined. You can sort them by the use case they belong to or by the name of scenario by updating the drop down menu on top of the list.
3. Click on a scenario to open it.



Open a use case scenario

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- YouTube Video - How to Create Scenario-Based Wireframe?
- YouTube Video - How to Present Wireframes?
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Requirement diagram

Requirement diagram lets you visualize system functions as well as the ways to test the functions. This chapter teaches you how to work with requirement diagram.

Creating requirement diagram

This page shows you how to create requirements in requirement diagram, specify requirements body, relate requirements and create test cases.

Customizing requirement types

You will see how to define your own requirement type in this page.

Requirement grid

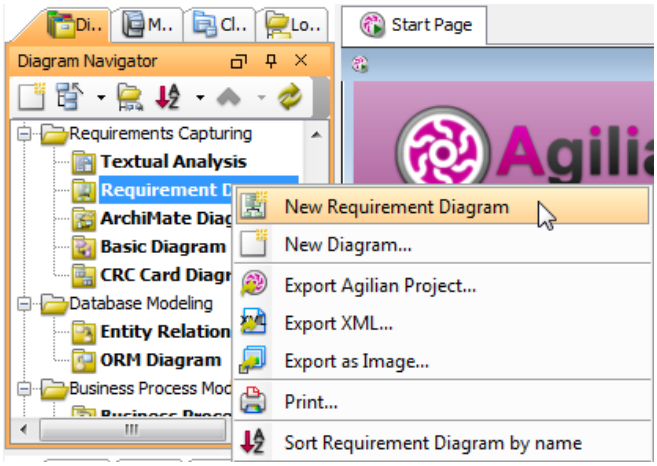
Instead of creating requirements in diagram, you can also create from grid - a table that lists requirements and their properties.

Creating requirement diagram

Requirement diagram lets you visualize system functions as well as the ways to test the functions. It is one of the SysML diagram types. You can create your own requirement type for business specific needs. The customization part will be covered in the next page.

Creating requirement diagram

- Click on **Requirement** on toolbar and select **Requirement Diagram** from the drop down menu **.**
- Right click on **Requirement Diagram** in **Diagram Navigator** and select **New Requirement Diagram** from the popup menu.
- Select **File > New Diagram > Requirements Capturing > Requirement Diagram** from the main menu.

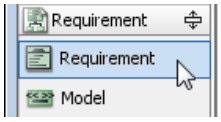


Create requirement diagram

Alternatively, select **File** > **New Diagram** > **Requirements Capturing** > **Requirement Diagram** from the main menu to create a requirement diagram.

Creating requirement

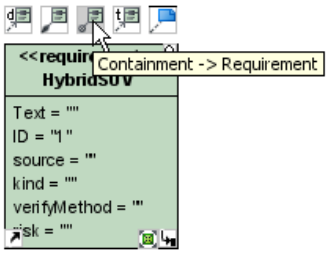
To create a Requirement, click the **Requirement** button on the diagram toolbar and then click on the diagram.



*Create **Requirement***

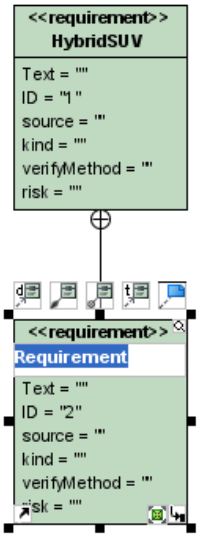
Decomposing requirement

To decompose a Requirement, click the **Containment** -> **Requirement** resource and drag.



Decompose Requirement

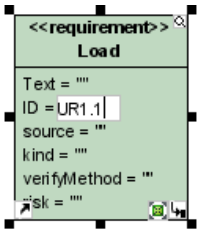
Move the mouse over empty space of the diagram and then release the mouse button, a Requirement together with a Containment relationship will be created.



Requirement and Containment created

Inline editing requirement properties

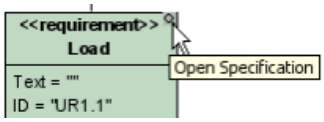
To inline edit the property of a Requirement (e.g. ID), double-click on the property, enter new value and press Enter to confirm.



Inline editing Requirement properties

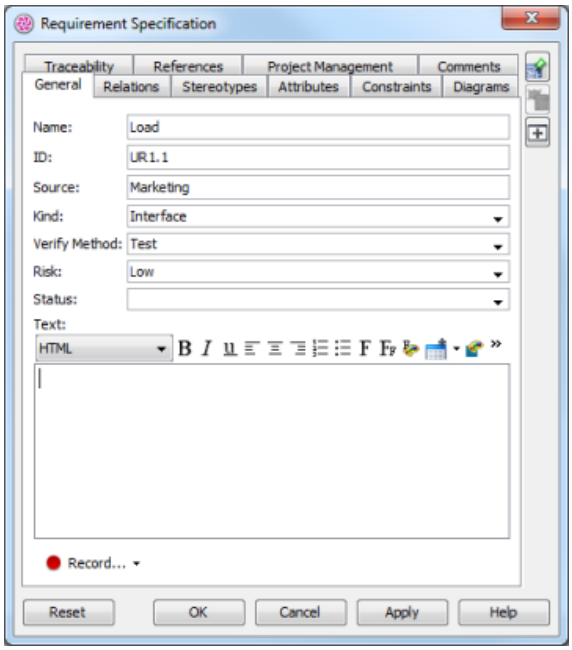
Editing requirement properties with specification dialog box

You can also open specification dialog box of a Requirement to edit its properties. Click **Open Specification** resource of the Requirement.



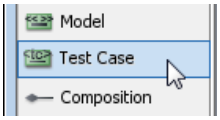
Open specification of Requirement

The **Requirement Specification** dialog box shows. Edit the properties and click **OK** button to apply the changes.



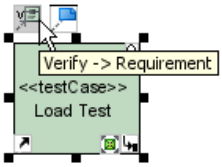
Creating test case and link to requirement

To create a Test Case, click the **Test Case** button on the diagram toolbar and then click on the diagram.



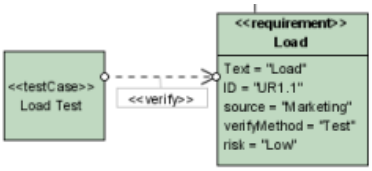
*Create **Test Case***

Click **Verify** -> **Requirement** resource of Test Case and drag.



Link Requirement with Test Case

Move the mouse over a Requirement and then release the mouse button, a Verify relationship will be created from the Test Case to the Requirement.



Verify relationship created

Related Resources

The following resources may help you learn more about the topic discussed in this page.

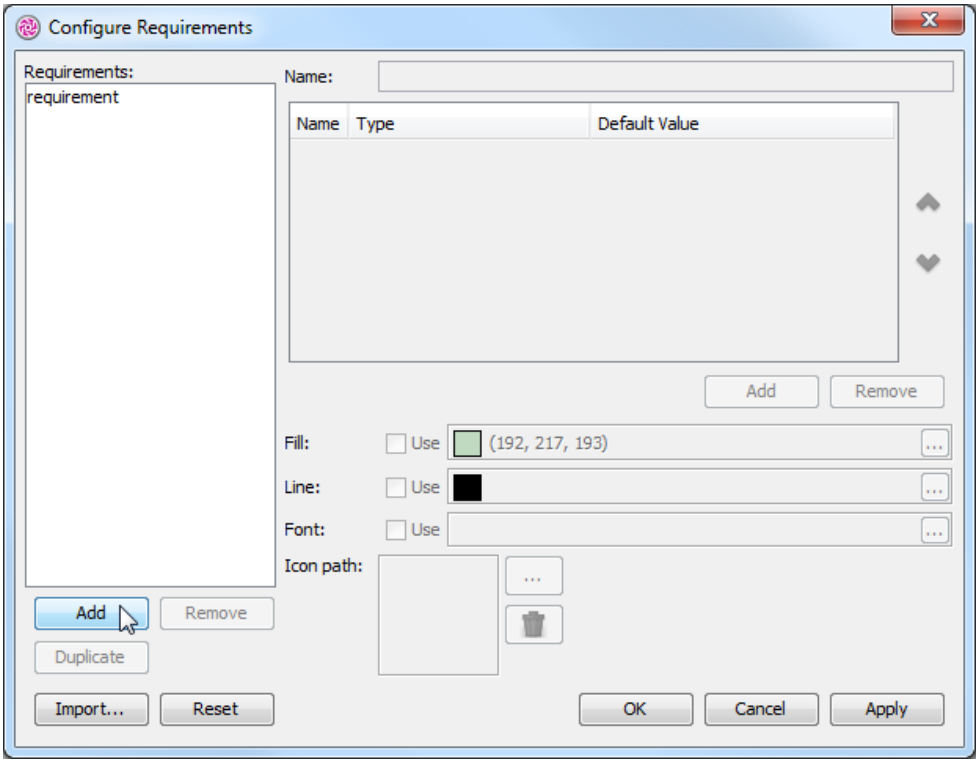
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Customizing requirement types

Users can record and present requirements as boxes visually through requirement modeling. The name of requirements summarizes the requirement while a set of attributes defines the requirement. The default requirement box allows users to specify general attributes, such as ID, source, kind, verify method, risk and status. Moreover, you are allowed to customize your own requirement types that contain attributes related to your domain.

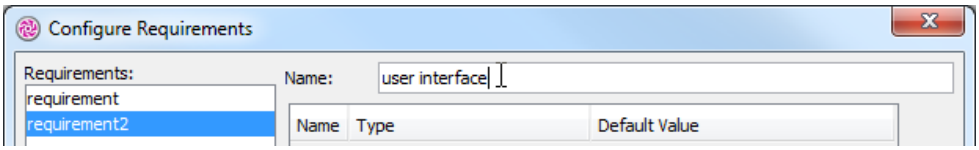
1. Create a new requirement diagram or open your target requirement diagram where you want to customize your our own requirement types. Select **Tools > Configure Requirements** from the main menu.

2. When the **Configure Requirements** dialog box pops out, you will see the default requirement type *requirement* lists under **Requirements**. Click **Add** button to add a new requirement type.



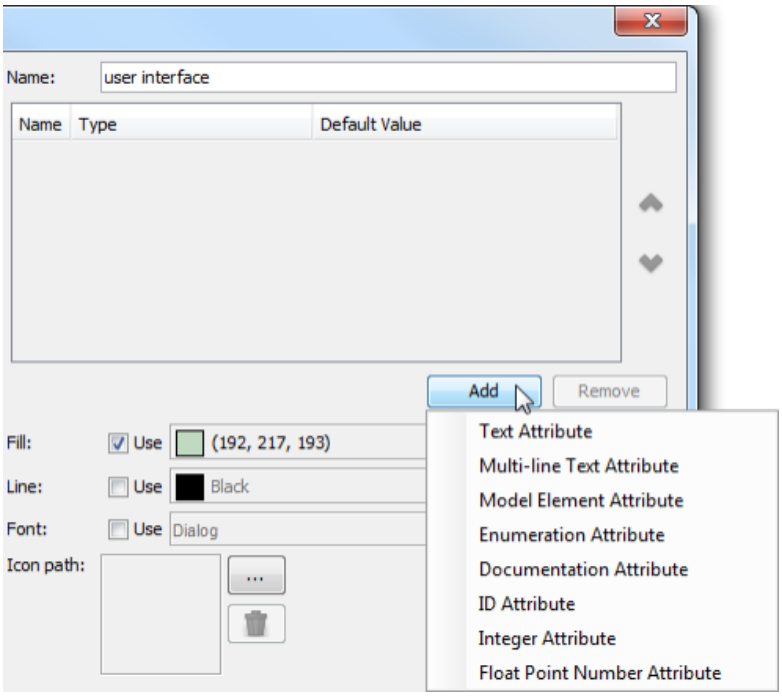
*Add a new requirement type in the **Configure Requirements** dialog box*

3. Name the newly created requirement type in the **Name** field.



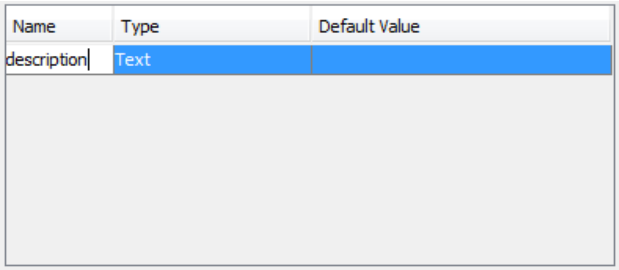
Enter name for requirement type

4. Next, add attributes for the requirement type to make it meaningful. Click **Add** button and select an attribute from the drop-down menu.



Select an attribute

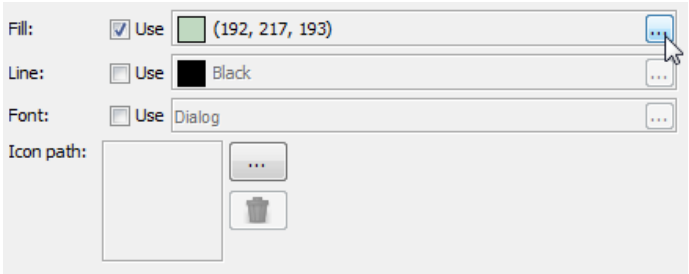
5. Name the newly created attribute. Create as much as attribute you need by following the previous step.



Enter name for attribute

> **NOTE:** If you select **Enumeration Attribute** from the drop-down menu, **Edit Enumeration...** button will appear. Click **Edit Enumeration...** button to edit it.

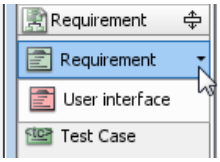
6. Besides defining attributes, you can format the requirement type with fill, line and font. Click the **...** button of **Fill** if you want to customize a color for the requirement type.



Customize color for requirement type

> **NOTE:** Click the **...** button of **Line** if you want to customize its line property while click the **...** button of **Font** if you want to customize its font property.

7. Once you finish configuring requirement types, click **OK** button to return to your target requirement diagram.
8. Finally, you can see the customized requirement type is available on the diagram toolbar.



Related Resources

The following resources may help you learn more about the topic discussed in this page.

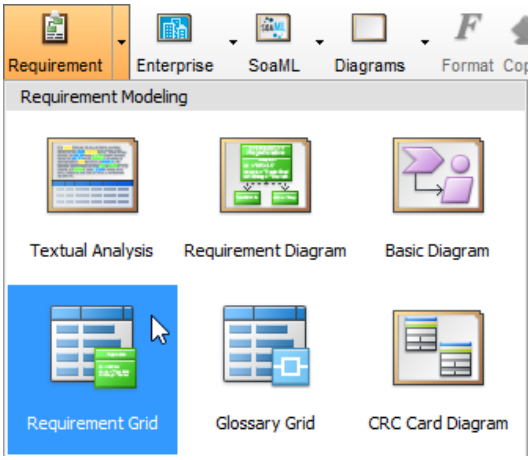
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Managing requirements with Requirement Grid

Requirements Grid is a table with requirements listed in it. It enables you to access all requirements in a project or diagram, lookup requirements by criteria and create requirements.

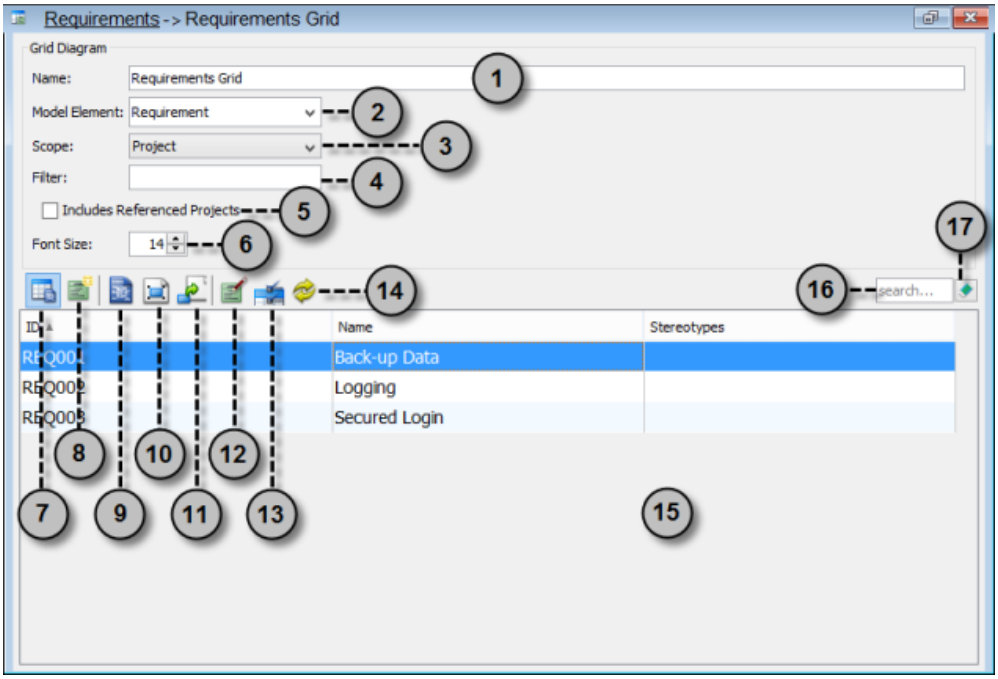
Creating the Requirement Grid

Click on **Requirement** on toolbar and select **Requirement Grid** from the drop down menu **.**



Open Requirement Grid

The overview of Requirement Grid



The Requirement Grid

No.	Name	Description
1	Name	The name of this Requirement Grid.
2	Model Element	Change the type of model element to be listed in grid. Although Requirement is selected by default, you may change it to other types.
3	Scope	The location to look for the requirements to list in grid. By default, requirements are found from the whole project. You can change to find requirements from specific model or package, or to find only requirements right at the root level. You can also restrict the scope to all diagrams, to within a specific diagram or to all requirements that has not been visualized in any diagram.
4	Filter	Apply filter to grid content. Text entered here is matched against the **Name** property of requirements listed in grid. Requirements that do not contain the entered text in their name are hidden.
5	Include Referenced Projects	Check it to list also requirements in referenced projects, in Requirement Grid.
6	Font Size	Click to adjust the font size of text in Requirement Grid.

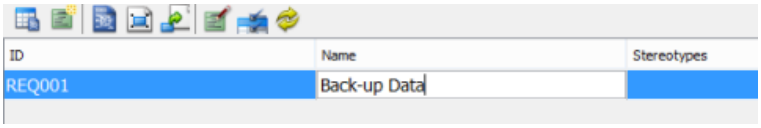
7	Configure Grid	Click to show/hide the grid configuration panel, which allows you to enter the name of grid, the model element to be listed in grid, the scope and to apply filter to grid content.
8	New Requirement	Click to create a requirement.
9	Open Specification...	Select a requirement in Requirement Grid and click this button to open its specification.
10	Show View...	Select a requirement in Requirement Grid and click this button to list the diagrams that contains the view of the selected requirement.
11	Visualize...	Select a requirement in Requirement Grid and click this button to show it in a new or existing diagram.
12	Configure Requirements	Click to create or edit requirements types.
13	Configure Columns...	Click to select the property(ies) of requirements to be listed in the grid, as columns.
14	Refresh	Click to refresh the grid content by showing the most updated information of requirements listed.
15	List of requirements	Requirements are listed here.
16	Search	Find requirement(s) by entering search criteria.
17	Clear	Click to clear the text entered in **Search** box.

The fields in Requirement Grid

Creating requirement in Requirement Grid

To create a requirement in **Requirement Grid**:

1. Click on **New Requirement** above the **Requirement Grid**.
2. Enter the name of requirement.



Creating requirement in Requirement Grid

3. Press **Enter** to confirm editing.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Textual analysis

Textual analysis is a tool for recording customers' needs. Furthermore, it lets you extract key terms from a passage you recorded, and transform the terms to model elements or put them into glossary to build a project -based dictionary.

Recording requirements

Document customers' needs by performing textual analysis.

Identifying important terms

Shows you how to identify glossary term from a passage recorded by textual analysis.

Identifying candidate objects

Shows you how to identify candidate model element from a passage. You selectively convert candidate to actual model element and visualize it in diagram.

Forming diagram from candidate objects

Shows you how to visualize candidate elements in a diagram.

Candidate pane view

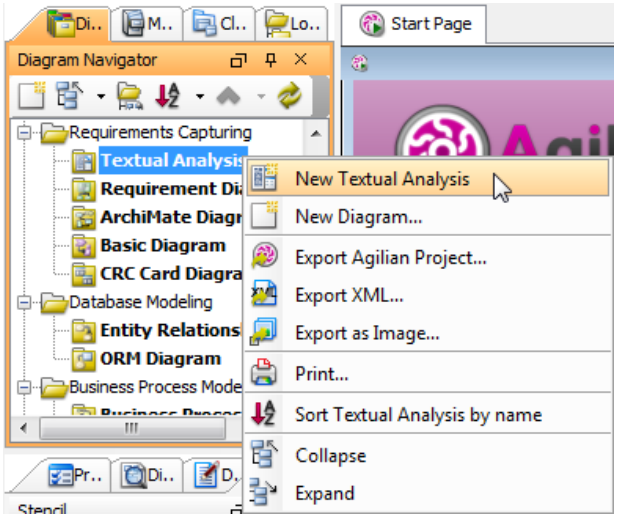
A view that shows candidate elements in visualized form - as boxes.

Recording requirements

Before you start business process modeling, you usually have to discuss with your customers about their needs and to familiarize yourself with their company's operations as well as their problems. During the meeting you can collect useful information from customer, which include the conversation log, documents. You can make use of textual analysis , a text-based editor, to help record those textual information. In addition to a plain text editor, you can identify important terms or objects (e.g. class, use case) from the problem description.

Creating textual analysis

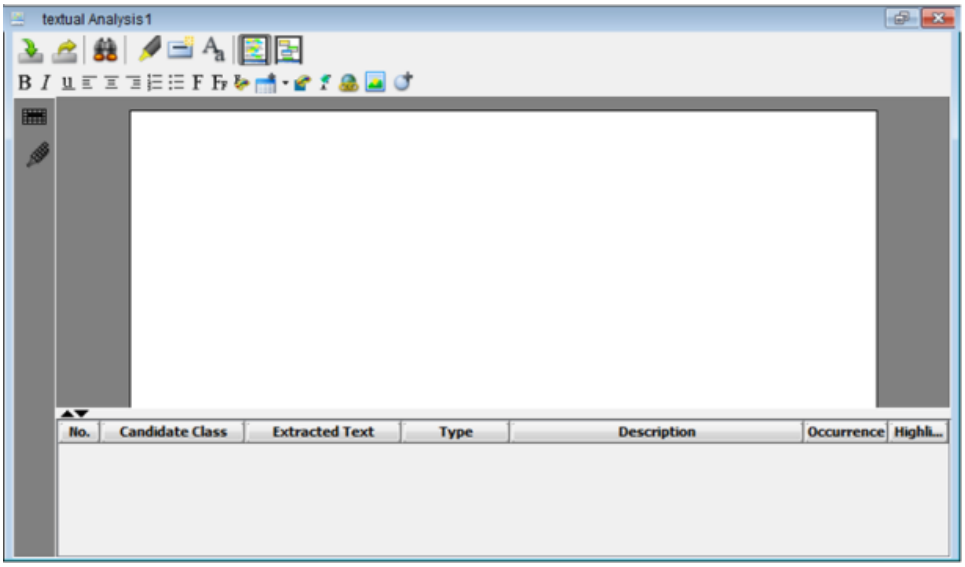
- Click on **Requirement** on toolbar and select **Textual Analysis** from the drop down menu **.**
- Right click on **Textual Analysis** in **Diagram Navigator** and select **New Textual Analysis** from the popup menu.
- Select **File > New Diagram > Requirements Capturing > Textual Analysis** from the main menu.



Create a textual analysis

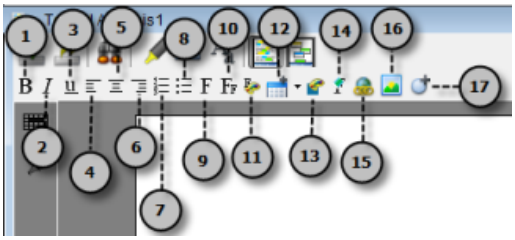
Problem statement editor

The problem statement editor is where you can record the textual information you obtain from your customers.



Problem statement editor

All buttons on editor's toolbar are depicted in the following table:



Editor's toolbar

No.	Name	Description

1	Bold	Set the highlighted text to bold.
2	Italic	Set the highlighted text to italic.
3	Underline	Underline the highlighted text.
4	Left Justify	Set the alignment of highlighted text to the left.
5	Center Justify	Set the alignment of highlighted text to the center.
6	Right Justify	Set the alignment of highlighted text to the right.
7	Ordered list	Add a numbered list.
8	Un-ordered list	Add a list with bullet points.
9	Font	Select the font family of highlighted text.
10	Font size	Select the size of highlighted text.
11	Font color	Select the color of highlighted text.
12	Table	Add a table.
13	Background color	Select the background color of highlighted text.
14	Clear formats	Clear formats of the whole editor to convert the content to plain text.
15	Link	Add a hyperlink.
16	Image	Add an image.
17	Add Model Element...	Insert an existing model element or create a new one.

The description of buttons on problem statement's toolbar

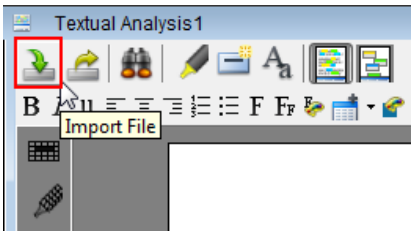
Entering problem statement

Three ways of entering problem statement are provided in Agilian .

- Typing on the editor
- Importing an external text file
- Copying and pasting from an external source

To type in the editor, type the problem statement directly on the editor.

To import a text file, click **Import File** on the toolbar.



Import file

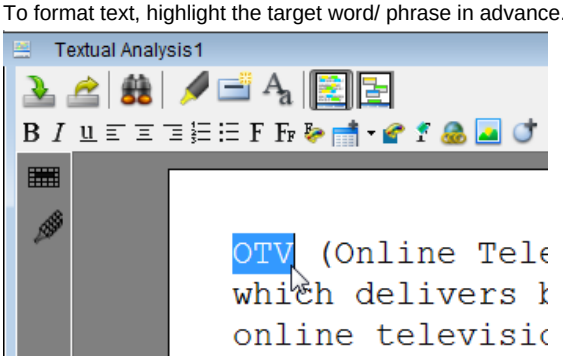
When the **Open** dialog box pops out, select a text file to import. As a result, the imported problem statement will be shown on the text area.

To copy and paste from an external source, press **Ctrl + C** on the selected text and press **Ctrl + V** for pasting it on the editor.

Formatting text

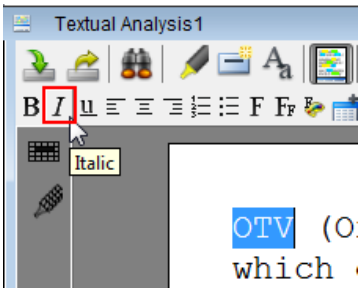
Since Agilian supports rich text format (RTF), you can format the problem statement on the editor, such as making it bold, italic, or inserting a table.

1. To format text, highlight the target word/ phrase in advance.



Highlight OTV

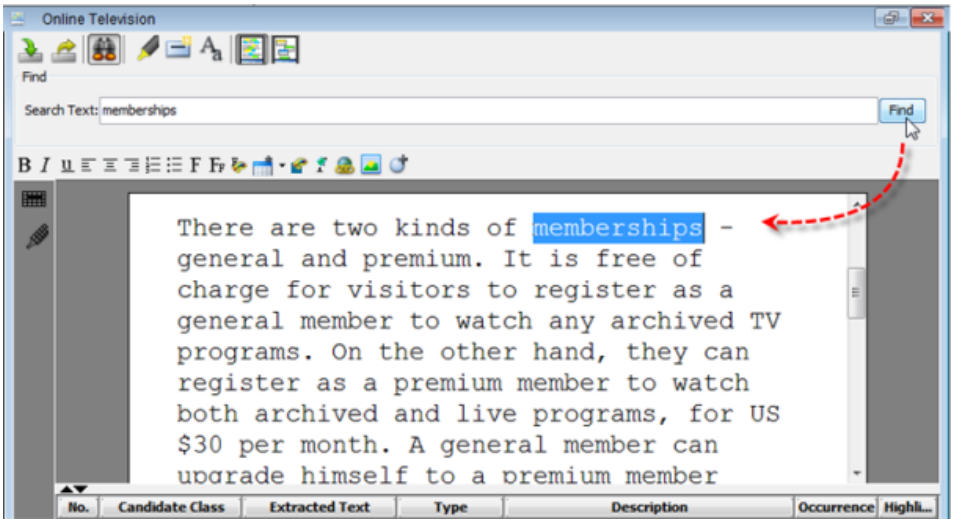
2. Next, click a corresponding button on the toolbar. i.e. Click **Italic** button to make the target word italic.



*Click **Italic** button*

Finding a keyword

You can search your target word/phrase in problem statement in shortcut through **Find** feature. Click **Find** button on the toolbar and then enter the word/ phrase in **Search Text**. Finally, click **Find** button next to **Search Text**. As a result, the word/ phrase matching you typed will be highlighted.



The word memberships is highlighted in problem statement

Related Resources

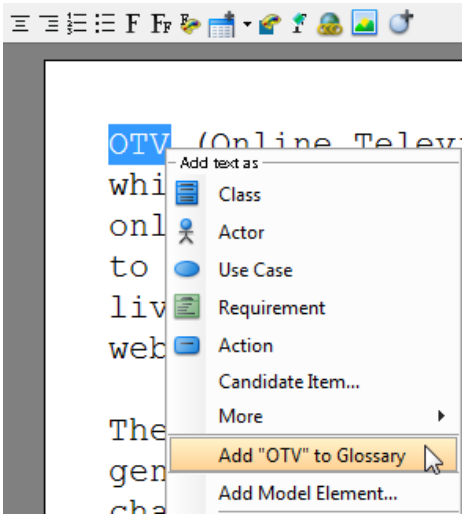
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Identifying important terms

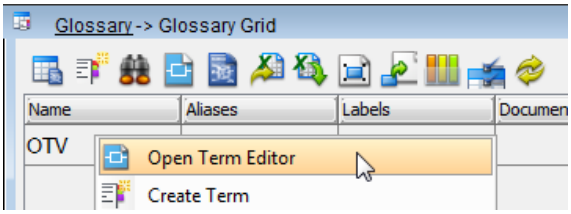
A word usually can have various meaning under different domains. To clarify and standardize the meaning of your specific word, you can extract it from textual documentation to define it as a glossary term. After adding the word as glossary term, you can define its aliases and enter its documentation to provide additional information. In textual analysis, you can define a specific word by highlighting it on problem statement editor and add it to glossary. After that, define aliases and enter documentation for the glossary term in term editor.

1. Highlight the specific term on problem statement editor, right click on it and select **Add "[the highlighted term]" to Glossary** from the pop-up menu.



Add OTV to glossary

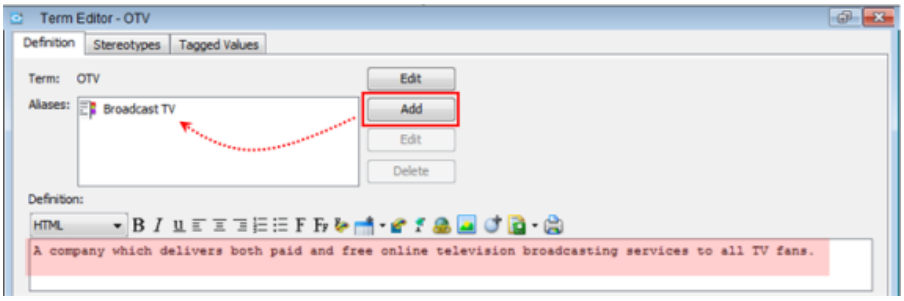
2. When the **Glossary Grid** page is opened, right click on the newly created term and select **Open Term Editor** from the pop-up menu.



Open Term Editor

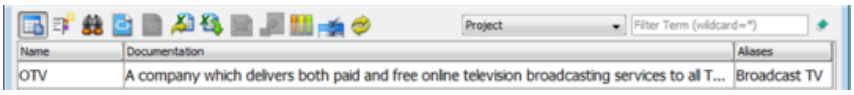
3. In the **Term Editor** page, open **Definition** tab.

4. You can define aliases for the term and enter documentation as definition for the term. To insert an alias, click **Add** button to type the alias in the pop-up **Input** dialog box. To enter the definition of the term, enter under **Definition** directly.



Define aliases and enter documentation

As a result, the columns of **Aliases** and **Documentation** are filled when you return **Glossary Grid** page.



OTV is defined

> **NOTE:** If the **Aliases** column is hidden, click **Configure Columns...**, open the **Properties** tab and select it under the **Details** folder.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube

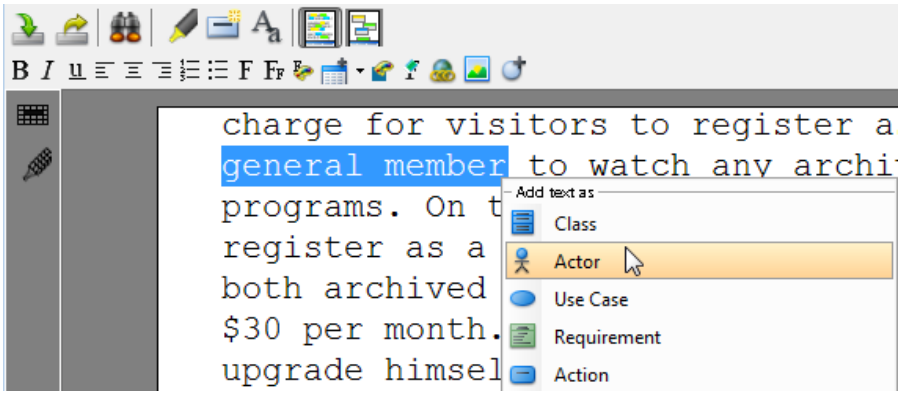
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Identifying candidate objects

By studying the problem statement, you can extract words or phrases that are relevant to the system and convert them into model elements, such as classes, use cases (system goals) and action, etc. Those objects are regarded as candidate objects. You can extract words or phrases from problem statement to become specific type of candidate objects, and edit their properties when necessary.

Identifying candidate objects

Highlight the word/ phrase from the problem statement and select **Add text as [model element type]** from the pop-up menu.



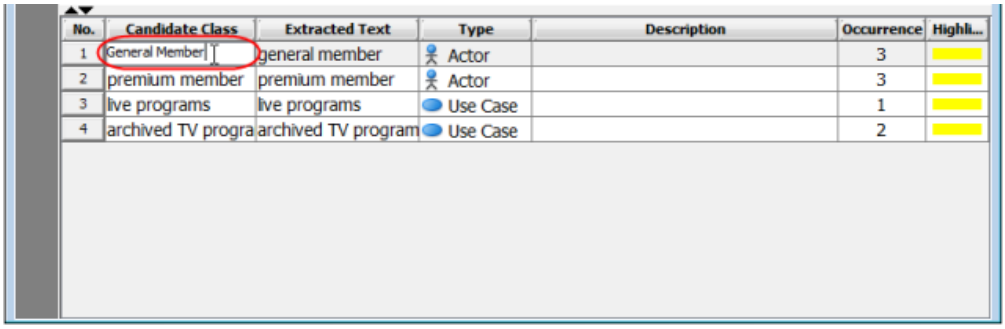
Select actor as its type

Editing candidate objects

You can rename candidate objects, change their type, write their description and change their color of highlight in the grid at the bottom of textual analysis .

To rename the candidate object:

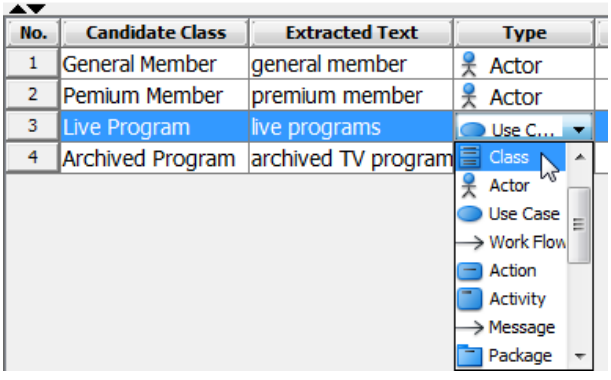
Double click on the **Candidate Class** cell and rename the candidate object.



Rename candidate object

To change the candidate object's type:

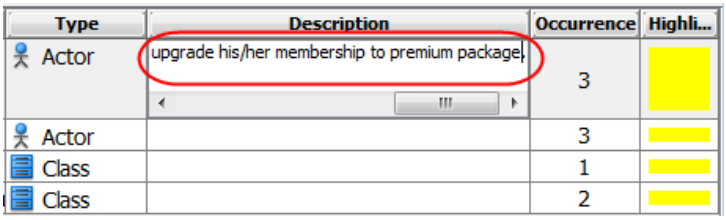
Double click on the **Type** cell and select a type from the combo box.



Select class as its type

To add description for the candidate object:

Double click on **Description** cell and type text inside the cell.

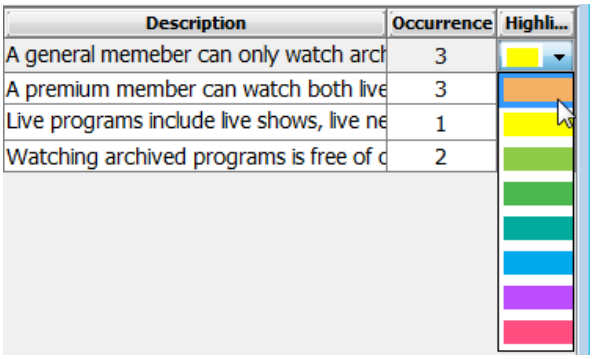


Enter description

NOTE: The text you typed in **Description** cell will become the documentation of the corresponding model element.

To change the highlight color of candidate object in problem statement:

1. Click the target **Highlight** cell and press its inverted triangle.
2. Select a color from the combo box.



Select the highlight color

Related Resources

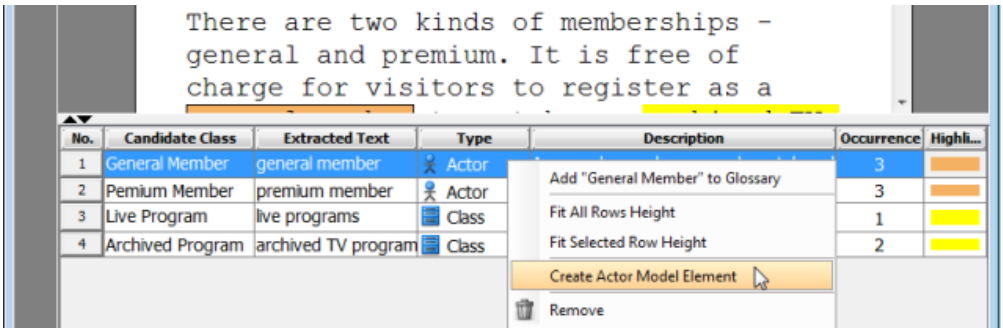
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Forming diagram from candidate objects

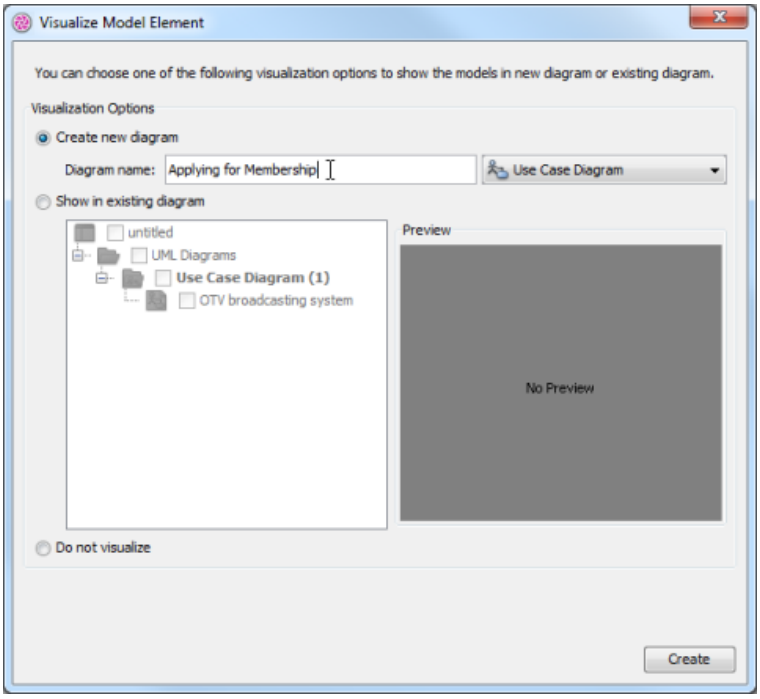
In problem statement editor, you can form a diagram from candidate objects, or show it in an existing diagram by visualizing it.

1. Right click on the target candidate object and select **Create [candidate object’s type] Model Element** from the pop-up menu.



Create a model element

2. In the **Visualize Model Element** dialog box, either check **Create new diagram** to show your model element on a new diagram or check **Show in existing diagram** to show on an existing diagram. Finally, click the **Show** button to proceed.



*Check an option in **Visualize Model Element** dialog box*

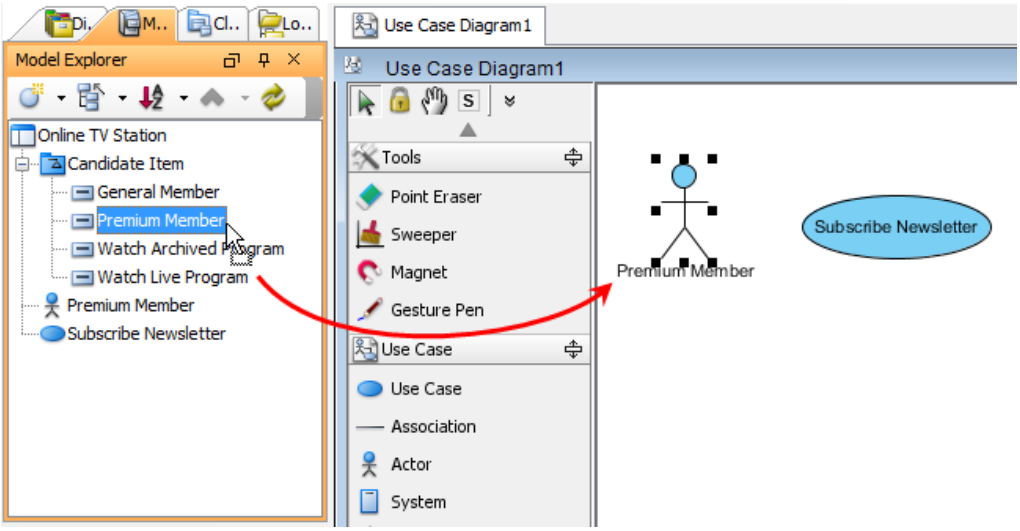
As a result, the model element will be shown on the selected diagram.

NOTE: If you have already made a model element for the candidate object, the **Create Model Element** option will be hidden even after you right click on it.

Dragging and dropping candidate objects

You can visualize existing candidate objects by dragging from **Model Explorer** and dropping on the diagram.

To visualize a candidate object or several candidate objects, select a candidate object (or a few candidate objects) from **Model Explorer**, drag and drop it(or them) on the target diagram. As a result, the view of the selected candidate object(s) will be shown on diagram.



Drag from Model Explorer and drop on the diagarm

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Candidate pane view

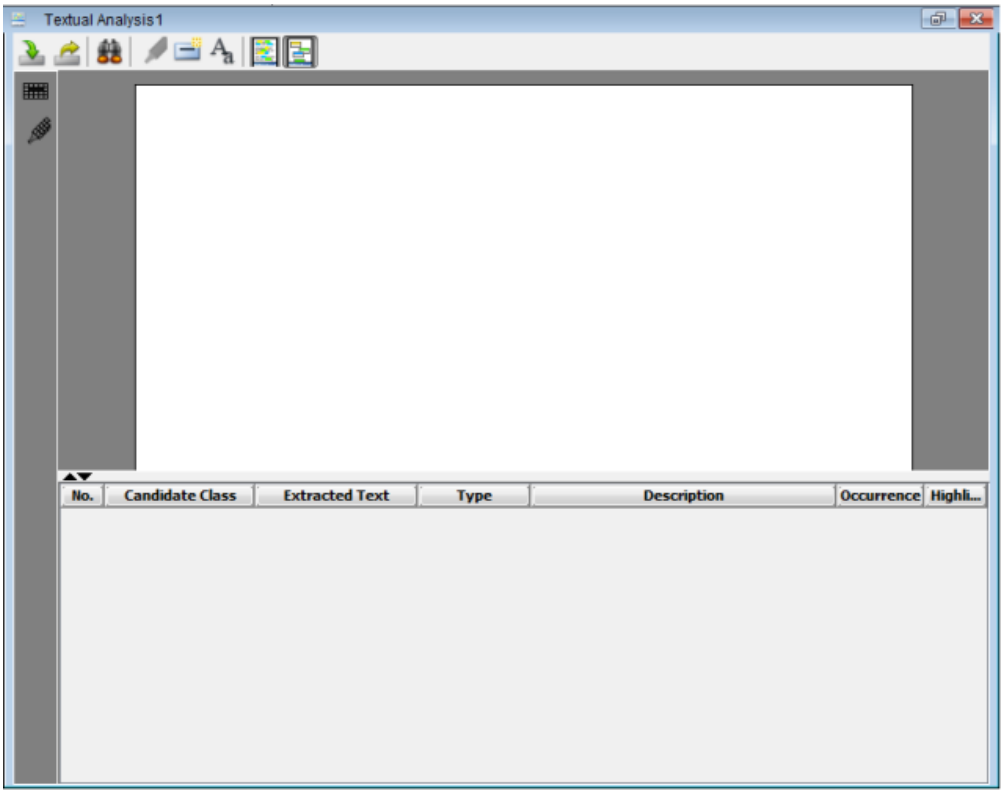
Textual analysis can be divided into two views: problem statement view and candidate pane view. While you can edit problem statement and format text in problem statement view, you can edit and organize candidate objects in candidate pane view. The main characteristic of candidate pane view is, you can visualize candidate objects as boxes on candidate pane for easy arrangement.

To switch to candidate pane view:

Click **Candidate Pane View** button.

*Click **Candidate Pane View** button*

The overview of candidate pane view

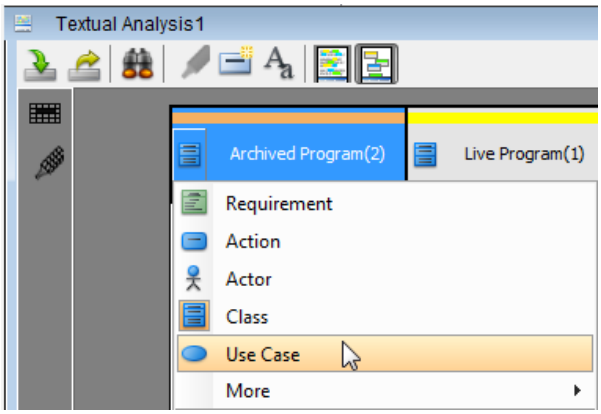


Candidate pane view

Editing candidate objects

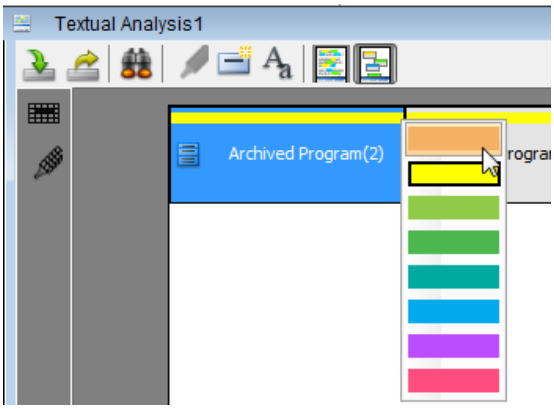
Candidate pane view is similar to problem statement view where you can rename candidate objects, enter their description, change their highlight color and type. Except editing in the grid at the bottom, you can also edit through candidate object's box on candidate pane.

To change the model element type of a candidate object, move the mouse over the target candidate object's box. Click the inverted triangle next to the model element's icon when it reveals. Select a model element type from the pop-up menu.



Change model element type

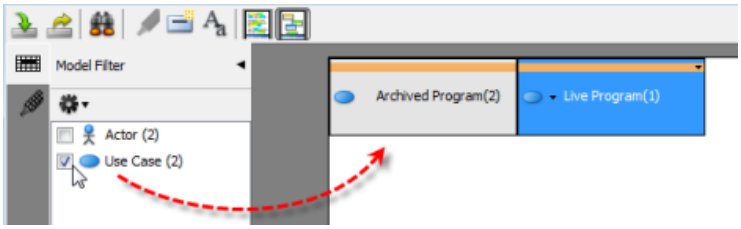
To change the highlight color of a candidate object, move the mouse over the target candidate object's box. Click the inverted triangle on its top-right corner when it reveals. Select a color from the pop-up menu.



Change highlight color

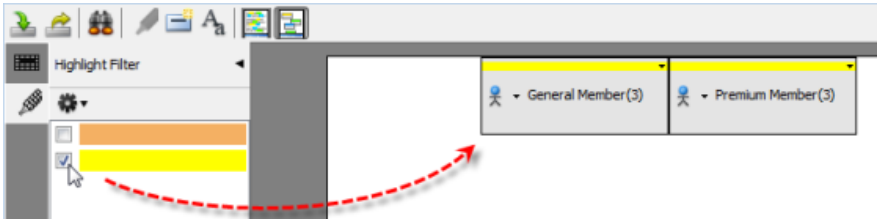
Filtering candidate objects

To filter specific model element of candidate objects, click **Model Filter** button. Check target model element(s) from the pop-up menu you want to view on candidate pane.



*Check **Use Case***

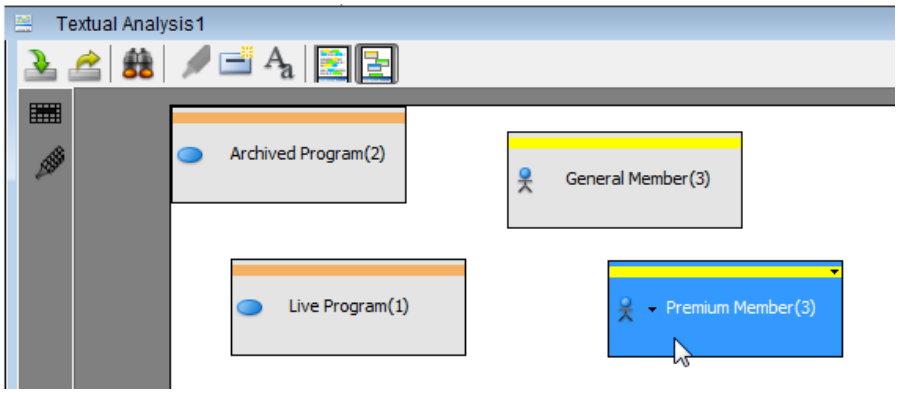
To filter specific highlight of candidate objects, click **Highlight Filter** button. Check target highlight from the pop-up menu you want to view on candidate pane.



Check yellow

Freely moving candidate objects

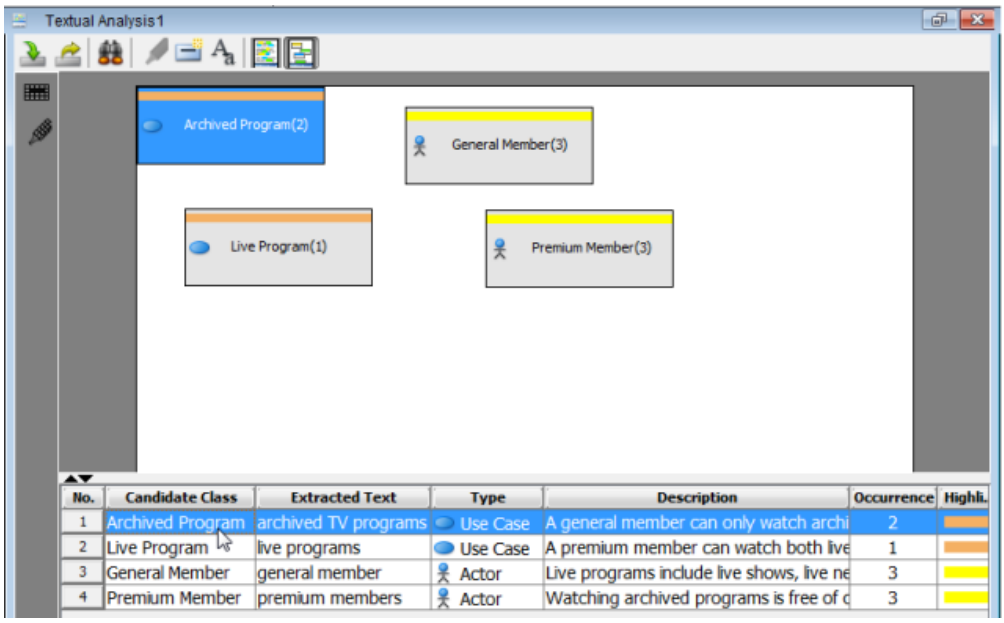
To move candidate object's box, just press the target box and drag it to your preferred location.



*Press and drag **Premium Member***

Selecting candidate objects

When you click a specific candidate object in the grid at the bottom of candidate pane, the corresponding candidate object's box will be selected on candidate pane, and vice versa.



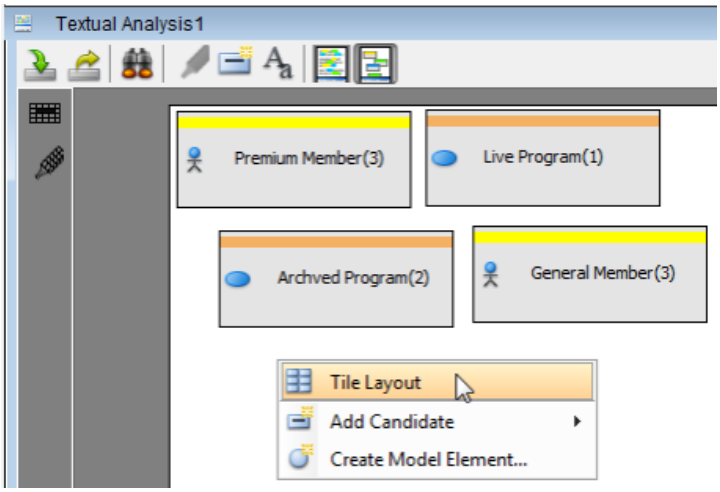
Click candidate object in grid

Setting tile layout

Tile layout refers to the selected objects are arranged in horizontal row.

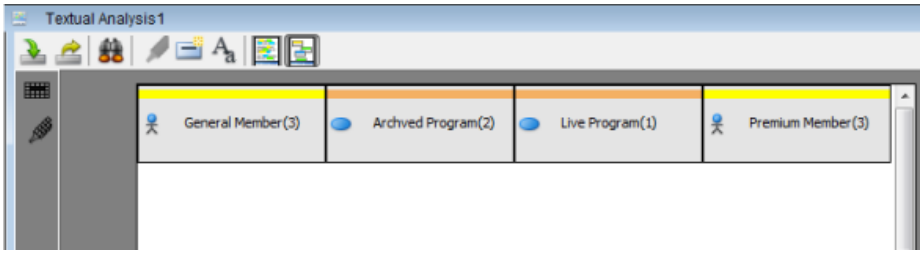
To set tile layout for all candidate object's boxes on candidate pane :

Right click on candidate pane's background and select **Tile Layout** from the pop-up menu.



Set tile layout

As a result, all candidate objects are arranged in tile layout.



Tile layout

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

CRC card diagram

Class-Responsibility Collaborator (CRC) card visualizes classes in card-like presentation. In this chapter, you will learn CRC card diagram, and see how to draw it.

Creating CRC card diagram

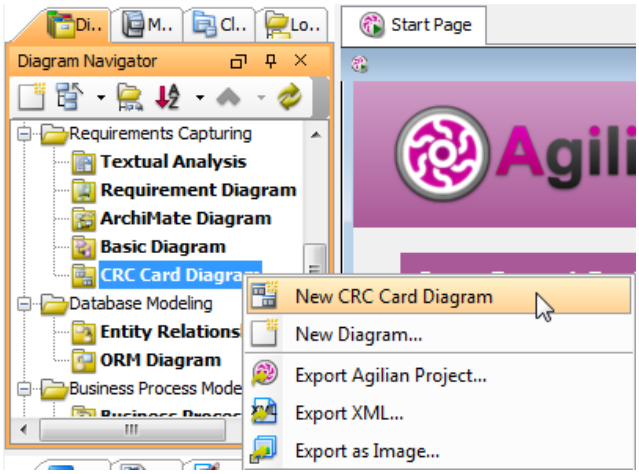
Teaches you how to create CRC card diagram.

Creating CRC card diagram

Class-Responsibility Collaborator (CRC) card visualizes classes in card-like presentation. Each CRC card contains information like the description of class, its attributes and responsibility. A CRC card diagram is a holder of these cards.

Creating CRC card diagram

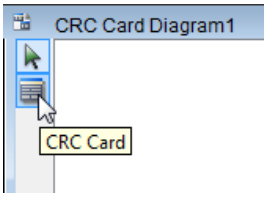
- Click on **Requirement** on toolbar and select **CRC Card Diagram** from the drop down menu **.**
- Right click on **CRC Card Diagram** in **Diagram Navigator** and select **New CRC Card Diagram** from the popup menu.
- Select **File > New Diagram > Requirements Capturing > CRC Card Diagram** from the main menu.



Create CRC card diagram

Creating CRC card

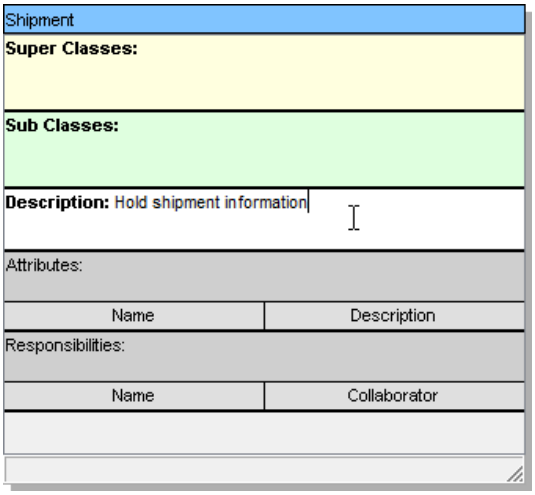
Select **CRC Card** from the diagram toolbar. Click on the diagram to create a CRC card. You can create multiple CRC card on a diagram by repeating this step.



Create CRC card

Editing CRC card properties

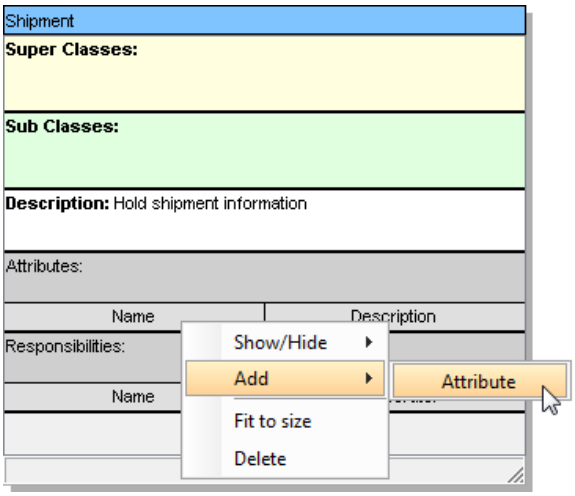
All properties in a CRC card must be edited inline. To edit, double click on the desired field, update its value, and click on the diagram background to confirm editing.



Edit description

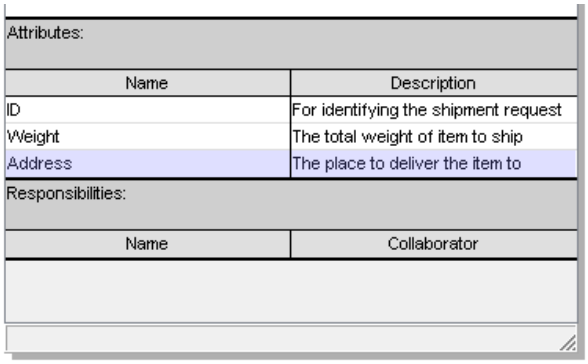
Adding attributes

Right-click on the **Attributes** heading and select **Add** > **Attribute** from the popup menu.



Add attribute

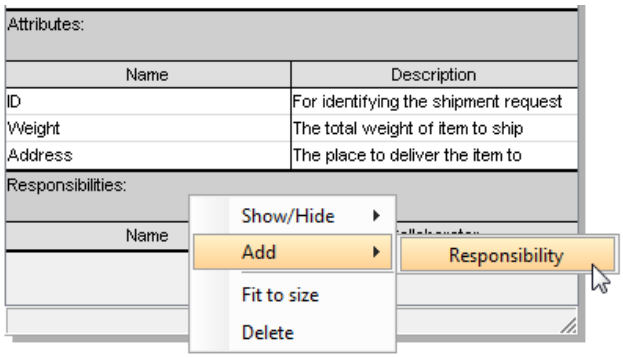
Enter the name and description. Repeat this step until all attributes are added.



All attributes added

Adding responsibilities

Right-click on the **Responsibilities** heading and select **Add** > **Responsibility** from the popup menu. Like creating an attribute, enter the name and collaborator of each responsibility to show the relationship with other parties.



Add responsibility

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Working with glossary

Glossary is a place where you can read and define project/domain -wide terminologies. This chapter shows you how to identify/define term, and how to add alias. The use of glossary grid will be covered, too.

Identify glossary term

Besides defining a term from scratch, you may extract it from documentation of model elements or from problem statements in textual analysis. You will see how to identify terms from various source.

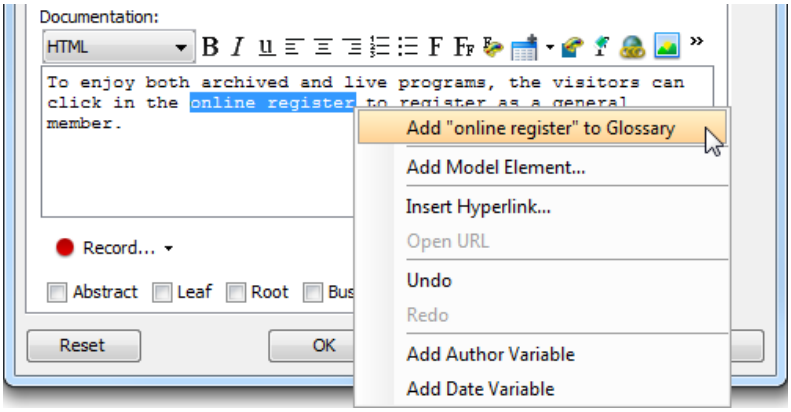
Glossary grid

The glossary grid is the primary area where you can read and define terms. You will learn how to create term in grid, and how to work with functions like label configuration and Excel exporting.

Identify glossary term

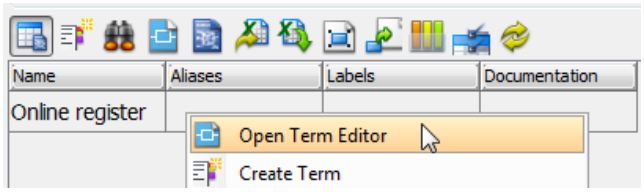
You can identify specific terms by adding them to glossary, and clarify them by defining aliases and entering documentation in any textual documents.

1. Highlight the specific term on **Documentation** editor, right click on it and select **Add "[the highlighted term]" to Glossary** from the pop-up menu.



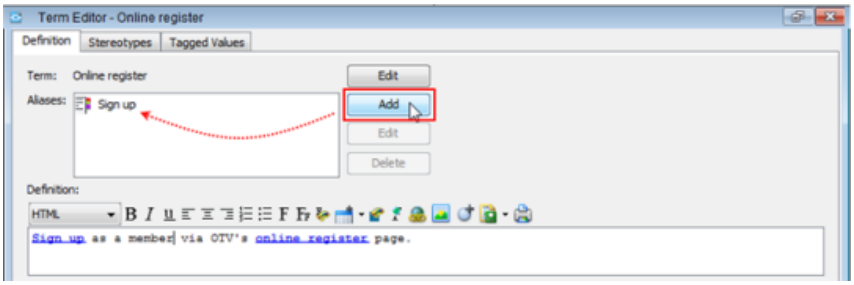
Add "online register" to glossary

2. When the Glossary Grid page unfolds, enter name for the glossary grid and then right click on the newly created term and select **Open Term Editor** from the pop-up menu.



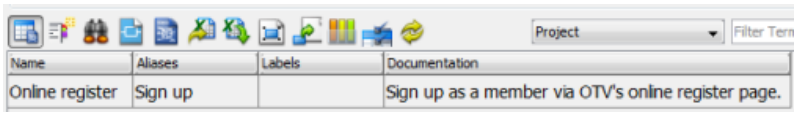
Right click to open term editor

3. In the **Term Editor** page, open **Definition** tab.
4. You can define aliases for the term and enter documentation as definition for the term. To insert an alias, click **Add** button to type the alias in the pop-up **Input** dialog box. To enter the definition of the term, enter under **Definition** directly.



Define aliases and enter documentation

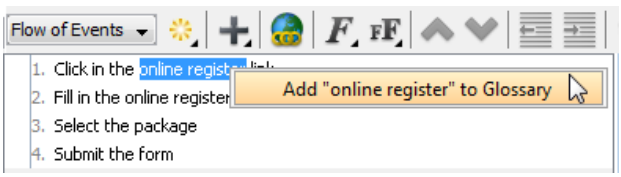
As a result, the columns of **Aliases** and **Documentation** are filled when you return **Glossary Grid** page.



Completed glossary grid

Identify term from flow of events

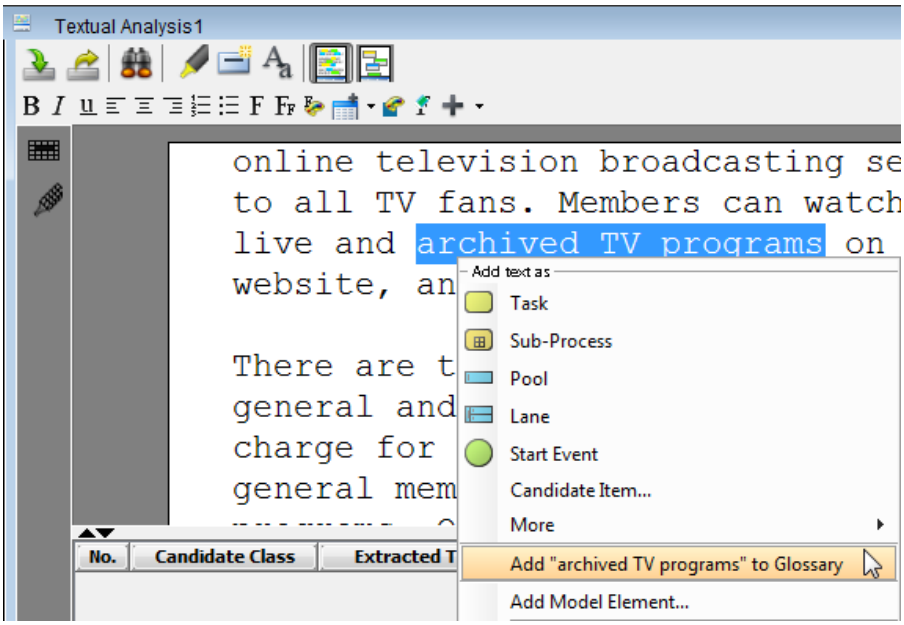
Highlight the specific term on flow of events editor, right click on it and select **Add "[the highlighted term]" to Glossary** from the pop-up menu.



Add "online register" to Glossary

Identify term from textual analysis

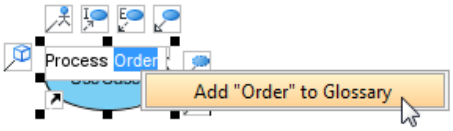
Highlight the specific term on textual analysis, right click on it and select **Add "[the highlighted term]" to Glossary** from the pop-up menu.



Add "archived TV programs" to Glossary

Identify term from shape name

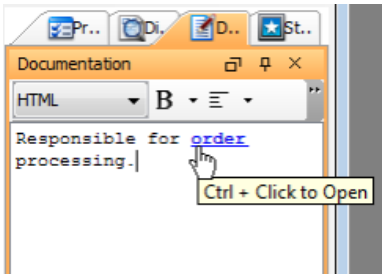
Highlight the specific term when editing a shape inline. Right click on it and select **Add "[the highlighted term]" to Glossary** from the pop-up menu.



Add term to glossary when renaming shape

Opening term

To read the definition of a term, press the **Ctrl** key and click on the term from documentation/flow of events content/shape name. By doing so, the glossary grid will be opened, with the selected term highlighted.



To open a term

Related Resources

The following resources may help you learn more about the topic discussed in this page.

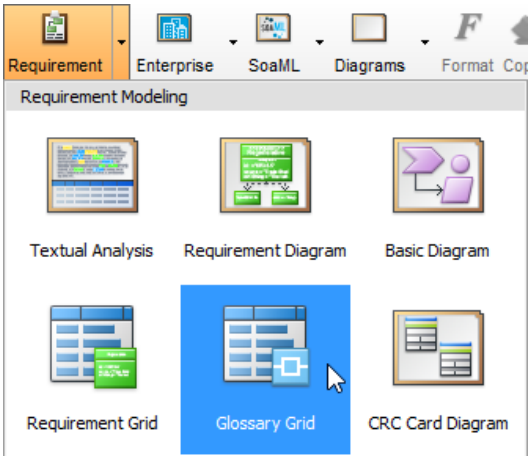
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Using Glossary Grid

Glossary grid is a table where you can identify specific glossary term. In addition, you can define aliases and enter documentation for the glossary term. You can categorize the terms by defining and assigning label(s) to them.

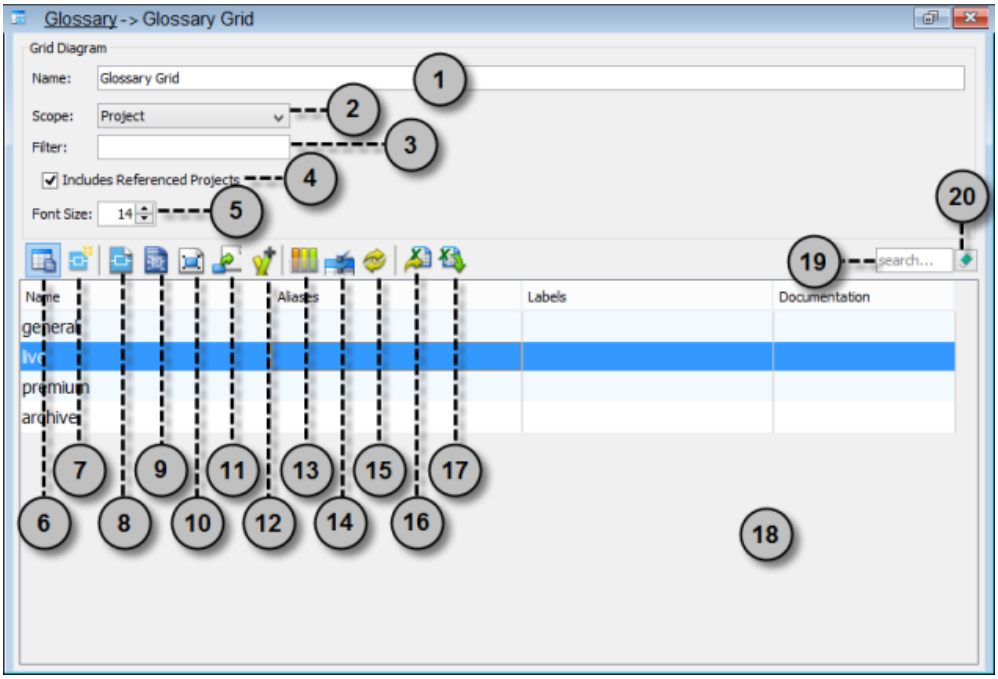
Creating the Glossary Grid

Click on **Requirement** on toolbar and select **Glossary Grid** from the drop down menu **.**



*Open **Glossary Grid***

The overview of Glossary Grid



The Glossary Grid

No.	Name	Description
1	Name	The name of this **Glossary Grid**.
2	Scope	The location to look for the terms to list in grid. By default, terms are found from the whole project. You can change to find terms from specific model or package, or to find only terms right at the root level. You can also restrict the scope to all diagrams, to within a specific diagram or to all terms that has not been visualized in any diagram.
3	Filter	Apply filter to grid content. Text entered here is matched against the **Name** property of terms listed in grid. terms that do not contain the entered text in their name are hidden.
4	Include Referenced Projects	Check it to list also terms in referenced projects, in **Glossary Grid**.
5	Font Size	Click to adjust the font size of text in **Glossary Grid**.
6	Configure Grid	Click to show/hide the grid configuration panel, which allows you to enter the name of grid, the model element to be listed in grid, the scope and to apply filter to grid content.
7	New Term	Click to create a term.

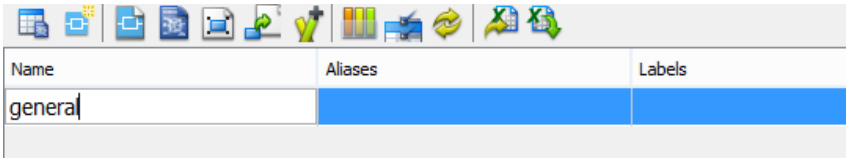
8	Open Term Editor	Select a term in **Glossary Grid** and click this button to open the term editor for editing it.
9	Open Specification...	Select a term in **Glossary Grid** and click this button to open its specification.
10	Show View...	Select a term in **Glossary Grid** and click this button to list the diagrams that contains the view of the selected term.
11	Visualize...	Select a term in **Glossary Grid** and click this button to show it in a new or existing diagram.
12	Add Label to Selected Term(s)	Select a term in **Glossary Grid** and click this button to add labels to it.
13	Manage Label...	Click to add/edit/delete labels.
14	Configure Columns...	Click to select the property(ies) of terms to be listed in the grid, as columns.
15	Refresh	Click to refresh the grid content by showing the most updated information of terms listed.
16	Export to Excel	Click to export grid content to Excel file.
17	Import from Excel	Click to import grid content from exported Excel file.
18	List of terms	Terms are listed here.
19	Search	Find term(s) by entering search criteria.
20	Clear	Click to clear the text entered in **Search** box.

The fields in Glossary Grid

Creating term in Glossary Grid

To create a term in **Glossary Grid**:

1. Click on **New Term** above the **Glossary Grid**.
2. Enter the name of term.



Creating term in Glossary Grid

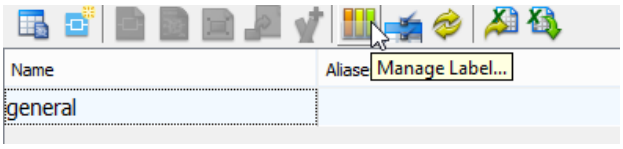
3. Press **Enter** to confirm editing.

Organizing terms with labels

You can categorize the terms by defining and assigning label(s) to them.

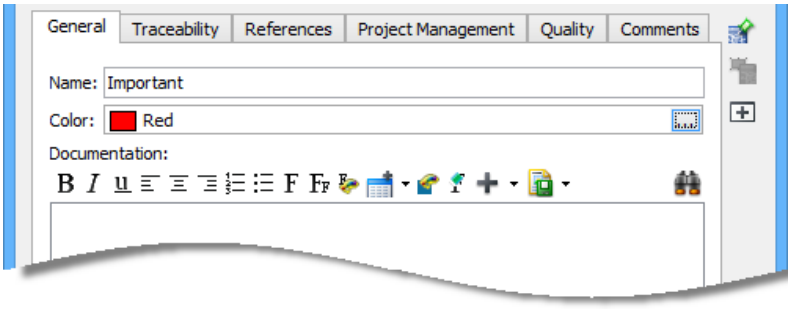
Creating label

1. Click on **Manage Label...** at the top of the **Glossary Grid**.



Manage Label

2. In the **Manage Label** window, click **Add...**.
3. In the **Glossary Label Specification** window, enter the name of label and give it a unique color.

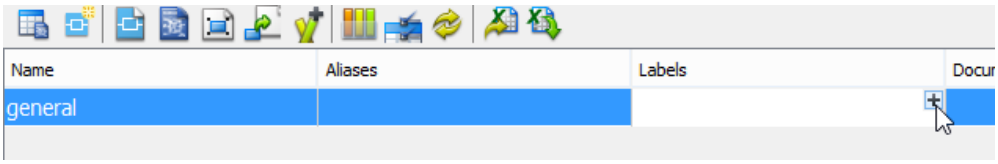


Entered label's details

4. Click **OK** to confirm editing.

Adding label to a term

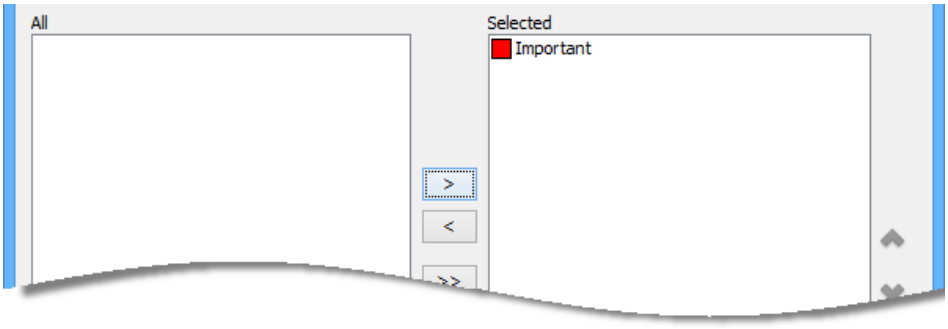
1. Select the desired term in the **Glossary Grid**.
2. Click the **+** button under the **Labels** column.



Add label

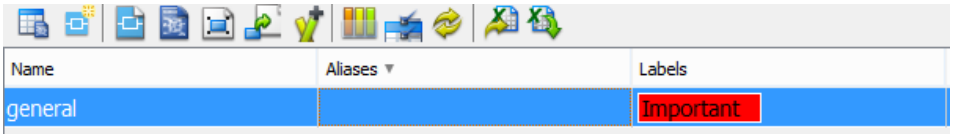
> **NOTE:** If the Labels column doesn't appear, click **Configure columns...**, open the **Properties** tab and select it under the **Others** folder..

3. Select the label(s) to add to the term and click **>**.



Add label

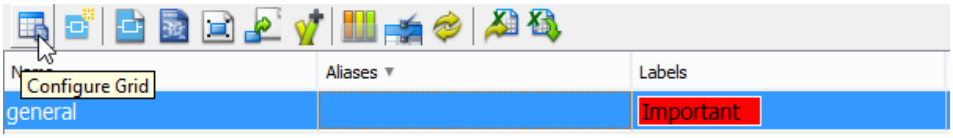
4. Click **OK** to apply and return to the **Glossary Grid**.



Label added to term

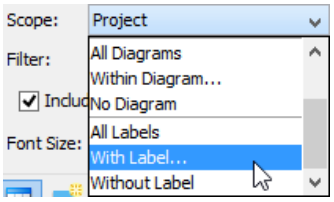
Listing terms by their label

1. Click on **Configure Grid...** at the top of the **Glossary Grid**.



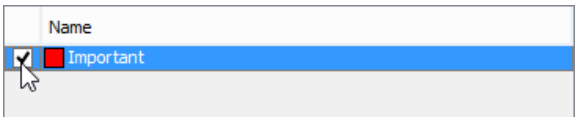
Configure Grid

2. Under the drop down menu **Scope**, select **With Label...**.



Adjust scope

3. In the **Scope** window, select the label to be included into the scope.



Selecting label

4. Click **OK** to confirm and return to the **Glossary Grid**. From now on, only terms that contain the selected labels will be listed in the grid.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Grid diagram

A grid is capable in listing model elements in tabular form, with them appear as row and properties as columns. This chapter will teaches you how to create and configure grid.

Creating grid diagram

The steps required to create a grid.

Creating element in grid

Shows you how to create model element in a grid.

Configuring property columns

As mentioned before, columns of model elements are presented as columns in grid. You will see how to add and remove property columns.

Setting the scope of grid content

Shows you how to set the scope (e.g. project, model, diagram) of grid content.

Filtering and searching in grid

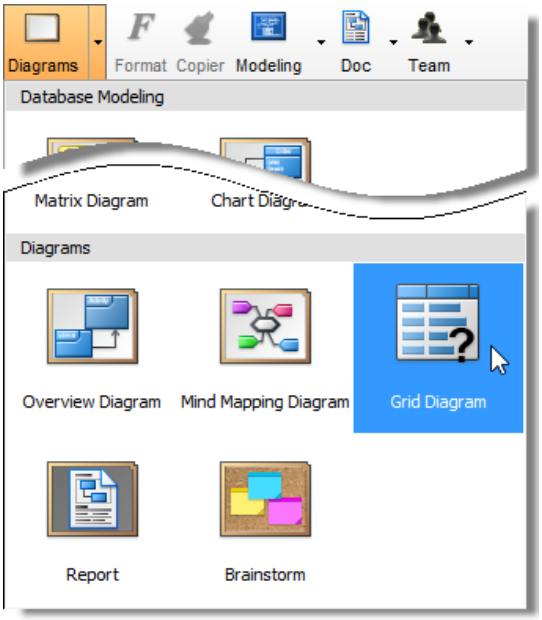
Filter model elements in grid by using filter and search.

Creating grid diagram

The Grid Diagarm provides a convenient way to view specific types of model elements within your project. You can create a grid to list whatever model element type you like, and customize the properties of model element being shown inside the grid.

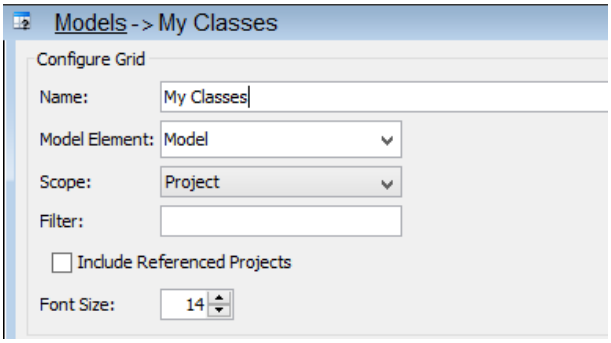
To create a grid:

1. Click on **Diagrams** on toolbar and select **Grid Diagram** from the drop down menu **.**



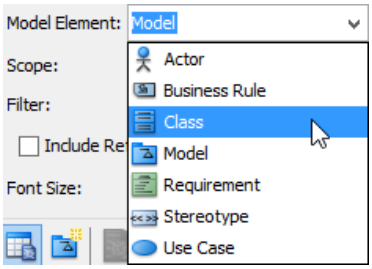
Open Grid Diagram

2. The **Grid Diagram** is opened. Enter the name of the grid.



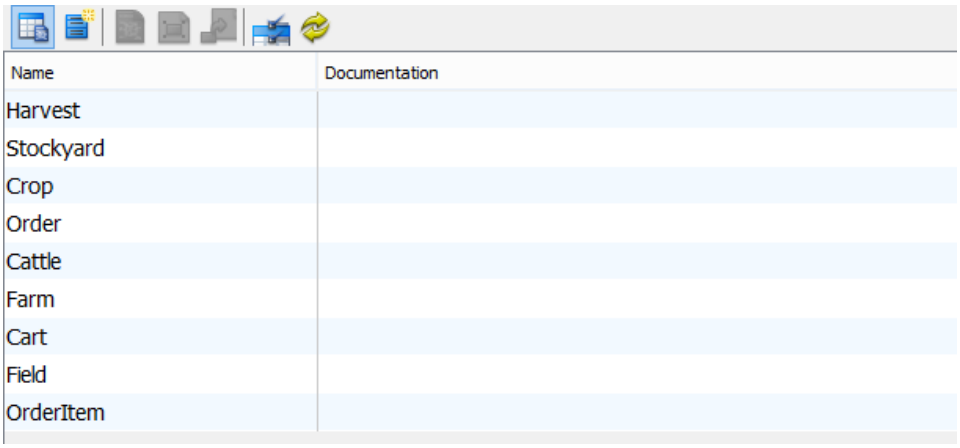
Enter grid's name

3. Select a model element type from the drop down menu of **Model Element**. Click **More...** to list all the available choices.



Select Class

The model element of selected type will then be shown on the grid.



Name	Documentation
Harvest	
Stockyard	
Crop	
Order	
Cattle	
Farm	
Cart	
Field	
OrderItem	

The model element of selected type is shown

Related Resources

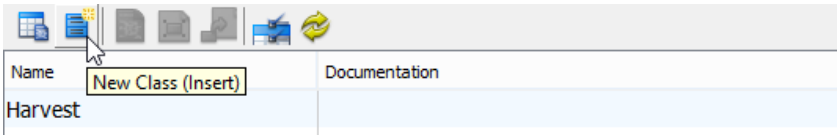
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Creating element in grid

Besides creating elements by drawing them on diagram, you can also create them in grid. Creating elements in grid gives you an overview on all elements of same type.

1. Click on the **New [Element]** button, where **Element** refers to the type of model element you have chosen in the **Model Element**'s drop down menu.



Create a model element in grid

2. Enter name for the newly created model element and then press **Enter** to confirm.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

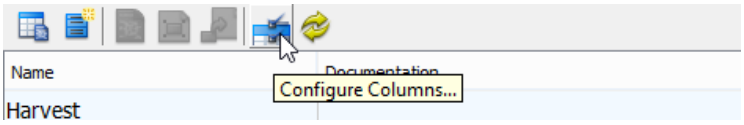
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Adding/removing property columns

In grid, rows represent model elements while the columns show their properties. Name and documentation columns are shown by default, in addition, you can optionally add or remove columns to display the data you are interested in.

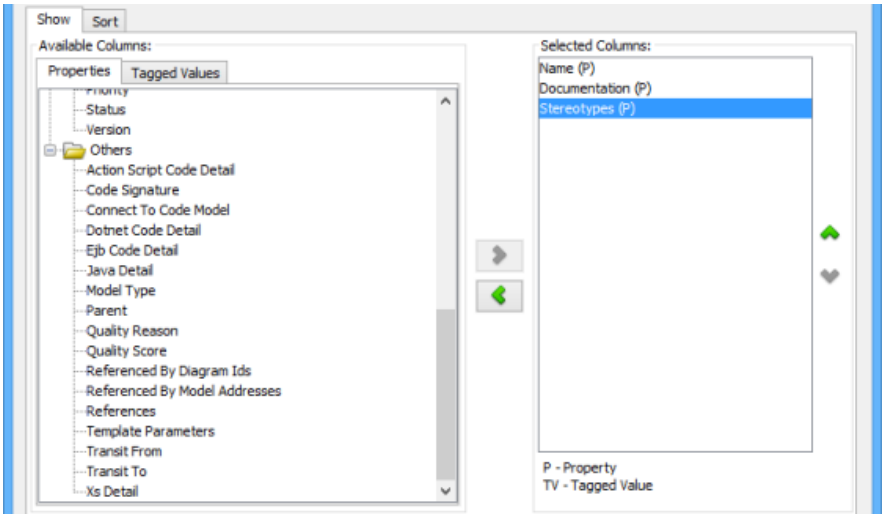
Adding extra property column

1. Click **Configure Columns...** on top of the grid.



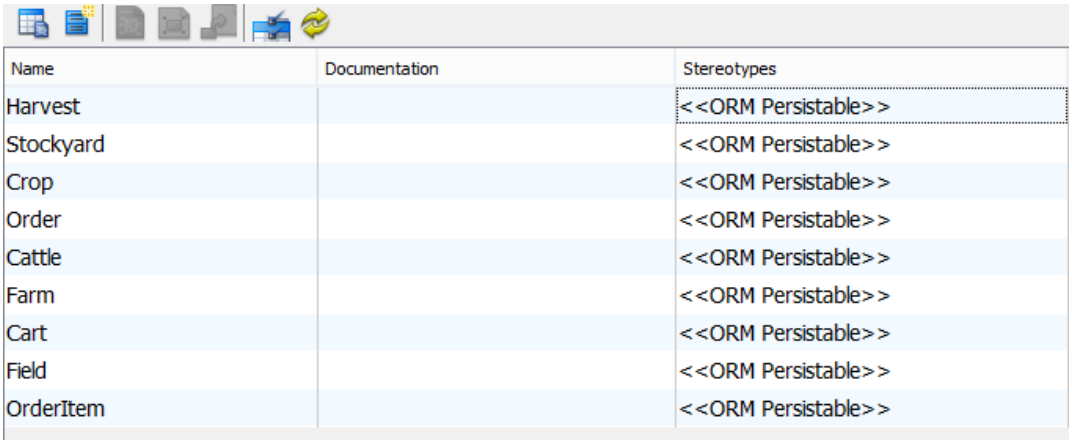
Configure Columns

2. In the **Configure Columns** dialog box, select your preferred property under **Properties** tab and then click **OK** button.



Choose properties

3. Click **OK** to confirm editing and return to the grid.



Name	Documentation	Stereotypes
Harvest		<<ORM Persistable>>
Stockyard		<<ORM Persistable>>
Crop		<<ORM Persistable>>
Order		<<ORM Persistable>>
Cattle		<<ORM Persistable>>
Farm		<<ORM Persistable>>
Cart		<<ORM Persistable>>
Field		<<ORM Persistable>>
OrderItem		<<ORM Persistable>>

Grid updated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

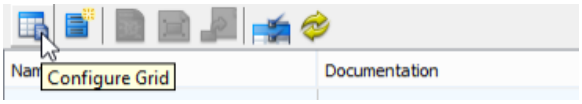
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Setting the scope of grid content

In grid, you can customize the scope of grid content after you have created various types of model elements. After that, your desired model elements within a particular scope will be shown on the grid.

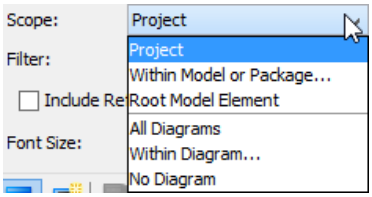
To set the scope of grid content:

1. Click **Configure Grid...** on top of the grid.



Configure Grid

2. Select your desired scope from the combo box of **Project**.



Select scope

> **NOTE:** **Within Model or Package**: Select this to show all selected model element types within a particular model/ package.
> **Root Model Element**: Select this to show all selected model element types under root node.
> **All Diagrams**: Select this to show all selected model element types within all diagrams.
> **Within Diagram**: Select this to show all selected model element types within a particular diagram.
> **No Diagram**: Select this to show all selected model elements without diagram.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

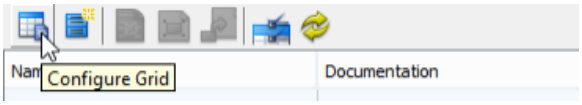
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Filtering and searching in grid

To locate a specific model element or model elements in grid, you can make use of the filtering or searching feature. Filtering excludes rows that do not match with a specified criteria, while searching highlights the rows that matches a specified criteria.

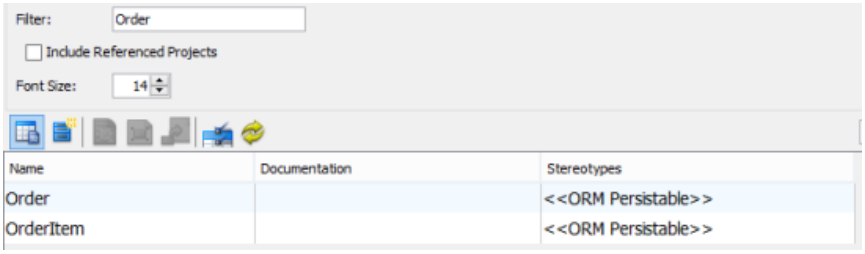
Filtering

1. Click **Configure Grid...** on top of the grid.



Configure Grid

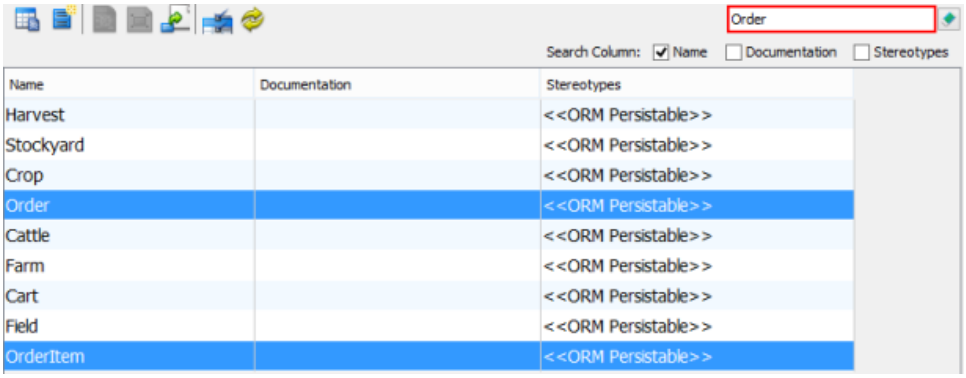
2. Enter in the **Filter** field the name of the model element(s) that you want to find, or part of its name. You can use an asterisk (i.e. the * character) to indicate wildcard characters. Upon typing, the grid is updated to exclude the rows that do not match with the entered text.



Grid content filtered

Searching

Enter in the **Search** field the key words of the model element(s) that you want to find. Key words refer to words that are contained by the properties (columns) listed in grid. You can use an asterisk (i.e. the * character) to indicate wildcard characters. Upon typing, the grid is updated to highlight the rows that match with the entered text.



Searching in grid

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Mind Mapping Diagram

Mind mapping is a tool to help you in brainstorming and organizing ideas, concepts, words, tasks through a visual note-taking way. It sometimes helps to capture requirements and business process, too. You will see how to draw and edit mind mapping diagram in Agilian.

Drawing mind mapping diagram

Draw mind mapping diagram and create mind mapping nodes in diagram.

Formatting nodes

Decorate nodes for better organization of similar ideas.

Linking nodes

Relating concepts with link connectors.

Reference to resources

Maintain reference between node and other artifacts in project.

Relocating a branch

Restruct mind mapping diagram by relocating a branch of nodes.

Layout diagram

Make diagram looks tidier by performing layout.

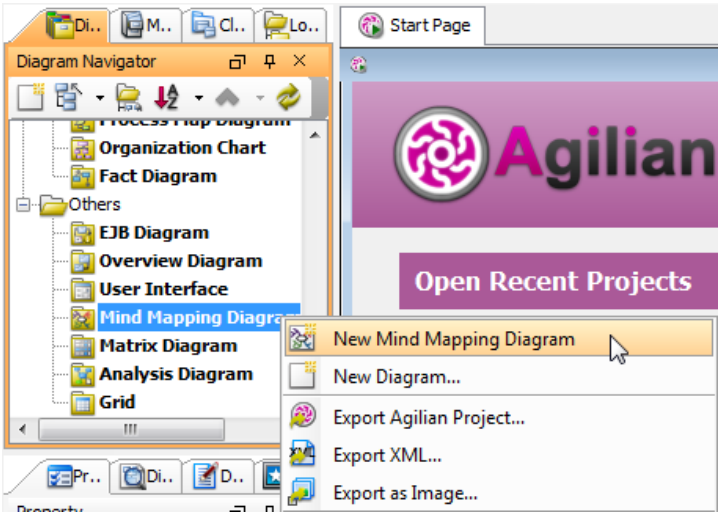
Drawing mind mapping diagram

Mind mapping is a tool to help you in brainstorming and organizing ideas, concepts, words, tasks through a visual note-taking way. It sometimes helps to capture requirements and business process, too. Modelers can create and link model element (such as task, use case, classes) with mind mapping node. The traceability can be kept between initial idea (mind mapping node) and detail design elements (e.g. class).

Creating mind mapping diagram

To create a mind mapping diagram:

1. Right click on the node **Mind Mapping Diagram** in **Diagram Navigator** and select **New Mind Mapping Diagram** in popup menu.



Create a mind mapping diagram

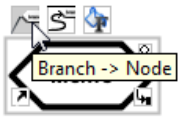
2. This create a mind mapping diagram with a central idea node appear in it. Immediately name the central idea node and press **Enter** to confirm. You can then start drawing the diagram by branching nodes from the central idea nodes.

Naming central idea node

Creating branch with resource centric interface

Mind mapping is formed by nodes that represent ideas or concepts. They are connected with each other, showing the flow of thinking. To create a new branch of nodes from an existing node:

1. Move the mouse pointer over a node. Press on the resource icon **Branch -> Node**.

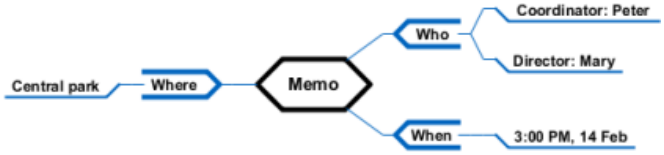


To create a node

2. Drag it out. Release the mouse button to create the node.

A node is created

3. Name the node and press **Enter** to confirm. Create the other nodes by repeat the same steps.



A mind mapping diagram is created

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

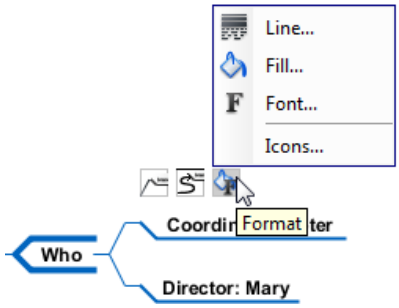
Formatting nodes

You can set colors to nodes to represent different kinds of idea and concepts. You can also set icon(s) to a node to represent the nature of a node, such as a telephone icon for concepts related to contacting somebody.

Changing the line, fill or font style of node

1. Select the node(s) that you want to format. Multiple node selection can be made by a range selection or by pressing the **Ctrl** key and select the nodes subsequently.

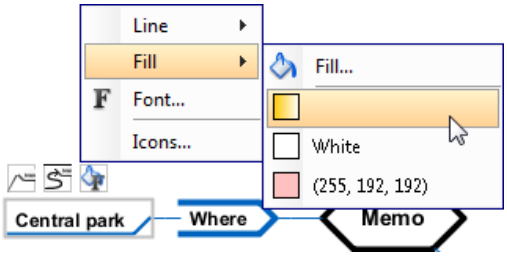
2. Move the mouse pointer to a node within the selection. Click on the **Format** resource icon.



To format nodes

3. Select either **Line...**, **Fill...** or **Font...** from the popup menu to change specific type of format. (Read the coming sub-sections for details about line, fill and font settings)

Once you have confirmed your selection, your choice will be memorized. When you want to apply the settings on other nodes, you can select the new nodes, re-open the same popup menu, and select the setting through the popup menu.



Choosing a color for node

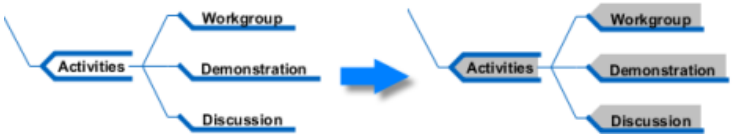
Line

Line settings control the appearance of border around node(s). You can adjust the style (e.g. dash, solid), the weight, which is the thickness of line, the color and the level of transparency.

Nodes with brown border

Fill

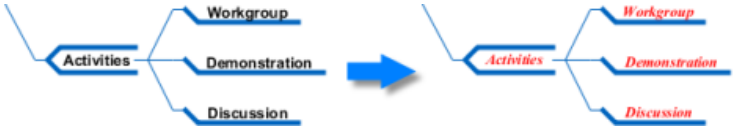
Fill settings control the background color of node(s). You can apply solid and gradient colors, as well as to control the transparency.



Nodes with gray background

Font

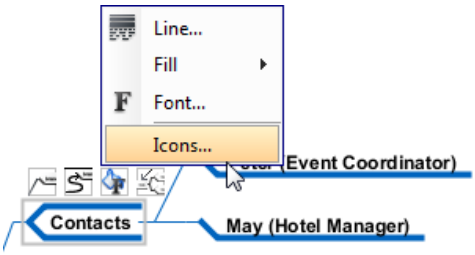
Font settings control the font style, size, type and color of text appear on a node.



Nodes with Times new roman text

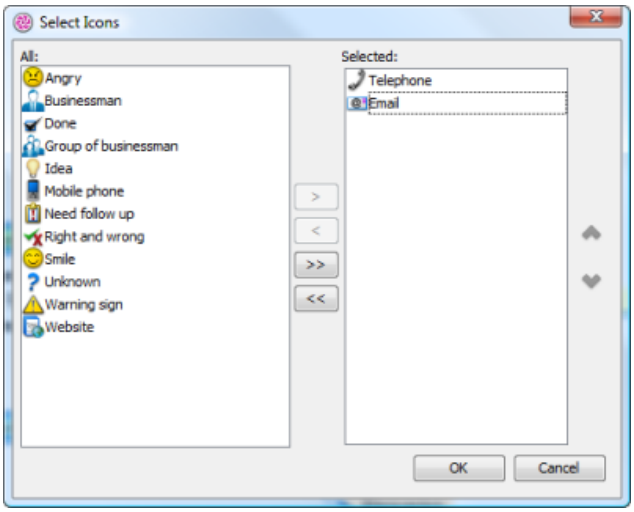
Changing icon of node

1. Select the node(s) that you want to set icon. Multiple node selection can be made by a range selection or by pressing the Ctrl key and select the nodes subsequently.
2. Move the mouse pointer to a node within the selection. Click on the **Format** resource icon, then select **Icons...** from the popup menu.



To edit icons for a node

3. In the **Select Icons** dialog box, select the icon to set and click **>** to assign it to the node(s). Click **OK** to confirm.



To select icons for chosen node(s)

A node with icons

Related Resources

The following resources may help you learn more about the topic discussed in this page.

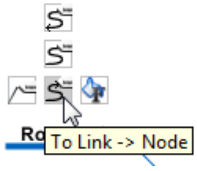
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Linking nodes

Other than the traditional branch connector that represents a generation of idea, you can link related ideas and concepts by using a link connector. There is no exact definition about how two nodes can said to be related. It is up to the designer whether to link the nodes or not. As long as you want to represent that two nodes and related, and the relationship is meaningful, you can add a link between them.

To link nodes:

1. Move the mouse cursor over the source node. Press on any of the resources: Link, To Link, From Link. To and From links are directed relationship, which shows an arrow to indicate the flow from source to target node.



To link to antoher node

2. Drag to the target node and release the mouse button.

> **NOTE:** Unlike traditional resource icons, the Link resources must be released on an existing node. Releasing on diagram will not result in creating a new node.

3. Optionally enter the name of link. Press **Enter** to confirm editing.

Link is created between nodes

Related Resources

The following resources may help you learn more about the topic discussed in this page.

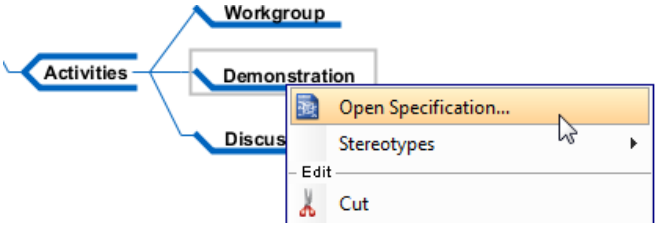
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Reference to resources

You can add references to node, to reference to both internal and external resources such as a shape, a diagram, a file, a URL, etc. For example, to make a node *Prepare Agenda* link to a document of agenda template. This makes a mind map more informative by providing additional information from a mind map which might be casually developed.

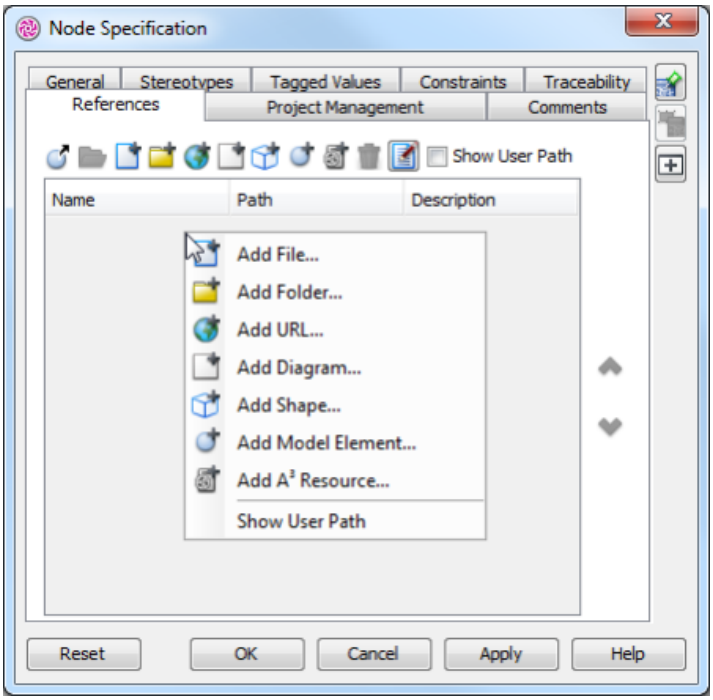
To add a reference:

1. Right click on the node you want to add reference and select **Open Specification...** from the popup menu.



Opening node specification

2. In the node specification, open the **References** tab. Right click on the center of pane and select the type of reference to add from the popup menu.



Going to add a reference

Type of reference	Description
File	An external file.
Folder	An external folder.
URL	A URL. For example, http://www.visual-paradigm.com
Diagram	A diagram in the opening project, such as a requirement diagram.
Shape	A shape in the opening project, such as a use case shape on a use case diagram.
Model element	A model element in the opening project, such as a use case.
A³ Resource	A URL of A³ Platform.

Description of different kinds of references

3. Supply the information of reference such as the file path of a file reference, a diagram for a diagram reference.
4. Click **OK** to confirm.

Once a reference has been added, you can open it from the **References** tab by right clicking on it and selecting **Open...** from the popup menu.

Opening a referenced file

Related Resources

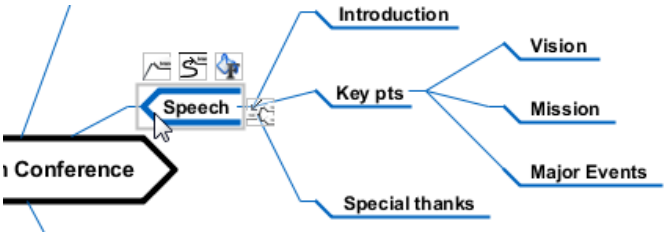
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Relocating a branch

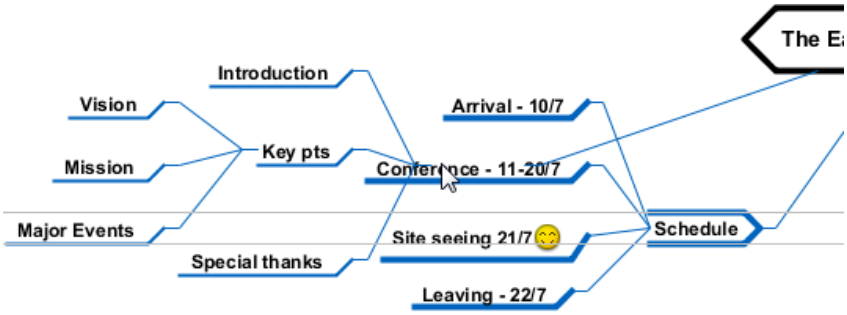
In case a branch of nodes is mis-positioned, you can reposition it to under another node through drag and drop. Here are the steps:

1. Press on the pointer end of the first node of a branch that you want to reposition.



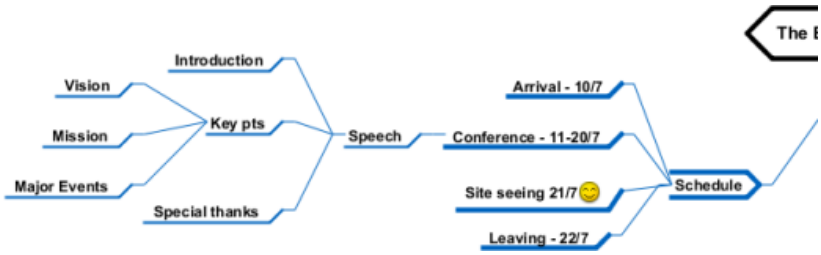
Pressing on the pointer end of a node

2. Drag to node that you want to move the branch to.



Dragging over the target node

3. Release the mouse button.



Mouse button released

Related Resources

The following resources may help you learn more about the topic discussed in this page.

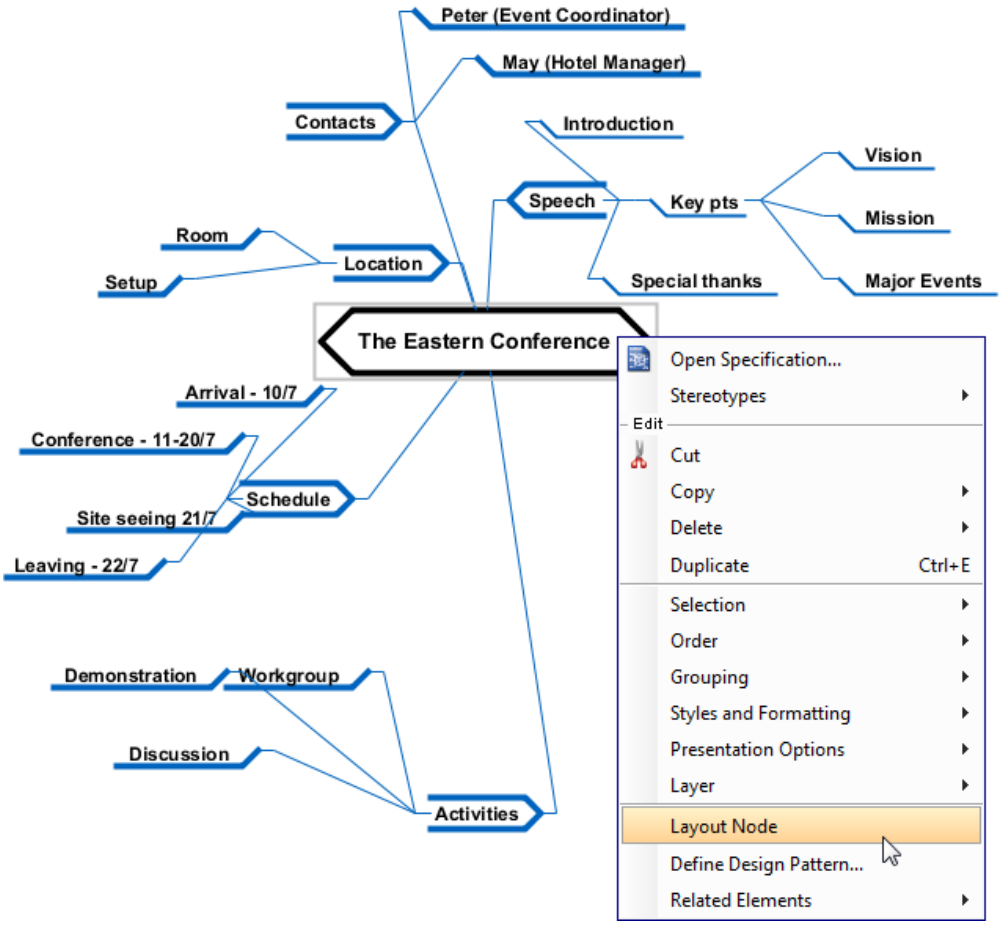
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Layout diagram

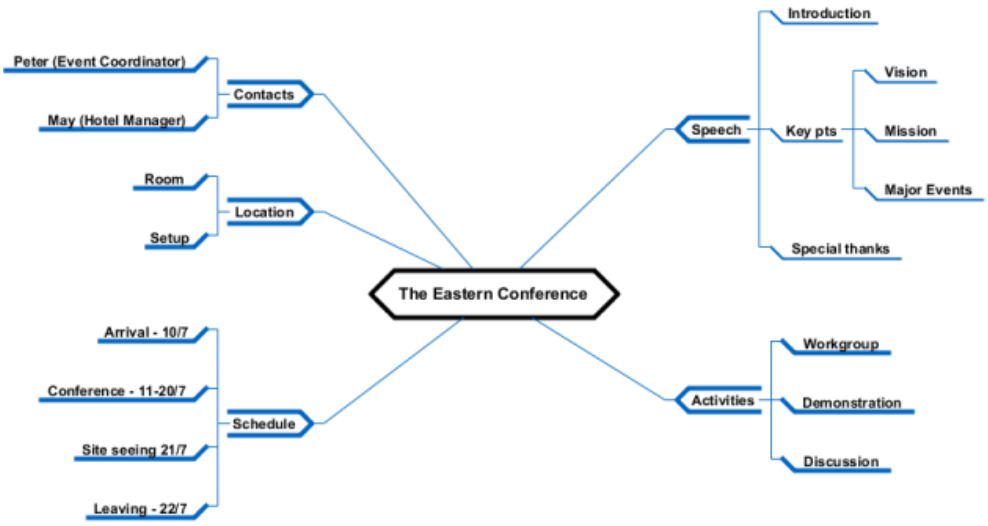
In a mind mapping diagram , ideas are stretching out across, which leads difficulties in tracing nodes with different ideas due to the unorganized nodes. It will be time consuming to rearrange the idea nodes manually. This also affects our brainstorming procedure by caring the tidiness of diagram. By performing a layout, you can keep brainstorming and drawing the diagram without caring about the tidiness of the diagram. You can perform a layout once the diagram is drawn. Any nasty diagrams can be well-organized in a breeze.

Diagram based

By performing a diagram based layout, all idea nodes in diagram are included in the range of layout. To perform a diagram based layout, right click on the central idea node and select **Layout Node** from the popup menu.



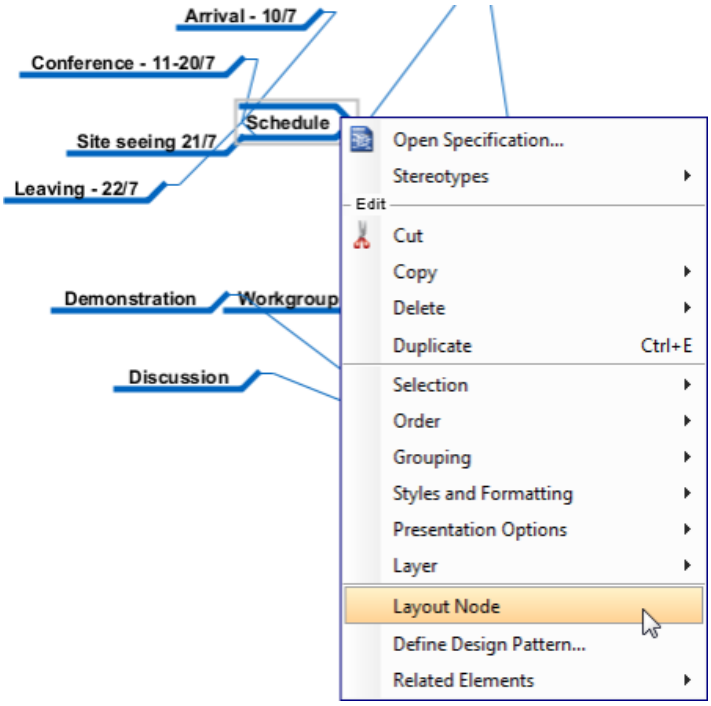
To layout nodes on a diagram



Result of diagram based layout - all nodes are layout-ed

Using resource-centric interface

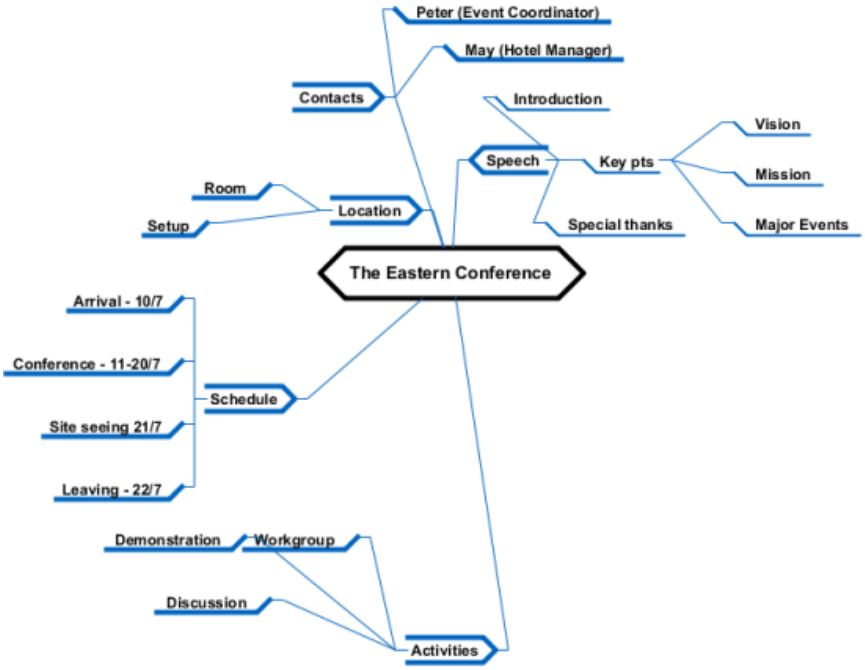
Mouse gesture enables you to layout shapes on a diagram through the movement of mouse. To perform a layout with mouse gesture, right press on the diagram background, sketch a "L" like gesture path and release the mouse button to execute layout.



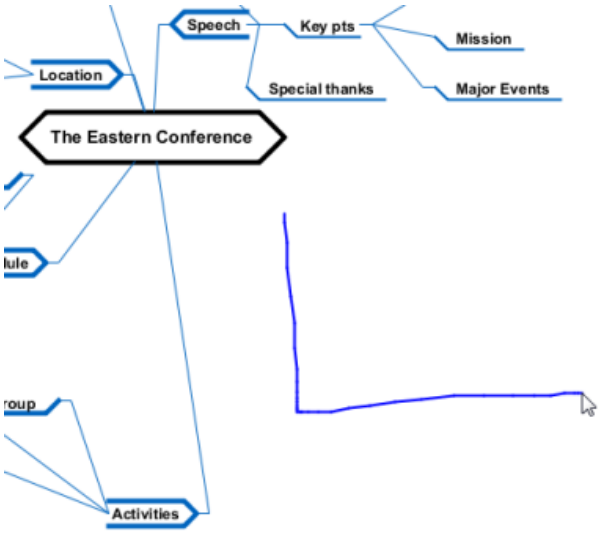
Layout diagram with mouse gesture

Node based

By performing a node based layout, only the chosen node and its descendant nodes are included in the range of layout. To perform a node based layout, right click on the idea node and select **Layout Node** from the popup menu.



To perform a node based layout



Result of node based layout - only Schedule node is layout-ed

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Brainstorm

Agilian enables you to record important ideas during a meeting through a note-taking feature called "Brainstorm". During the meeting, you create note (shapes) in a corkboard -like diagram. When the meeting ends, you may organize the notes, and derive a diagram from them.

Using Brainstorm

Introduce to Brainstorm and shows you how to create a Brainstorm diagram.

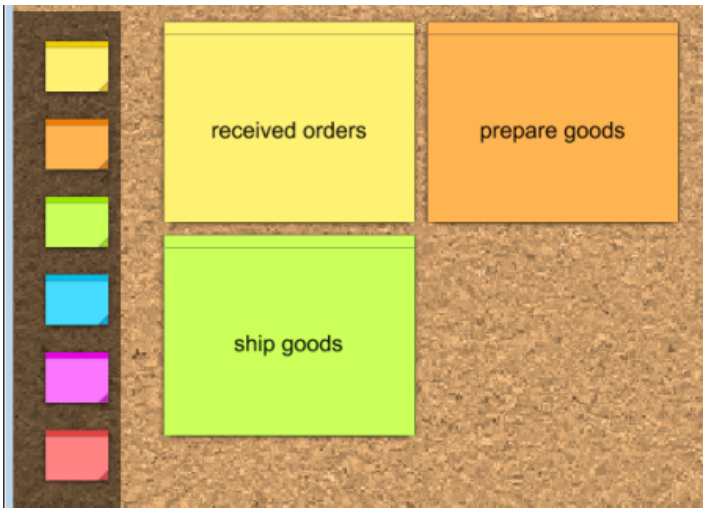
Realize Brainstorm Notes

Teaches you how to realize notes into model elements.

Using Brainstorm

Very often, you model a system not just by imagination, but with facts, knowledges and customers' ideas. You may meet with users, understand how they work, identify their requirements and proceed to visualize their needs with models.

Agilian enables you to record important ideas during a meeting through a note-taking feature called "Brainstorm". During the meeting, you create note (shapes) in a corkboard -like diagram. When the meeting ends, you may organize the notes, and derive a diagram from them. This helps ensure all important thoughts from users are well recorded and won't be lost when constructing a model.



A sample Brainstorm diagram

Creating Brainstorm diagram

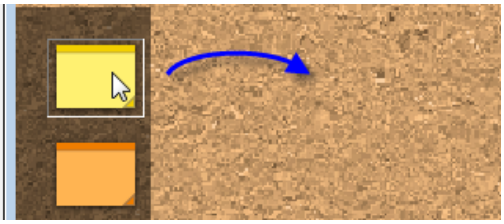
In order to start taking notes, you need to create a Brainstorm diagram. There are three ways you may take to create a Brainstorm diagram:

- Diagram Navigator - Right click on **Brainstorm** in Diagram Navigator and select **New Brainstorm** from the popup menu.
- Toolbar - Click on the **Diagrams** button in toolbar and select **Brainstorm** from the drop-down menu.
- Menu - Select **File > New Diagram > Others > Brainstorm** from the main menu.

Creating a note

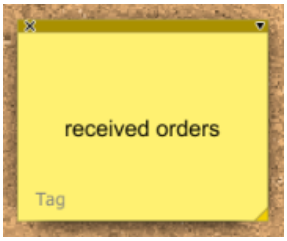
Brainstorm diagram applies a corkboard theme. You may add notes to it to record ideas and thoughts collected. To create a note:

1. Press and drag in diagram toolbar a note with the desired color.



Creating a note

2. Release the mouse button on the diagram to create a note.
3. Enter the note content. Press **Ctrl-Enter** to finish editing.



A note is created

Imagine when you are having a meeting with user and taking notes with Brainstorm, you may not be perfectly sure whether the notes you create are utlimately important or not. As long as you think that the information may help you model, it is worthwhile to note it down for now. The key idea is to add notes in a casual manner. Do not spend too much time on judging the importance of the content. Otherwise, you may miss out information that is really important during the process of strenuous sorting.

Deleting a note

If a note has been created by mistake or the note text is no longer correct/meaningful, you may want to delete the note. To delete a note, you may select it and press the **Delete** key. Alternatively, press on the cross button at top left of note shape to perform a deletion.

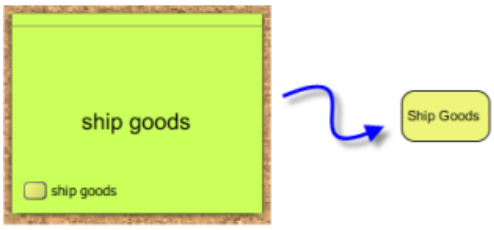
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Realizing Brainstorm Notes

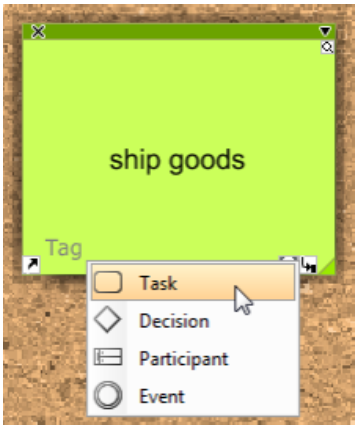
Agilian enables you to record important ideas during a meeting through a note-taking feature called "Brainstorm". During the meeting, you pay attention to what the participants say and create note (shapes) in a corkboard -like diagram to record the key points. After the meeting, you may make use of the notes collected to help you construct the model. To optimize the process, Agilian supports a realize function to transform notes into model elements.



Producing a task from a note

To realize a note:

1. Look for the note that can help you construct a model element. For example, when you see a note with text "ship goods", you may want to create a task *Ship Goods* from it.
2. Click on **Tag** at bottom left of note body. Select the type of element from the popup menu, such as Task.



Select note type

3. Click the tag at bottom left and select **Realize...** from the popup menu.
4. In the **Transit Model Element** window, enter the properties of the model element you are going to produce. Click **OK** at bottom right to continue.
5. The **Visualize Model Element** window enables you to show the model element on a diagram. You may show it on a new diagram by selecting Create new diagram and entering the diagram name. Or, show it in an existing diagram by selecting Show in existing diagram and selecting the diagram to show. Or, not to show it on any diagram by selecting Do note visualize. Click **Create/ Show/ Close** at bottom right.

Changing note tag

If you have selected a tag for a note and you want to change it, click on any tag at bottom left of note shape and select **Type > [New Tag]** from the popup menu. This will clear the previously select tag and apply the newly selected one. Note that once a tag has been realized, you cannot change it to another type anymore.

Adding note tag

You may add multiple tags to a note and realize multiple model elements from different tags (one model element per tag). To add another tag, click on any previously added tag at bottom left of note shape and select **Add > [New Type]** from the popup menu.

Deleting note tag

If you think that a note tag is no longer suitable for the note, you may delete it. To delete a tag, click on the tag you want delete from note and select **Delete** from the popup menu. Note that if the tag has been realized, deleting the tag would not delete the realized model element. And if you add the tag again, the realization relationship will be maintained.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Creating diagrams

In this chapter, you will learn not only how to create diagrams, but also some of the frequently used functions that aid in diagramming, including resource-centric interface, using tagged values to add custom properties and spell checking.

Creating diagrams

You will see how to create a diagram, how to create shapes and draw freehand shapes.

Model element and view

When model element is seen as data, view is a representation of data. This page tells you more about the relationship between model element and view.

Master view and auxiliary view

Multiple views can be added for model elements. Among the views, one master view can be set. The rest are called auxiliary views. This page provides you with more information about views.

Resource centric interface

Resource-centric interface provides you with a set of function icons around shapes, for triggering functions in quick. You will learn how to make use of resource-centric interface to create and connect shapes, and know the differences between types of resource icons.

Tagged values

Tagged values can be used to add domain specific properties to shapes. There are several types of tag, such as text, integer, floating point number, etc. You will see how to create tagged values.

Spell checking

Spell checking is an automatically enabled feature for ensuring the accuracy of written content such as shape name and documentation. You will see how to configure spell checking.

Creating diagrams

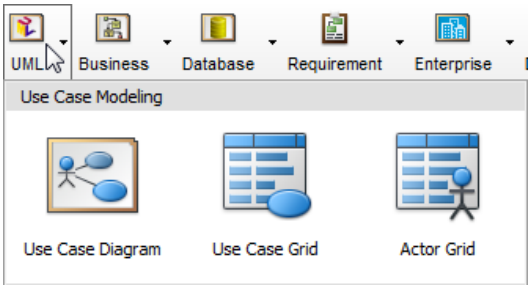
You can create diagrams to help visualize what you did, what you are doing and what you need to do on your target system or application. There are different types of diagrams to fulfill different needs, categorized as UML Diagrams, requirements capturing, database modeling, business process modeling and others.

Ways to create diagram

There are three ways of creating diagram: toolbar, menu and Diagram Navigator.

Toolbar

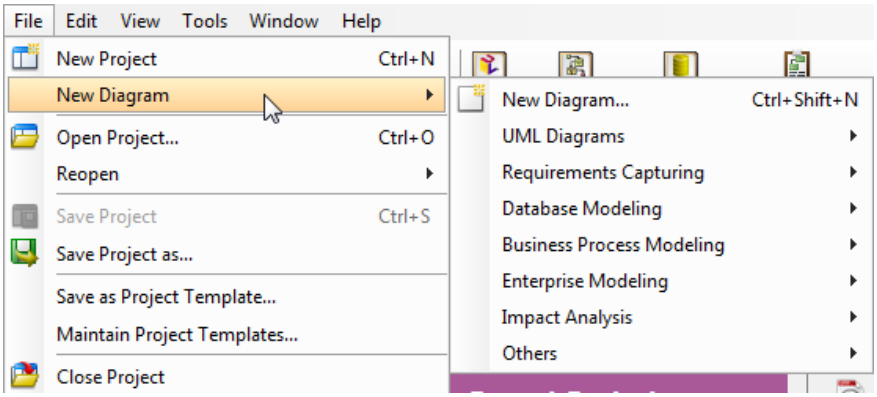
Click on a button of category in toolbar and select the type of diagram to create in the drop down menu.



Create diagram through toolbar

File Menu

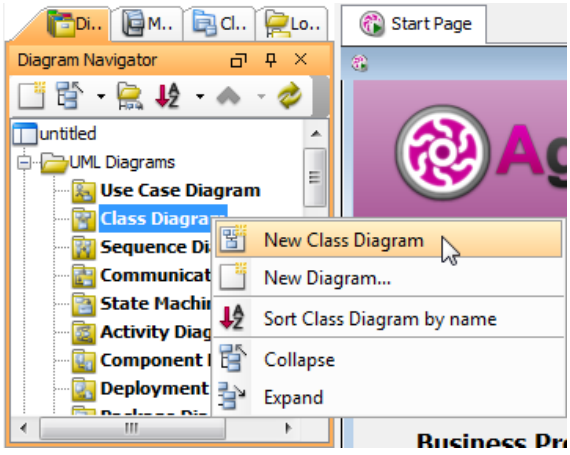
Select **File > New Diagram** from the main menu. Then, select the diagram category and the type of diagram to create.



Create diagram through menu

Diagram Navigator

Right click on the type of diagram to create and select **New [Type] Diagram** from the popup menu (e.g. **New Class Diagram**).



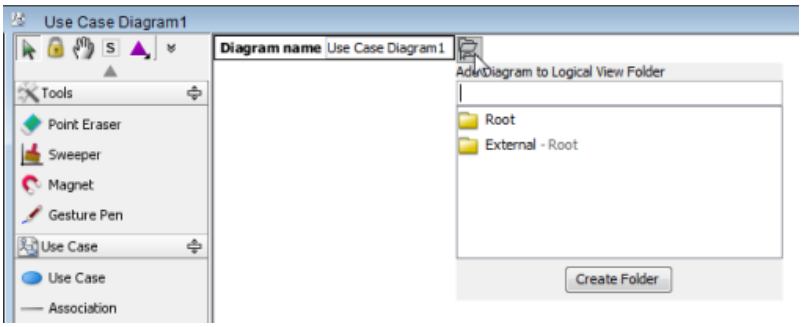
Create diagram through Diagram Navigator

Creating and place diagram in a logical view

Logical view enables you to categorize diagrams in your own way by using view folders. When you create diagram, you may enter diagram name immediately. At the same time, if there is at least one logical view in your project, you may select the view to store the diagram.

To select a logical view when creating diagram:

1. Click on the button next to the box of diagram name.



Create and place diagram in a logical view

2. Select the view folder. If you want to create a new folder to store the diagram, click **Create Folder**.

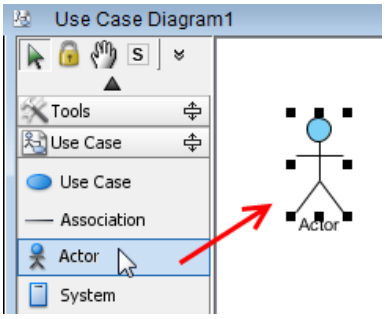
Drawing shapes

After creating a new diagram, diagram elements can be created as well through the diagram toolbar. In this section, these techniques of drawing shapes will be explicated:

- Creating Shapes
- Creating Connectors
- Creating Self-Connection

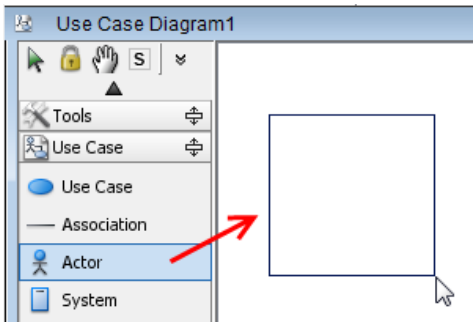
Creating shapes

To create a shape, click a diagram element from the diagram toolbar and click it on the diagram pane for creating. The generated element will appear to have a default size.



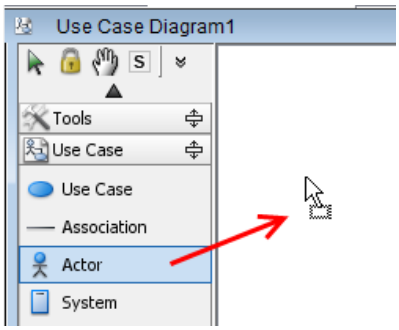
Click an actor on the diagram pane

For defining a specific shape size, drag a specific boundary with the mouse after clicking a diagram element from the diagram toolbar.



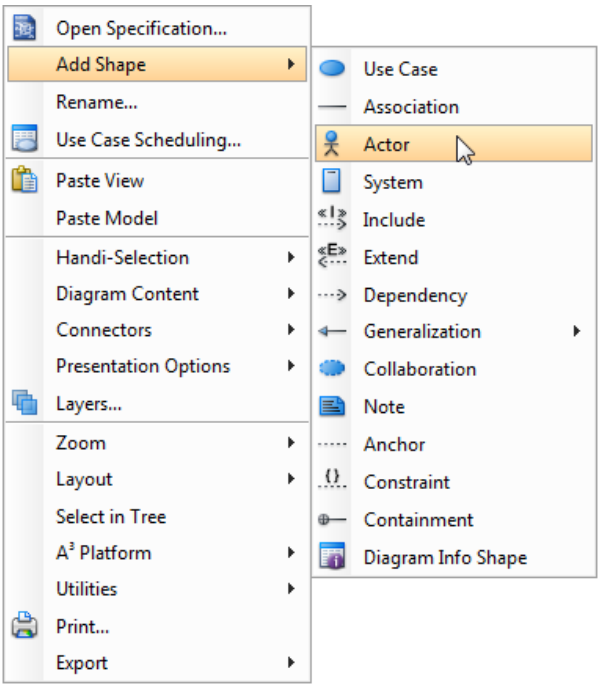
Drag a specific boundary with the mouse

Alternatively, a diagram element can be created by dragging the diagram element and then dropping it on the diagram pane.



Drag and drop to create a shape

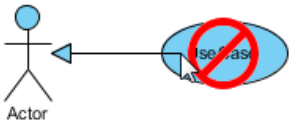
In addition, you can add a shape through the pop-up menu of diagram. Right click on the diagram background, select **Add Shape** and then a specific shape from the pop-up menu.



Create a shape through the pop-up menu of diagram

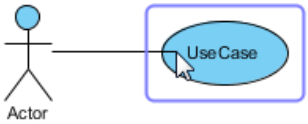
Creating connectors

To create a connector, select the desired connector from the diagram toolbar, drag the connector from the source shape to the destination shape. Since Agilian provides a continuous UML syntax checking, if you create an invalid connection, a stop sign will be pop-out. For instance, you are not allowed to connect an actor and a use case with a generalization relationship.



An invalid connection is created

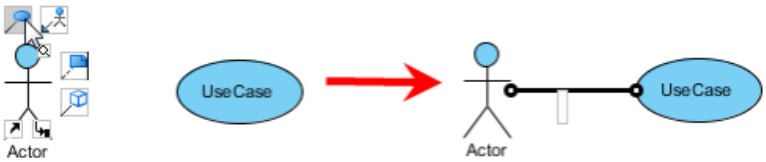
If the connection is valid, a blue rectangle surrounding the destination shape can be seen.



A valid connection is created

Moreover, connectors can be created through resource icons:

Move the mouse over the source shape, press one of its resource icons and drag it to the destination shape.

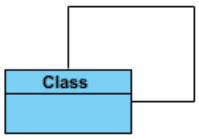


Press a resource icon to connect shapes

If you release the mouse on an empty space, a new shape will be created with a connector.

Creating self-connection

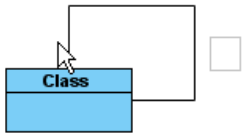
Some shapes can make a connection for itself, for example, **Self-Association** of a **Class** in class diagram and **Self-Link** of an **Object** in communication Diagram. To create a self-connection, select the connector from the diagram toolbar and then click on the target shape. Alternatively, you can press the target shape's resource icon directly.



Self-Connection is created

Creating self-connection

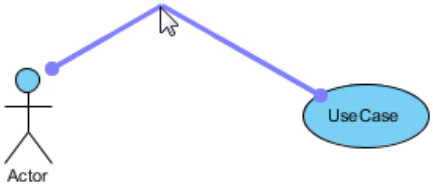
Some shapes can make a connection for itself, for example, **Self-Association** of a **Class** in class diagram and **Self-Link** of an **Object** in communication Diagram. To create a self-connection, select the connector from the diagram toolbar and then click on the target shape. Alternatively, you can press the target shape's resource icon directly.



Self-Connection is created

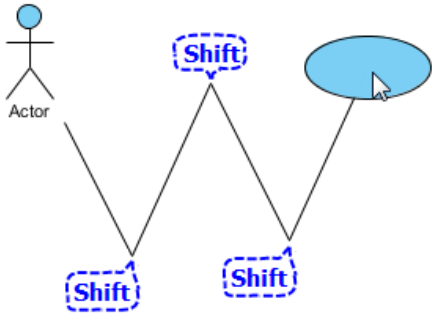
Creating turning point on connector

A turning point is a point on a connector where a bending take place. To create a turning point on an existing connector, press on the connector and drag to bend the connector.



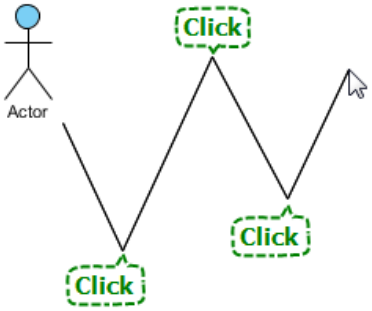
Create turning point on existing connector

You can also create a turning point when creating a creator through the resource centric interface. When dragging out a resource-icon, press the Shift button at where you want to create the turning point.



Create turning points when creating shape by dragging a resource

If you try to create connector by clicking on a resource icon, click at where you want to create a turning point to create it.



Create turning points when creating shape by clicking a resource

Resource-centric interface

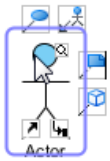
Visual Paradigm is the first vendor to introduce the resource centric diagramming interface. The resource centric interface greatly improves the efficiency of modeling. You don't have to traverse between the toolbar and the diagram to create diagram elements, make connections and modify the diagrams. The resource centric interface can make sure the modeler is able to create a diagram with correct syntax more quickly.

There are tree types of resource icons:

- Connection Resource

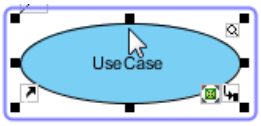
- Manipulation Resource
- Branching Resource

Connection resource
It is designed for creating elements and making connections.



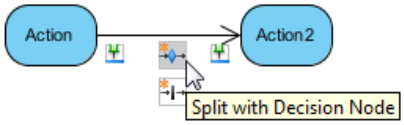
Create association with connecting a new or existing use case

Manipulation resource
You can use Manipulation Resource to modify properties or appearance of elements. For example, you can show or hide compartments, add references, add sub-diagram and fit size.



Fit size through manipulation resource

Branching resource
Branching Resource helps you to create decision structure in diagram. An orange asterisk appears on the split resource icon.

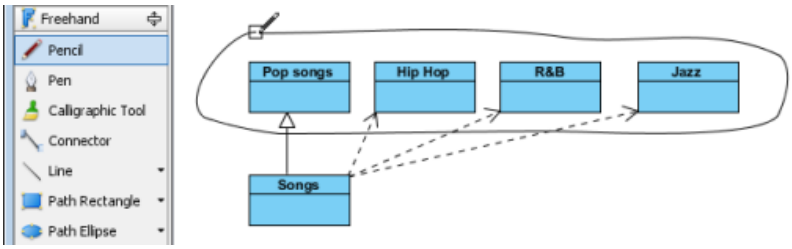


Create decision structure through branching resource

Drawing freehand shapes

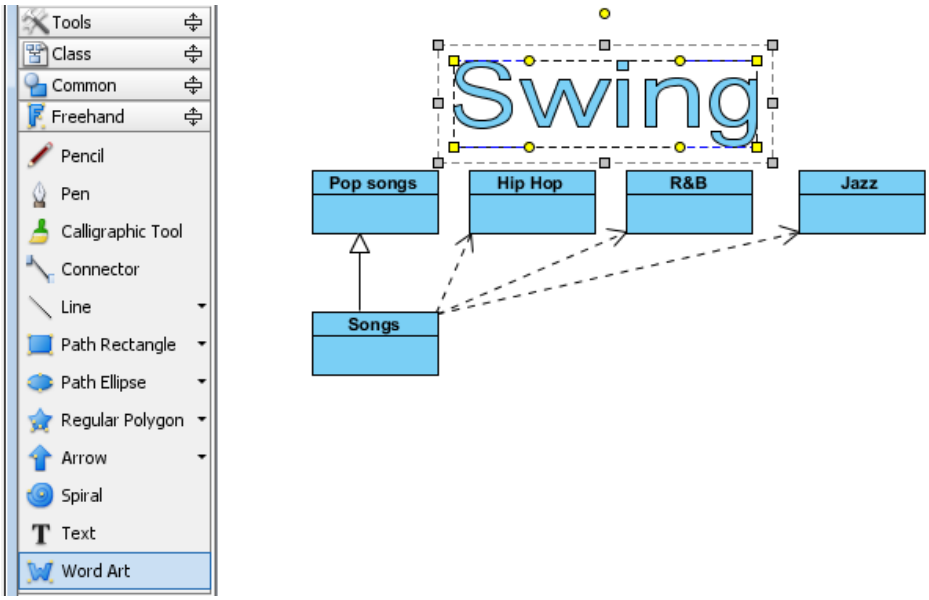
Freehand shape is a kind of general graphic shapes. Pen shapes, pencil shapes, and some regular shapes like circle, rectangle and arrow are available. Freehand shape can be used for annotating diagram. For example, you can use freehand shapes to emphasize some shapes.

A specific shape can be highlighted with a pencil shape.



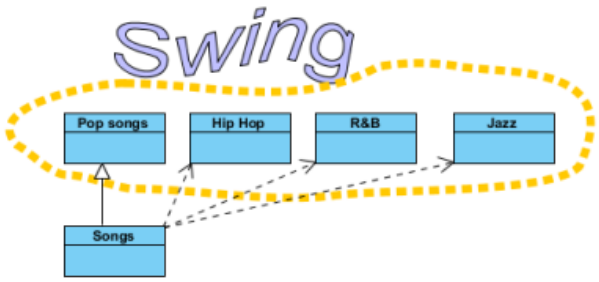
Pencil

An outstanding text can be shown with word art.



Word Art

A freehand shape style can be formatted in order to address your information.



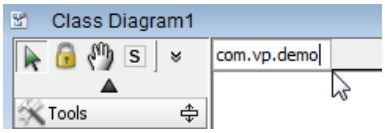
Formatted freehand shapes

Changing package header

You can specify the parent package of any diagram through Package Header.

Package header when diagram created

When a diagram is created, the package header will be unfolded as it allows you to specify the parent package of the diagram. Specify the package by entering the fully qualifier name of the package.



Specify parent package in package header

After entering the name of parent package, you will find that the diagram name is the same as the name of parent package.

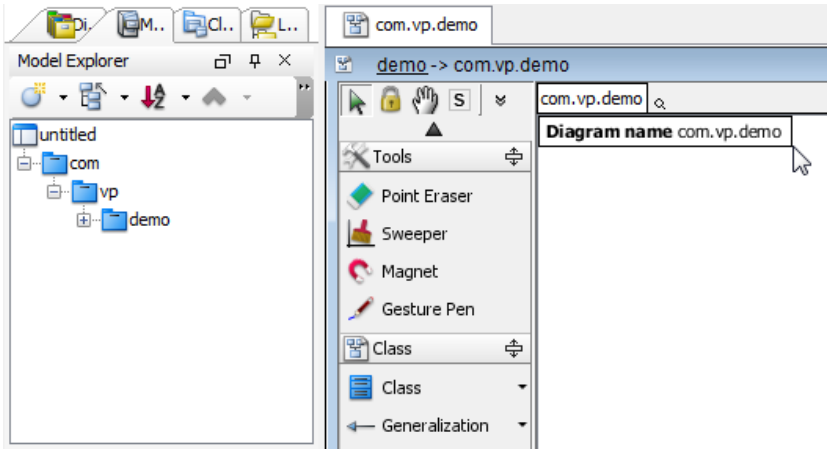
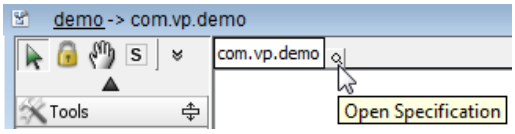


Diagram name will be same as fully qualify of parent package

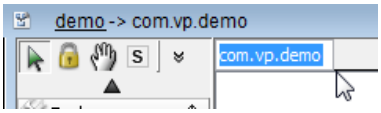
The diagram name can be renamed. However, the name of parent package won't be changed following with the diagram name.

You can open specification of parent package by pressing the **Open Specification** button on the right-hand side of the parent package name.



Open Specification

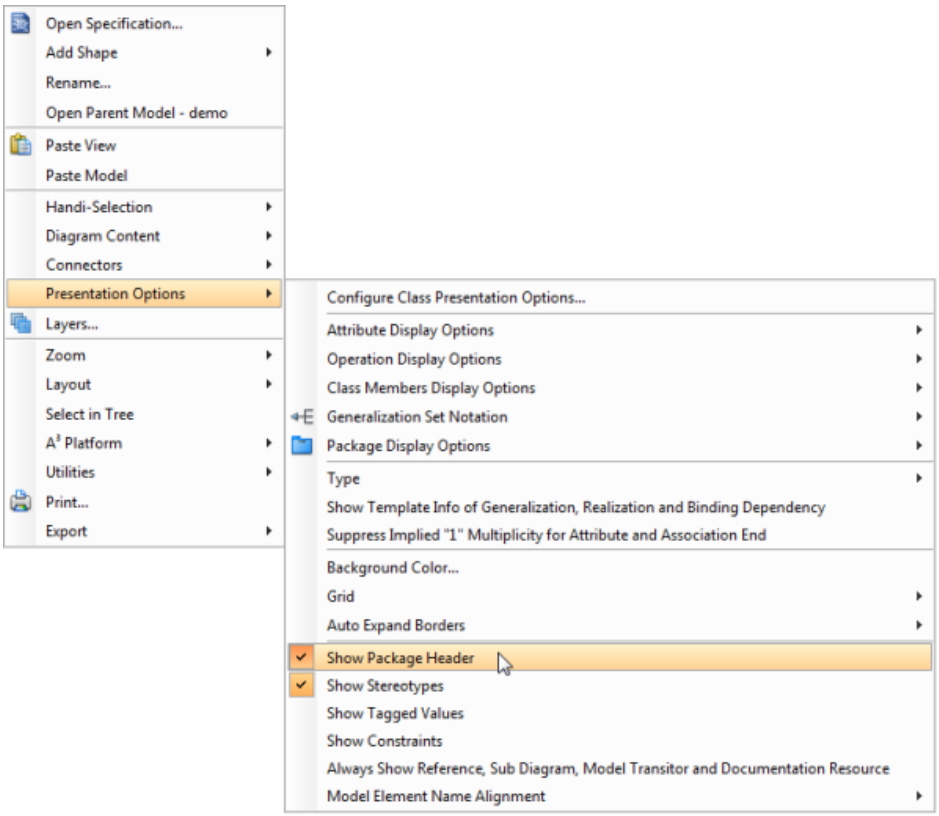
You can rename the parent package of the diagram by double clicking on it.



Double click on the parent package

A new package will be created if you enter a completely new package name. If the previous parent package does not contain elements, it will be deleted. That means the documentation (or other properties) of previous parent package will be lost.

A package header can be either shown or hidden through the pop-up menu of diagram. Right click on the diagram background and select **Presentation Options > Show Package Header**from the pop-up menu.



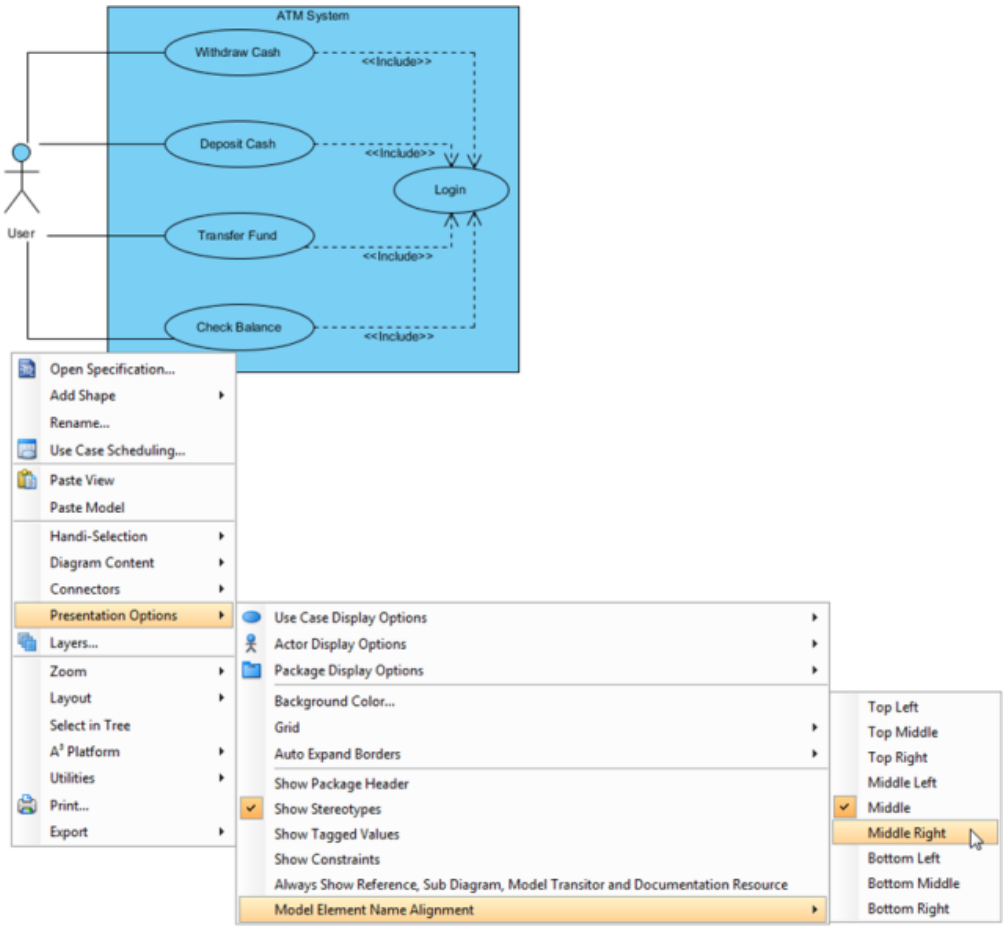
Show/hide package header

Justify shape name

In Visual Paradigm, a shape name is aligned center horizontally, and top or middle vertically, depending on the characteristic of shapes. However, the shape name can be realigned. Even if a language, such as Modern Hebrew, that is written from the right to the left can be displayed on a shape clearly.

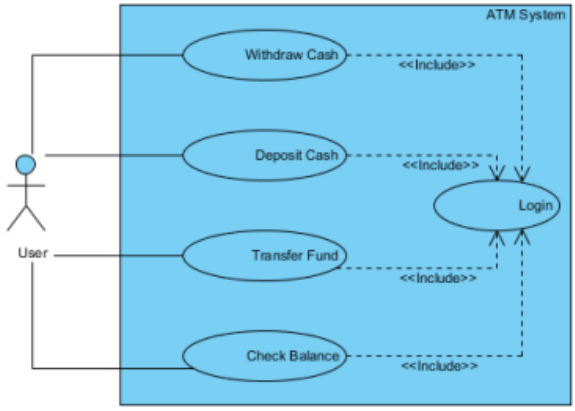
Adjusting shape name's position

1. Right click on the diagram background, select **Presentation Options > Model Element Name Alignment** and then select a specific alignment option from the pop-up menu.



Select an alignment option from the pop-up menu

2. As a result, all shapes' name in the whole diagram will turn into the alignment option you set previously.



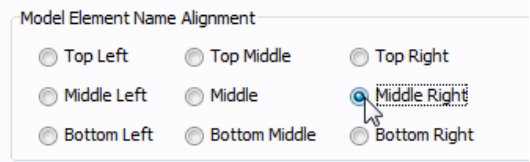
All shapes' names turn into middle right

Apart from the whole diagram setting, a specific shape can also be set:

1. Right click on a shape, select **Presentation Options > Model Element Name Alignment** and then select a specific alignment option from the pop-up menu.
2. As a result, the specific shape will turn into the alignment option you set previously.

In addition to the current diagram, future diagrams can also be set:

1. Select **Tools > Project Options...** from the main menu.
2. In the **Project Options** dialog box, select the **Diagramming** category, check an option out of **Model Element Name Alignment** section under the **Appearance** tab.



*Check an alignment option in the **Options** dialog box*

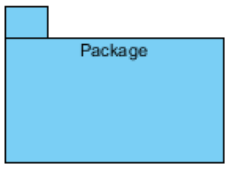
Exceptions

Although most shape' name can be justified, some are exceptional. Two main kinds of shapes that their name cannot be justified are introduced as follows:

On one hand, shapes neither with floating name label (freely movable) nor with a label outside the shape can be justified. Actor, Initial Pseudo Node and BP Start Events are examples of this kind of shape.

Floating name label

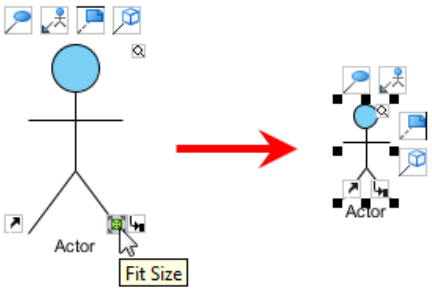
On the other hand, the names of container shapes are not available for positioning. Since their “bodies” are used for containing other shapes, thereby, they have a limited scope of displaying names. Package, State and System are example of this kind of shape.



Container shape

Enable and disable minimum shape size

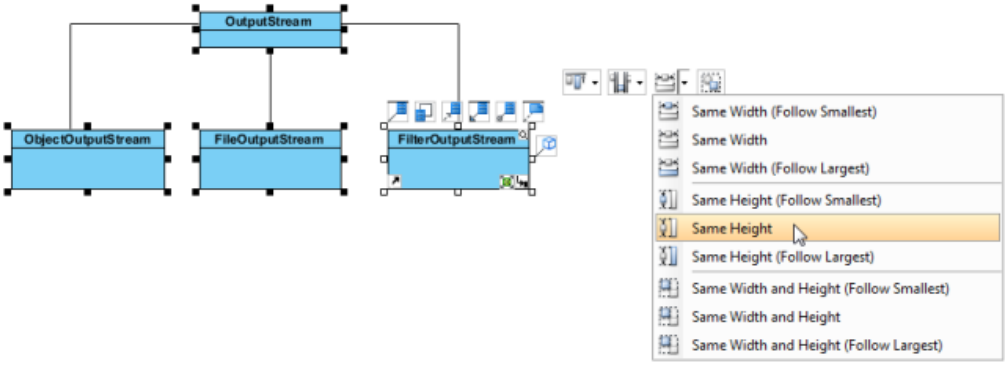
Since all shapes have their own default minimum size, users are not allowed to resize them to smaller than the minimum size. The default setting helps to ensure those compact shapes are clear enough to be seen on a diagram under normal circumstance. The minimum size of a shape can be determined by pressing its fit size button.



The minimum size of a shape

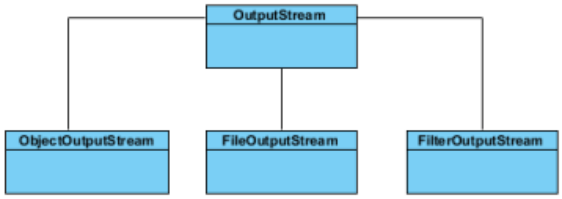
Now, it is possible to disable such setting, so that shapes can be resized to extremely small in size, despite their minimum size:

1. To disable the function of the minimum size checking, select **Tools > Project Options...** from the main menu.
2. In the **Project Options** dialog box, select the **Diagramming** category and uncheck **Enable minimum shape size when resizing shape** under **Appearance** tab. Click **OK** to confirm.
3. After that, you can resize a shape to the size as small as you want.
4. Furthermore, select other shapes and select an option from the drop-down menu of resource icon **Same Width**.



Select an option from the drop-down menu of resource icons

5. As a result, other shapes will turn into the same size as the shape you have done previously.



All shapes are turned into the same small size

Undo and redo

During the process of editing a diagram, you may make some careless mistakes, such as accidentally deleting a shape. You can use the **Undo** function to cancel the previous action. On the contrary, you may re-perform the action through the **Redo** function. Note that the undo/redo feature in Agilian is diagram based.

Undo

You can roll back undesirable changes by performing **Undo**. Undo function can be executed in the following three ways:

- Select **Edit > Undo** from the main menu.
- Click the **Undo** button on toolbar.
- Press **Ctrl-Z**.

Redo

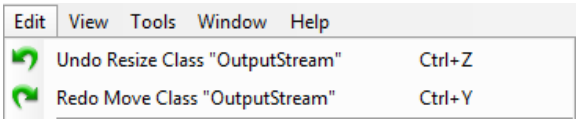
This feature is to re-perform actions that have just been undone. Redo function can be executed in the following three ways:

- Select **Edit > Redo** from the main menu.
- Click the **Redo** button on toolbar.
- Press **Ctrl-Y**.

Showing name for undo and redo action

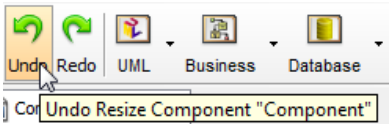
It's hard for us to remember all actions we have done previously. By Agilian, we can recall the actions we have done before.

You can find an action name of undo/ redo by clicking **Edit** from the main menu.



Menu shows Undo/Redo name

On the other hand, you can also find the action name of undo/ redo on toolbar button's tooltip. Move the mouse over the **Undo** or **Redo** button and then a tooltip with **Undo**/ **Redo** action name will appear.



Toolbar button's tooltip shows undo action

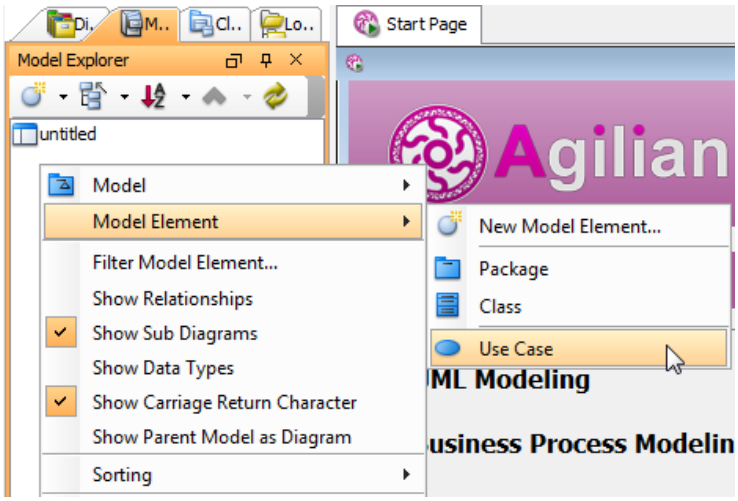
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

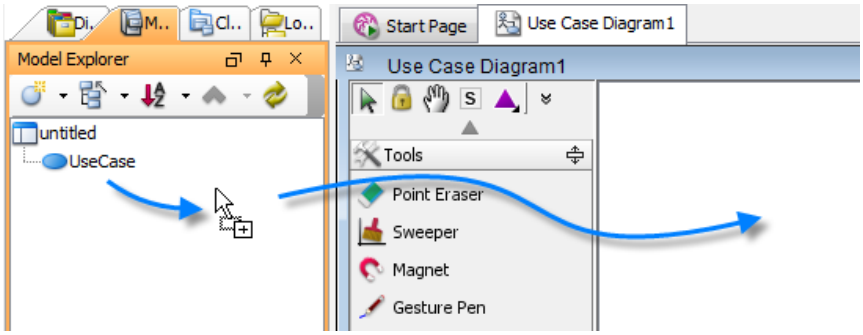
Model element and view

A model element is an elementary component of a model. It will be created when you create a shape on a diagram, or when we create one directly through the Model Explorer. An example of model element is use case.



*Creating a model element through **Model Explorer***

You can visualize an existing model element by dragging and dropping a model element from **Model Explorer** to diagram. We call the visualized form of model element a view, or a shape, depending on whether we want to emphasize the differences against model elements, or we want to focus on diagramming operations.



Creating a view from an existing model element through drag and drop

When developing context-based diagrams, you will reuse a model element in different diagrams, resulted in creating multiple views. Each model element can associate with zero to multiple views. When you make specification-level change, such as changing name, on any view, the change will be applied to all views.

Showing a view of a model element

If you want to open a diagram from a model element, right click on the model element in **Model Explorer** and select **Show View...** from the pop-up menu. Then, select the view to open in the **Show View** dialog box and click **Go to View** to open the diagram.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Master view and auxiliary view

When your project is simple, you are able to express all of the design ideas with just a few diagrams. The diagrams are simple and self-explanatory. Each of them represents a distinct design idea and there is no overlapping between diagrams.
When you are dealing with a complex project, you may need to draw multiple diagrams to represent different contexts. You need to borrow shapes from a diagram to make them appear in other diagrams (i.e. contexts). In fact, this is extremely common when modeling with class diagram and business process diagram. Take UML class diagram as an example, there may be a domain diagram that presents all the entity classes and, another diagram that presents the associations and dependencies between a specific controller class and its related entity classes. So in this case, both diagrams contain the same set of entity classes.

Instead of re-creating those classes again and again in different diagrams, Visual Paradigm allows you to "re-use" them. Through simple copy and paste (**Ctrl-C** and **Ctrl-V**), you can easily copy a shape from one diagram to another. Each shape is formally known as a "view". So with this, you can create multiple views for a model element in representing different contexts. Changes made on a shape are all synchronized to other instances that appear in other diagrams without extra effort. This is great, but there is a drawback though.

A master view is simply the specific view of model element that decides the placement of that model element within a model hierarchy. It can be a shape on a diagram, or the representation in **Model Explorer**. When you create a model element the first time, either by drawing a shape or by creating a model element under Model Explorer, the created view will be treated as the master. Subsequent views are all known as auxiliary views. When you attempt to move a master view to another parent shape, you are updating the real model structure as well (as reflected in **Model Explorer**). However, when you move any auxiliary view to a different parent shape, there will be no change at all on the model structure.

Selecting a master view

A model element can have multiple views. Yet, it can only have one master view. You can change the master view of a model element. By doing so, the original master view will become auxiliary, and the assignment of parent element will be updated immediately base on the new master view.

1. Open the **Show Views** window. In **Model Explorer**, you can open it by right clicking on the target model element and selecting **Show View...** from the popup menu. In diagram, you can open it by right clicking on the target view and selecting **Related Elements > Show Other Views...** from the popup menu.
2. In the window you can see a list of views of the selected model element/view. Click **Set Master View...**at bottom left corner.
3. This shows the **Set Master View** window. You can select on the left hand side the **Model Explorer** or a specific diagram to be the master view of selected element. To select **Model Explorer** means that the placement of model element will always follow the hierarchy as presented in **Model Explorer**. Any re-positioning made in views in any diagram will not affect the model hierarchy. Click **OK** to confirm your selection.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Resource centric interface

Visual Paradigm is the first vendor to introduce the resource centric diagramming interface. The resource centric interface greatly improves the efficiency of modeling. It can make sure the modeler is able to create a diagram with correct syntax more quickly. The utmost importance thing is that you don't have to go back and forth between the toolbar and the diagrams for creating diagram elements, making connections and modifying the diagrams.

Resource centric interface on action

NOTE: To check or uncheck the options (including **Resources, Group Resources**, **Extra Resources** and **Generic Resources Only**) on resource centric interface, select **View > Resource Centric** from the main menu.

Creating shapes through resource centric interface

Instead of clicking shapes one by one from the diagram toolbar, you can create another shape from an existing shape on the diagram.

1. Move the mouse on a shape and select a resource icon out of resource centric interface.

Create a use case with resource centric interface

2. Drag the resource icon and release it until reaching to your preferred place.
3. As a result, another shape is created and linked with the previous shape.

Release the mouse to create use case

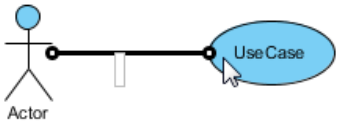
Connecting shapes through resource centric interface

For relating two shapes together, you have to link a shape to another. With resource centric interface, you can not only link them up, but also select an appropriate relationship for them.

1. Select two shapes from the diagram toolbar and drag them on the diagram individually. Move the mouse on one shape and select a resource icon out of resource centric interface.

Press actor's resource icon

2. Drag the resource icon and release it until reaching to another shape.
3. As a result, two shapes are linked together.

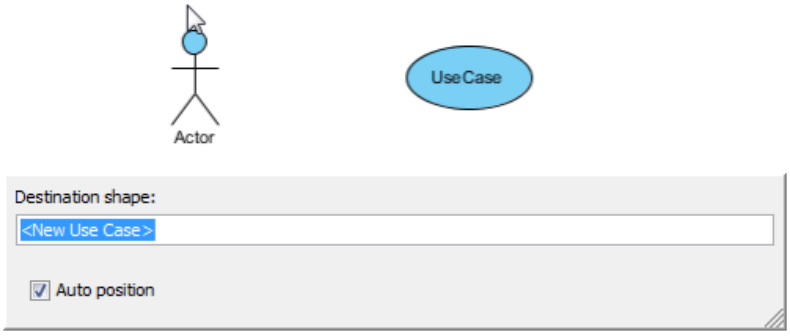


Release the mouse to connect with the use case

Using quick connect

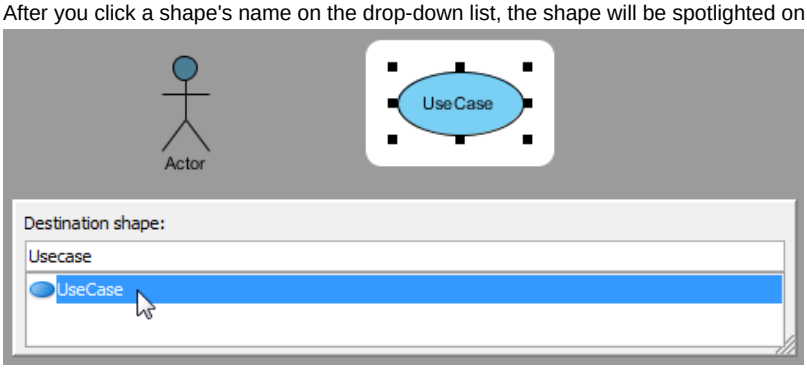
With quick connect, you are able to search the preferred shape and connect to it on the diagram accurately and quickly.

1. After creating some shapes on diagram, move the mouse on a shape and click a resource icon out of resource centric interface. The **Quick Connect** dialog box will be shown subsequently.



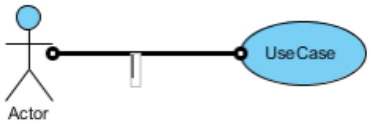
***Quick Connect** dialog box*

2. You can either create a new shape or select an existing shape by typing its name in the text field of **Quick Connect** dialog box. To select an existing shape, enter its full name directly or just type the first letter of its name. As a result, a list of shapes which match with the word(s) you typed will display.
3. After you click a shape's name on the drop-down list, the shape will be spotlighted on the diagram immediately.



Use Case is spotlighted

4. Confirm the selection, press **Enter**. The two shapes will be linked automatically.



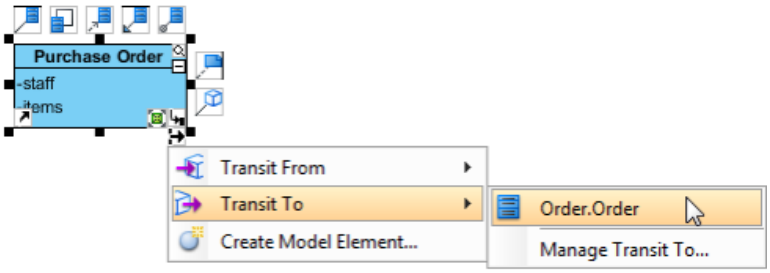
An actor is linked with a use case

> **NOTE:** You should click the resource icon in accordance with the existing shapes on your diagram.
> If the shape does not exist on the diagram, the **Quick Connect** dialog box will not pop out even when you click the resource icon.

Managing transition of shapes

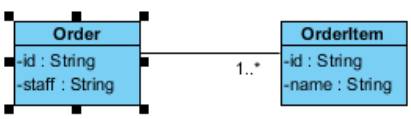
Model transitor enables you to trace between model elements across different phrases of development. Once a transition is created between two shapes, you can navigate between them through the resource centric interface.

Move the mouse on a shape and select **Model Transitor> Transit To** and then select the shape's name you would like to transit to/from from the pop-up menu.



Transit to another shape

As a result, the vision will be diverted to the selected shape after transition is selected. Thereafter, you can transit to/ from between these two shapes.



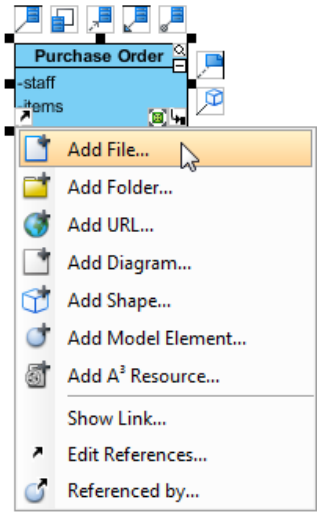
Shape is transited

NOTE: Transition does not only apply on two shapes on the same diagram, but also two shapes in different diagrams.

Adding and opening reference

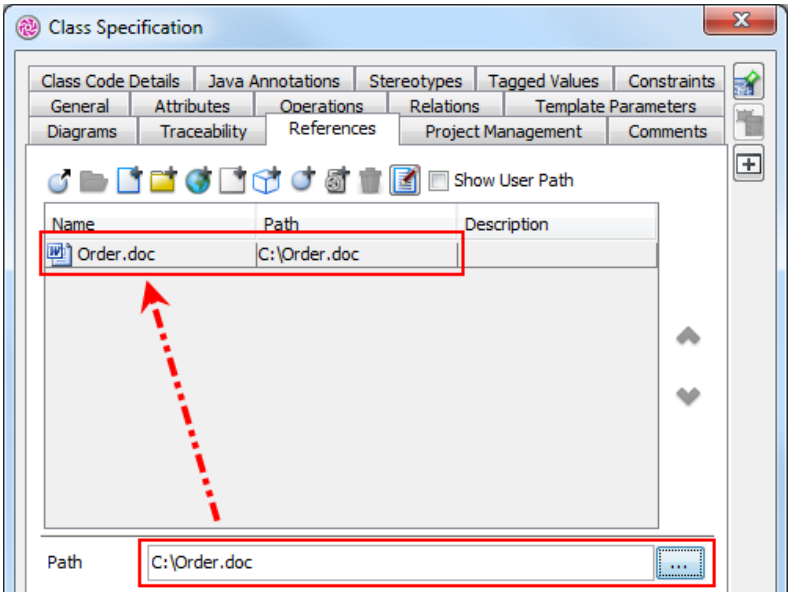
For providing extra information to shapes, you can insert both internal and external resources, such as a shape, a diagram, a file, a URL through the resource centric interface. After editing references, they can be opened throughout the resource centric interface.

1. Move the mouse on a shape that you would like to insert references for, press resource icon **Resources** at the bottom left corner and select a reference option from the pop-up menu.



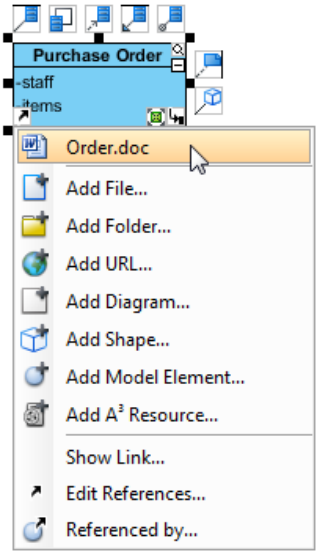
*Select **Add File...** from the pop-up menu*

2. Insert the corresponding resource in **References** tab of your shape's specification dialog box.



*Insert a file in **Class Specification** dialog box*

3. After adding reference, you can click resource icon **Resources** again and the reference you have just created will be revealed on the pop-up menu.



The newly created reference

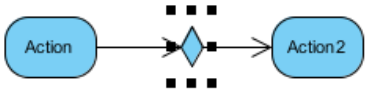
Splitting connection by shape

Resource centric supports some connectors (e.g. Control Flow in Activity Diagram) with splitting the connector by adding another shape.

1. Move the mouse on the connector between two shapes and select a split resource icon out of resource centric interface.

Select a split resource icon

2. As a result, an extra shape is inserted between two existing shapes.



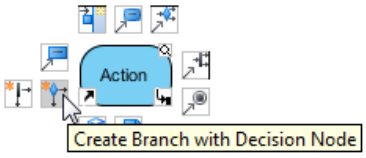
*Control Flow is split by **Decision Node***

> **NOTE:** There is an orange asterisk on the split resource icon.

Creating structure

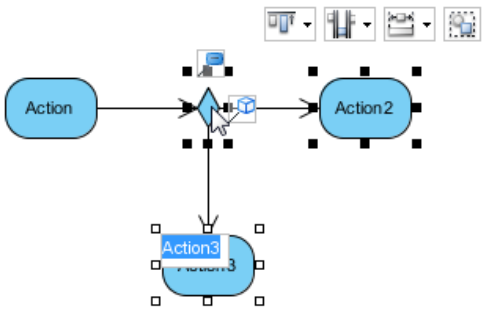
Resource centric interface supports some models (e.g. Action in Activity Diagram) with creating a structure of models. Instead of creating shapes one by one, branch resource icon helps speed up the shape creation process; you thereby can create several shapes simultaneously.

1. Move the mouse on a shape and select a branch resource icon out of resource centric interface.



*Select **Create Branch with Decision Node** on resource centric interface*

2. Drag the resource icon and release it until reaching to your preferred place.
3. As a result, two shapes and a decision node are created and connected.

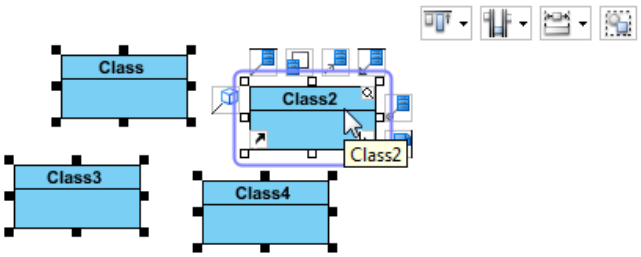


NOTE: There is an orange asterisk on the branch resource icon.

Group selection resource

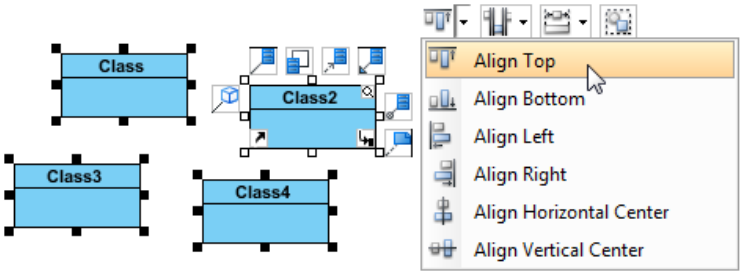
When a great amount of shapes are displayed on the diagram disorderly, resource centric interface can support alignment and grouping after selecting several shapes.

1. Select several shapes on diagram.
2. Move the mouse on the last selected shapes and group selection resources will be shown instantly.



Group Selection Resources are shown

3. Press the reversed triangle of **Align Top** and select an alignment option from the drop-down menu.



*Select **Align Top** from the drop-down menu*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

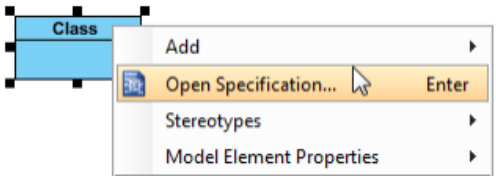
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Tagged values

In Agilian , tagged values are properties defined by user and can be added to a model element. Agilian supports either defined tagged values in stereotype or tagged values without stereotype. In UML 1.1, stereotypes and tagged values were used as string-based extensions that could be attached to UML model elements. In UML 2.x, stereotypes and tagged values will be defined in Profile that can provide more structure and precision to the definition.

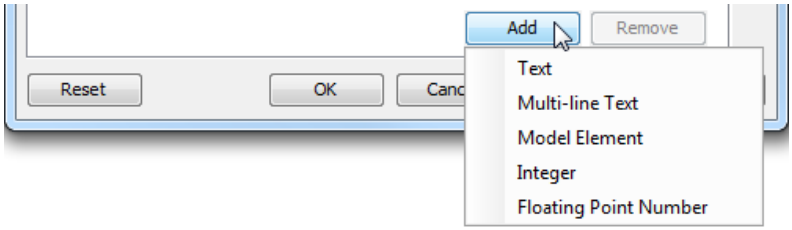
Adding user-defined tags

1. Right click on the selected shape and select **Open Specification...** from the pop-up menu.



*Right click to select **Open Specification...***

2. In **Specification** dialog box, select **Tagged Values** tab and click **Add** button to select an option of value type.



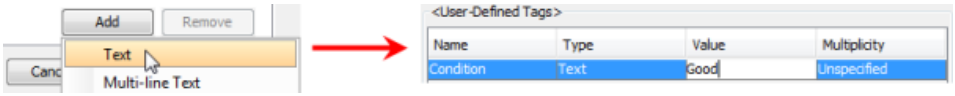
Add an option of value type

Different types of tagged values

Text

Text supports string-based value.

In **Specification** dialog box, select **Tagged Values** tab, click **Add** button and select **Text** from the pop-up menu. Enter the text in the text field directly.

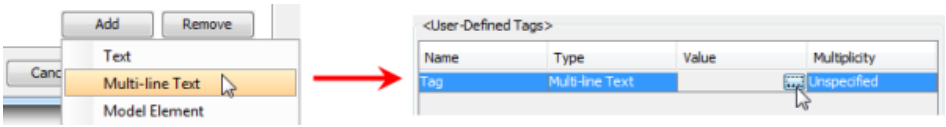


Add text

Multi-line Text

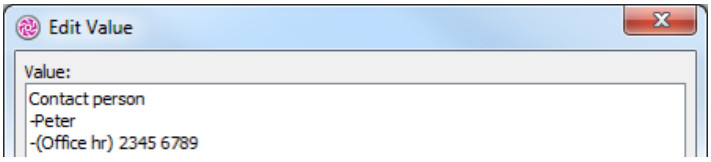
Multi-line Text supports multi-line string.

1. In **Specification** dialog box, select **Tagged Values** tab, click **Add** button and select **Multi-line Text** from the pop-up menu. Click the **...** button of **Value**.



Add multi-line text

2. When **Edit Value** dialog box pops out, enter the multi-line text. Click **OK** button to finish editing.

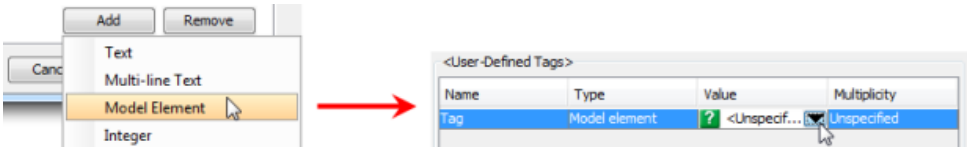


*The **Edit Value** dialog box*

Model Element

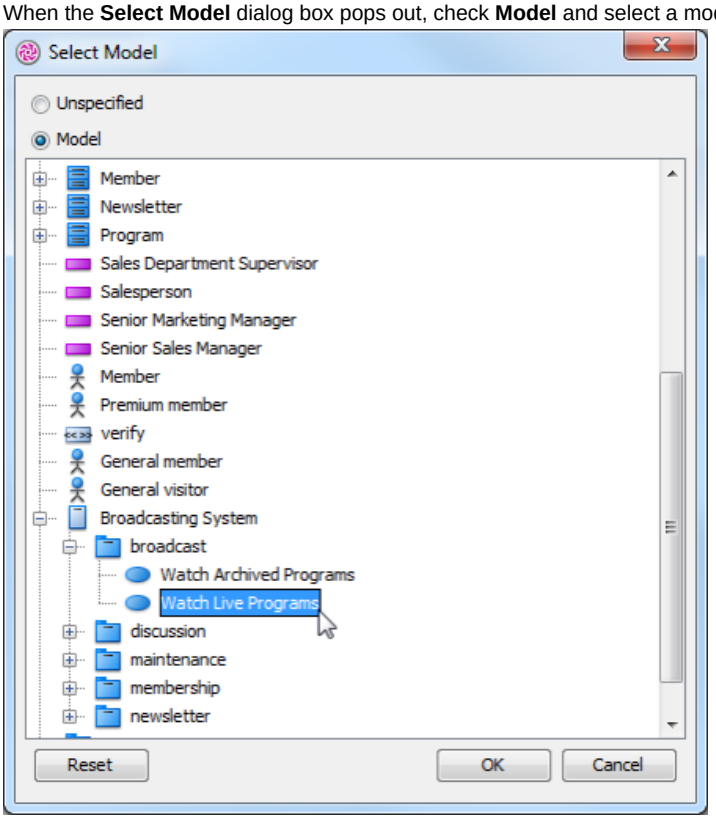
Model Element supports reference(s) of model element.

1. In **Specification** dialog box, select **Tagged Values** tab, click **Add** button and select **Model Element** from the pop-up menu. Click the reverse triangle button of **Value**.



Add model element

2. When the **Select Model** dialog box pops out, check **Model** and select a model element. Click **OK** button to confirm.

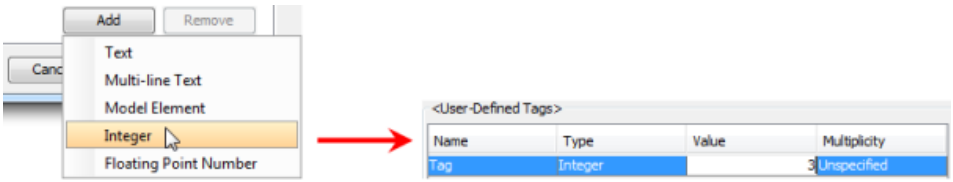


Select a model element

Integer

Integer supports values with numbers only.

In **Specification** dialog box, select **Tagged Values** tab, click **Add** button and select **Integer** from the pop-up menu. Enter a number in **Value**.

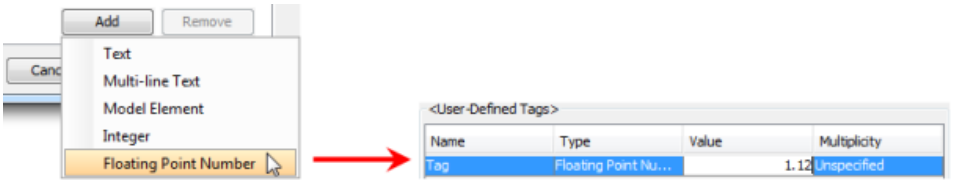


Add integer

Floating Point Number

Floating Point Number supports values with decimal places.

In **Specification** dialog box, select **Tagged Values** tab, click **Add** button and **Floating Point Number** from the pop-up menu. Enter a number with decimal places in **Value**.

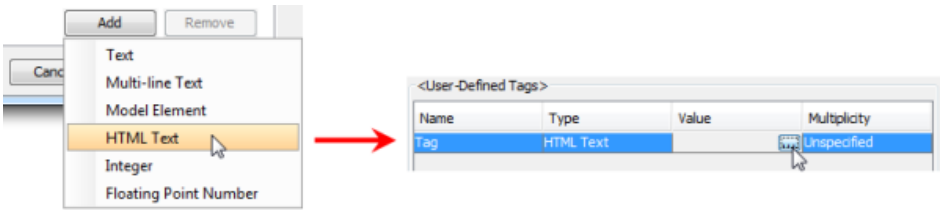


Add floating point number

HTML

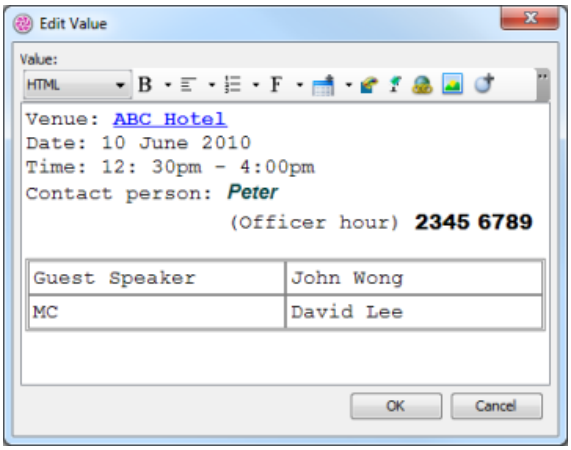
HTML is a hidden tagged value. To reveal this option, select **Tools > Options...** from the main menu. When the **Options** dialog box pops out, select **Diagramming > Environment** tab and check **Stereotype support HTML tagged value**.

In **Specification** dialog box, select **Tagged Values** tab, click **Add** button **> HTML Text** from the pop-up menu and click **...** button in **Value** when the option for **HTML** is revealed.



Add HTML text

When **Edit Value** dialog box pops out, enter the text. You can format the HTML text content with the toolbar, for example, changing the font color of selected text and underlining selected text.

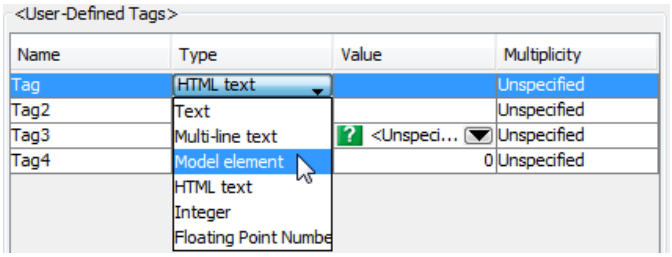


Format the HTML text content

Click **OK** button to finish editing.

Editing tagged values

You can enter the user-defined tagged value's name, select its type and enter its value.

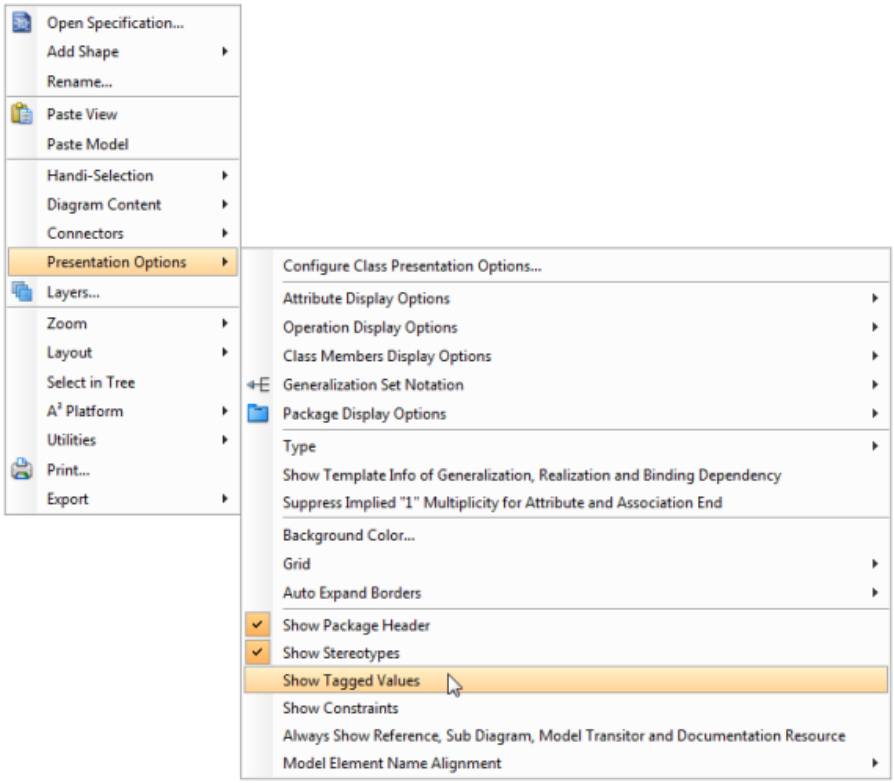


Edit name, type and value on user-defined tagged value

The value of stereotype can be edited, however, its name and its type cannot be edited as they are defined in stereotype.

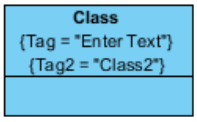
Visualizing tagged values on diagram

Right click on the diagram background and select **Presentation Options > Show Tagged Values** from the pop-up menu.



Show tagged values

If it is defined, the tagged values will be seen within the shape(s) on the diagram.



Tagged values are shown

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Spell checking

You will never find it hard to avoid making mistakes in your diagram after using spell checking . It can help you to correct both typing mistakes and spelling mistakes, however, it is slightly different from other spelling and grammar checking tools you have used before. It doesn't check your whole diagram automatically, but underlines wrong words with a red curve line.

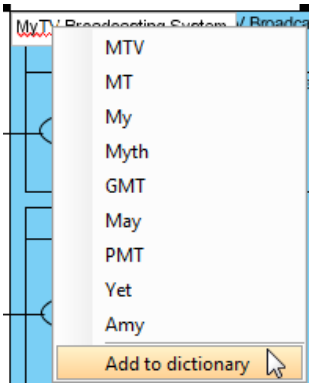
Correcting a word

If you type an incorrect word mistakenly, you can:

- Either re-type the word correctly or
- Right click on the wrong word and then select one out of the suggest words.

Adding a new word to dictionary

Sometimes, the dictionary cannot recognize the word you type and it doesn't always mean you type an incorrect word . It may be a rare word or a new word that you create, for example, your company's name. You can simply right click on the wrong word and select **Add to dictionary** to add a new word to dictionary. You type this word again next time, it won't be marked as a wrong word.

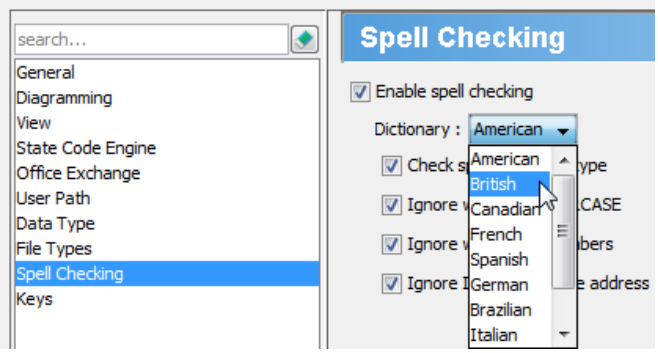


Add a new word to dictionary

Changing the language of dictionary

The dictionary usually defaults as American English for spell checking. If you want to change the language of dictionary, you just need to:

1. Click **Tools** > **Options...** from the main menu.
2. In the **Options** dilaog box, click **Spell Checking**. Select your target language from the drop-down menu of **Dictionary**.

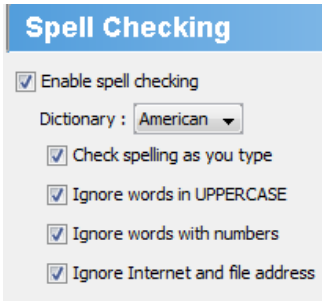


Change the language of dictionary

NOTE: There are a few more languages in dictionary that can be used, such as French and German.

More options of spell checking

There are a few more options of spell checking, for example **Ignore words with numbers** and **Ignore Internet and file address**. Check the item you want to be included in spell checking while uncheck the item you don't want to be included.



Check other options of spell checking

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Project management properties

Project management properties are a set pre-defined properties, made for recording additional management-level information for all kinds of project data (e.g. diagrams, model elements, diagram elements).

Using project management properties

You will see a list of project management properties with their description.

Configuring project management properties look-ups

Add available value selection for project management properties, and set defaults.

Using project management properties

Project management properties are a set pre-defined properties, made for recording additional management-level information for all kinds of project data (e.g. diagrams, model elements, diagram elements). Here are all the project management properties you can find:

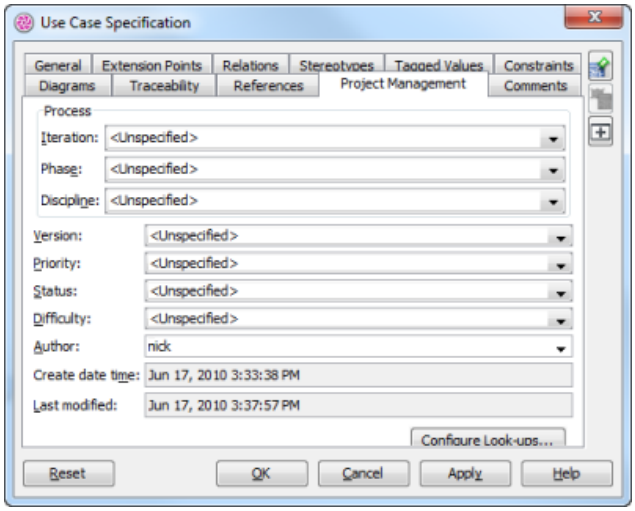
Property name	Description
Process	Specify the part of the process where the editing element is involved. Three sub-properties for further specification. They include: Iteraction, Phase and Discipline.
Version	The stage of the editing element. For example, you may have two class diagrams for modeling the a system from design and implementation angles respectively. The two diagrams will have version 1.0 and 2.0, to show the progress of work.
Priority	The importance of editing element.
Status	The status of editing element. It is particular useful for use case and BPMN activity shapes, for setting their status such as Proposed, Planned, Tested, etc.
Difficulty	How difficult it is to complete the goal as modeled by the edting element.
Author	The person who created the editing element. This is particular useful in a team working environment. Once the author is known you can contact that person in case you have questions about a part of a model.
Create date time	The date and time when the editing element was created. This property is a read-only property that is filled automatically to all elements when creation.
Last modified	The date and time when the editing element was modified. This property is a read-only property that is filled automatically to all elements when the project file is being saved .

A list of project management properties

Further to recording project management properties, you can print those properties in report, too.

Editing project management properties

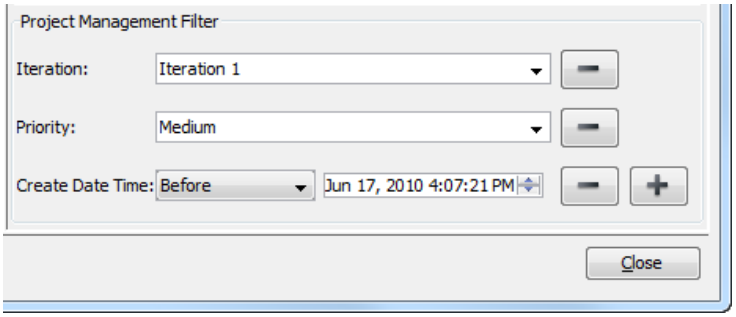
Like all the other specification level properties, project management properties can be edited through the specification dialog box of all diagrams, model and diagram elements. Select the desired diagram/model element/diagram element. Right click and select **Open Specification...** from the popup menu. Under the tab **Project Management** you can find the properties of the chosen element.



Editing project management properties

Project management properties in report generation

Like most other properties, project management properties will be presented in PDF, HTML and Word report, too. Furthermore, you can filter the elements to present in report by project management properties when editing template.



Editing project management filter in editing report template

Related Resources

The following resources may help you learn more about the topic discussed in this page.

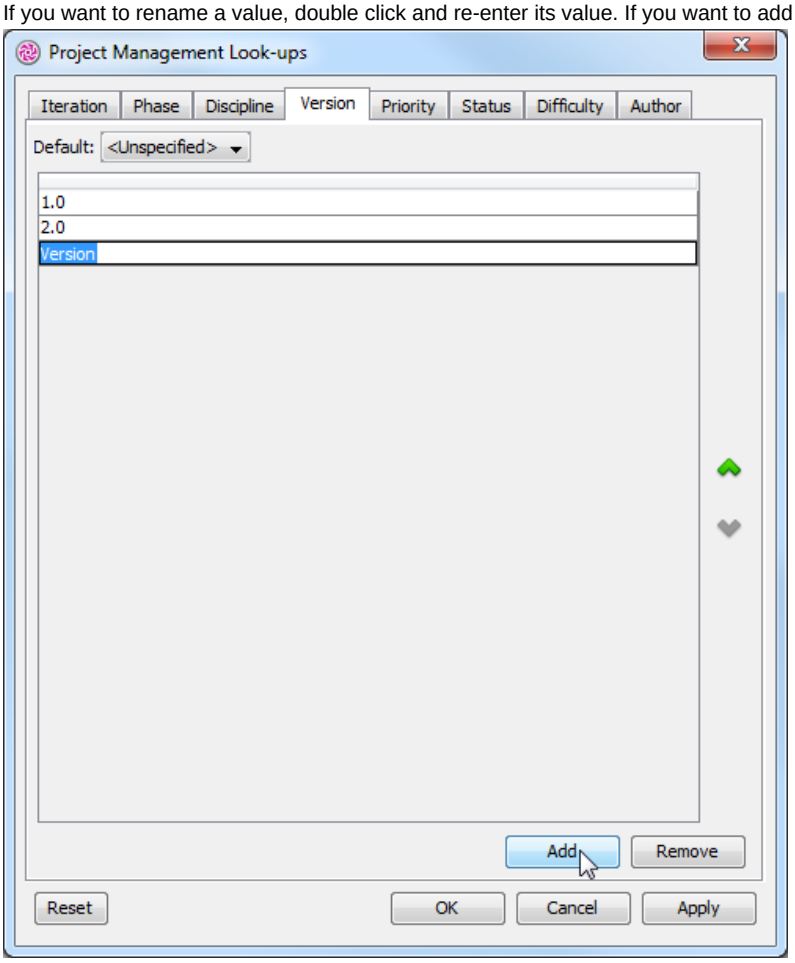
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Configuring project management properties

For each project management property , there are several default values available for selection. For example, values "1.0" and "2.0" can be selected for property **Version**. You can edit an existing value, or add additional values to a property by editing the look-ups.

To configure a property:

1. Select **Modeling > Project Management Look-ups...** from the main menu.
2. In the **Project Management Look-ups** dialog box, open the tab of the property that you want to edit its look-ups.
3. If you want to rename a value, double click and re-enter its value. If you want to add a lookup value, click **Add** at the bottom right corner.



Add a version

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Documenting model elements

You can document your model through the documentation editor. In this chapter, you will learn how to use the RTF documentation editor and how to record voice for shapes.

RTF documentation

Rich text format documentation can be entered to shapes. You will see a description about different formatting functions, as well as the steps for defining a template.

Voice documentation

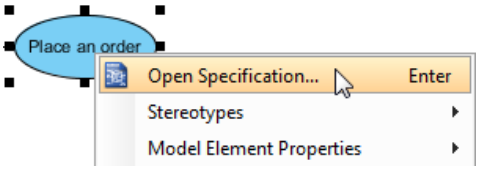
Instead of text, you also can record audio clip as a voice documentation. You will see how to do this in the page *Voice documentation*.

RTF documentation

With rich text format (RTF), you can format the text in documentation editor, such as making it bold, italic, or adding tables within the text. In addition to formatting, RTF supports users to add linkage to another model element, save documentation as template and furthermore reuse the template.

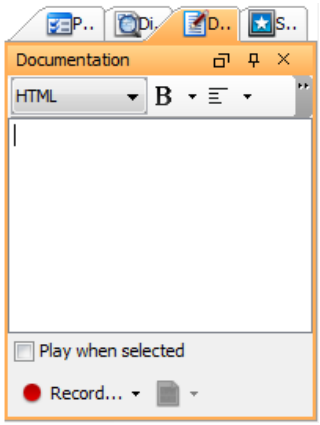
Viewing and editing the documentation of model element

To view or edit the documentation of model element, right click on the model element and select **Open Specification...** from the pop-up menu.



Open specification

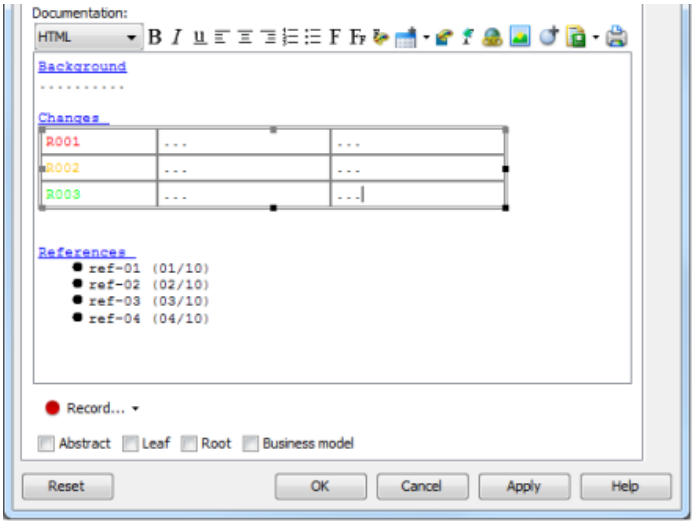
Alternatively, click the **Documentation** pane after select a specific model element on the diagram pane.



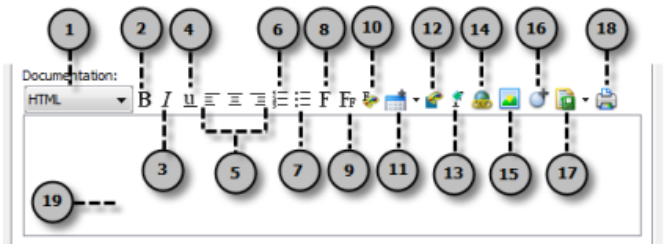
***Documentation** pane*

Documentation editor

The documentation pane is where you can describe the selected model element. With the editor's toolbar, you can format the content, save as template, reuse the existing template, add model elements and print the content.



The overview of documentation editor



The toolbar on documentation editor

No.	Name	Description
1	HTML	**HTML** - Read and edit the real content. **HTML Source** - Read and edit the HTML source of content. **Plain Text** - Read and edit plain text of content without formatting.
2	Bold	Set the highlighted text to bold.
3	Italic	Set the highlighted text to italic.
4	Underline	Underline the highlighted text.
5	Alignments	Set the alignment of highlighted text to the left, the center or the right.
6	Ordered list	Add a numbered list.
7	Un-ordered list	Add a list with bullet points.
8	Font	Select the font family of highlighted text.
9	Font size	Select the size of highlighted text.
10	Font color	Select the color of highlighted text.
11	Table	Add a table.
12	Background color	Select the background color of highlighted text.
13	Clear formats	Clear formats of the whole editor to convert the content to plain text.
14	Link	Add a hyperlink.
15	Image	Add an image.
16	Add Model Element	Insert an existing model element or create a new one.
17	Template	**Save as Template...**: Save the current documentation content as a template. **Manage Template...**: Delete the saved template.
18	Print	Print the documentation content.
19	Editor pane	Enter and start editing the documentation content.

The description of documentation editor

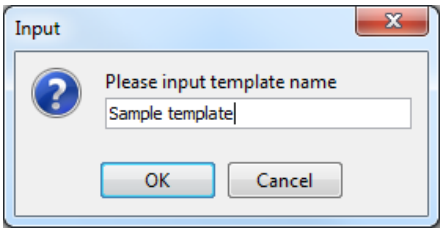
Editing documentation

Saving a template

1. The documentation you typed can be saved as template in the documentation editor. Choose **Save as template...** from the drop-down menu of **Template**.

*Choose **Save as template...***

2. In the **Input** dialog box, enter the template name and click **OK** button to confirm.



Enter template name

Reusing a template

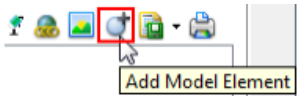
1. In the documentation editor, an existing template can be reused. Select a saved template from the drop-down menu of **Template**.

Select a template

2. As a result, the selected template will be shown on the documentation editor.

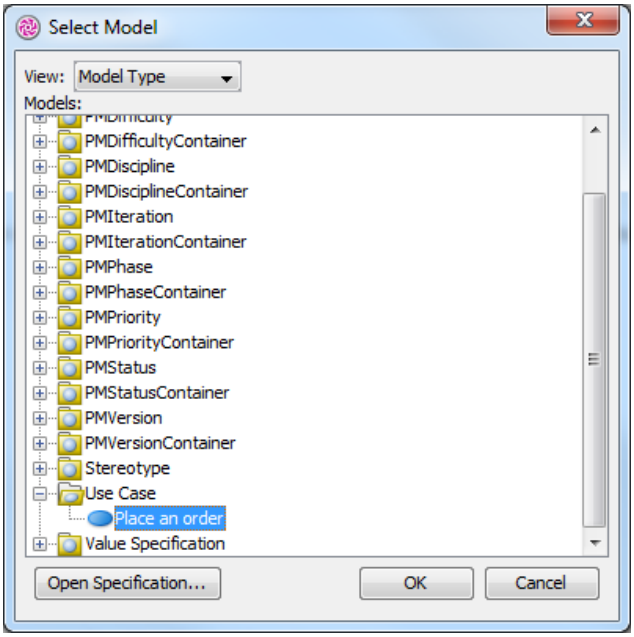
Adding model element

1. Click **Add Model Element** button on the editor's toolbar after decided a place for inserting a model element.



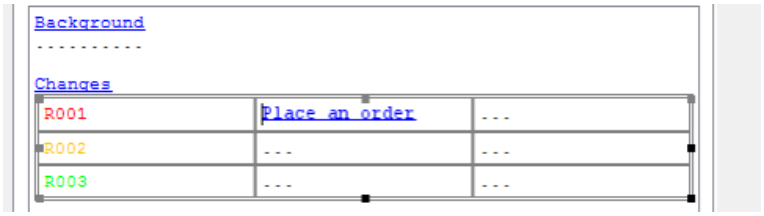
*Click **Add Model Element** button*

2. In **Select Model** dialog box, select an existing model element on the list to add it on the document. If you want to modify the selected model element, you can click **Open Specification...** button.



***Select Model** dialog box*

3. Finally, click **OK** button to confirm.
4. Consequently, the name of inserted model element will be shown on the documentation pane with underline.



The selected model element is shown

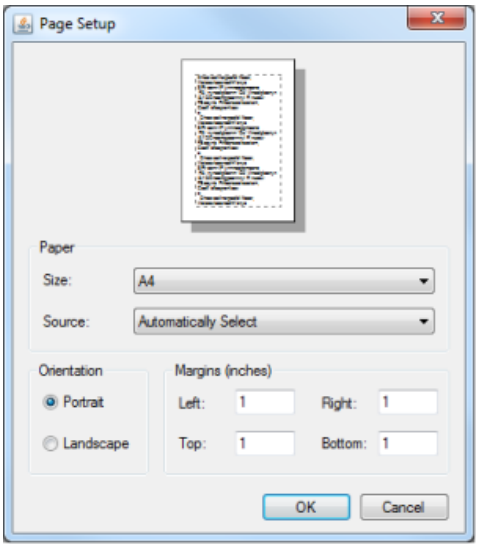
If you want to preview the inserted model element, you can right click on its name and select **Show View...** from the pop-up menu. After the **Show View** dialog box pops out, you can preview it in the **Preview** window. If you want to view the actual model element on the diagram, you can click **Go to View** button.

Printing documentation

1. After clicking the **Print** button, the **Page Setup** dialog box will pop out.

*Click **Print** button*

2. In **Page Setup** dialog box, select size and source for paper, check an option under **Orientation** and enter the inches for the paper margins.



*The **Page Setup** dialog box*

3. Click **OK** button to confirm printing.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

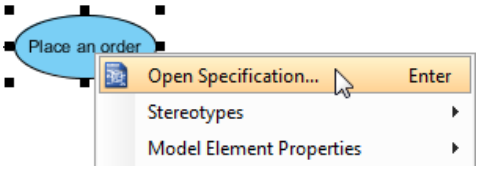
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Voice documentation

In addition to textual description for your model elements, you can record voice documentation , or embed audio files.

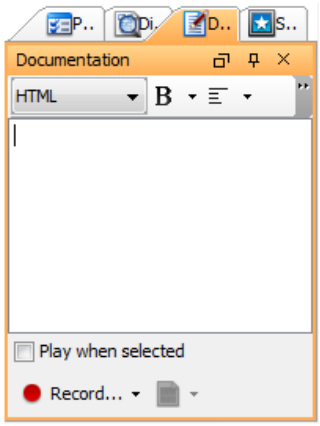
Viewing and editing the documentation of model element

To read or edit the documentation of model element, right click on a model element and select **Open Specification...** from the pop-up menu to unfold **Specification** dialog box.



Open specification

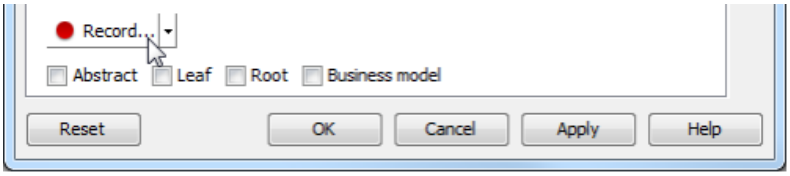
Alternatively, click the **Documentation** pane after select a model element on the diagram.



***Documentation** pane*

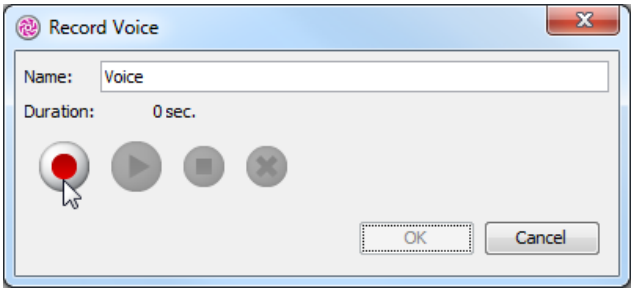
Recording voice

1. Right click on a model element and select **Open Specification...** from the pop-up menu.
2. In **Specification** dialog box, click **Record...** button on the bottom left corner of dialog box.



*Click **Record...** button*

3. In the **Record Voice** dialog box, click the **Record** button to start recording.



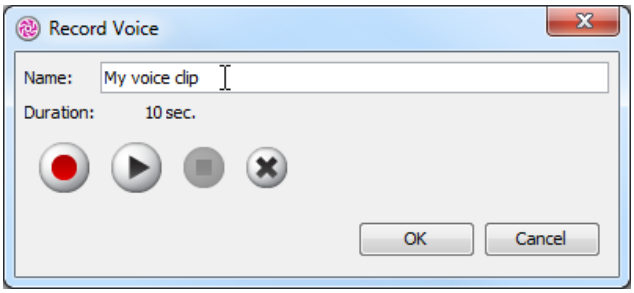
Start recording

> **NOTE:** Make sure your audio input device is active before operate voice documentation.

4. Click the **Stop** button when you want to end the recording.

> **NOTE:** Play the recorded voice by pressing the **Play** button; record again by pressing the **Clear** button, and rerun the previous steps.

5. Enter the name for recorded voice clip in the text field of **Name**.

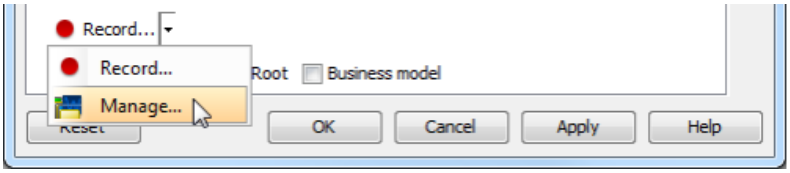


Name voice clip

6. Click **OK** button to confirm recording.

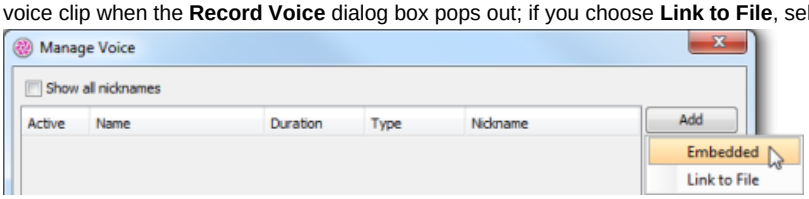
Linking voice to documentation

1. Right click on a model element and select **Open Specification...** from the pop-up menu.
2. In **Specification** dialog box, choose **Manage...** button on the bottom left corner of dialog box.



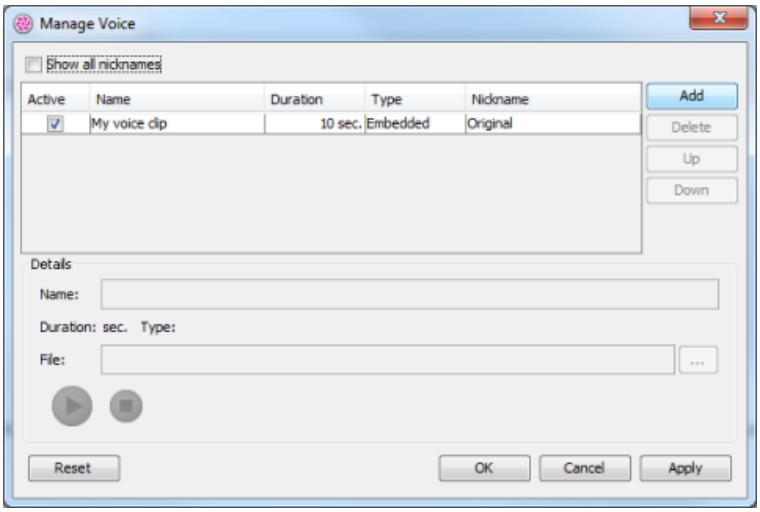
*Choose **Manage...** button*

3. In the **Manage Voice** dialog box, click the **Add** button, and choose either **Embedded** or **Link to File**. If you choose **Embedded**, record a voice clip when the **Record Voice** dialog box pops out; if you choose **Link to File**, select an audio file when the **Open** dialog box pops out.



*Choose **Embedded** in **Manage Voice** dialog box*

4. After select, the voice clip is shown on the list. If you want to rename the voice clip, click the voice clip and then enter in the text field of **Name** under **Details**.



The voice clip is shown

5. Click **OK** button to confirm.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Style and formatting

You can change shapes' appearances freely by editing their fill color, font and line style. In this chapter, we will walk through in detail.

Applying fill, line and font styles on diagrams

Shows how to edit the fill, line and font of shapes.

Managing and applying styles

You can save formatting properties as a style to aid in reusing in other shapes. This page tells you how to save a style, and reuse it on another shape.

Setting line style

Controls how connector routes. There are five options: rectilinear, oblique, curve, round oblique and round rectilinear.

Setting line jumps options

When two connectors intersect with each other, the line jump option determines how the intersection will be rendered.

Setting connector caption orientation

Controls how a caption of connector appear against different connector orientations.

Format copier

Shows you how to copy the formatting properties of a shape and copy to another.

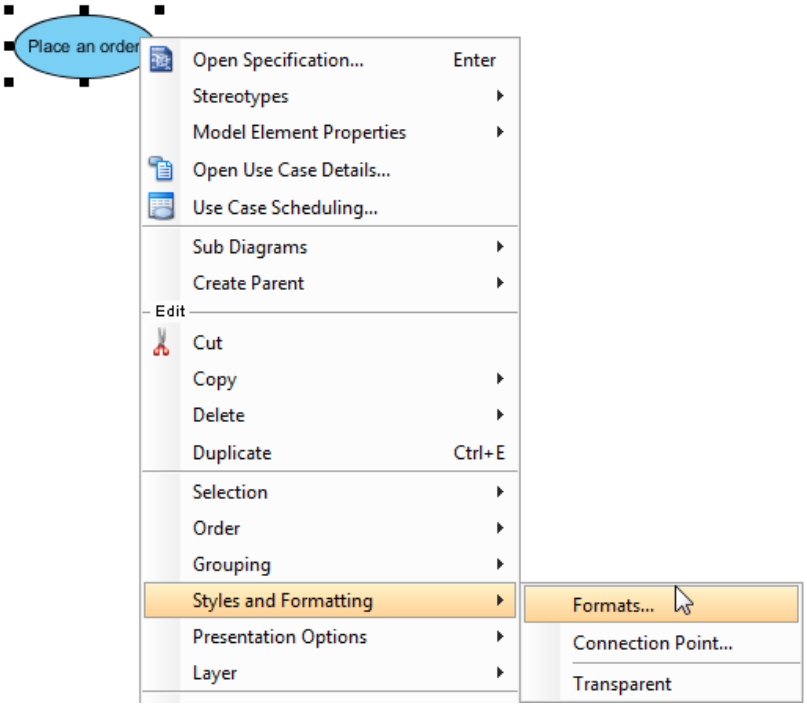
Set connection point

Determines how to position the end point of connector, whether to point to the shape's border or point into the center.

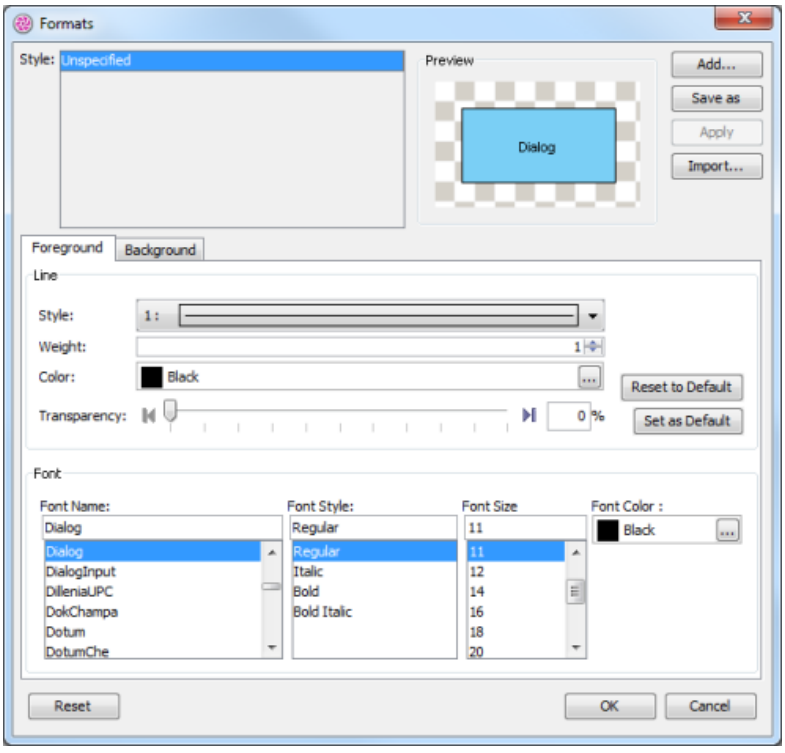
Change diagram elements styles

To emphasize the importance of a particular diagram element or make it outstanding, you can format the diagram elements with your preference.

To unfold the **Format**s dialog box, right click on the target shape and select **Styles and Formatting** > **Formats...** from the pop-up menu.



*Select **Formats...** from the pop-up menu*



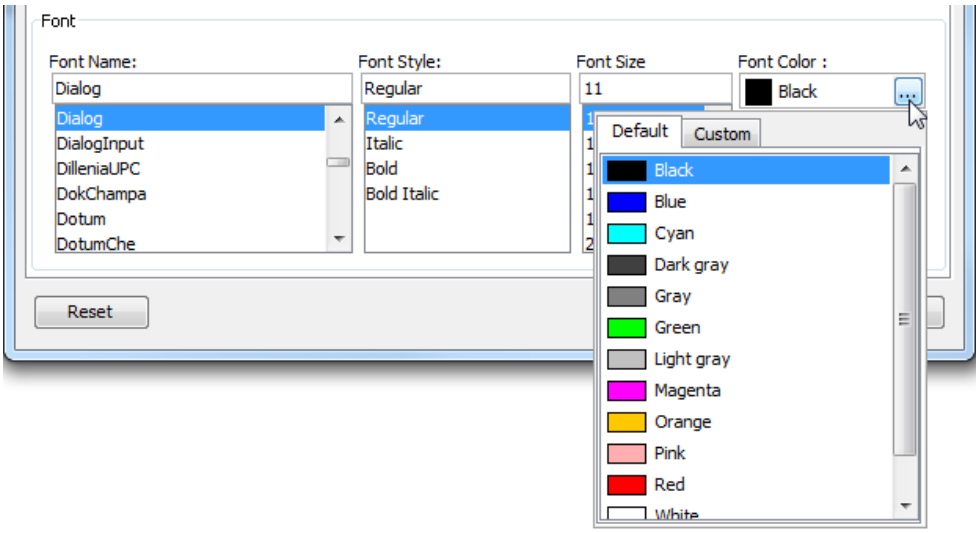
***Formats** dialog box*

You can change the following settings from the **Formats** dialog box:

- Changing shapes foreground style
- Changing shapes font style
- Changing shapes background style

Changing shapes foreground style

In the **Format** dialog, you can change the foreground style under the **Font** section. Click the **...** button in **Font Color** to select a color either from the **Default** page (which shows pre-defined colors) or from the **Custom** page (which shows a variety of colors and allows you to define any custom colors).



Color pane

If you want to specify a custom color, you can switch to the **Custom** pane. After change the font color, the preview will update automatically.

Changing shapes font style

In the **Font** section, you can also change the font name, style and size.

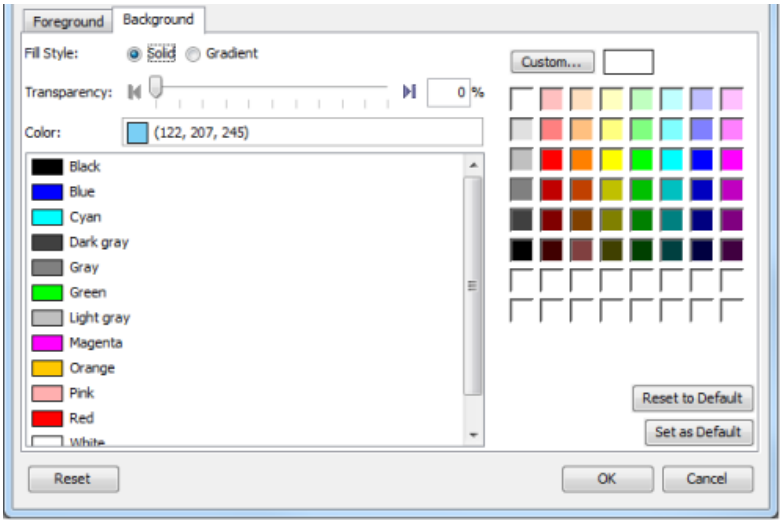
Font section

Field	Description
Font Name	Select a type for font. The amount of fonts depends on the fonts available in your computer.
Font Style	Select a style for font. You can select one of the 4 styles and a preview will be shown for each of the style items.
Font Size	Select the size for font. You may either select a font size by rolling the scrollbar or enter the font size in the text fields.

The Preview pane will display the selected font format.

Changing shapes background style

You can click on the **Background** tab to custom the background style.



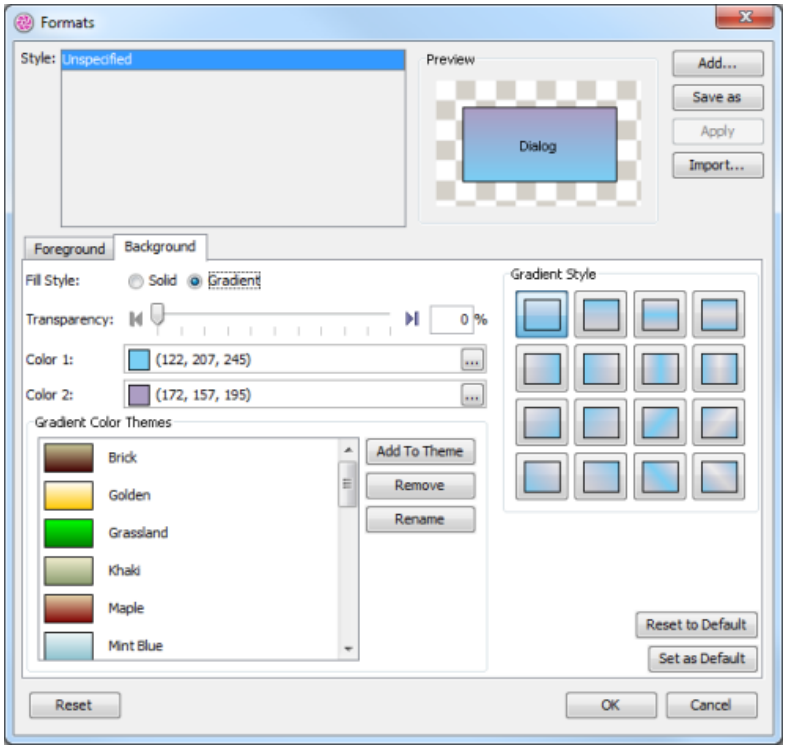
***Background** tab*

In the **Background** tab, it allows you to select a solid fill color or a gradient fill color as well as define its transparency.

Field	Description

Fill Style	Select the fill style of the fill color. It can either be **Solid** (a single color) or **Gradient** (a fill color that is mixed with two colors).
Transparency	Specify the transparency of the fill color. The greater the value, the more transparent the shape will become. 0 (zero) transparency makes the fill color completely opaque, while 100 (one hundred) transparency makes the fill color completely transparent. You can adjust the transparency by dragging the slider, or by typing the value in the text field. Alternatively, you can click the Opaque button to set the fill color to opaque, or click the Transparent button to set the fill color to transparent.
Preview	The Preview pane displays a rectangle that is filled with the editing fill color. The background is checked so that you can also preview the transparency of the fill color as well.
Save as Default	To save the current fill color as the default fill color for new shapes, click the **Set as Default** button.
Reset to Default	To reset the current fill color to the default fill color, click the **Reset to Default** button.

Upon checking **Gradient** in **Fill style** field, you will see the detail pane for formatting a gradient fill color.



Gradient Fill style

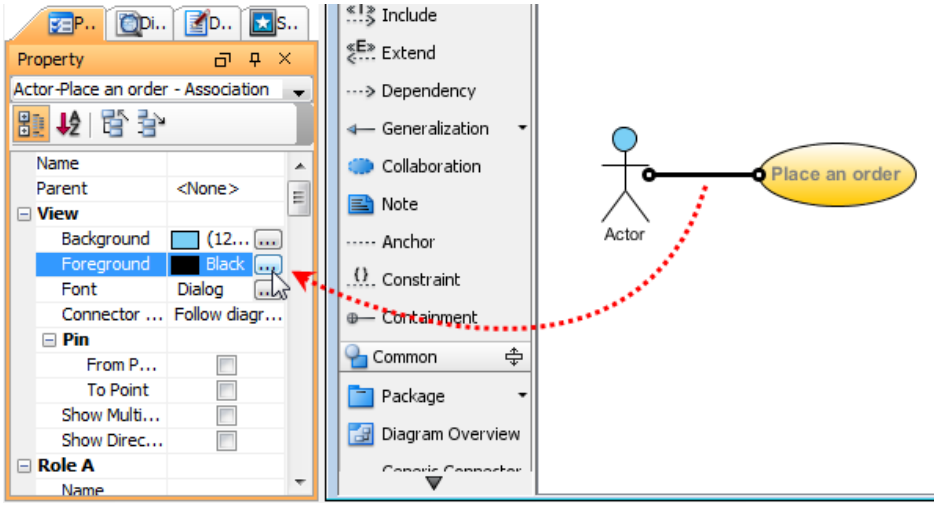
Field	Description
Color 1	You can select the first color of the gradient from the **Color 1** field. To select a color click the **...** button or double-click on the color editor. A color chooser will appear for you a select a color.
Color 2	You can select the second color of the gradient from the **Color 2** field. To select a color click the **...** button or double-click on the color editor. A color chooser will appear for you to select a color.
Gradient Color Themes	The Gradient Color Themes pane displays a list of pre-defined gradient color themes. To add a new color theme select **Color 1** and **Color 2** then click the **Add to Themes...** button. Please note that you must select a combination of colors that does not already exist in the color themes. To rename a theme click on the **Rename...** button or double-click on the desired theme. An input dialog will appear for you to enter a new name. To remove a theme select the theme and click on the **Remove** button, or use the Delete key instead.
Gradient Style	The Gradient Style pane allows you to select the gradient style of the gradient fill color (the angle of how the gradient color is drawn). There are sixteen pre-defined gradient styles, which are shown as toggle button in the Gradient Style pane. To select a gradient style to use click on one of the styles.

After you finish setting, click **OK** to apply to the selected shape.

Formatted shape

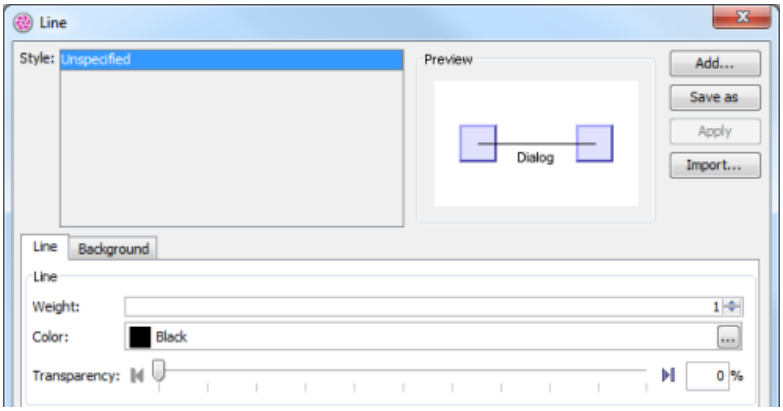
Changing connector line style

To change the connector line style through **Property** pane, select the target connector on the diagram, click the **...** button in **Foreground** row of the **Property** pane.



*Click the **...** button in the **Property** pane*

You can format the line style in the line section. It allows you to adjust its weight (thickness), color and transparency.



Line section

Field	Description
Weight	Adjust the weight (thickness) of a line. The greater the value, the thicker the line will become. You can use the up/down button to increase/reduce the line weight, or you can type directly into the text field. The line weight ranges from 1 to 20.
Color	Specify the line color. Click on the **...** button next to the **Color** field to select a color, either from the **Default** page (which shows predefined colors) or from the **Custom** page (which shows a larger variety of colors, and allows you to define any custom colors). **NOTE:** You can only enter integer values for line weight. If you type 2.8 in the text field, 2 will be applied instead.
Transparency	Specify the transparency of the line. The greater the value, the more transparent the line. 0 (zero) transparency makes the line completely opaque, while 100 transparency makes the line completely transparent. You can adjust the transparency either by dragging the slider, or by typing the value in the text field. Alternatively you can click on the Opaque button to set the fill color to opaque, or click on the Transparent button to set the fill color to transparent.
Preview	The Preview pane displays the line with the selected line format applied.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

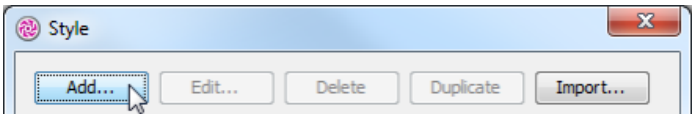
Managing and applying styles

You can format shapes and connectors in Agilian by changing their attributes, such as line style, weight, color and transparency. Moreover, you can apply your preferred styles or remove them after creating. Since adding and applying styles of shapes and connectors is as simple as clicking few clicks, the newly created format settings will be applied on the selected shapes/ connectors easily and instantly.

Adding style

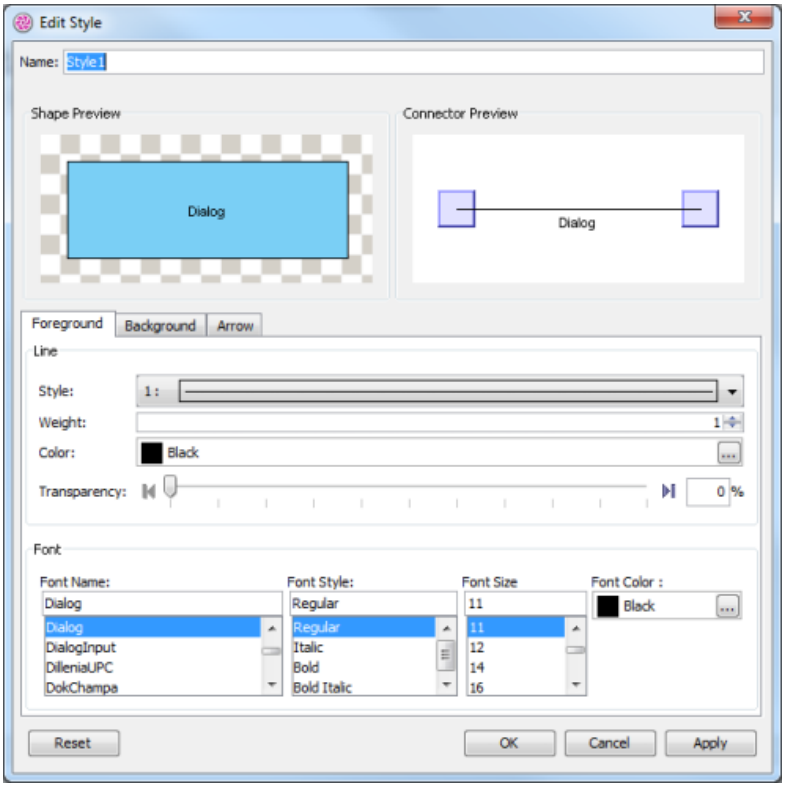
To open the style dialog box, Select **View > Styles...** from the main menu.

In the **Style** dialog box, click **Add...** to create and edit a new style.



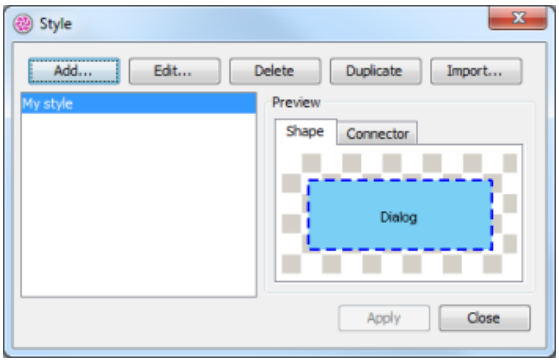
*Click **Add...** button in **Style** dialog box*

In the **Edit Style** dialog, you can change the name, foreground line style, font style, background style and arrow style. Click **OK** button after you finish editing the setting.



***Edit Style** dialog box*

As a result, the style is created.

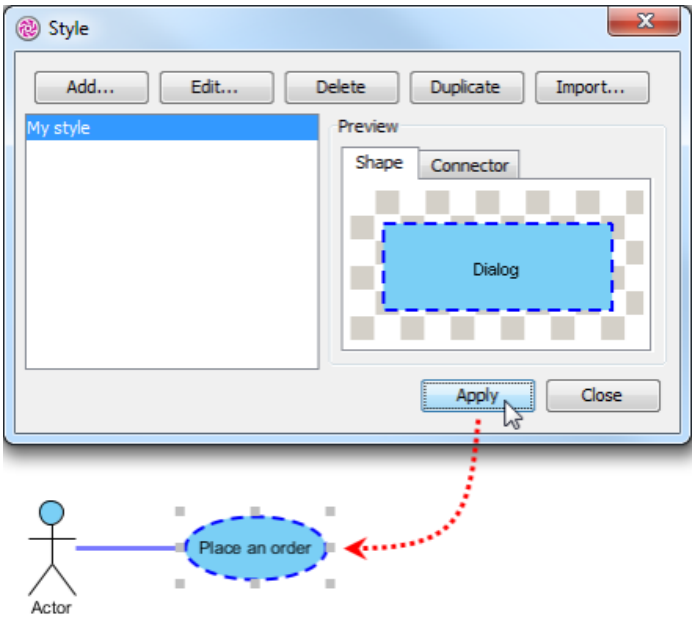


Style added

Applying style

Upon keeping the **Style** dialog box open, select a target shape on the diagram and click **Apply** in the **Style** dialog box.

As a result, the shape is changed into the newly created style.



Apply style to selected shape

Related Resources

The following resources may help you learn more about the topic discussed in this page.

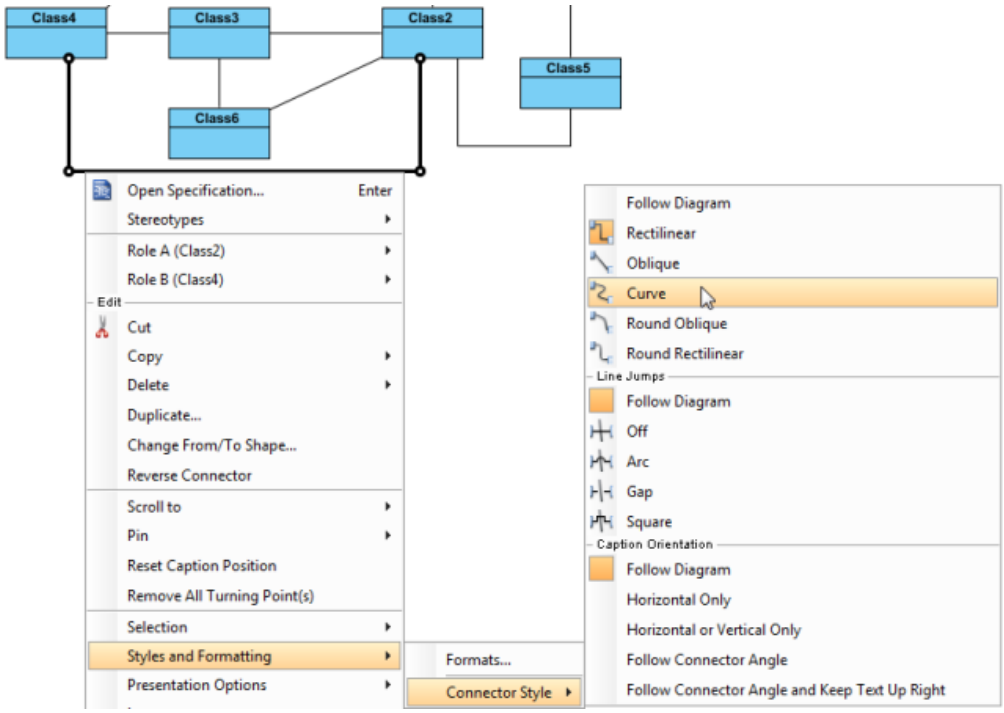
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Setting line style

Connectors are the lines that connect two shapes. When more shapes are created and more connectors appear, you may find that it is difficult to handle the straight spaghetti-like connectors. To overcome this problem, Agilian provides five connector styles to help you handle the connectors, namely **Rectilinear**, **Oblique**, **Curve**, **Round Oblique** and **Round Rectilinear**.

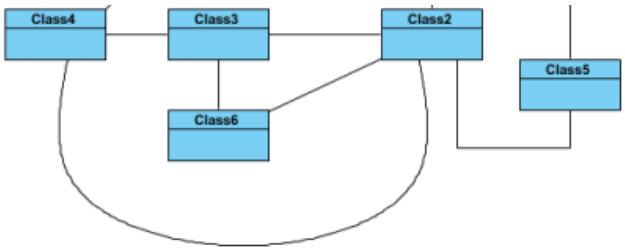
Setting connector line style

To change the line style, right click on the target connector and select **Style and Formatting > Connector Style** and a line style option from the pop-up menu.



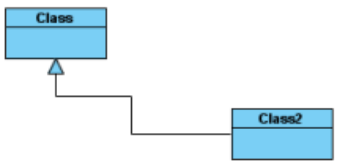
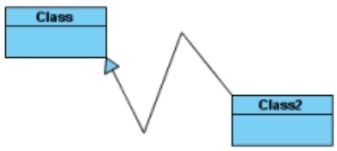
Change line style

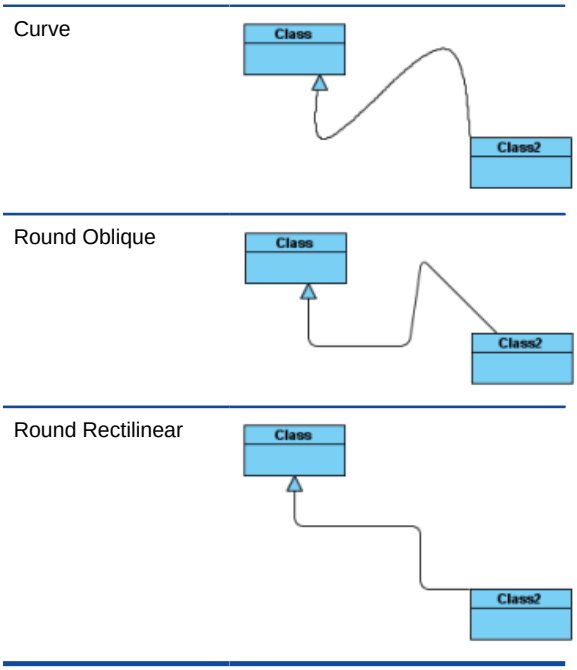
As a result, the connector is changed into the selected line style.



Change line style

Line style options

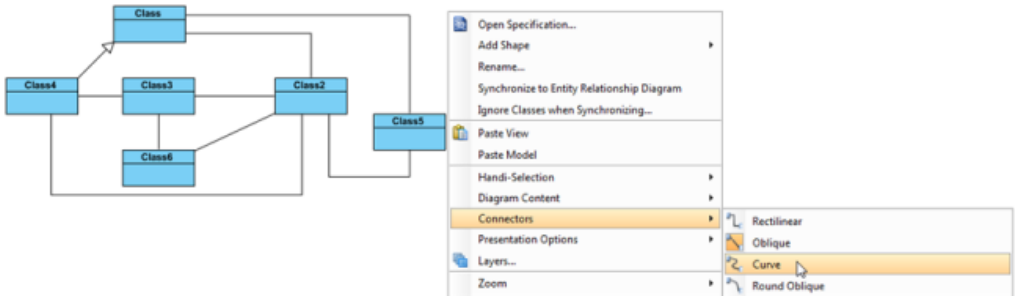
Name	Sample
Rectilinear	
Oblique	



Setting diagram base line style

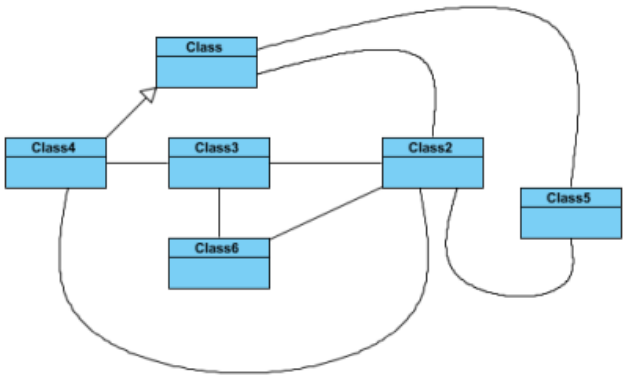
Beside the five styles mentioned above, there also have **Follow Diagram** feature, you don't have to set the connector one by one if you want to change all connectors in the diagrams which defined as **Follow Diagram**.

To change the style of all lines on diagram, right click on the diagram background, select **Connectors** and a line style option from the pop-up menu.



Change diagram line style

As a result, all lines on the diagram are changed into the selected line style.



All lines are changed into curve

Related Resources

The following resources may help you learn more about the topic discussed in this page.

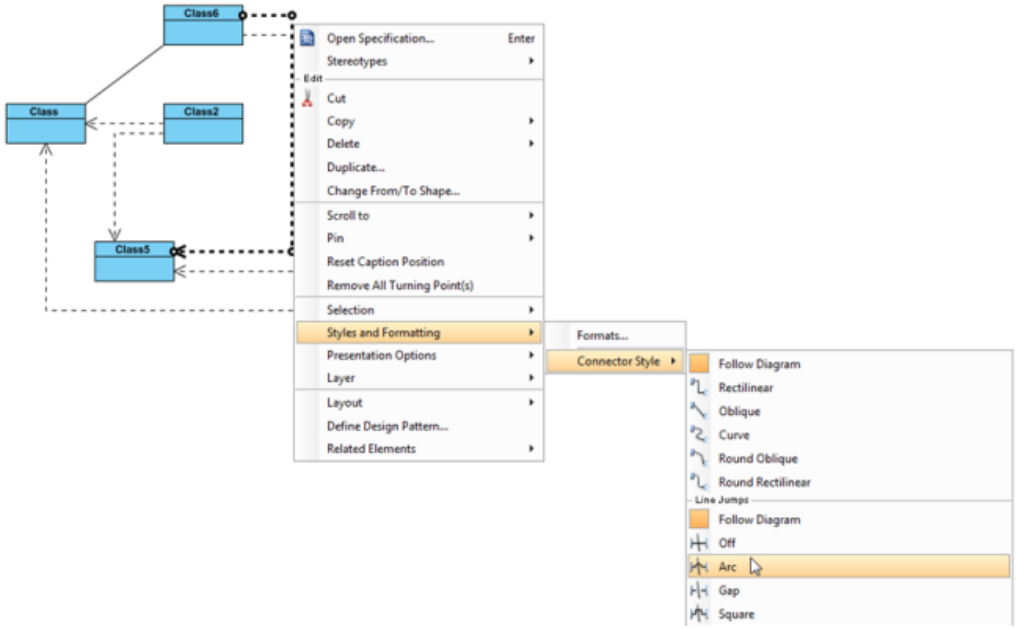
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Setting line jumps options

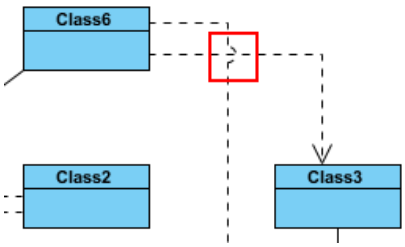
As more diagram elements on your diagram, more miscellaneous connectors are overlapped with each other. It is impossible to explicate on those intersections which connector you indicate. The advantage of Line Jumps is making one of the two connectors different to another to indicate that which connector links with which diagram element clearly. Agilian provides four line jumps options to help you to distinguish connectors. Furthermore, the enhanced feature of line jumps in Agilian enables you to set different size of line jumps.

Setting connector line jumps options

To change the jumps option of a connector, right click on the connector, select **Style and Formatting > Connector Style** and then select one out of four options under **Line Jumps**.



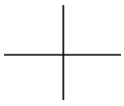
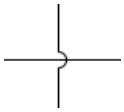
Change Arc line jumps

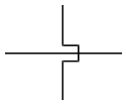


Arc sample

NOTE: The line jumps options are available only when two connectors are overlapped.

Line jump options

Name	Sample
Off	
Arc	
Gap	

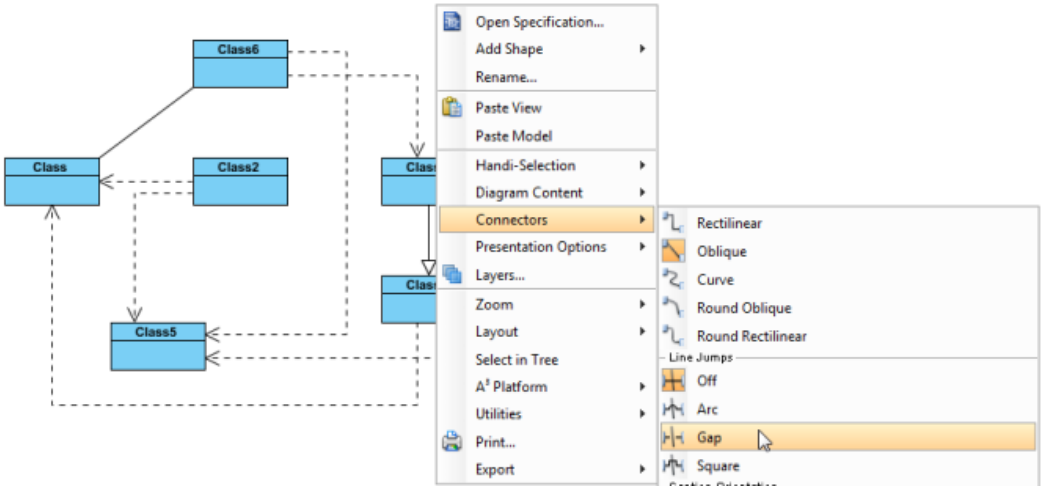
Square


The description of 4 line jump options

Setting diagram base line jumps options

In addition to the 4 options mentioned above, **Follow Diagram** is another choice for altering the connectors. The main feature of **Follow Diagram** is, instead of setting it one by one, all connectors in the diagrams can be changed simultaneously instead of setting one by one.

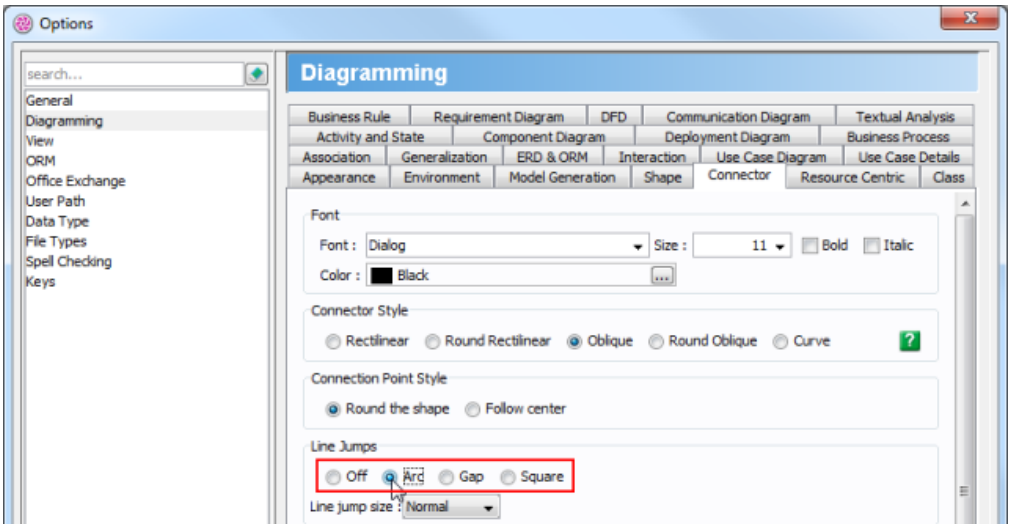
Right Click on the diagram background, select **Connectors** and select one out of four options under **Line Jumps** from the popup menu.



*Select **Gap** from the pop-up menu*

Setting Line Jump for New Project

1. Select **Tools > Project Options** from the main menu.
2. In the **Options** dialog box, open the **Diagramming** page, open the **Connector** tab and select the **Line Jumps** style, or select **Off** to disable it. At last, click **OK** to confirm the changes.



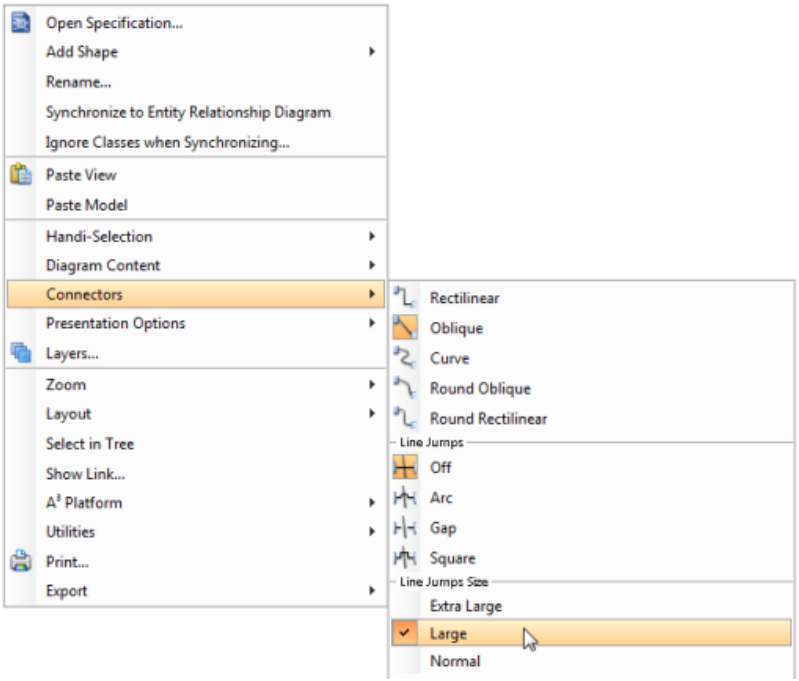
*Select **Arc** in **Options** dialog box*

Setting different line jump size

You can enlarge the line jump size to make the selected line jump (or all line jumps) on the connector(s) more obvious. Furthermore, the size of line jump can be customized in either the current diagram or the future diagram. If you only want to set to the connectors of current diagram, right click on the diagram's background, select **Connectors** and then a size option from the pop-up menu. Otherwise, if you want to set to the connectors of future diagram, set it through **Options** dialog box. Three size are availabe for chooising. Normal is the standard size by default while extra large is the maximum size.

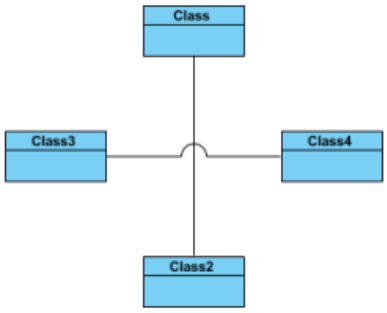
Setting line jump size for current diagram

1. Right click on the diagram's background, select **Connectors** and then a size option from the pop-up menu.



*Set **Large***

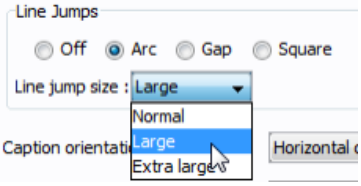
2. The line jump(s) on the current diagram will turn into the size you selected.



Arc in large size

Setting line jump size for future diagram

1. Select **Tools > Project Options** from the main menu to open the **Options** dialog box.
2. In the **Options** dialog box, click **Diagramming** page and open **Connector** tab.
3. Under the **Line Jumps** section, check an line jump option and then select a line jump size option from **Line jump size**'s drop-down menu.



*Select **Large***

Related Resources

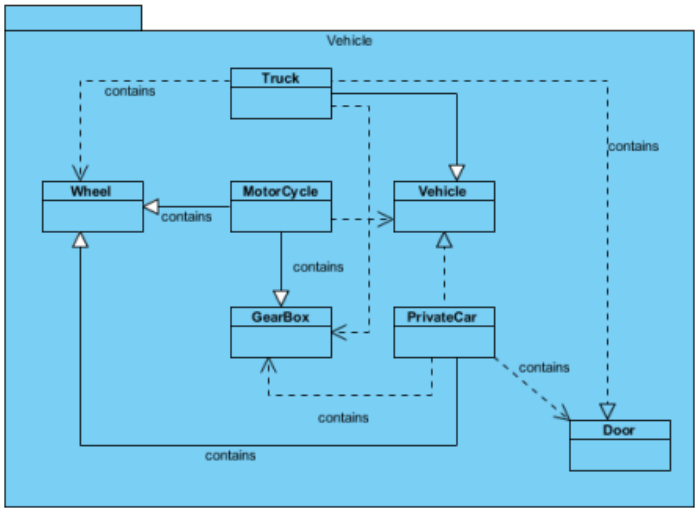
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Setting connector caption orientation

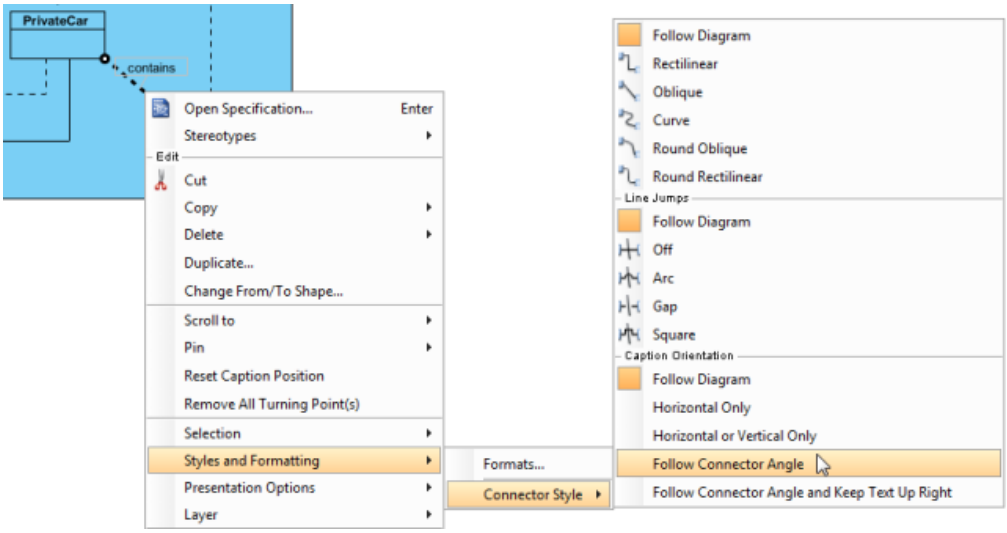
Visual Paradigm supports a number of ways of aligning connector caption , which apply on different modeling preferences. By default, the caption of a connector is aligned horizontal only, but you also can customize it to **Follow Diagram**, **Horizontal Only**, **Horizontal or Vertical Only**, **Follow Connector Angle**, and **Follow Connector Angle and Keep Text Up Right**. You can either customize it one by one or change all connectors in the diagram which defined **Follow Diagram**.

Setting connector caption orientation

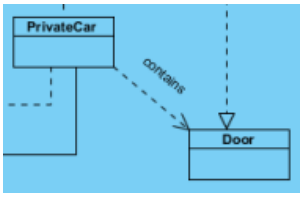


Sample class diagram

To customize the caption orientation option of a specify connector, select the connector, right click and select **Style and Formatting > Connector Style**, and then select one out of four options under **Caption Orientation**.



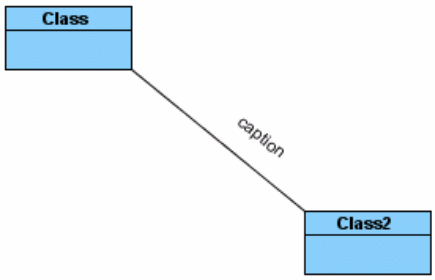
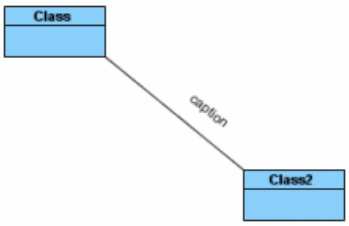
Change caption orientation to Follow Connector Angle



Follow Connector Angle sample

Caption orientation options

Name	Sample	Description

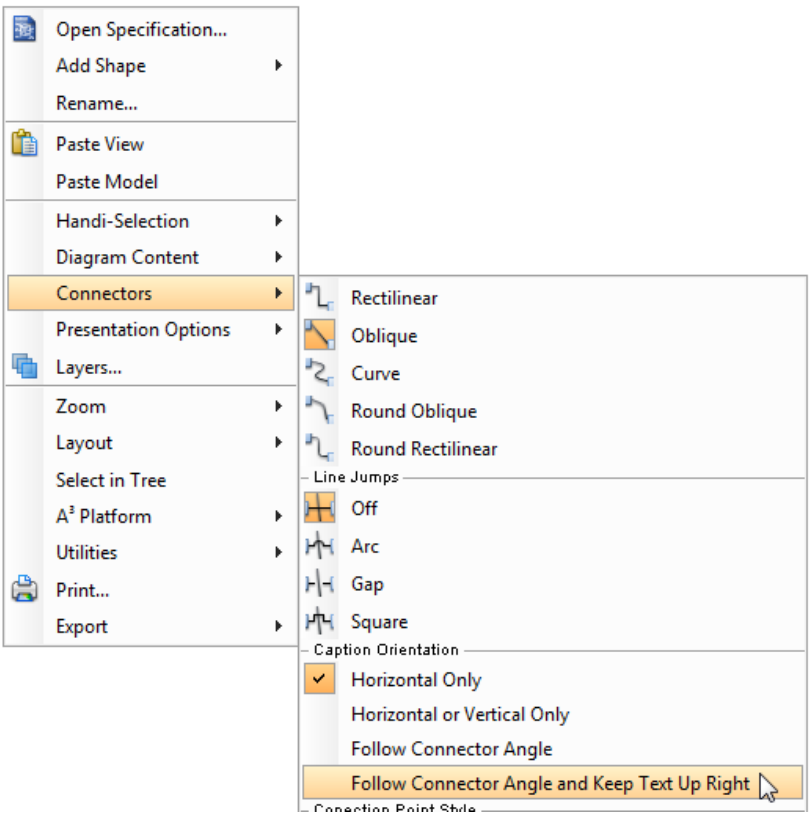
Horizontal Only		**The caption of connector is aligned horizontally.**
Horizontal or Vertical Only		**The caption of connector is aligned either horizontally or vertically, according to the connector.**
Follow Connector Angle		**The caption of connector is aligned the diagonal angles of both shapes.**
Follow Connector Angle and Keep Text Up Right		**The caption of connector is aligned the diagonal angles of both shapes, but keeps top right.**

The description of 4 caption orientation options

Setting diagram base connector caption direction

In addition to the 4 options mentioned above, **Follow Diagram** is another choice for altering the connector. The main feature of **Follow Diagram** is, all connectors in the diagrams can be changed simultaneously instead of setting one by one.

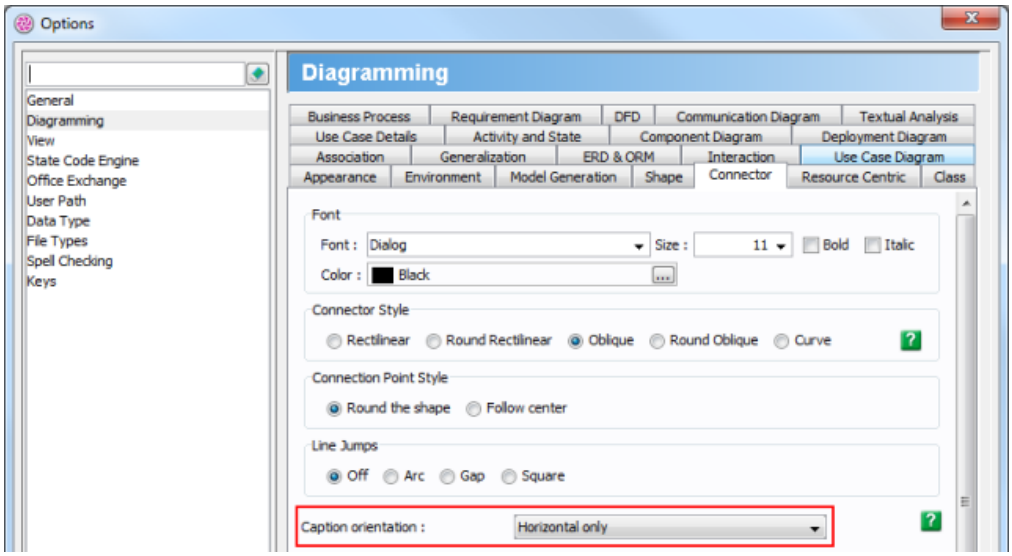
Right click on the diagram background, select **Connectors** and select one out of four options under **Caption Orientation** from the popup menu.



Change caption orientation by diagram pop-up menu

Workspace wide

Workspace wide setting affects the new connectors being created in projects created or will be created under the current workspace, including the opening project. To set, select **Tools > Project Options** from the main menu. In the **Project Options** dialog box, open the **Diagramming** page, switch to the **Connector** tab and select the desired way of aligning caption under the **Caption Orientation** drop down menu. At last, click **OK** to confirm the changes.



*Select **Horizontal only** in **Options** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

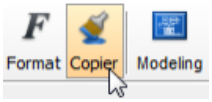
Format copier

Format is defined as the properties for a shape in terms of fill, line and font. Shapes are formatted for two major reasons: making your project more attractive and giving emphasis on the meaning of shapes. However, it would be troublesome and time-consuming to repeat the same action when you need more than one shape to have exactly the same format as the previous one you have already done. Format copier can deal with this problem for you. It's so handy that you can clone the formatting properties from one shape to another or even more.

Copying format to another shape

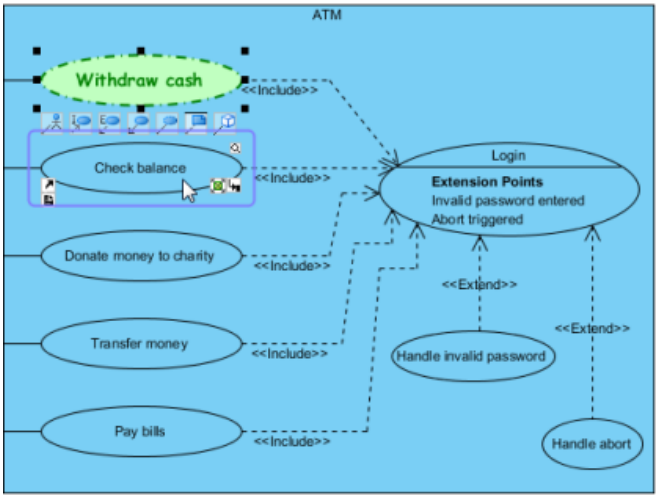
If you want another shape to have exactly the same formatting properties as the previous one you have done, you can simply:

1. Click on the shape that you want its format to be cloned.
2. Click on the **Copier** button on toolbar.



*Click **Copier***

3. Click the shape you want to format.



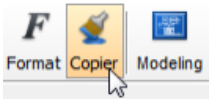
Clone the format property from one shape to another

NOTE: You can copy formatting to another type(s) of shape.

Copying format to multiple shapes

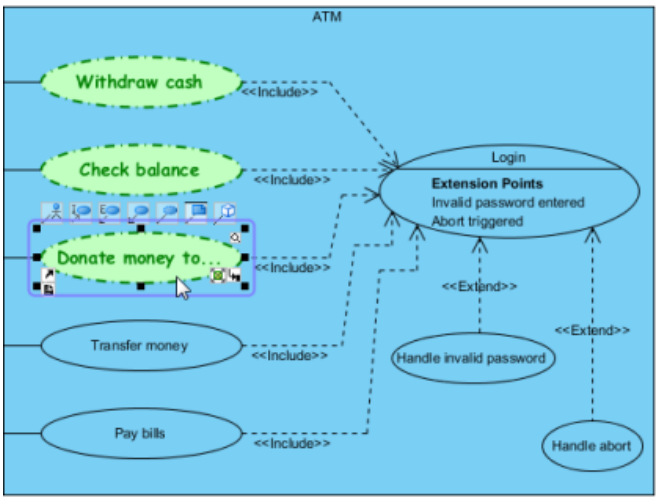
If you want the format properties of your previous shape to be cloned to more than one shape, you should:

1. Click on the shape that you want its format to be cloned.
2. Double click the **Copier** button on toolbar.



Double click Copier

3. Click the shape you want to format.



Clone the format property from one to multiple shapes

NOTE: If you don't want the format properties to be cloned to other shapes any more, you should cancel cloning by clicking **Copier** once again.

NOTE: You can only copy format to shapes within the same diagram.

Related Resources

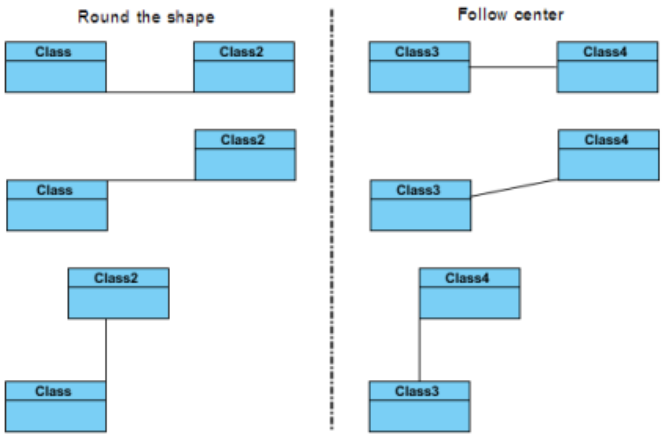
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Set connection point style

The connection point of a connector is used to connect from the original shape to the target shape using a connector. In Agilian , you can choose one of two kinds of connection point style: either Round the shape or Follow center for each shape. Round the shape allows you to set the last connection point of the connector moving along the boundaries of the original shape while Follow center refers the last connection point of the connector depends on the center of the original shape. The most attractive point of connection point style feature is that the animation of a connection point style will be playing repeatedly once you select the corresponding connection point style.

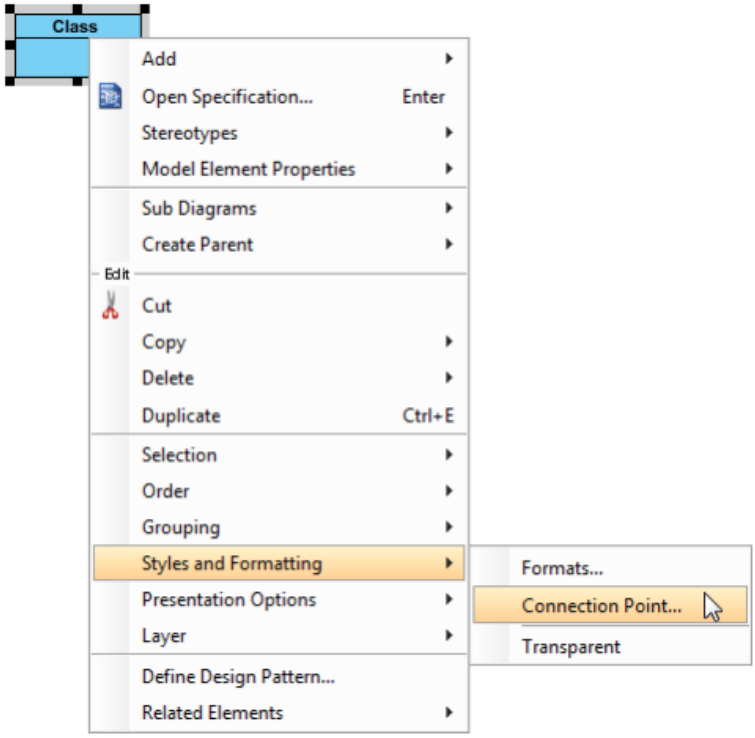
You can compare the differences of two kinds of connection point style shown as follows:



Compare two kinds of connection point style

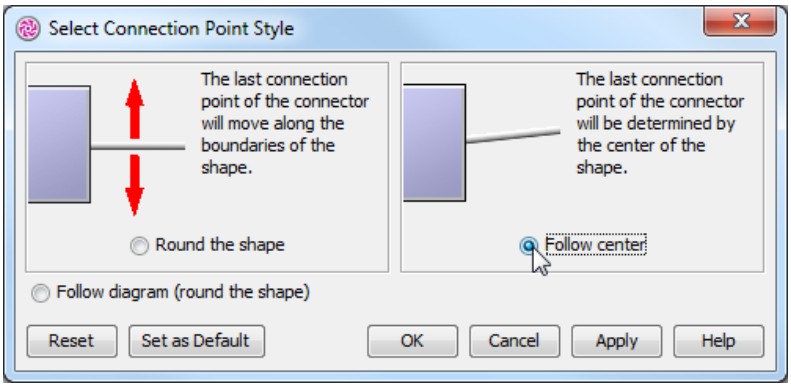
To set a connection point style:

1. Right click on shape and select **Styles and Formatting > Connection Point...** from the pop-up menu.



*Open **Select Connection Point Style** dialog box*

2. In the **Select Connection Point Style** dialog box, choose a connection point style option. Click **OK** button.



*Choose **Follow center** option*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Present shape as primitive shape

In addition to the standard appearance of notations (e.g. stickman for actor, box for class, oval for use case, etc), here you have one more option - the primitive option. The primitive option enables you to present any kind of shape as simple rectangle/oval/rounded rectangle shape. You can specify custom text to such "primitive shape", which is particular useful for modeling for general-purposes. A sample use would be to present an actor that represents a computer as a rectangle, and then describe its role with custom text and have it presented on diagram.

To show a shape as primitive shape:

1. Right click on the shape and select **Presentation Options > Primitive Shape...** from the popup menu.
2. Adjust the appearance of shape by setting **Shape type**.
3. Adjust the content of shape by setting **Text**. The **Name** option makes the name of shape presented. The **Tagged value** option enables you to display the value of a specific tag. You need to enter the tag name in the drop down menu, or select from the list of existing tags. The **Custom** option enables you to display whatever text you want.
4. Click **OK** when ready.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

General modeling techniques

This chapter covers all the general model techniques that increase the productivity of your work.

Automatic diagram layout

Helps you reorganize shapes and connectors on diagram to make diagram tidy. Several options are provided to produce different results of layout.

Fit shape size

To fit the size of shape against the update of shape content, or to fit manually.

Diagram element selection

Description of various ways of shape selection including traditional range selection and handi-selection.

Copy and paste

Common copy-and-paste editing features. Instead of pasting inside Agilian you can paste to external applications like MS Word, or to paste as XML to aid in interoperability.

Alignment guide

Helps to make sure shapes are aligned well through the visible guide line.

Reverse connector direction

To immediately reverse the direction of connector without deleting and recreating it.

Visualize model elements on diagram

Shows you how to show a model element on a diagram.

Visualize related model elements

Shows you how to show related elements of a shape on a diagram.

Adding comments

Shows how to add comments to shapes or diagrams.

Pinning connector ends

By pinning a connector's end(s) you freeze their position and make them point to the desired position.

Align and distribute diagram elements

Describes the steps about aligning and distributing shapes.

Adjusting caption's position and angle

For BPMN event and gateway, their captions can be configured to show in specific orientation. You can also make it draggable (default not).

Zooming diagram

Magnify or diminish diagram content by zooming in and out.

Diagram grids

Learn how to improve shapes' alignment and positioning with the help of diagram grids.

Automatic diagram layout

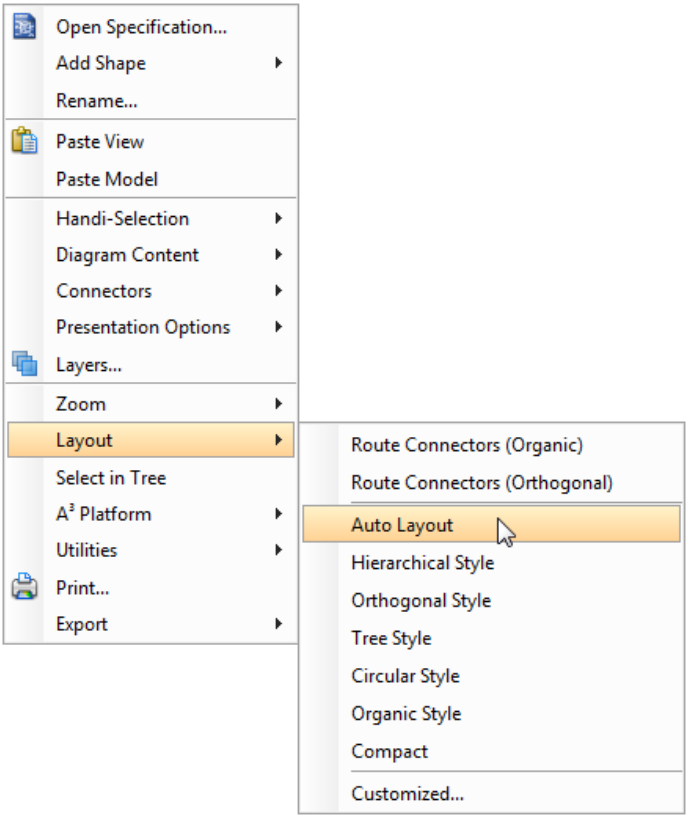
Agilian provides a layout facility for arranging diagram elements in diagrams. Diagram elements do not overlap and the relationship links do not cross over each another. Different layout styles and configurable options are provided, which allow extremely flexible and sophisticated layouts to be applied to diagrams.

Automatic layout diagram

There are a few different kinds of layouts: **Auto Layout**, **Orthogonal Layout**, **Hierarchic Layout**, **Directed Tree Layout**, **Balloon Tree Layout**, **Compact Tree Layout**, **Horizontal-Vertical Tree Layout**, **BBC Compact Circular Layout**, **BBC Isolated Circular Layout**, **Single Cycle Circular Layout**, **Organic Layout** and **Smart Organic Layout**.

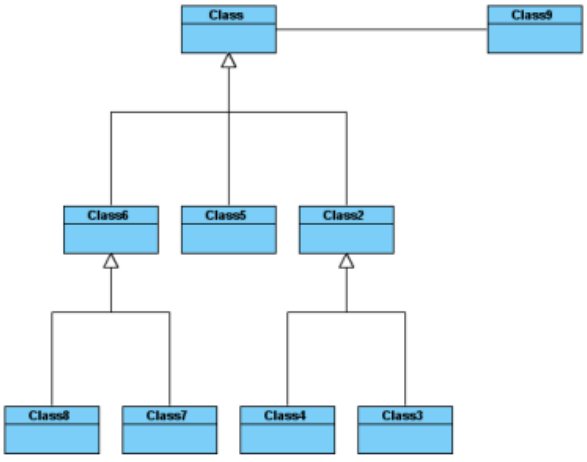
Auto layout

Selecting auto layout signifies that the most suitable layout is arranged for shapes automatically. It is the best choice for users when they have no preference in selecting a specific layout. To apply **Auto Layout** to the diagram, right click on the diagram's background and select **Layout > Auto Layout** from the pop-up menu.



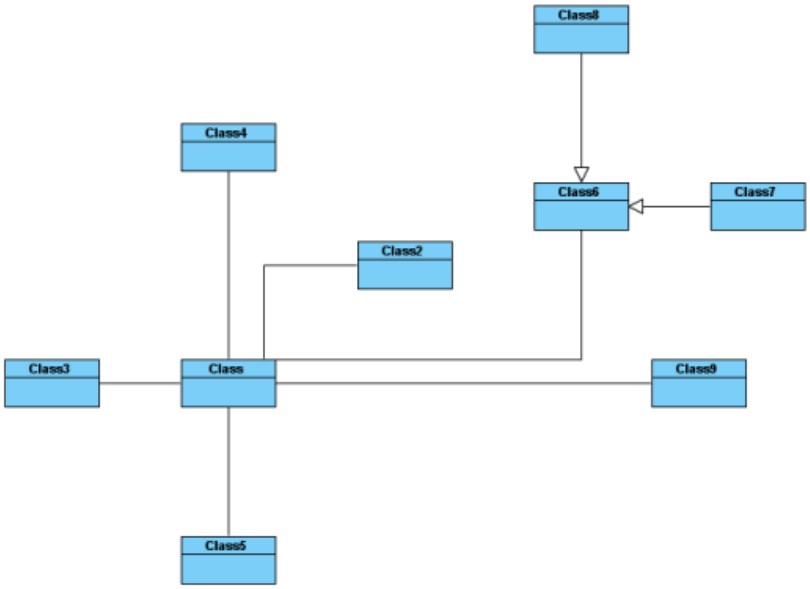
*Select **Auto Layout***

Class diagram (Hierarchy base / factory class diagram)



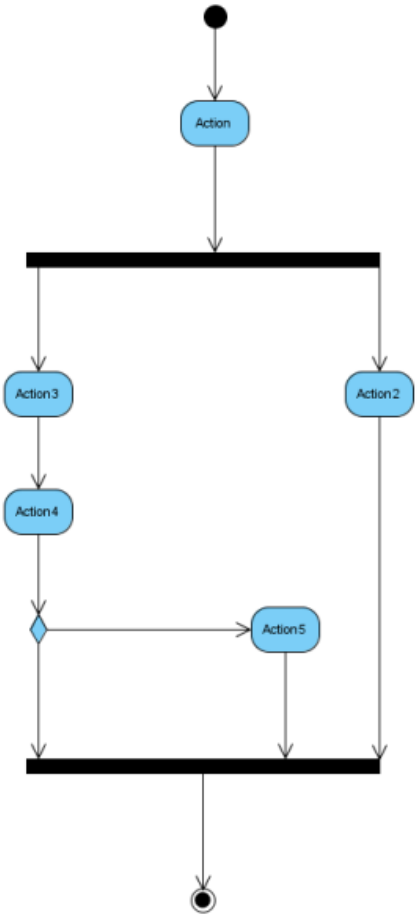
Hierarchy base (Factory class diagram)

Class diagram (Navigation base / mediator class diagram)



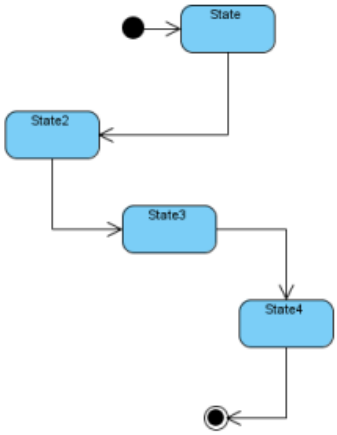
Navigation base (Mediator class diagram)

Activity diagram



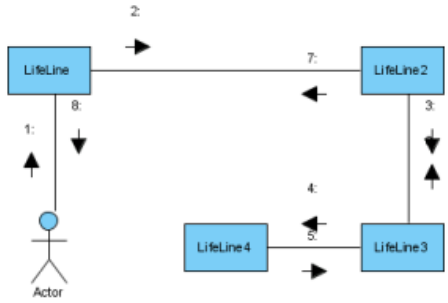
Auto Layout of activity diagram

State machine diagram



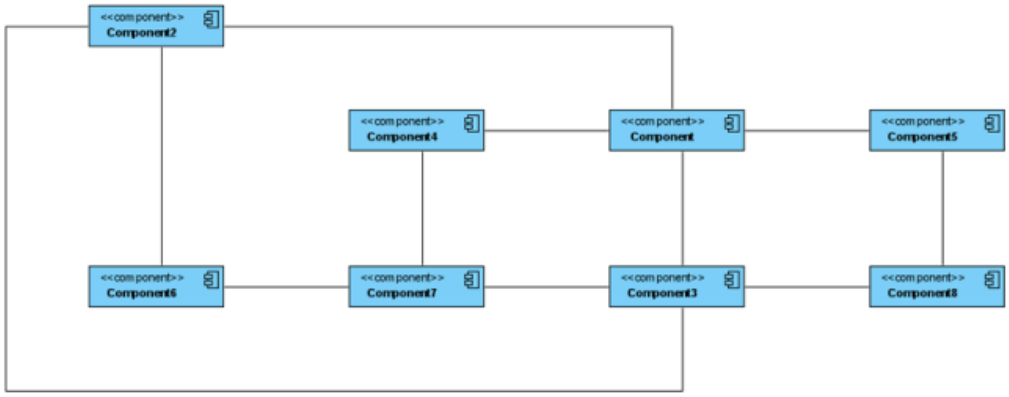
Auto layout of state machine diagram

Communication diagram



Auto layout of communication diagram

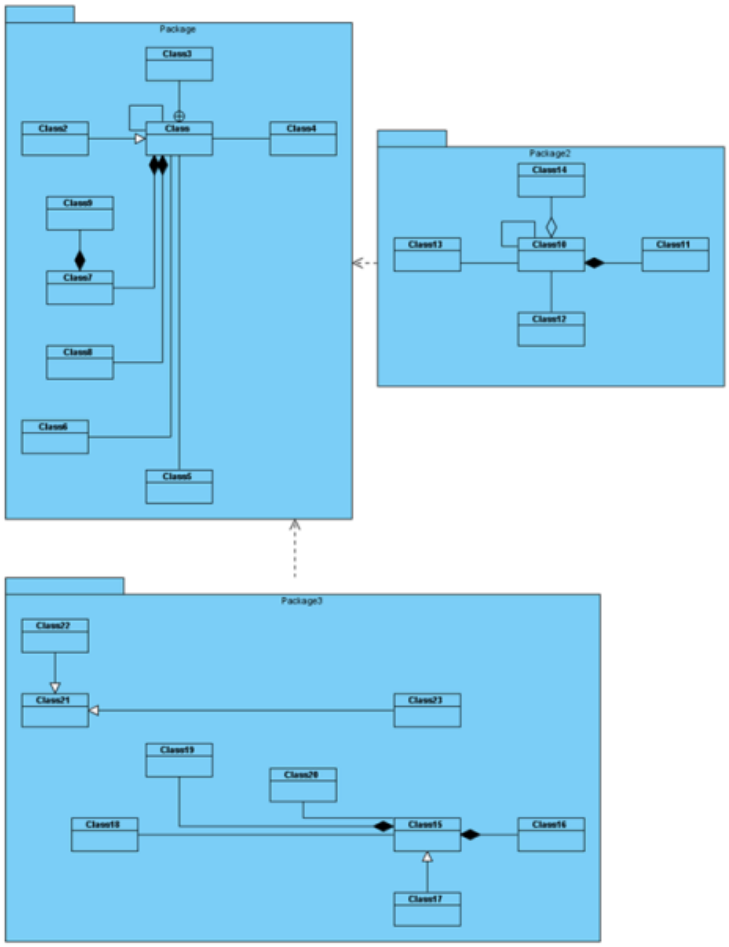
Other diagrams



Auto layout of other diagrams

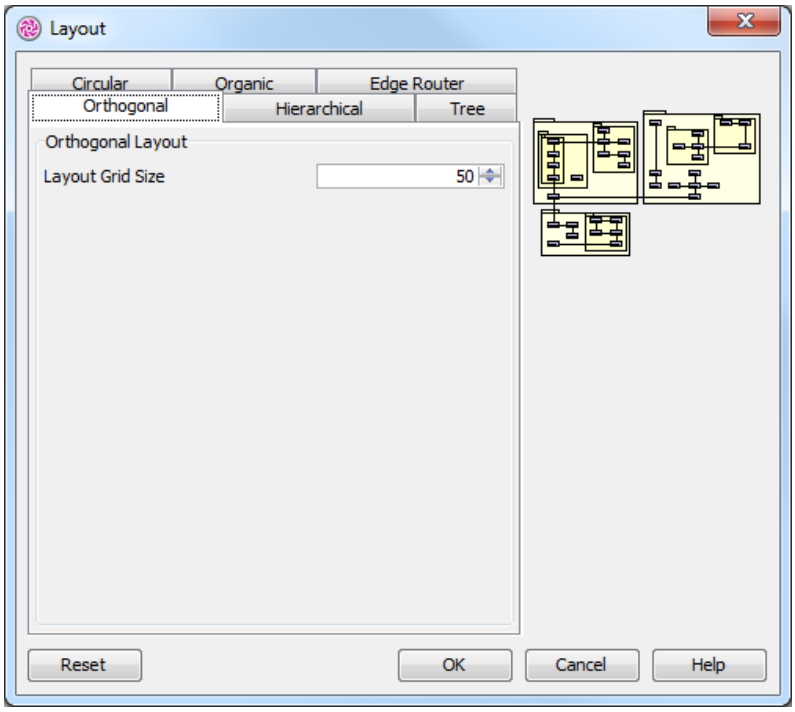
Orthogonal layout

Shapes are arranged based on the topology-shape-metrics approach in orthogonal layout. It is the best way for users to arrange shapes and connectors in Class Diagrams. As it is default layout in Agilian, every time you drag the models from the Model Tree to a diagram, the orthogonal layout will be applied to arrange the newly created shapes in the Class Diagram.



Orthogonal Layout

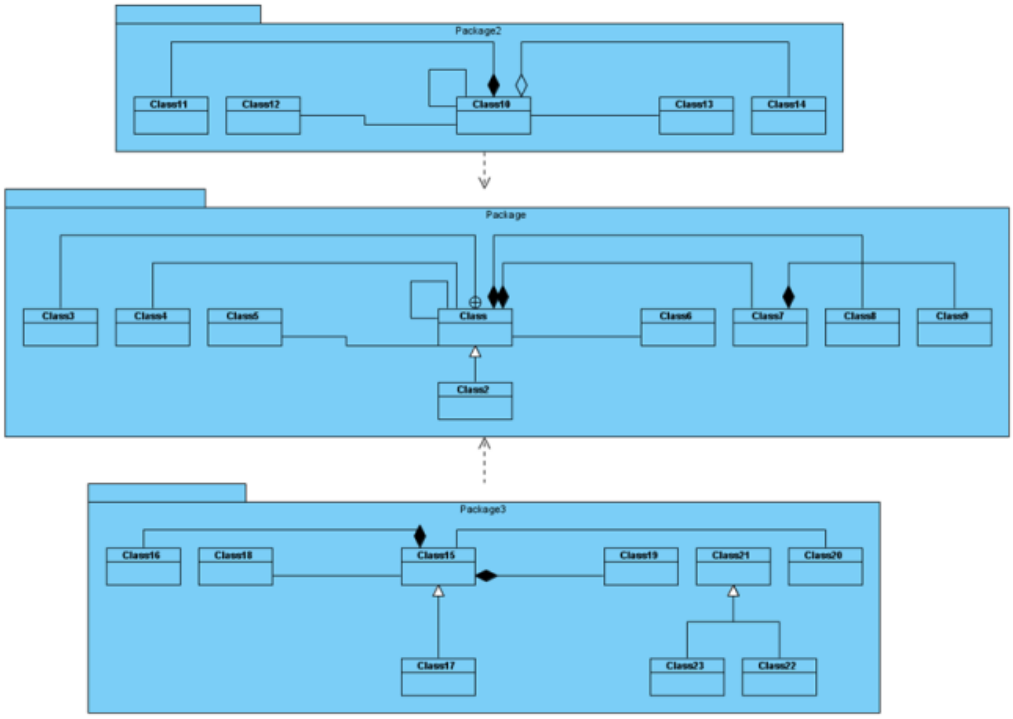
Layout Grid Size: the virtual grid size for layout. Each shape will be placed in accordance with its center point lays on a virtual grid point.



Orthogonal Layout setting

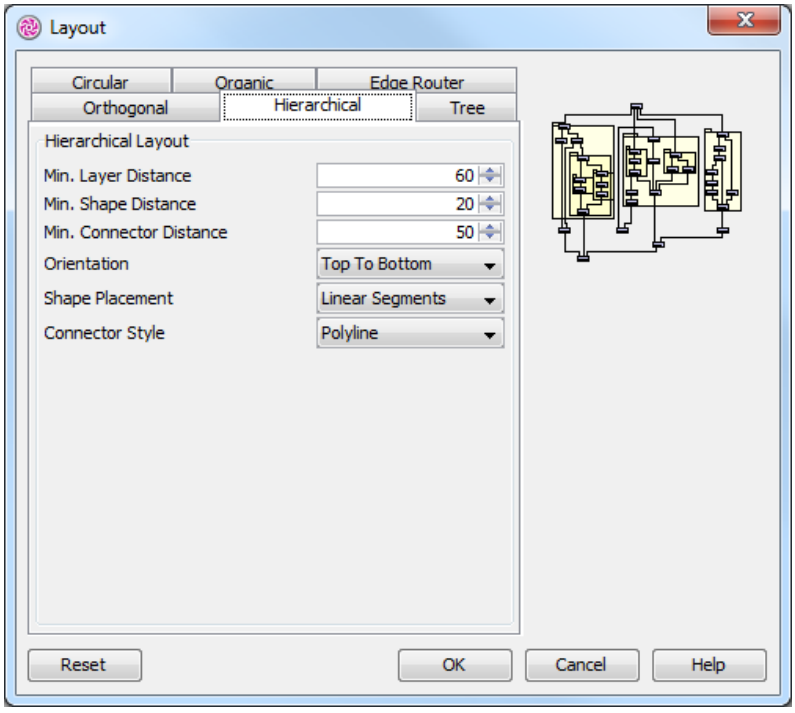
Hierarchic layout

Hierarchic Layout arranges shapes in a flow. It is the best way for users to arrange shapes that have hierarchical relationships, such as generalization relationships and realization relationships.



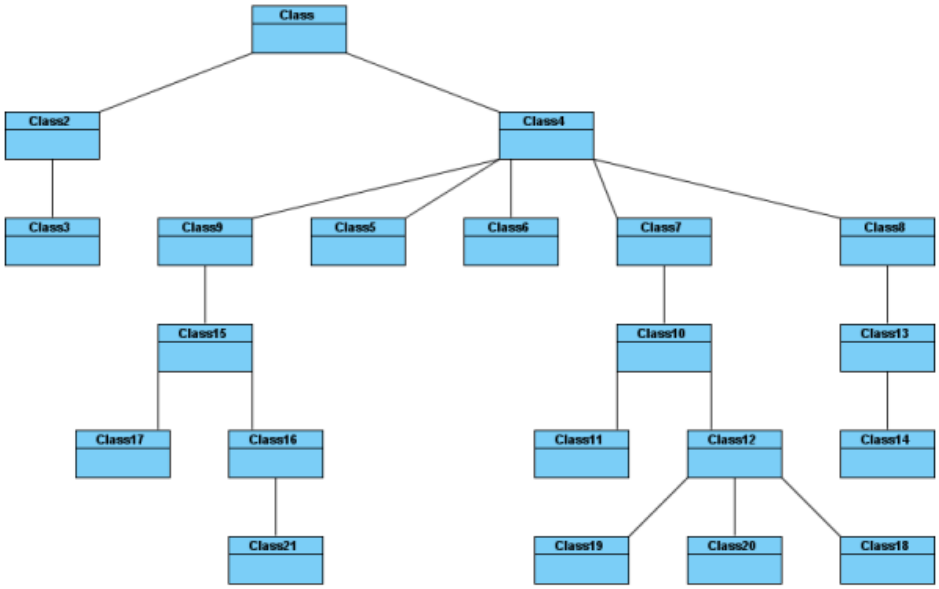
Hierarchic Layout

Min. Layer Distance: the minimal horizontal distance between the shapes.
Min. Shape Distance: the minimal vertical distance between the shapes.
Min. Connector Distance: the minimal vertical distance of the connector segments.
Orientation: the layout direction for arranging nodes and connectors from top to bottom, left to right, bottom to top, and right to left.
Shape Placement: affects the horizontal spacing between shapes, and the number of bends of the connectors -pendulum, linear segments, polyline, tree and simplex.



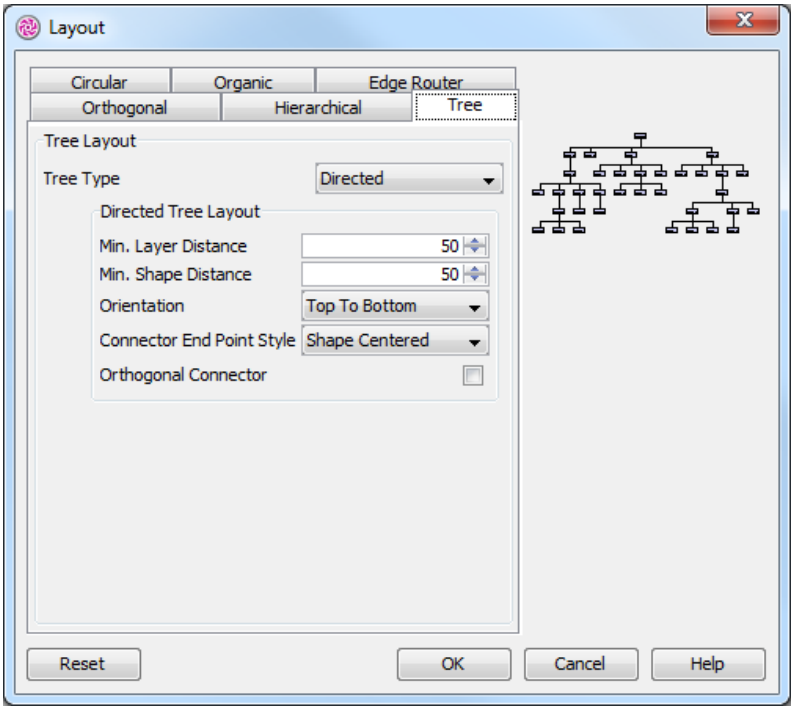
Hierarchical Layout setting

Directed tree layout
Directed Tree Layout, which is one of the tree layouts in Agilian, arranges shapes in a tree structure. It is the best way for users to arrange shapes except those which have hierarchical relationships, such as generalization relationships and realization relationships.



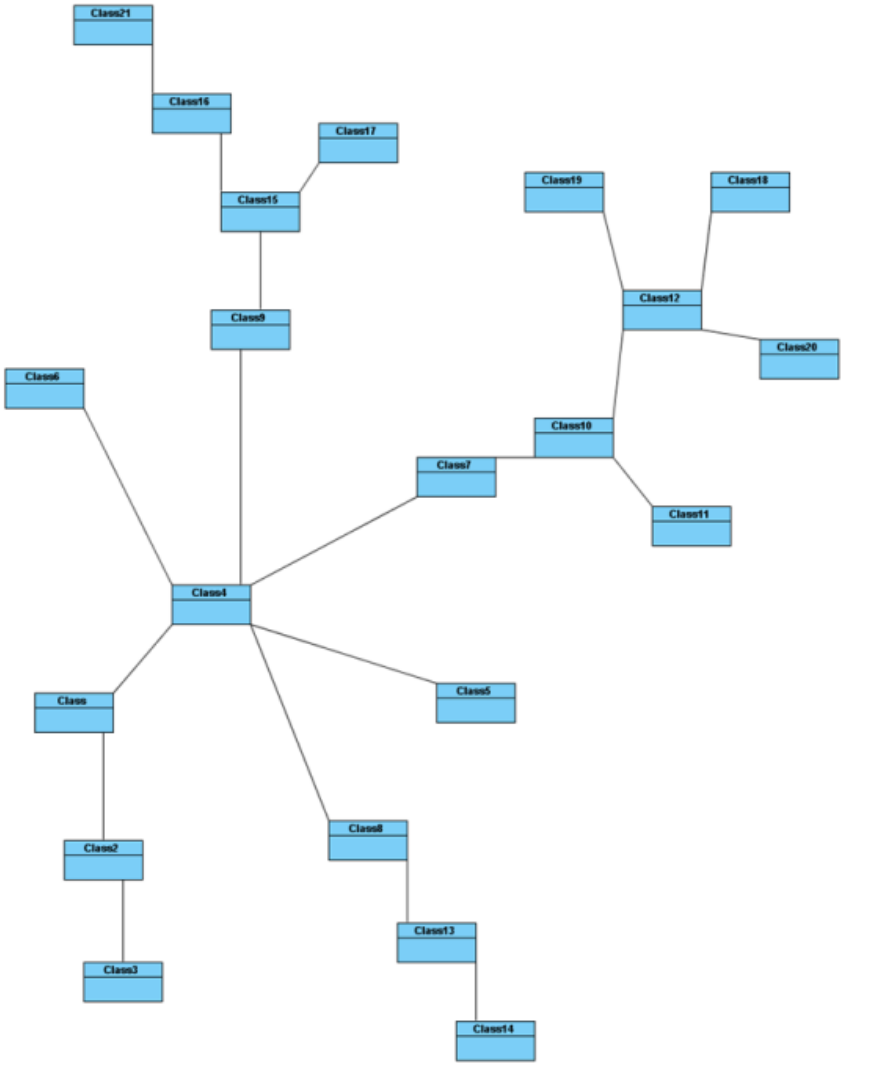
Directed Tree Layout

Min. Layer Distance: the minimal horizontal distance between the shapes.
Min. Shape Distance: the minimal vertical distance between the shapes.
Orientation: the layout direction for arranging nodes and connectors – top to bottom, left to right, bottom to top, and right to left.
Connector End Point Style: how the connector end points will be placed – shape centered, border centered, border distributed.
Orthogonal Connector: whether the connectors will be arranged in orthogonal.



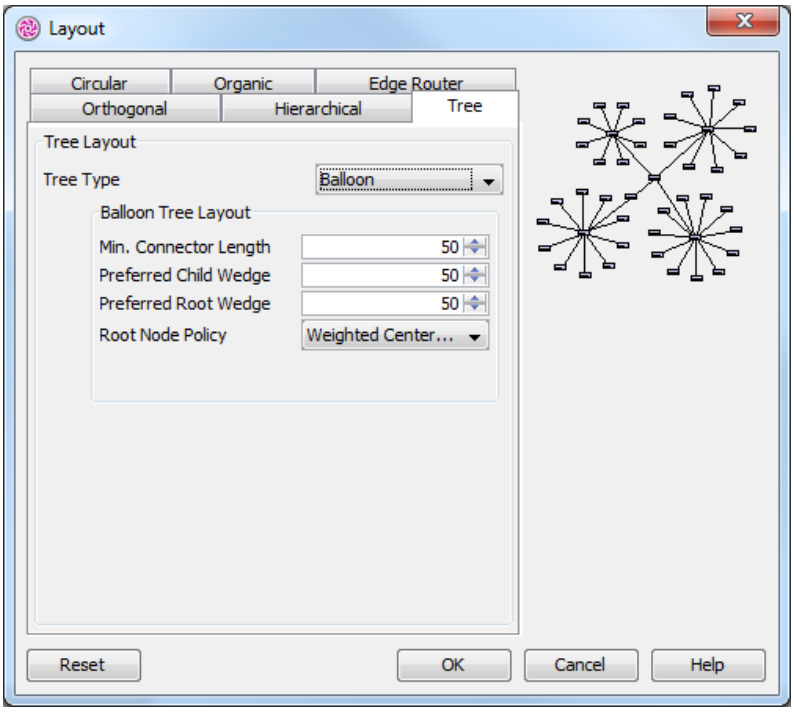
Directed Tree Layout setting

Balloon tree layout
Balloon Tree Layout, which is one of the tree layouts in Agilian, arranges shapes in a tree structure in a radial fashion. It is the best way for users to arrange large trees.



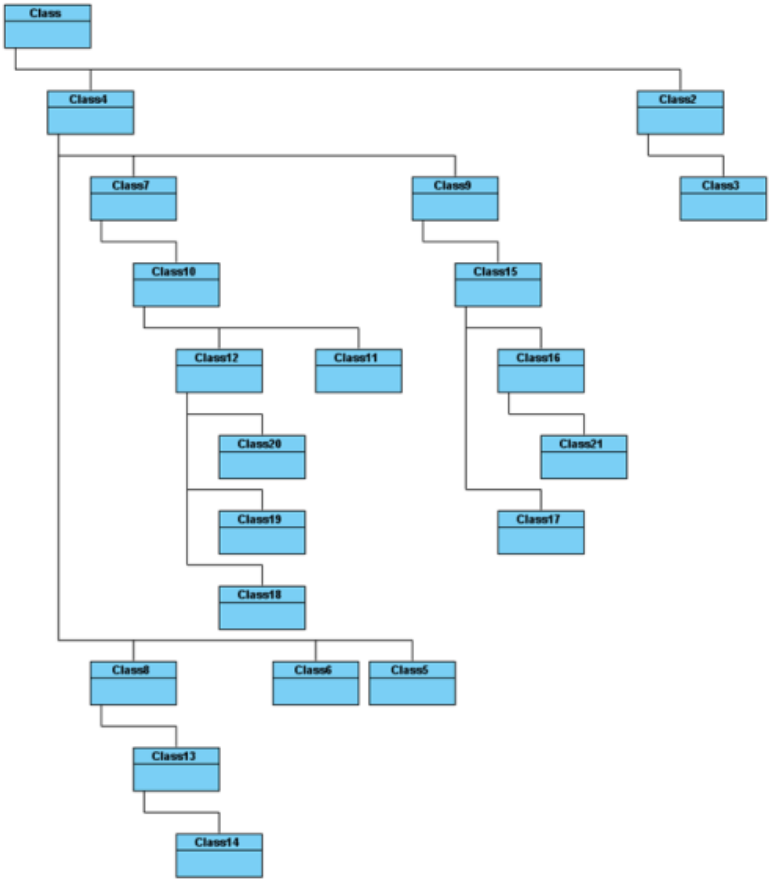
Balloon Tree Layout

Min. Connector Length: the minimal distance between the connectors and shapes.
Preferred Child Wedge: the angle at which the child node will be placed around its parent node.
Preferred Root Wedge: the angle at which a node will be placed around the root node.
Root Node Policy: determines which node is chosen as the tree root node for layout – directed root, center root, and weighted center root.



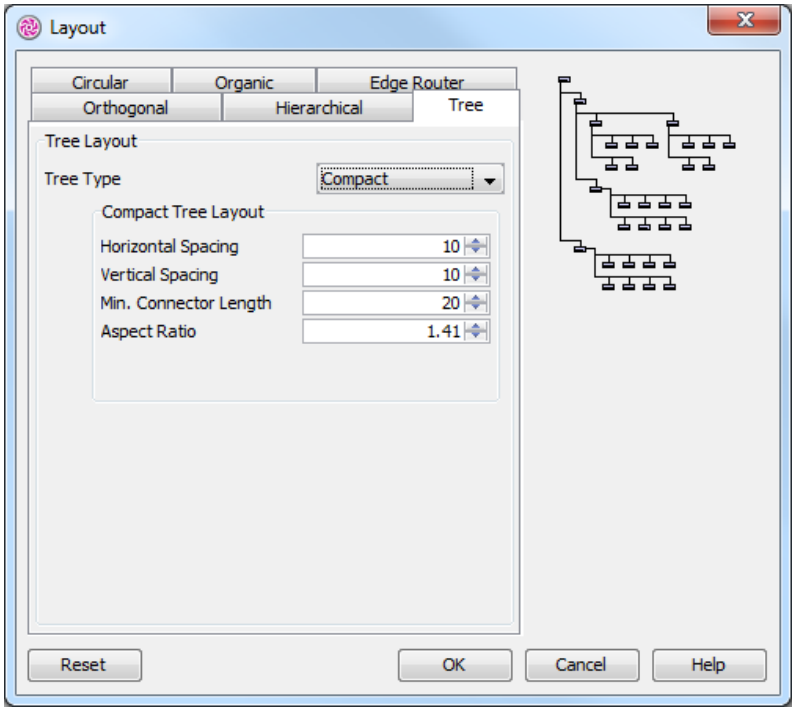
Balloon Tree Layout setting

Compact tree layout
Compact Tree Layout, which is one of the tree layouts in Agilian, arranges shapes in a tree structure. The aspect ratio (relation of tree width to tree height) of the resultant tree can be set.



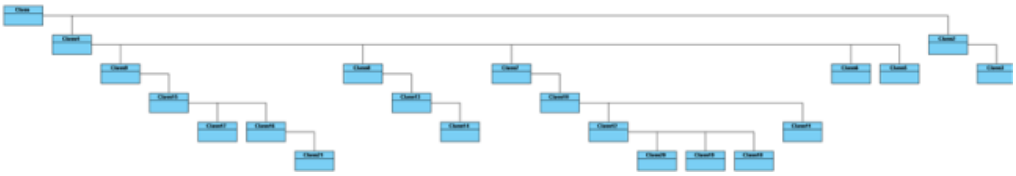
Compact Tree Layout

Horizontal Spacing: the horizontal spacing between the shapes.
Vertical Spacing: the vertical spacing between the shapes.
Min. Connector Length: the vertical distance of the connector segments.
Aspect Ratio: the relation of the tree width to the tree height.



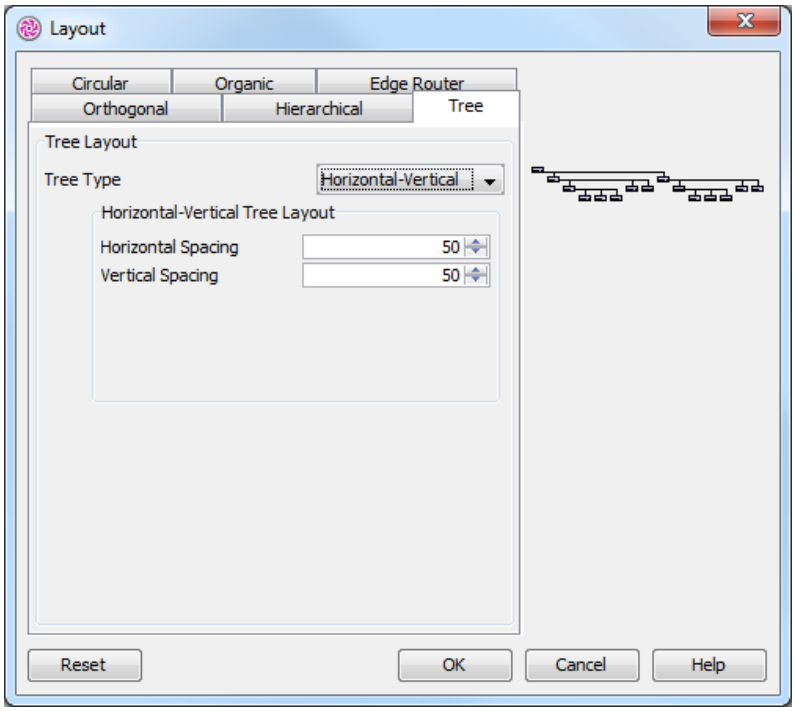
Compact Tree Layout setting

Horizontal-Vertical tree layout
Horizontal-Vertical Tree Layout, which is one of the tree layouts in Agilian, arranges shapes in a tree structure horizontally and vertically.



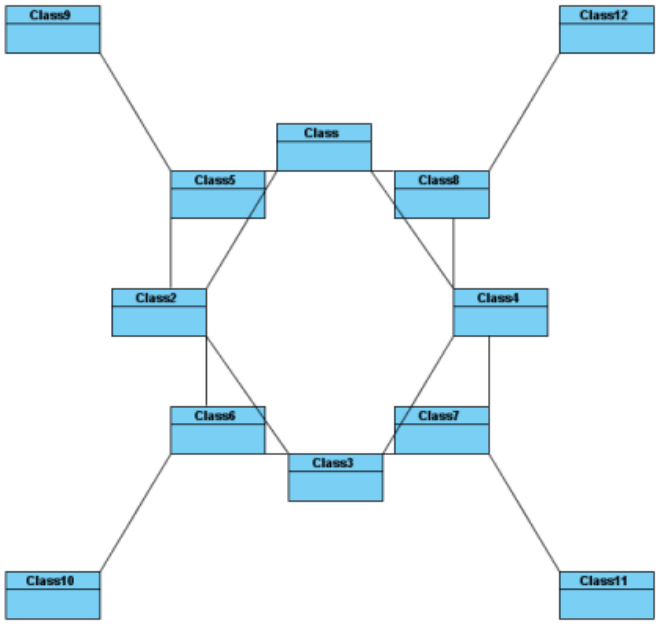
Horizontal-Vertical Tree Layout

Horizontal Spacing: the horizontal spacing between the shapes.
Vertical Spacing: the vertical spacing between the shapes.



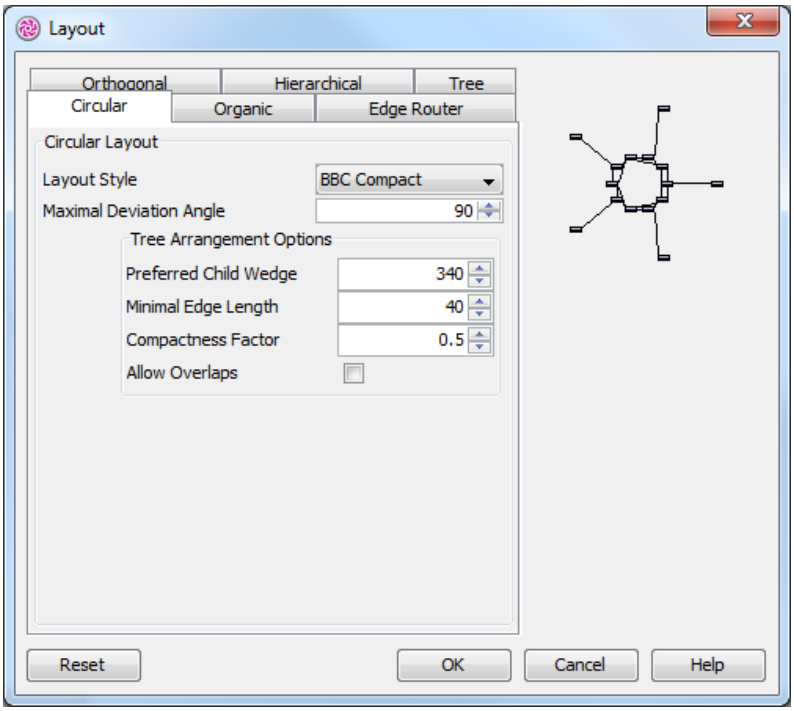
Horizontal-Vertical Tree Layout setting

BBC compact circular layout
BBC Compact Circular Layout, which is one of the circular layouts in Agilian, arranges shapes in a radial tree structure. The detected group is laid out on the separate circles. It is the best way for user to arrange shapes that belong to more than one group with a ring structure.



BBC Compact Circular Layout

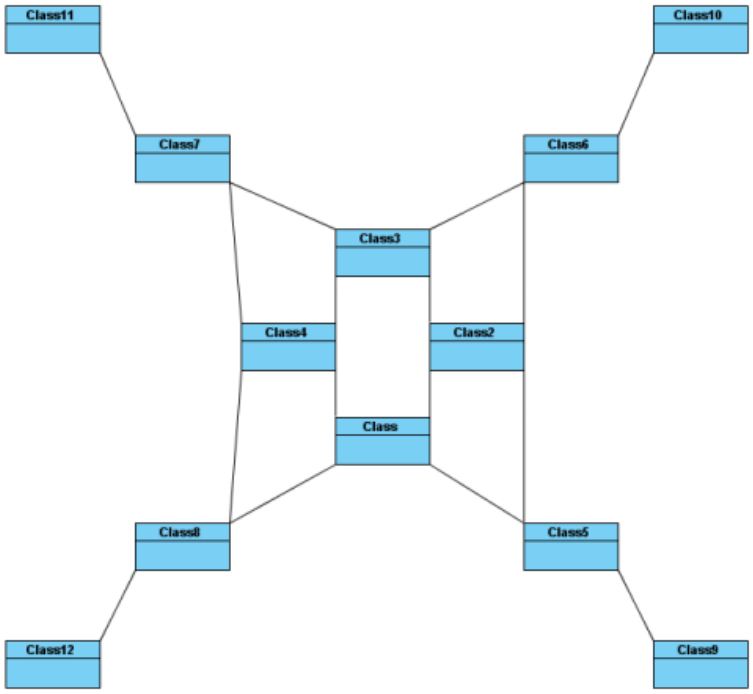
Maximal Deviation Angle: the maximal angle of deviation.
Preferred Child Wedge: the angle at which the child node will be placed around its parent node.
Minimal Edge Length: the minimal distance between the shapes.
Compactness Factor: the parameter that affects the length of connector. The smaller the compactness factor, the length of connectors will be shorter and the layout will be more compact.
Allow Overlaps: whether the shape can be overlapped.



BBC Compact Circular Layout setting

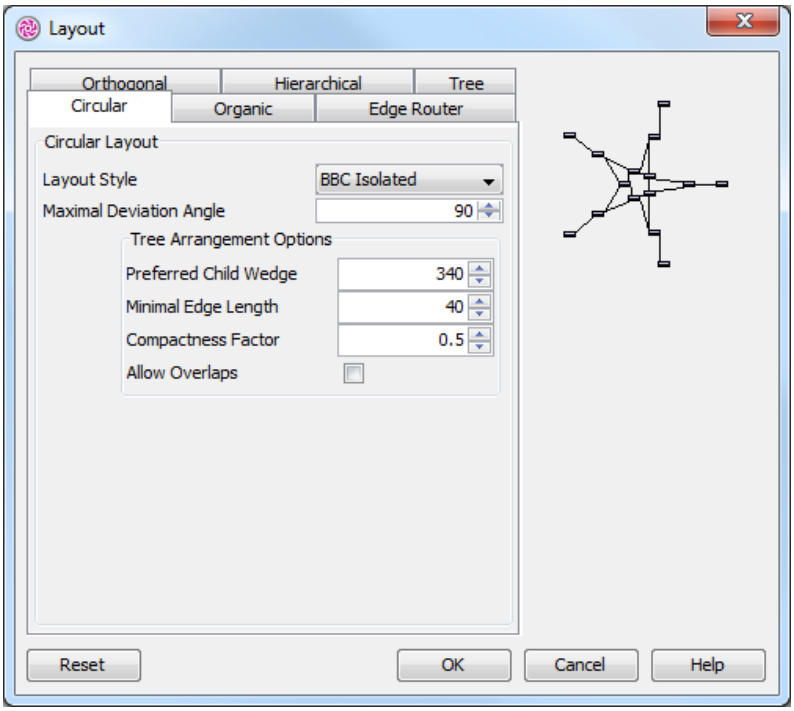
BBC isolated circular layout

BBC Isolated Circular Layout, which is one of the circular layouts in Agilian, arranges shapes into many isolated ring structures. It is the best way for users to arrange shapes that belong to one group with ring structure.



BBC Isolated Circular Layout

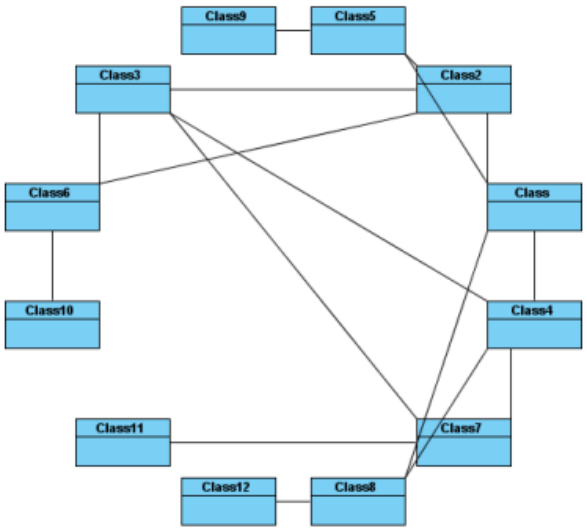
Maximal Deviation Angle: the maximal angle of deviation.
Preferred Child Wedge: the angle at which the child node will be placed around its parent node.
Minimal Edge Length: the minimal distance between the shapes.
Compactness Factor: the parameter that affects the length of connector. The smaller the compactness factor, the length of connectors will be shorter and the layout will be more compact.
Allow Overlaps: whether the shape can be overlapped.



BBC Isolated Circular Layout setting

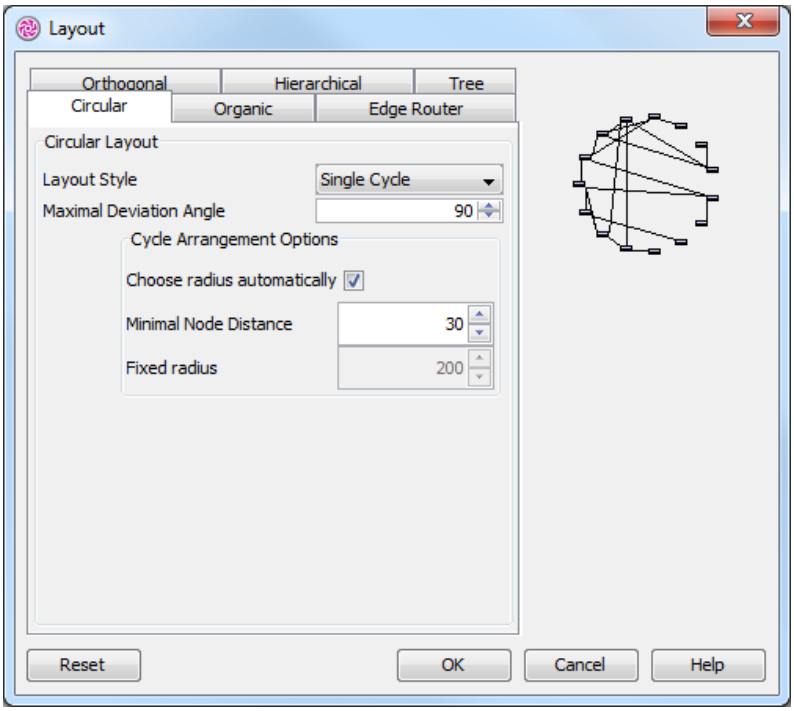
Single cycle circular layout

Single Cycle Layout, which is one of the circular layouts in Agilian, arranges shapes in circular structure in single circle.



Single Cycle Circular Layout

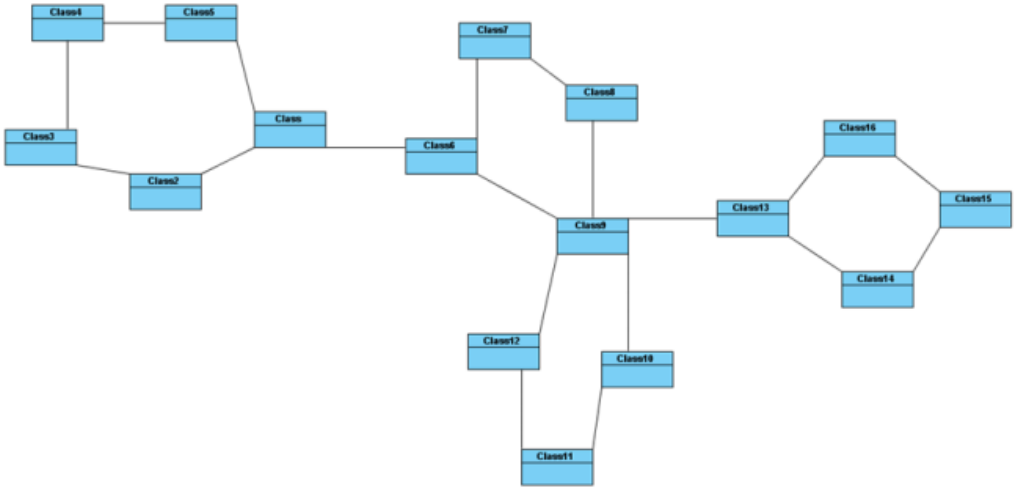
Choose radius automatically: determine the radius of circular structure automatically or manually.
Minimal Node Distance: the minimal distance between the nodes.
Fixed radius: the radius of circular structure.



Single Cycle Circular Layout setting

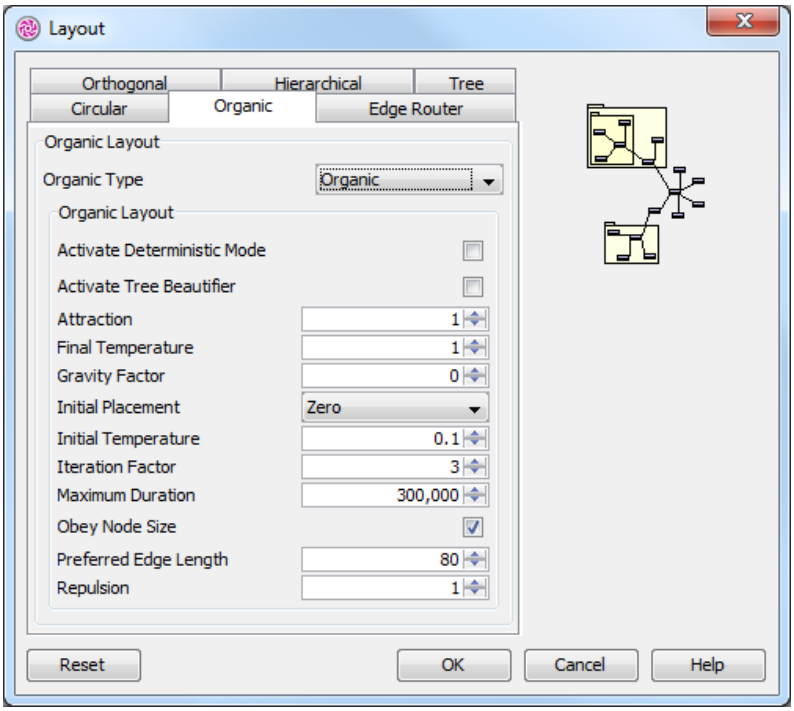
Organic layout

Organic Layout, which is one of the organic layouts in Agilian, arranges shapes in a star or ring structure. It is the best way for users to arrange the shapes that have highly connectivity relationship.



Organic Layout

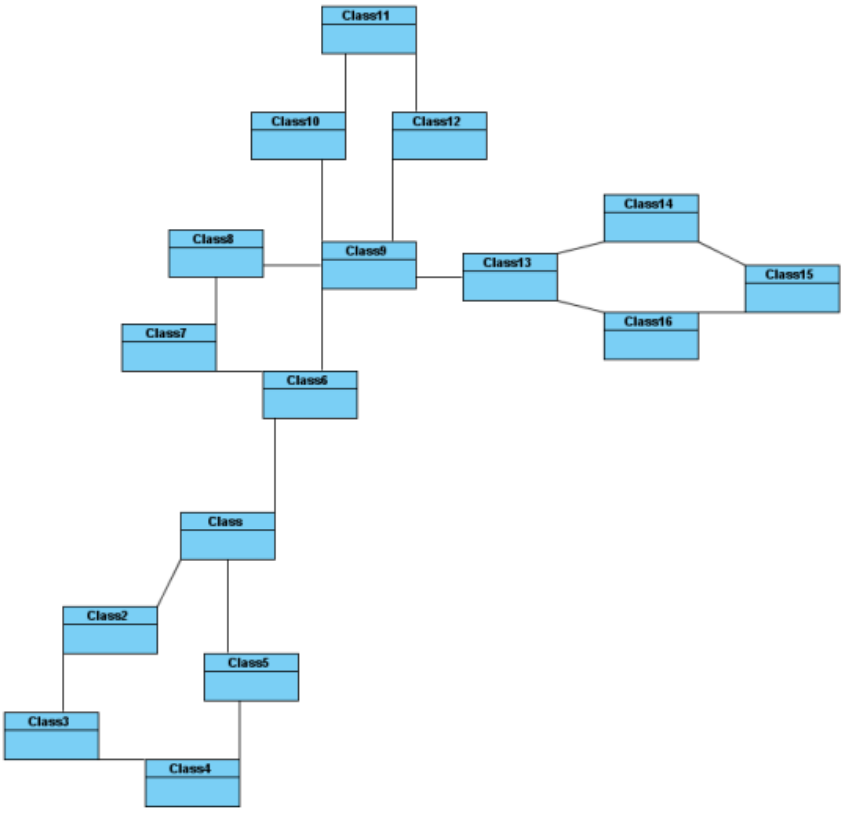
Activate Deterministic Mode: whether the layouter is in deterministic mode.
Activate Tree Beautifier: whether or not to activate the subtree beautifier.
Attraction: the degree of the attraction between shapes.
Final Temperature: the factor that affects the distance between shapes.
Gravity Factor: the factor that affects the distance between shapes and the center.
Initial Placement: the initial value of placement.
Initial Temperature: the initial value of temperature.
Iteration Factor: the degree of iteration.
Maximum Duration: the maximum degree of duration.
Obey Node Size: the size of obey shapes.
Preferred Edge Length: the preferred length between the nodes.
Repulsion: the factor that affects the distance between shapes which belong to the same ring or star structure.



Organic Layout setting

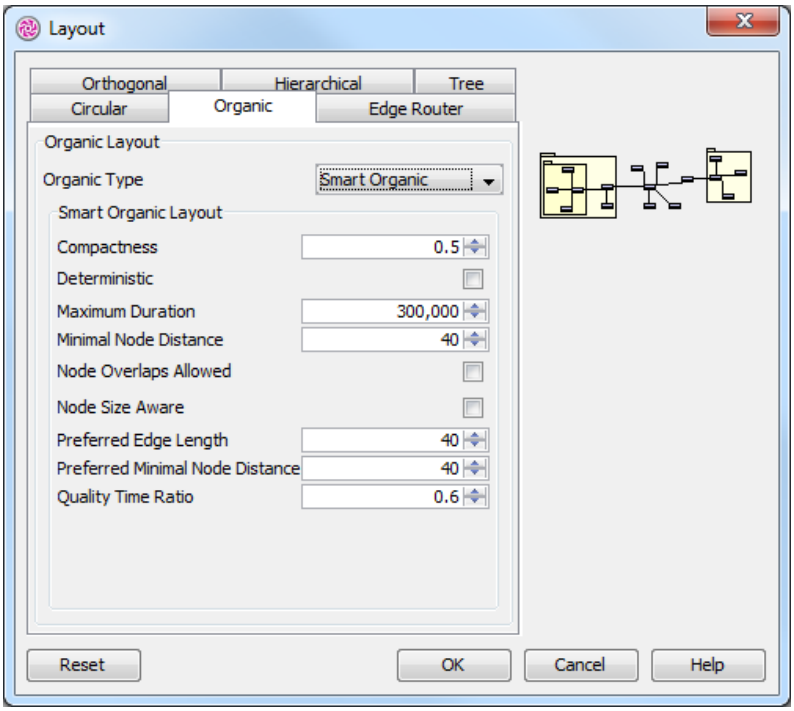
Smart organic layout

Smart Organic Layout, which is one of the organic layouts in Agilian, is a variant of the Organic Layout. It can set the ratio of the quality: producing time of layout and controls the compactness of layout.



Smart Organic Layout

Compactness: the factor that sets less/more compact layout.
Deterministic: whether the layouter is in deterministic mode.
Minimal Node Distance: the minimal distance between nodes.
Node Overlaps Allowed: whether the node can be overlapped.
Node Size Aware: whether the node size can be aware.
Preferred Minimal Node Distance: the preferred minimal distance between the nodes.
Quality Time Ratio: the ratio of the quality of layout to the producing time of layout.

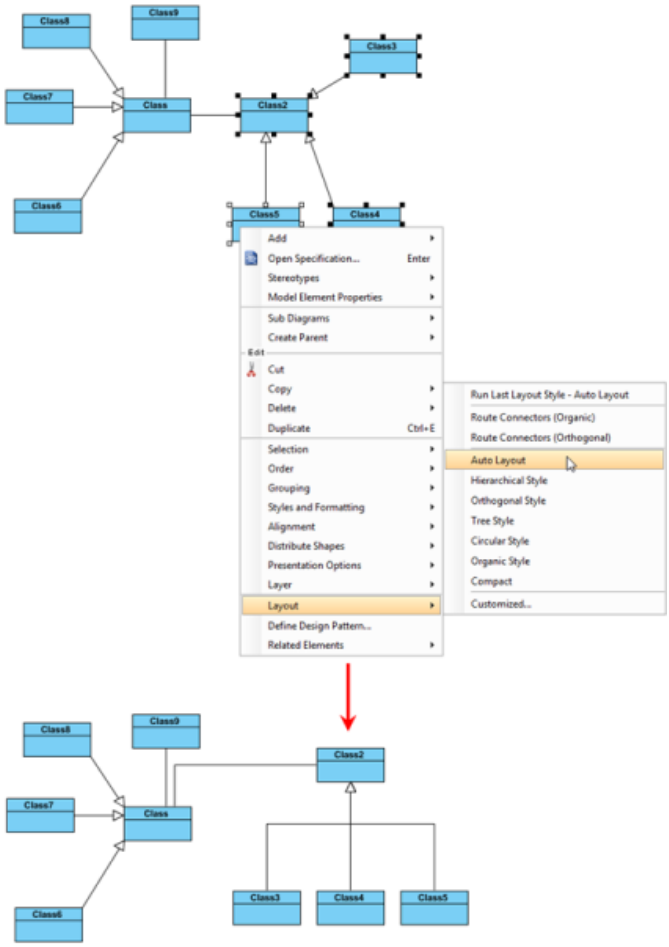


Organic Layout setting

Automatic layout selected shapes

To layout all the shapes in the diagram, right click on the diagram's background and select Layout from the pop-up menu.

To layout the selected shapes, right click a selected shape, select **Layout** and a layout option from the pop-up menu (when more than one diagram element is selected).



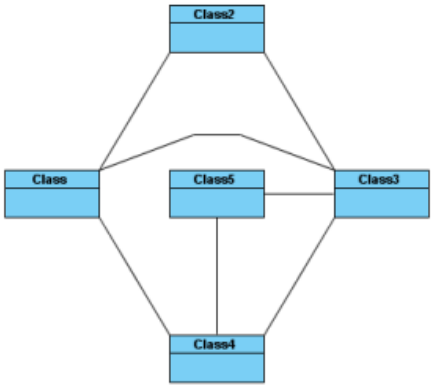
Perform layout with selected shapes

Automatic route connectors

There are 2 kinds of layouts which do not change the location of shapes, only change the connectors: **Organic Edge Route Layout** and **Orthogonal Edge Route Layout**

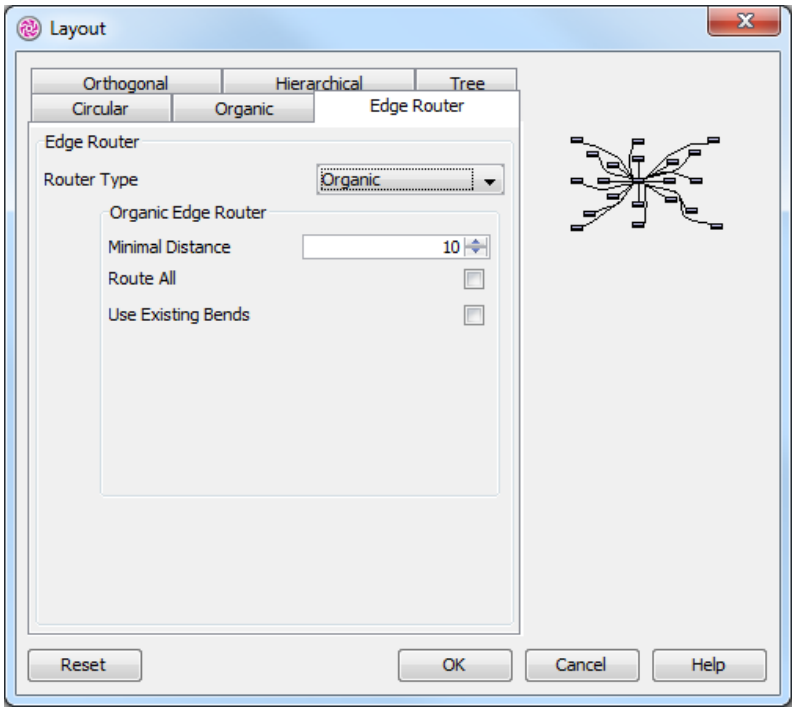
Organic edge route layout

Organic Edge Route Layout, which is one of the edge route layouts in Agilian, arranges the connectors without affecting the location of shapes. It can ensure that the shapes will not overlap and keep a specific minimal distance.



Organic Edge Route Layout

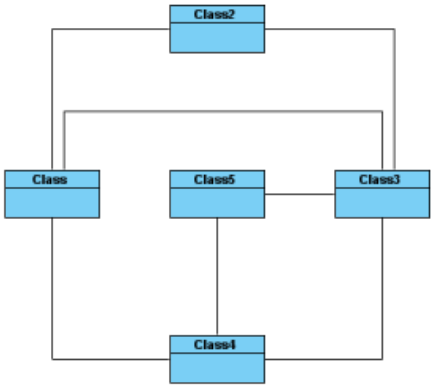
Minimal Distance: the minimal distance of the connectors.
Route All: whether all the connectors will be routed.
Use Existing Beans: whether using existing bends.



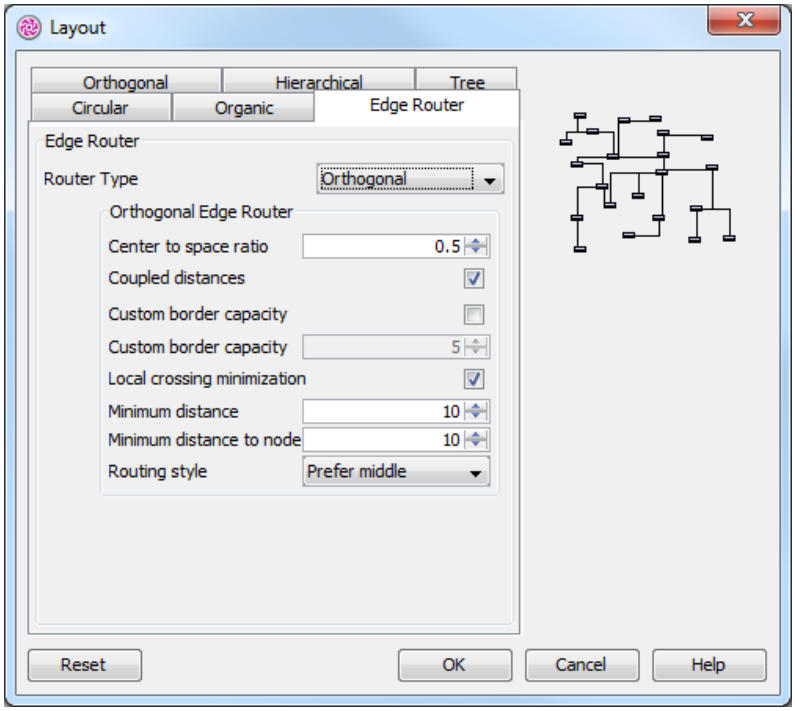
Organic Edge Route Layout setting

Orthogonal edge route layout

Route Connectors can arrange the connectors using vertical and horizontal line segments only. It is the best way for users to arrange the connectors that have complicated route.



Center to space ratio: the ratio of center to the distance between center and nodes.
Coupled distances: the distance between coupled nodes.
Crossing cost: the cost of crossing connectors.
Custom border capacity: the capacity of the border.
Local crossing minimization: whether the local crossing of connectors will be minimized.
Minimum distance: the minimum distance of connectors.
Minimum distance to node: the minimum distance between the shapes.
Rerouting: whether the connector that has many crossings will be rerouted.
Routing style: the style of routing.



Orthogonal Edge Route Layout setting

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

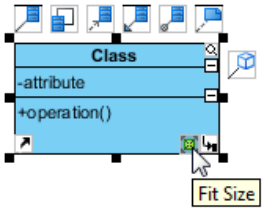
Fit shape size

In some cases, shapes are found oversized. For better presentation, you may need to resize them smaller. Fit size can help you to adjust shapes into the smallest size based on their content, such as the name of shape. This page will teach you how to fix the size of shapes manually and automatically.

NOTE: The size of shapes can be fixed automatically by right clicking on the diagram's background and checking **Diagram Content> Auto Fit Shapes Size**.

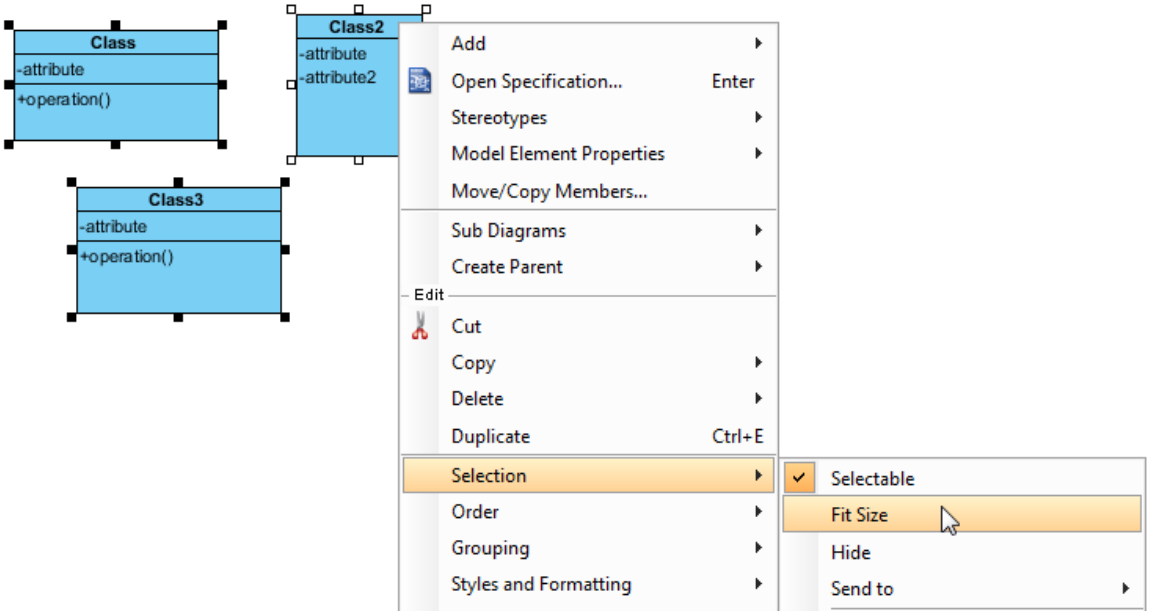
Fitting selected shapes size

To adjust a shape size, move the mouse on a shape and fit size resource centric interface will be shown. Click **Fit Size**resource icon at the bottom of the shape.



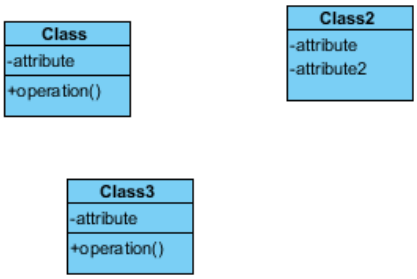
***Fit Size**resource icon*

To fit several shapes' size, select those shapes, right click on a selected shape and then select **Selection > Fit Size** from the pop-up menu.



Fit size for several shapes from the pop-up menu

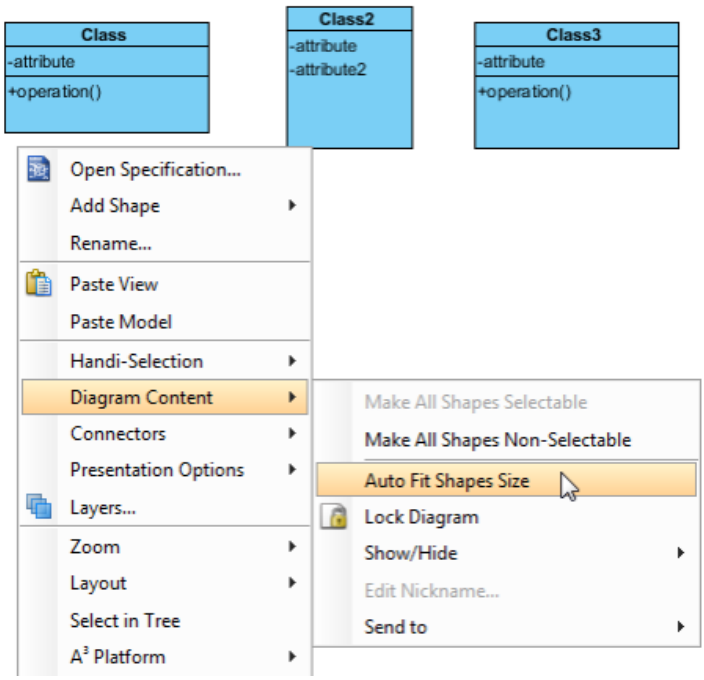
As a result, each shape will be adjusted to its fit size in accordance with its content, instead of fixing all selected shapes into the exactly same size.



Shapes are fitted size

Checking/unchecking automatic fit shape size mode

You can check/ uncheck the **Auto Fit Shapes Size** on diagram to make all the shapes on the diagram fitted size automatically. To do so, right click on the diagram's background, select **Diagram Content > Auto Fit Shapes Size** from the pop-up menu.



*Check **Auto Fit Shapes Size** from the pop-up menu*

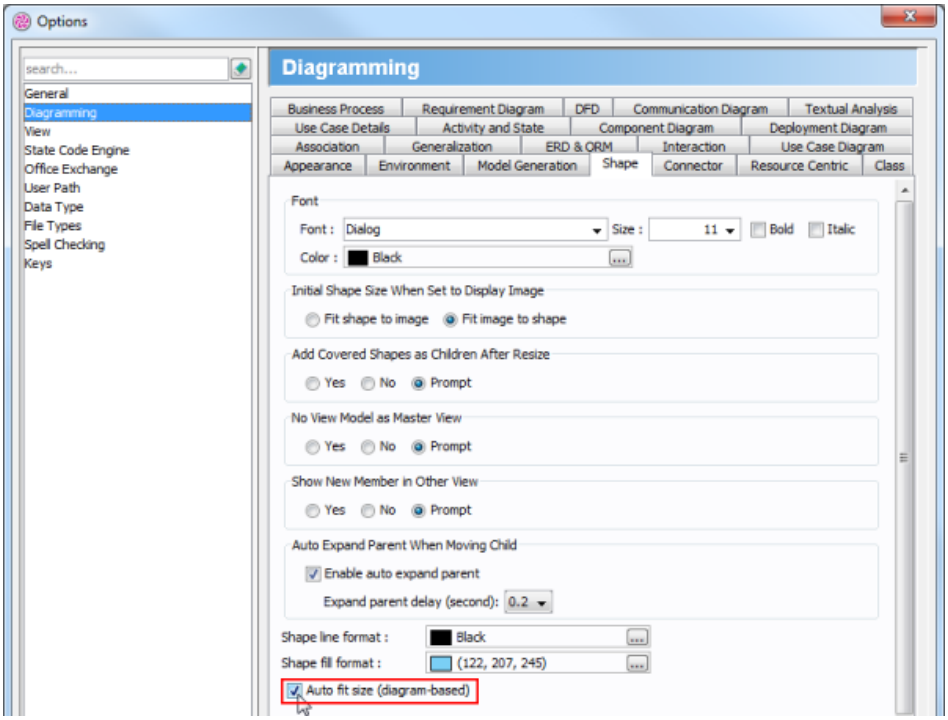
All the shapes are subsequently fitted size and they will become non-sizable.

If the content of shape is changed, the shape itself will be resized automatically.

Shape is resized automatically after new attribute added

You can also check **Auto Fit Size** for future usage.

1. Select **Tools > Project Options...** from the main menu to unfold **Project Options** dialog box.
2. In the **Options** dialog box, press the **Diagramming** category, select the **Shape** tab and check **Auto fit size (diagram-based)**.



*Check **Auto fit size (diagram-based)** in the **Options** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagram element selection

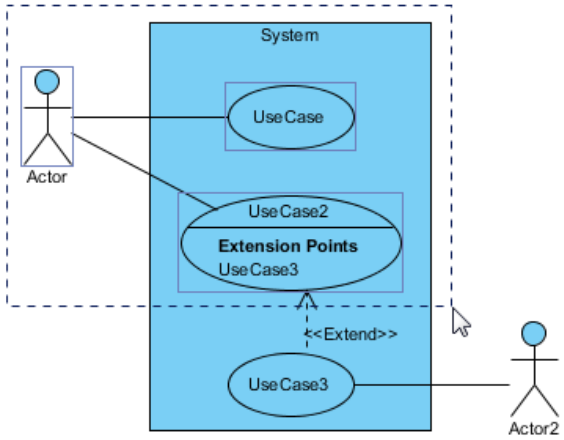
When a number of diagram elements need selecting simultaneously, diagram element selection supports this purpose. You can either click directly with hot keys or select a range of selection with the mouse. A specific type of diagram element(s) on a diagram can be selected as well.

Selecting multiple shapes

Multiple shapes can be selected by either selecting a range of shapes with the mouse on diagram or clicking shapes with pressing hot keys.

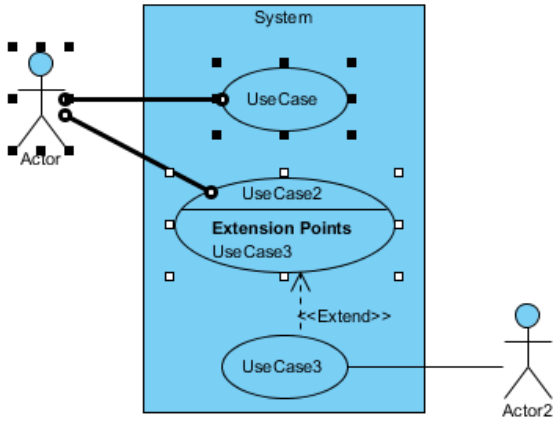
Selecting a range of shapes with the mouse

1. For selecting multiple shapes, drag them from corner to corner diagonally with the mouse. They will then be surrounded by a rectangle individually.



Select multiple shapes with the mouse

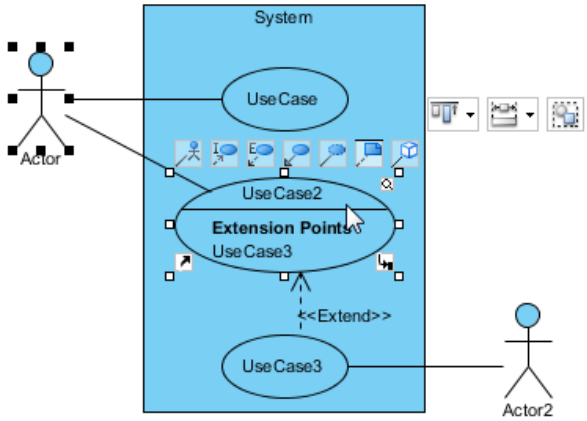
2. After releasing the mouse, those shapes will be selected.



Shapes are selected

Clicking with pressing ctrl/Shift key

Click a shape in advance and then click other shapes with pressing **Ctrl** or **Shift** key. As a result, those shapes will be selected.

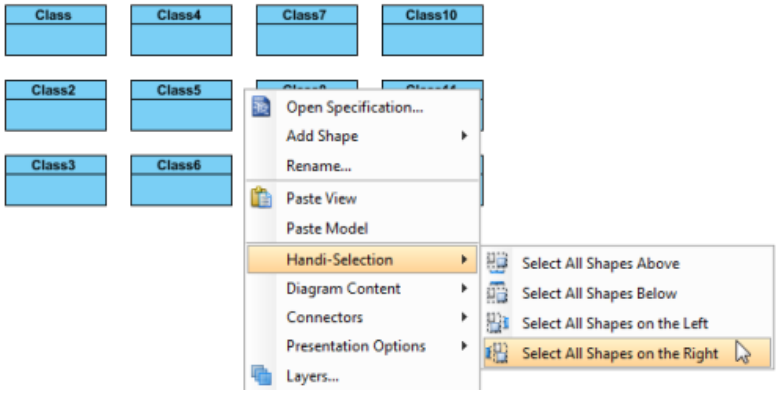


*Click shapes with pressing **Ctrl** or **Shift** key*

Handi-Selection

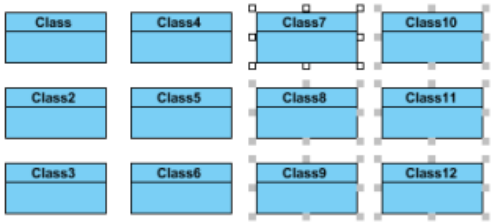
In some cases, the shapes are too complicated and more seriously, the whole diagram is extremely enormous that neither selecting a range of shapes with the mouse nor clicking shapes with pressing **Ctrl** or **Shift** key are the most suitable application. It is hard to drag the mouse on the large diagram, or is troublesome to click on many shapes. Using **Handi-Selection** is probably the best choice for you in this situation.

1. Right click on the diagram's background where is in the vicinity of those shapes you are going to select, select **Handi-Selection** and then select a scope for selecting shapes (i.e. above/ below/ left/ right) from the pop-up menu.



Select all shapes on the right from the pop-up menu

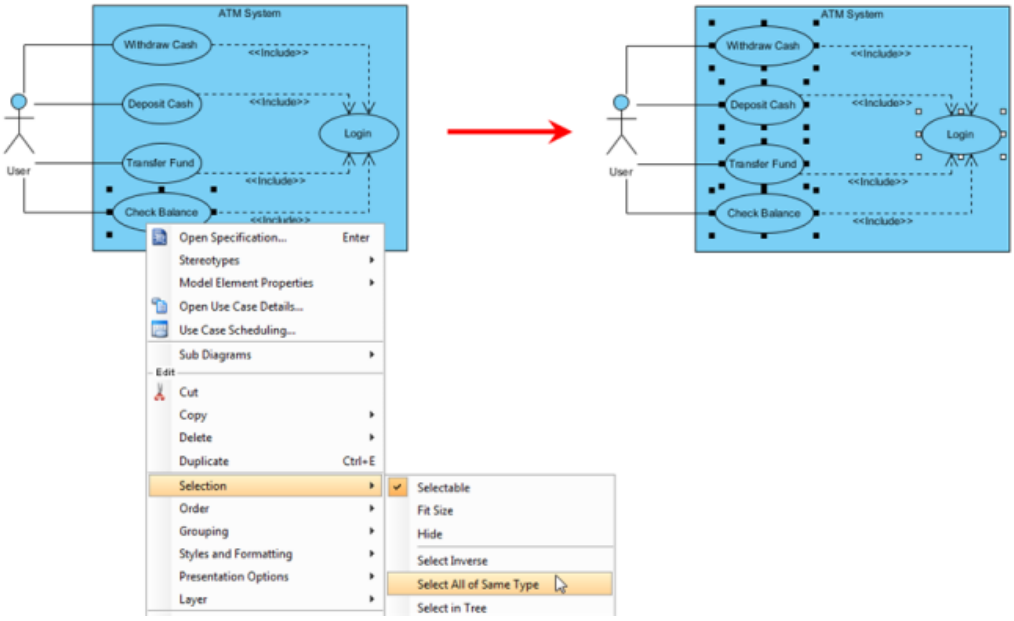
2. As a result, all the shapes of the particular scope will be selected.



All shapes on the right are selected

Selecting same type of shapes

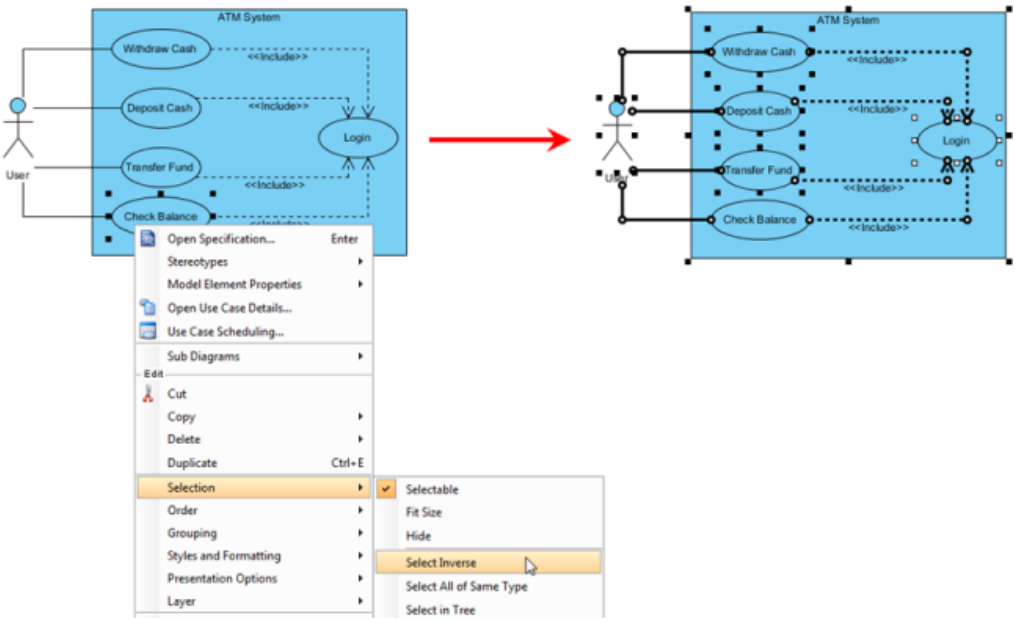
When you want to select a few shapes of same types on the diagram, right click on a shape and select **Selection > Select All of Same Type** from the pop-up menu. As a result, other shapes of same type as the shape you selected previously will be selected.



All shapes of same type are selected

Inverse selection

Shapes can be selected inversely. Right click on a shape that you don't want to be selected and select **Selection > Select Inverse** from the pop-up menu. As a result, all shapes will be selected except the shape you right clicked on previously.



Selection is inversed

Related Resources

The following resources may help you learn more about the topic discussed in this page.

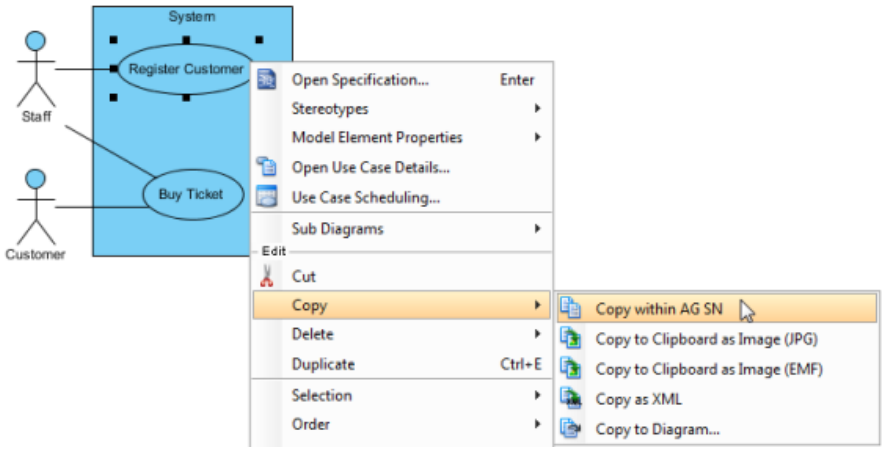
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Copy and paste

You can create a view of a model element by copying a view and pasting as view, while pasting as model creates a new model from the copied one.

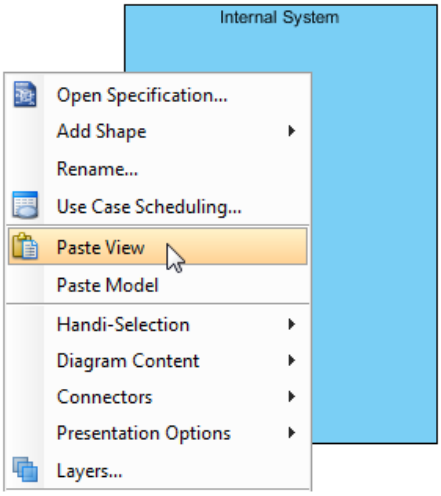
Copying within AG

1. Right click on the selected shape(s), select **Copy > Copy within AG** from the pop-up menu.



Copy selected shapes with AG SN

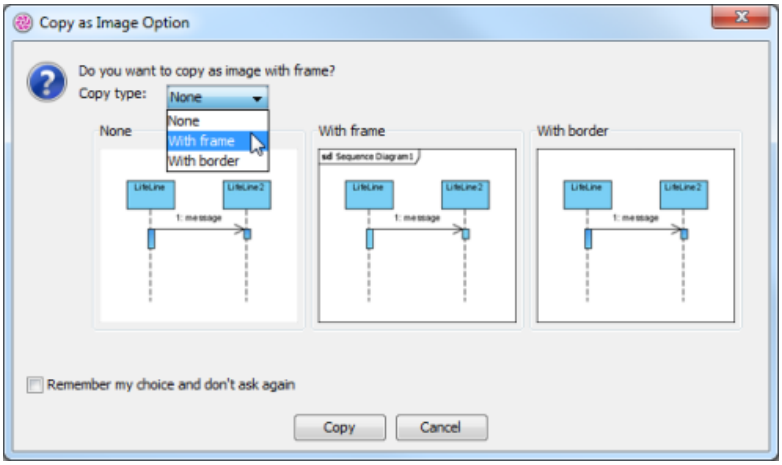
2. After you switch to the destination diagram, right click on the diagram background and select either **Paste View** or **Paste Model** from the pop-up menu. The feature of **Paste view** refers to present the same model element in another view under a new context. The shape is pasted without creating model; while the feature of **Paste Model** refers to duplicate the shape and present it in a new view. The model will be copied and pasted on the diagram. The new model will be named with appending a sequential number.



Paste the selected shape

Copying to clipboard as image (JPG)

1. Right click on the selected shape(s) and select **Copy > Copy to Clipboard as Image (JPG)** from the pop-up menu.
2. When the **Copy as Image Option** dialog box pops out, select an option from the drop-down menu of **Copy type**. Click **Copy** button to proceed.



*Select an option from the drop-down menu of **Copy type***

3. You should select a destination document (e.g. MS Word) for pasting the copied shape(s) to do further documentation. After you switch to the destination document, right click on the desired place and select **Paste** from the pop-up menu.
4. As a result, your selected shape(s) will be pasted on the destination document.

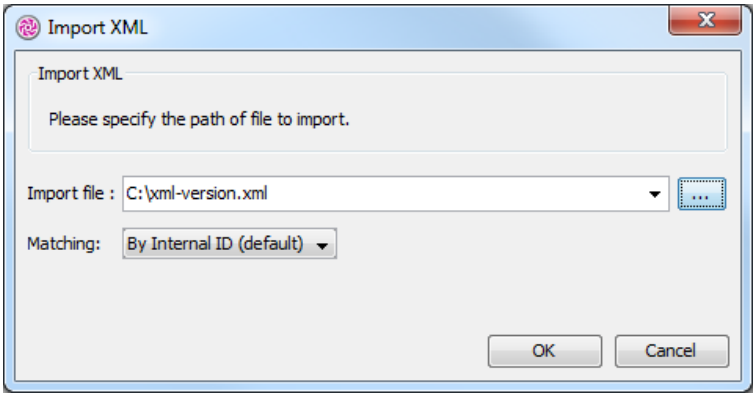
Copying to clipboard as image (EMF)

1. Right click on the selected shape(s) and select **Copy > Copy to Clipboard as Image (EMF)** from the pop-up menu.
2. When the **Copy as Image Option** dialog box pops out, select an option from the drop-down menu of **Copy type**. Click **Copy** button to proceed.
3. EMF is a kind of scalable image which can be pasted on a document for further documentation. After you switch to the destination document, right click on a desired place and select **Paste** from the pop-up menu.
4. As a result, your selected shape(s) will be pasted on the destination document.

Copying as XML

You can convert selected shapes into XML which contain the data of selected shapes in XML format. The XML data can then be imported into another project.

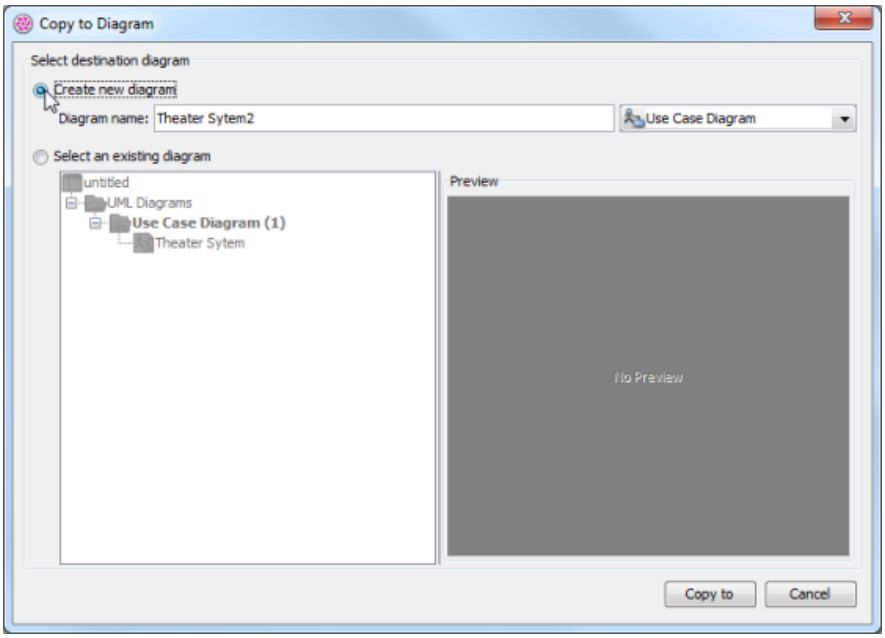
1. Right click on the selected shapes and select **Copy > Copy as XML** from the pop-up menu.
2. Open a text editor and create a new text file. Paste the XML there and save it as an XML file.
3. After that, the file can be imported to another project by selecting **File > Import > XML**.
4. When **Import XML** dialog box pops out, select the xml file path. Finally, click **OK** button to proceed.



Select the xml file path

Copying to Diagram

1. You can also copy the selected shape(s) to either a new diagram or an existing diagram. Right-click on the selected shapes, select **Copy > Copy to Diagram...** from the pop-up menu.
2. When the **Copy to Diagram** dialog box pops out, check **Create new diagram** if you want to paste to a new diagram or check **Select an existing diagram** if you want to paste to an existing diagram.



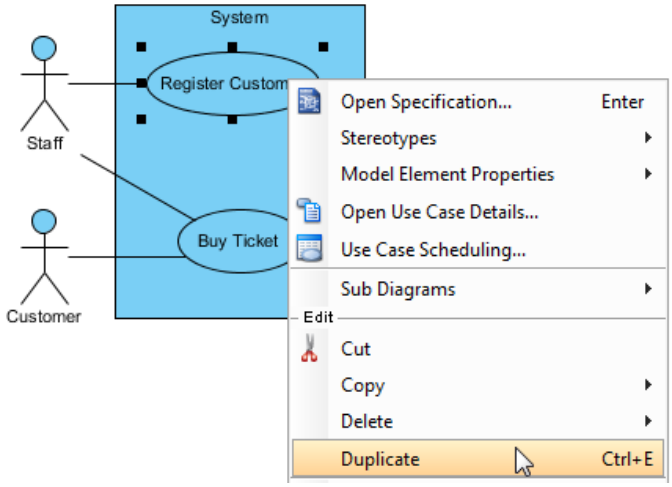
*Check **Create new diagram***

3. Click **Copy to** button. As a result, the selected shape(s) will be duplicated on the selected diagram.

Duplicating

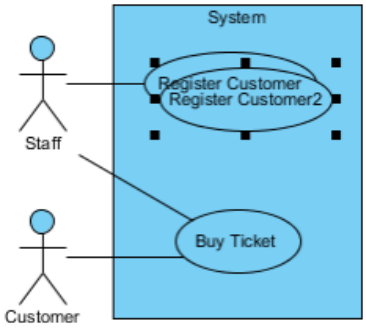
With the feature of duplicate, shapes can be duplicated on the same diagram instantly.

1. Right click on the selected shape(s) and select **Duplicate** from the pop-up menu.



Duplicate a selected shape

2. As a result, the selected shape(s) will be duplicated.



The shape is duplicated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

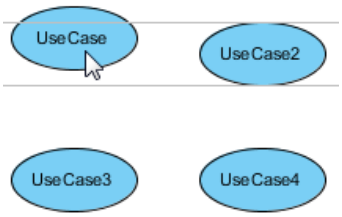
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Alignment guide

Alignment is the adjustment of an object in relation with other objects. Therefore, the diagram alignment means adjusting a shape's position with another's (or others') in a straight line or in parallel lines. When you drag shapes, alignment guide will appear to let you position the dragging content in accordance with existing shape(s).

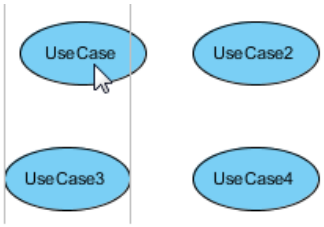
Showing edges alignment guide

When you drag a shape upward or downward, two horizontal lines will reveal on the both edge of the shape as an offer of assistance for adjusting the shape's position.



Drag upward

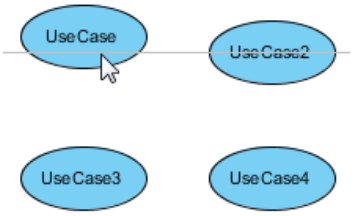
Similarly, when you drag a shape to the left or the right, two vertical lines will reveal on the both edge of the shape.



Drag to the right

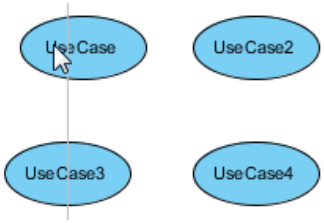
Showing center alignment guide

When you drag a shape upward or downward, a horizontal line will reveal in the center of the shape as an offer of assistance for adjusting the shape's position.



Drag upward

Similarly, when you drag a shape to the left or the right, a vertical line will reveal in the center of the shape.

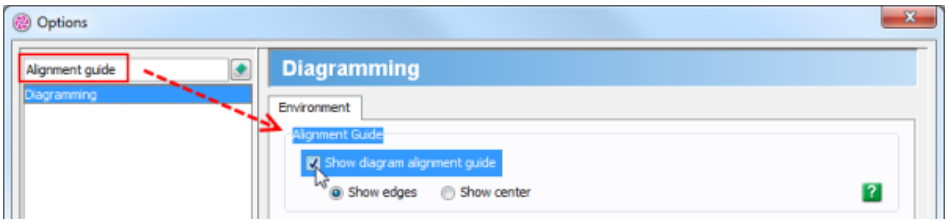


Drag to the right

Changing alignment guide options

If you want the alignment guide to help you to position when you drag shapes, check **Show diagram alignment guide**; on the contrary, uncheck it if you don't need a help. Two Choices are provided in alignment guide: **Show edges** and **Show center**. Choose **Show edges** if you want two horizontal lines to reveal on the both edge of the moving shape while choose **Show center** if you want a horizontal line to reveal in the center of the shape.

1. Select **Tool > Application Options...** from the main menu to unfold **Options** dialog box.
2. In shortcut, typing *Alignment guide* on top-left text field for searching. It will be shown that the option of **alignment guide** is on **Diagramming** category, **Environment** tab shortly. As you see, the option will be highlighted for your convenience.



*Change **alignment guide** options*

3. The option for **alignment guide** is checked as default. The next step you need to do is to choose either **Show edges** or **Show center** for your diagrams.
4. Uncheck **Show diagram alignment guide** if you do not want the assistance of alignment guide when dragging shapes.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

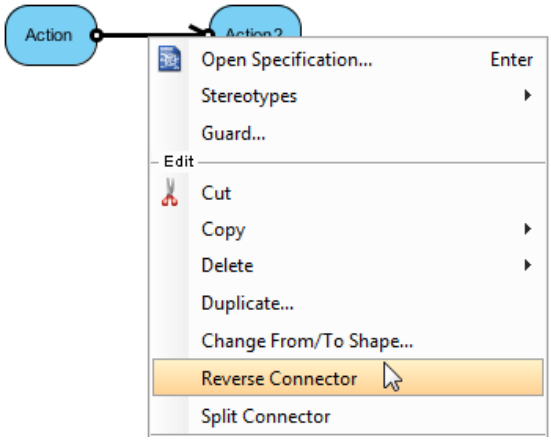
Reverse connector direction

The flow between shapes is represented by connectors, for example, a sequence message between two lifelines of *Student* and *StudentController* which represents the call from *Student* to *StudentController*. If the flow is created mistakenly, or the flows need reverting due to an updated data, the flows can be fixed by reverting connectors.

The function of reverse connector is not only for reverting the connector's direction, but also for repositioning the information contained by the end of connection. For connectors like association, each end contains specific information like multiplicity, role name, visibility, etc. Reverting connector will also swap the information.

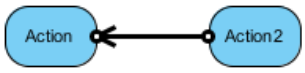
Reverse connector direction

1. Right click on your target connector and select **Reverse Connector** from the pop-up menu.



*Select **Reverse Connector** from the pop-up menu*

2. As a result, the flow of connector is reversed.



Connector is reversed

> **NOTE:** Connectors, such as message in sequence diagram, are not reversible.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

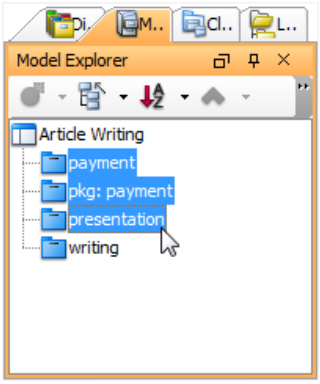
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Visualize model elements on diagram

Model elements are fundamental parts of a project. While a model element can be shown on more than one diagram, a diagram can also show a model element more than once. They, which can be found in **Model Explorer** or **Class Repository**, contain metadata and can be shown in diagram for visualizing the data. For example, a stick figure for an actor model element.

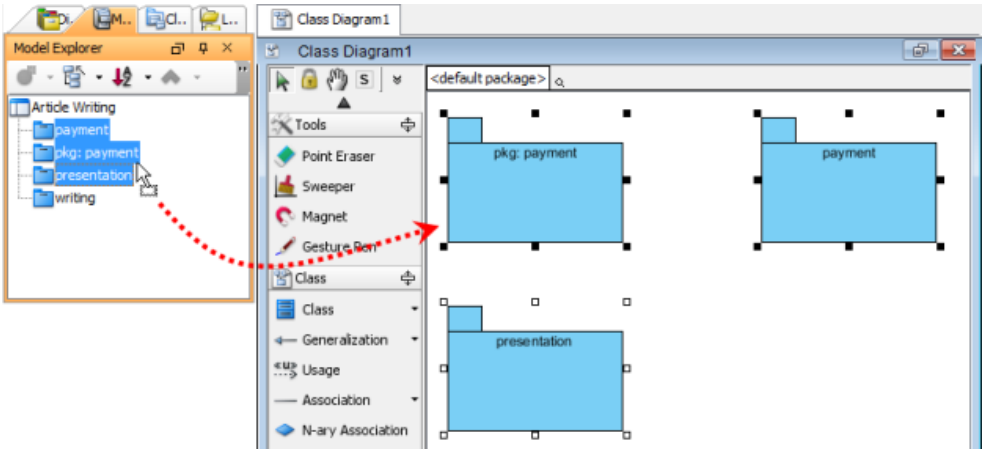
To visualize a model element or several model elements:

1. Select a model element (or as many as model elements you preferred) from **Model Explorer**.



*Select several model elements from **Model Explorer***

2. Hold on the mouse and drag it on the diagram pane. As a result, the views of the selected model element(s) will be shown on diagram.



Drag the model elements on diagram

Related Resources

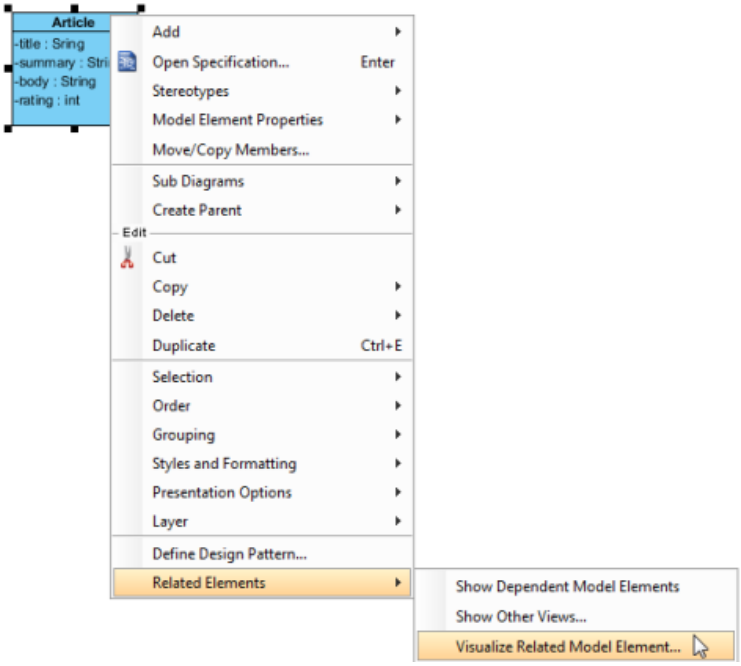
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Visualize related model elements

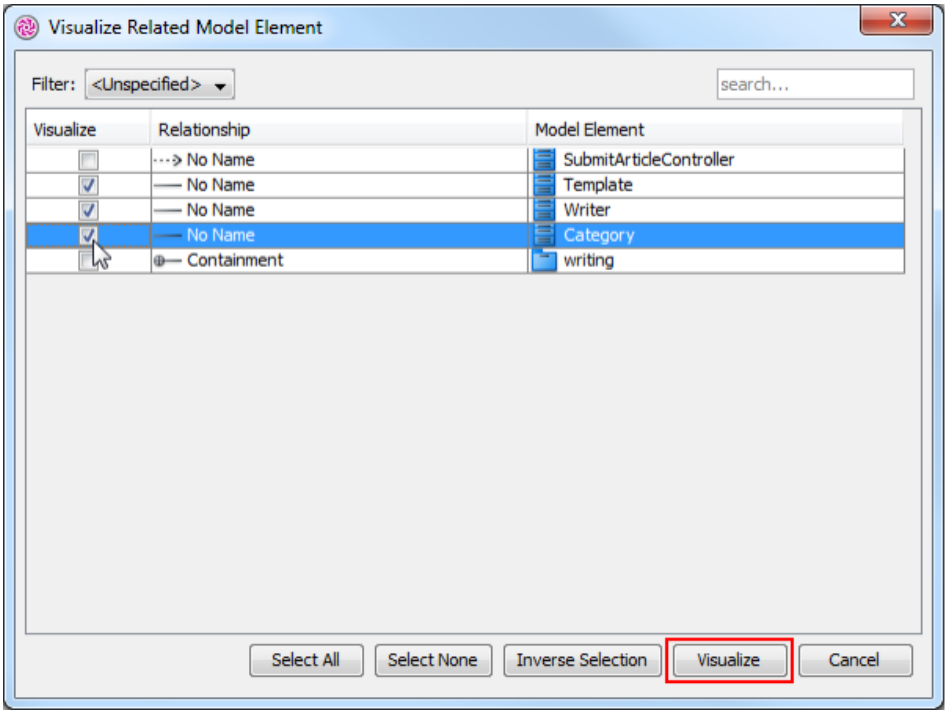
Sometimes, several model elements that related to the current model element(s) are hidden for various reasons. These related model elements, in fact, can be revealed through the feature of visualize related model element. With this feature, the relationship between model elements can be viewed thoroughly.

1. Right click on a model element and select **Related Elements > Visualize Related Model Element...** from the pop-up menu.



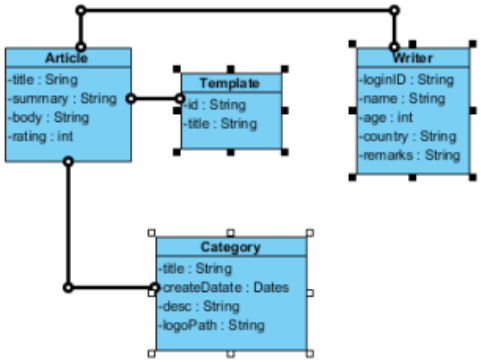
*Select **Visualize Related Model Element...***

2. When **Visualize Related Model Element** dialog box, check the related element(s) you want to be shown with the corresponding relationship on the diagram in **Visualize**. Click **Visualize** button to proceed.



***Visualize Related Model Element** dialog box*

3. As a result, the related models with connectors are shown on the diagram.



Related model elements are visualized

Related Resources

The following resources may help you learn more about the topic discussed in this page.

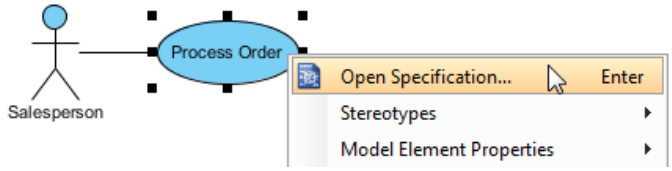
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Adding comments

Agilian supports comments on model elements. Since comments are usually used to record the progress and status of model elements, they are regarded as a textual annotation for model elements.

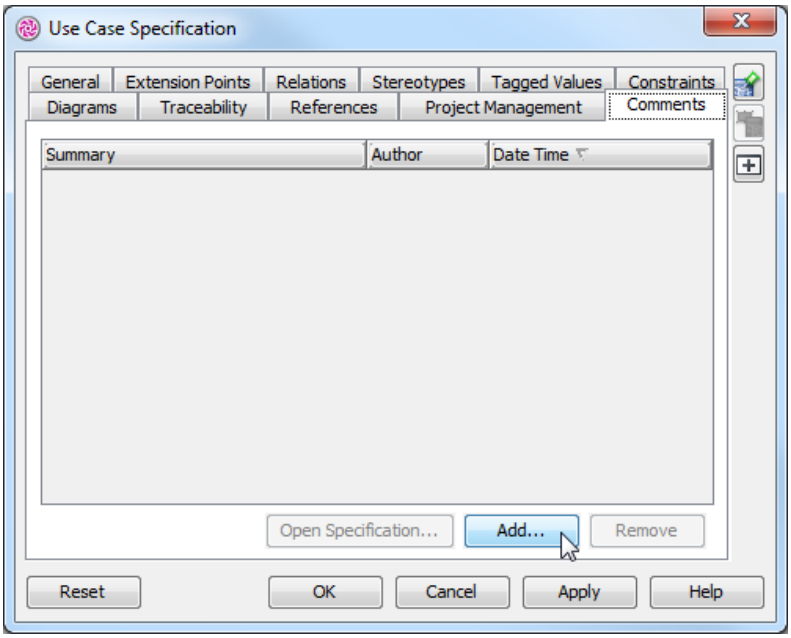
Adding comment to model element

1. For adding comment in a particular shape, right click on the shape and select **Open Specification...** from the pop-up menu.



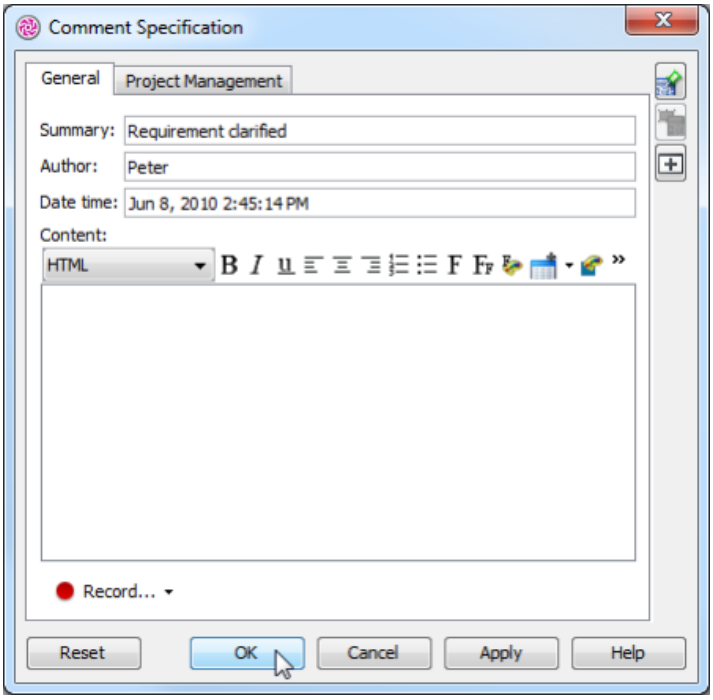
Open specification

2. In **Specification** dialog box, select **Comments** tab and click **Add...** button to create a comment.



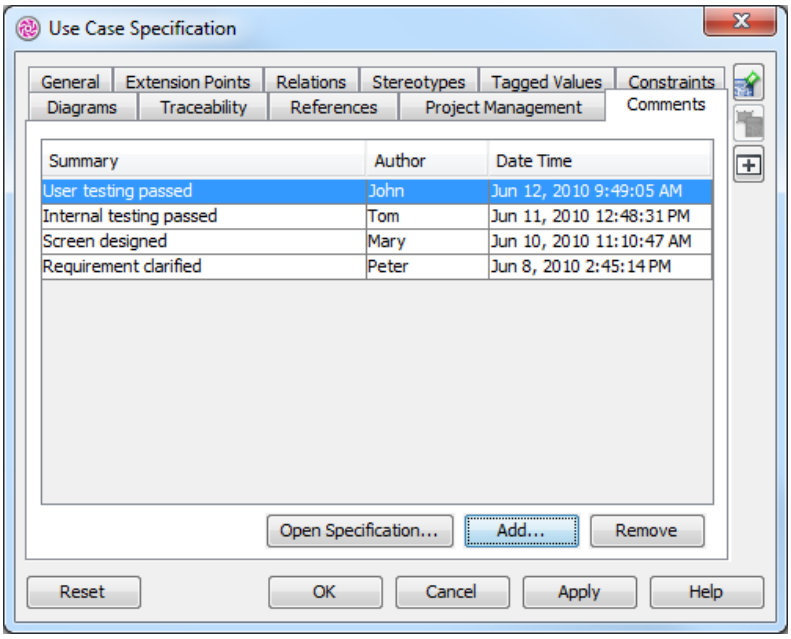
Add comment

3. In **Comment Specification** dialog box, enter summary, author and date time respectively. Select **OK** button to confirm editing.



*Enter information in **Comment Specification** dialog box*

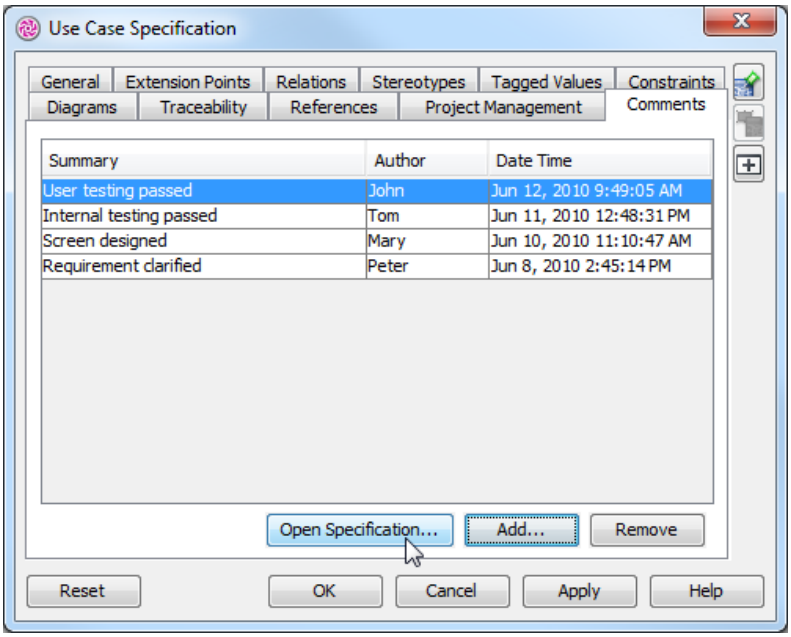
4. As a result, the comment you entered previously is shown on **Specification** dialog box.



Comment is shown

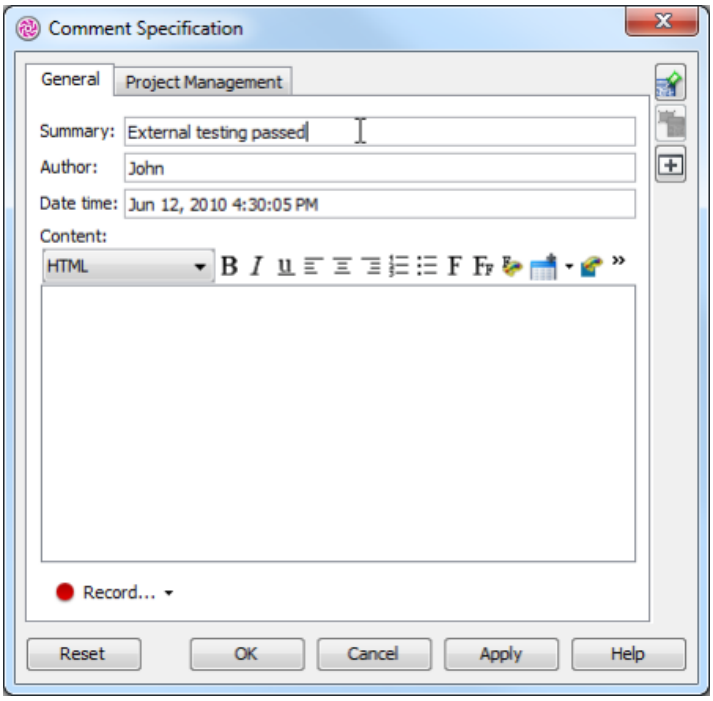
Managing comment of model element

1. To modify comment in a particular shape, right click on the shape and select **Open Specification...** from the pop-up menu.
2. In **Specification** dialog box, click **Comments** tab, select a specified summary comment and click **Open Specification...** button to proceed.



Open specification

3. In **Comment Specification** dialog box, modify information and select **OK** to confirm changing.



Modify information

Related Resources

The following resources may help you learn more about the topic discussed in this page.

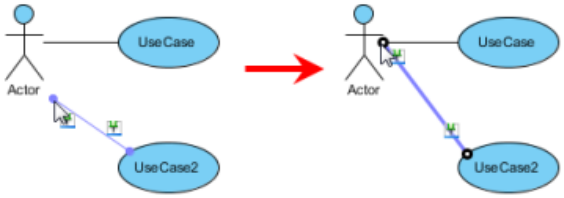
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Pinning connector ends

Connectors can be either pinned temporarily or pinned permanently. Connector ends help you to point out a specific position of shapes with pin.

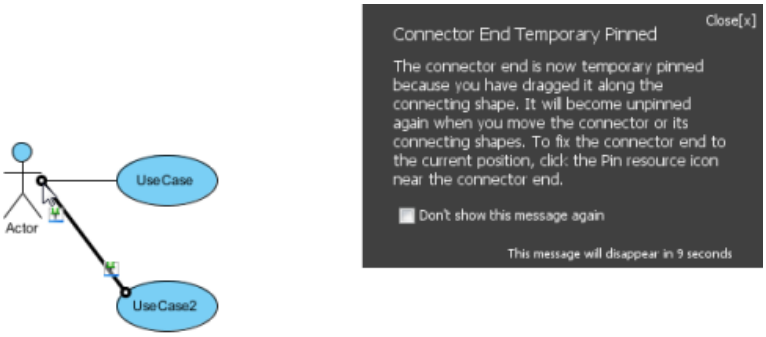
Adjusting connector ends temporarily

1. Connectors can be joined at same point of a shape on the diagram. To do so, drag one end of a connector to the shape.



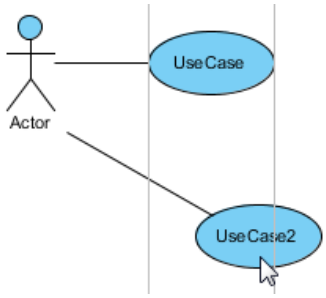
Drag one end of a connector to a shape

2. The connector is temporarily pinned. A dialog box will be shown on the top right corner of diagram to instruct you how to pin the connector.



Connector is temporarily pinned

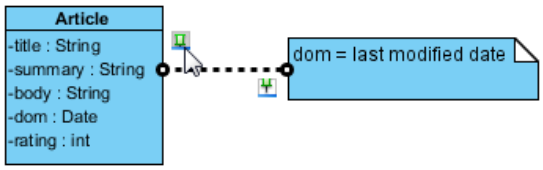
3. Since the connector that links from shape and to shape together is temporarily pinned, either from shape or to shape is moved, the connector between them will be unpinned.



Connector is unpinned

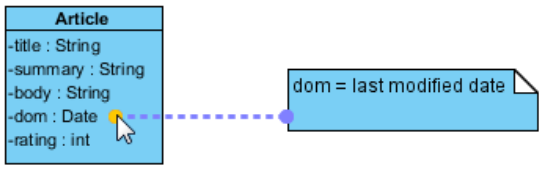
Pinning connector ends

1. Move the mouse over a connector and press its resource icon **Pin**.



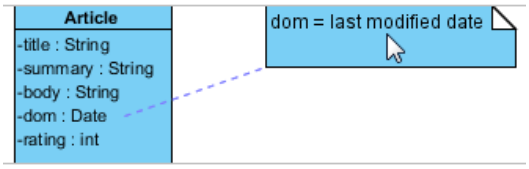
*Press resource icon **Pin***

2. Drag one end of connector to point out a specific position. Note that no dialog box of temporary pin will be shown this time.



Drag one end of connector to point out a specific position

3. Moving either from shape or to shape will not unpin the connector.



Connector is still pinned

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Align and distribute diagram elements

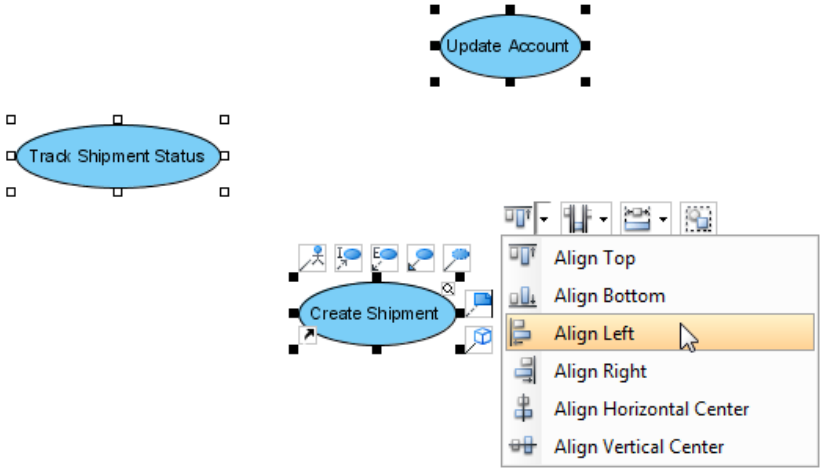
Agilian supports two types of positioning features: alignment and distribution which allow the positioning of selected shapes in accordance with the alignment/ distribution option through grouping resource icons. Alignment refers to the edges and the centers of selected shapes are aligned to each other while distribution refers to selected shapes are distributed in same direction based on their centers or edges.

Aligning diagram elements

Select a few model elements with the mouse on the diagram pane before executing alignment.

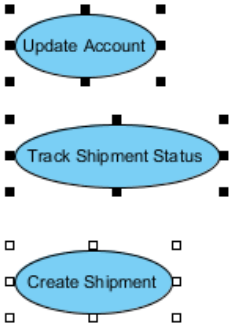
Using grouping resource icons

1. When move the mouse over one of the selected shapes, grouping resource icons will be shown.
2. Select an alignment option from the drop-down menu of **Align Top** on the grouping resource icons.



*Select **Align Left** through resource icon*

3. As a result, all selected shapes are aligned in accordance with the last selected shape.



Shapes are aligned

4. For turning the non-selected shape into the last selected shape, click a shape with pressing **Ctrl** key.

Setting model elements same width and height

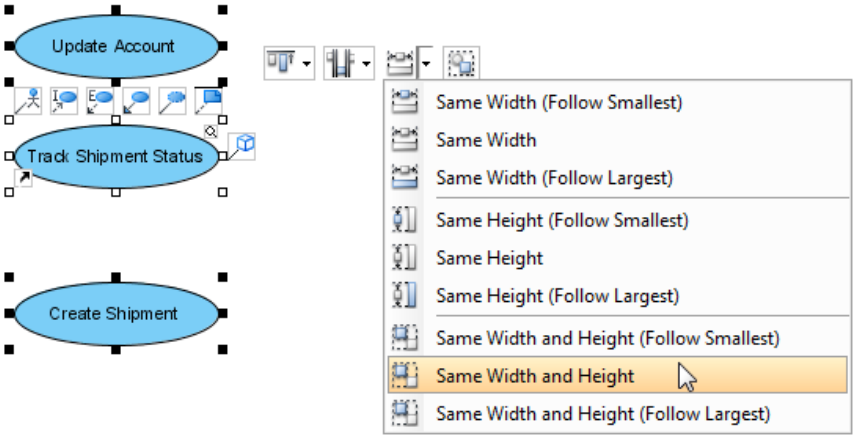
Besides aligning the shapes, shapes can also be resized through grouping resource icons or **Align Shapes Dialog**.

through main menu

Select **Edit > Align Shapes** and an alignment option from the main menu.

Using grouping resources

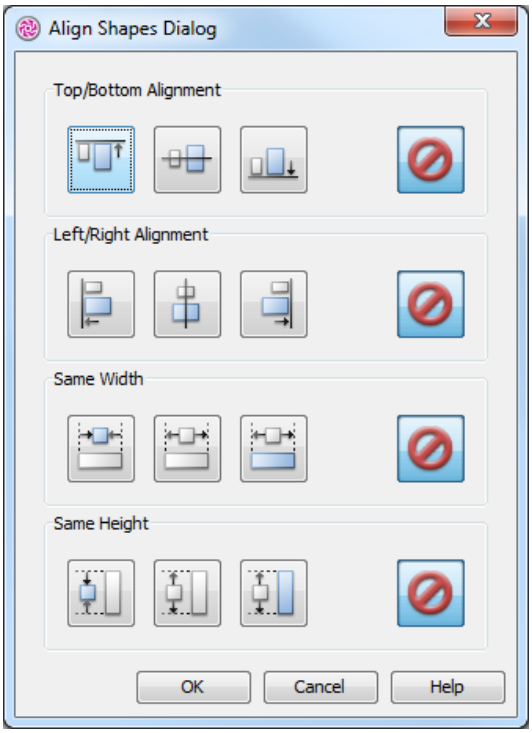
Select **Same Width, Same Height** and **Same Width and Height** from the drop-down menu of **Same Width** on grouping resource icons after select a few shapes on the diagram pane.



Resize through grouping resources

Using align shapes dialog

After select a few shapes on the diagram pane, select **Edit > Align Shapes > Align Shapes...** from the main menu to open **Align Shapes Dialog**. You can select an option by clicking the option button directly.



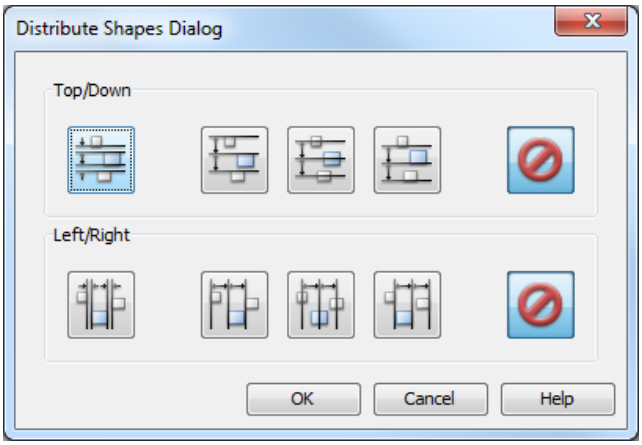
Align Shapes Dialog

Distributing diagram elements

In addition, model elements can be distributed in various directions through **Distribute Shapes Dialog**, the main menu or grouping resource icons.

through Distribute Shapes Dialog

After select a few shapes on the diagram pane, select **Edit > Distribute Shapes > Distribute Shapes...** from the main menu to unfold **Distribute Shapes Dialog**. You can select an option by clicking the option button directly.



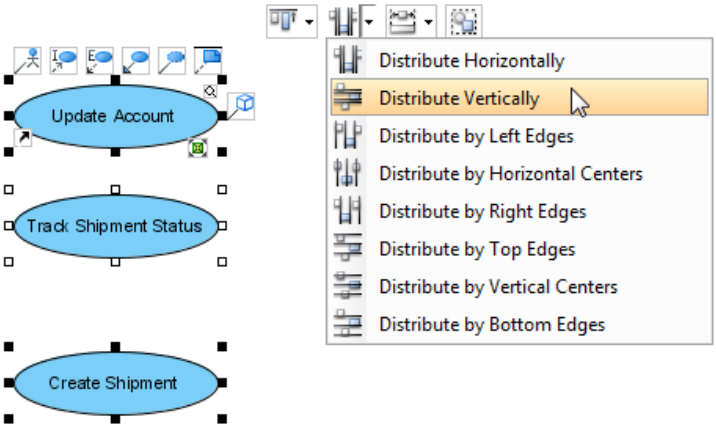
Distribute Shapes Dialog

through main menu

Select **Edit > Distribute Shapes** and then select an distribution option from the main menu after select a few shapes on the diagram pane.

Using grouping resource icons

Select a distribution option from the drop-down menu of **Distribute Horizontally** on grouping resource icons after select a few shapes on the diagram pane.



Distribute shapes on grouping resource icons

Related Resources

The following resources may help you learn more about the topic discussed in this page.

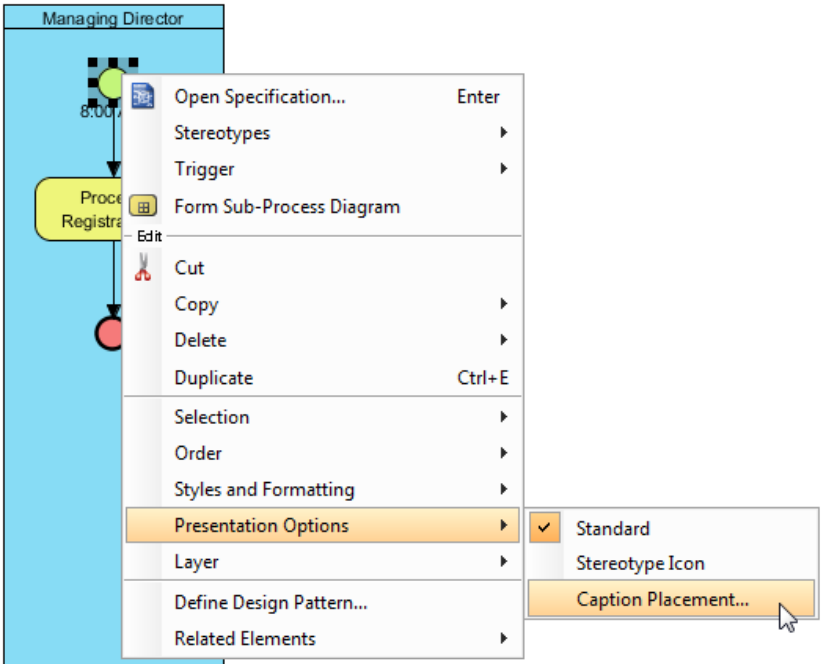
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Adjusting caption's position and angle

For shapes like event and gateway, their names are put outside and below the shape, which may overlap with outgoing sequence or message flow, making the name hard to read. To solve this problem, you can choose to place the caption elsewhere. Furthermore, you can rotate the caption to make it easier to read in print out.

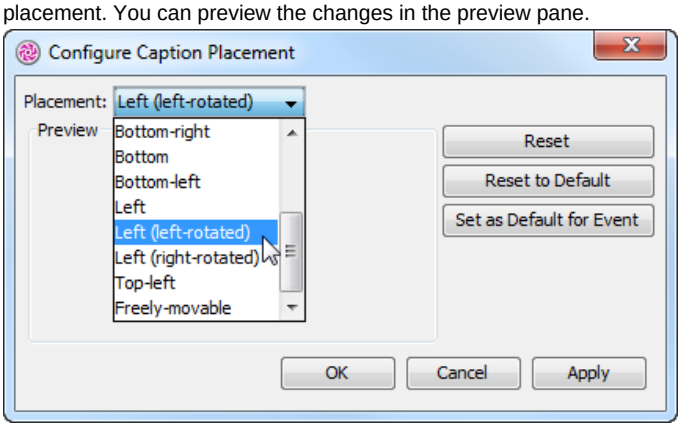
To set the position and angle of start, intermediate or end event, or gateway:

1. Right click on the shape and select **Presentation Options > Caption Placement...** from the popup menu.



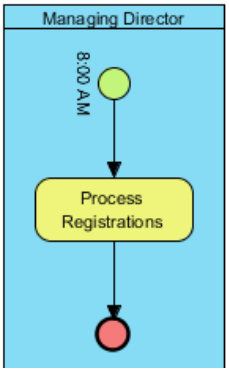
To change caption placement

2. Choose the placement, which is the position of caption. For some of the placement options, you can choose additionally the rotation of placement. You can preview the changes in the preview pane.



To choose a placement option

3. Click **OK** to confirm.



Caption position updated

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Zooming Diagram

An active diagram can be viewed in Diagram Overview automatically. If the diagram on diagram pane isn't clear enough, you can zoom in the diagram with your desired size through Diagram Overview .

1. Open Diagram Overview.

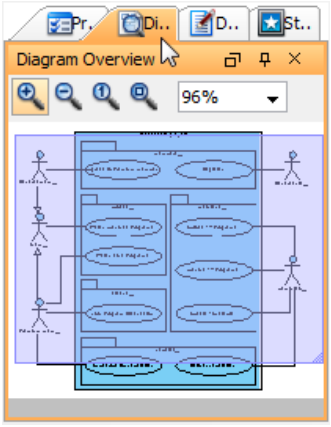
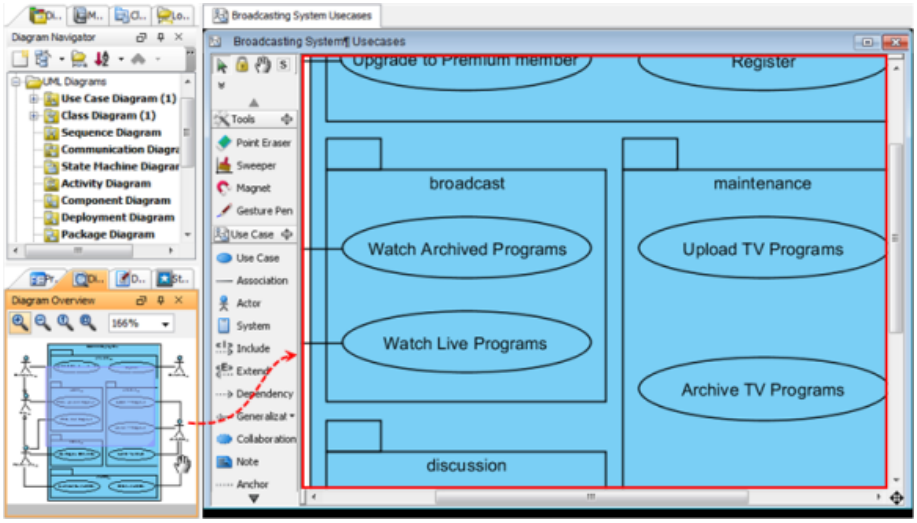


Diagram Overview

2. Drag the bottom right corner of the right purple box to zoom in/out the diagram. Alternatively, click on the buttons in toolbar or input the zoom ratio into the combo box to adjust the zoom ratio.



Roll up the mouse wheel

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagram grids

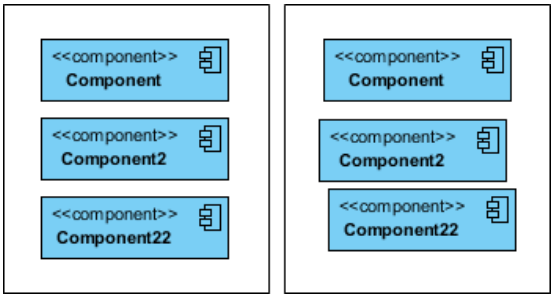
Diagram grids provide you with a precise visual cue to identify the position and boundary of shapes in diagram editors. You can make use of diagram grid to improve the positioning and alignment of shapes, which makes the diagram looks more neat, tidy and impressive. By default, the grid lines are not visible. In this article, we will see how to show them, and how to configure the various grid settings. We will use UML component diagram to explain the ideas but in practice, you can apply diagram grids on any diagram types - UML diagrams, BPMN diagrams, DFD, ERD etc.

Visibility of grid lines

Grid lines are not visible on diagrams by default. You can optionally show them by updating the grid settings. Here are the steps:

1. Right click on the diagram where you want to show/hide grid lines.
2. Select **Open Specification...** from the popup menu.
3. Open the **Grid Setting** tab.
4. Check/uncheck **Grid visible** and click **OK**.

Snap to grid

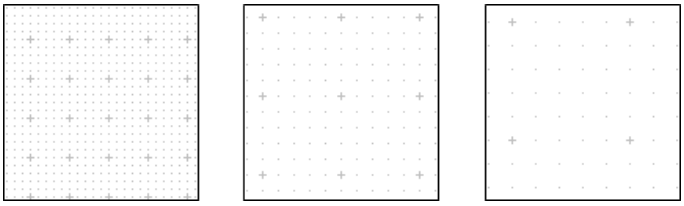


Effect of snap-to-grid on and off (left and right hand side)

The snap to grid function enables shapes to be positioned in an organized and well-aligned manner. When you draw, resize, or move a shape on a diagram, it will align to the nearest grid line (even when grid lines are not visible). This means you can make multiple shapes apply the same horizontal and/or vertical position, making the diagram looks more tidy. Note that the snap-to-grid option is turned on by default. If you want to change the setting:

1. Right click on the diagram where you want to enable/disable snap-to-grid.
2. Select **Open Specification...** from the popup menu.
3. Open the **Grid Setting** tab.
4. Check/uncheck **Snap to grid** and click **OK**.

Grid size

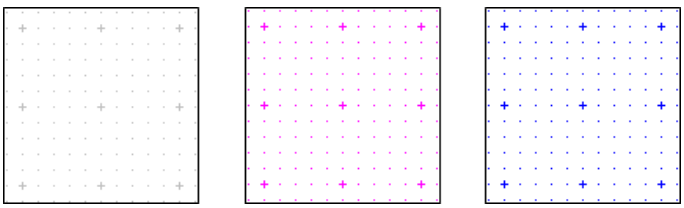


Different grid sizes - 5x5, 10x10 (default setting), 15x15

While the default spacing between grid line is set to 10 units, you can adjust the value to fulfill your diagramming needs. Please be reminded that when snap to grid is on, the shape you draw, resize or move will automatically align to the nearest grid line. This means that the value you input may influence the positioning of shapes. To update the grid size:

1. Right click on the diagram where you want to change the grid size.
2. Select **Open Specification...** from the popup menu.
3. Open the **Grid Setting** tab.
4. Edit the **Width** and **Height** of **Grid size** and click **OK**.

Grid color



Different grid colors - light gray (default setting), magenta, blue

Grid lines are not visible on diagrams by default. If you set them visible, you can see the light gray lines on diagram. If you want to change the color of grid lines, take the follow steps:

1. Right click on the diagram where you want to change the grid size.

2. Select **Open Specification...** from the popup menu.
3. Open the **Grid Setting** tab.
4. Pick up a **Grid color** and click **OK**.

Related Articles

- What to do If You Cannot Resize Shapes in a Diagram

Related Links

- Diagramming features of Agilian

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Advanced modeling techniques

This chapter covers less frequently used or comparatively complex modeling techniques.

Sweeper and magnet

Sweeper and magnet are handy tools for moving a group of shapes back and forth.

Mouse gestures

You can create shapes, connect shapes, or perform certain operations through pressing and dragging your right mouse button. You can gain more information in the *Mouse gestures* page.

Jumping to shape

When you are looking for a shape, a diagram or a model element, you can make use of the jump to feature to enter its name and jump to it immediately. It's like a commonly-known search function, but a faster approach.

Grouping diagram elements

You can more and format shapes by grouping them together. You will see how to group diagram elements on a diagram.

Show/hide diagram elements

You can optionally hide away some of the diagram elements on a diagram, or hide specific type of elements.

Layer

Layer provides a logical shape division in diagram. For example, a comment layer for annotation shapes. You can hide, lock and set active to a layer.

Making shape non-selectable

You can make shape non-selectable to avoid accidental movements for particular shapes. This is particular helpful when trying to move shapes in a container like package, without moving the package by mistake.

Showing model element in multiple diagrams

A model element can have multiple views. In this page you can see how to make use of drag and drop to create multiple views for a model element.

Using overview diagram

Overview diagram is best used to illustrate the relationship between diagrams, hence the content (e.g. interaction) they represent.

Changing Model Element Type

Shows you how to convert the type of a model element

Sweeper and Magnet

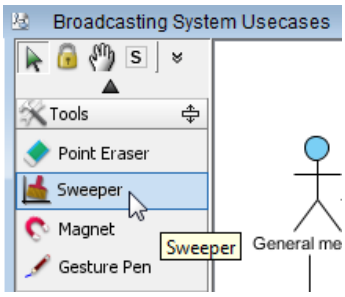
Modifying your diagrams from time to time is no longer a drag once you utilize sweeper and magnet. They can help you to modify your diagrams easily. Moreover, with them, you can move diagram elements easily without worrying too much about the layout. Sweeper can help you to expand more space between the diagram elements while magnet can help you to reduce the space.

Sweeper

The sweeper is one of the useful features for editing your diagrams. If you have ever experienced of moving the diagram elements without any tools, you probably understand how hard it is to manage the space between the diagram elements. Using sweeper to expand the space between the diagram elements allows you to move your diagram elements conveniently.

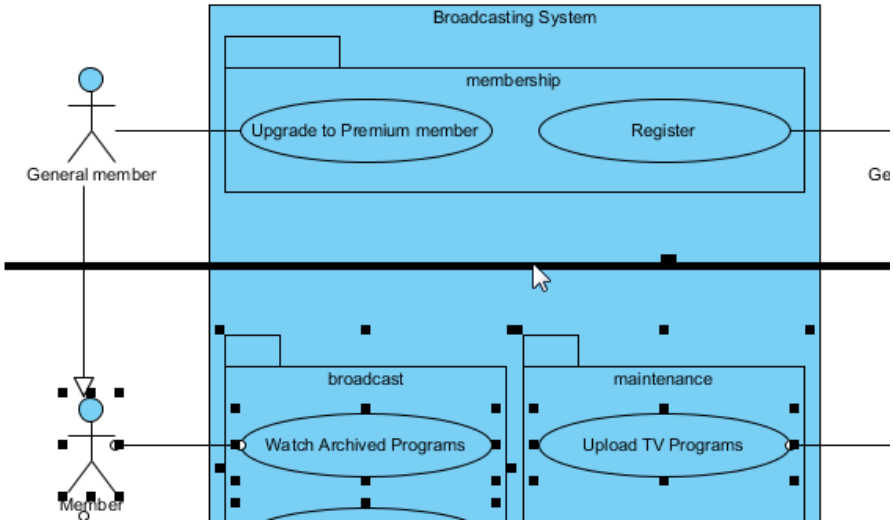
You can move the diagram elements by following the simple steps below:

1. Click **Sweeper** button from the diagram toolbar.



*Click **Sweeper** on diagram toolbar*

2. Move the mouse on the diagram pane where you would like to move diagram elements.
3. Hold onto your mouse and move the line toward your preferred direction to expand space for those diagram elements.

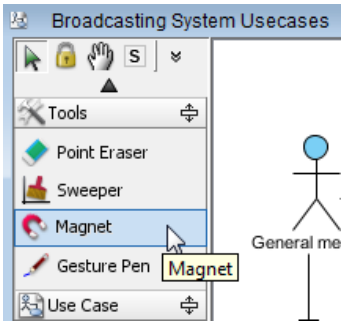


Moving the diagram element downward

Magnet

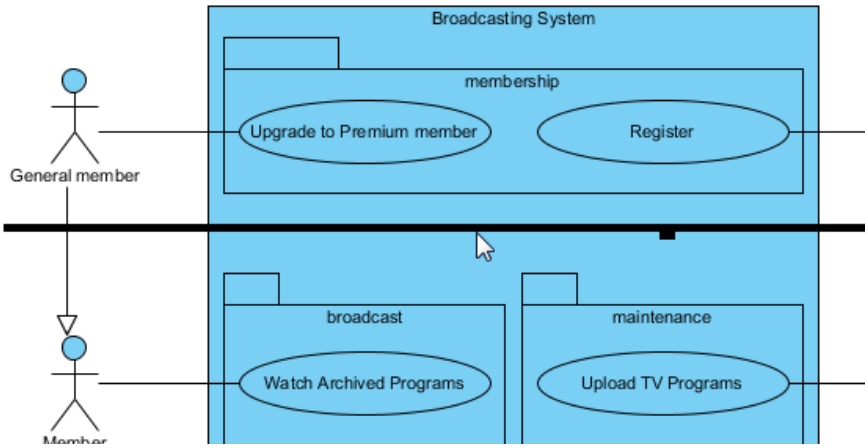
The magnet is another convenient feature for you to move your diagram elements. If you want to move a few diagram elements, you should try magnet. Its function is to reduce the space between the diagram elements and make your diagrams much tidier for printing. The steps of applying magnet on your diagram are shown as follows:

1. Click **Magnet** button from the diagram toolbar.



*Click **Magnet** on diagram toolbar*

2. Move the mouse on the diagram pane where you would like to move diagram elements.
3. Hold onto your mouse and move the line toward your preferred direction to reduce space for those diagram elements.



Moving the diagram element upward

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Mouse gestures

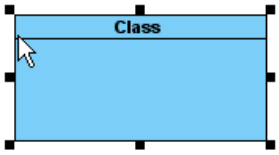
A variety of shapes and model elements can be created by sketching a path directly on the diagram pane with dragging the right mouse button to form a gesture. For your convenience and quick creation, mouse gestures allow you to execute common commands and create models within all diagrams.

Drawing shapes

1. To start using a mouse gesture, press the right mouse button and drag it until finish drawing a shape.

Draw a clockwise rectangle

2. When the shape is done, release the mouse. After the shape is created, the action description will be shown on top right corner of the diagram.



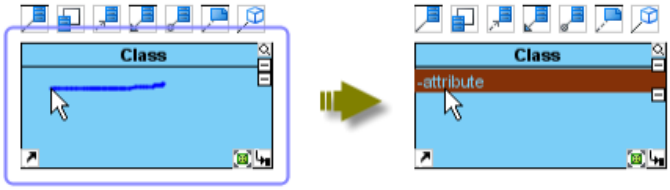
Class is created

Creating Class member

You can learn how to create attribute and operation within the class in the following sub-sections.

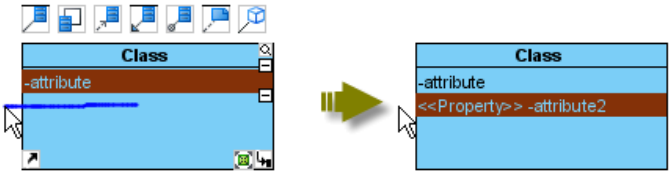
Creating an attribute

1. To create attribute, draw a line from the right to the left within the class. As a result, an attribute is created.



Attribute is created

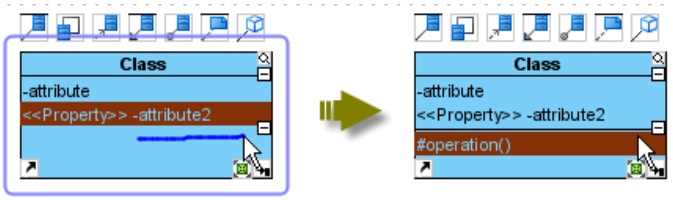
2. If you draw the line until outside the class, an attribute with <<Property>> stereotype will be created.



<<Property>> is created

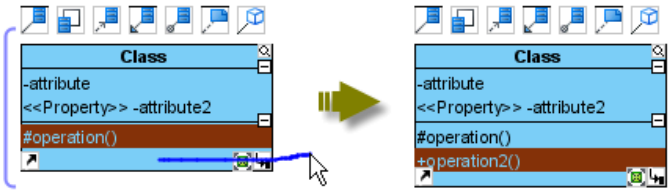
Creating an operation

1. To create operation, draw a line from the left to the right within the class, an operation with protected visibility is created.



Operation is created

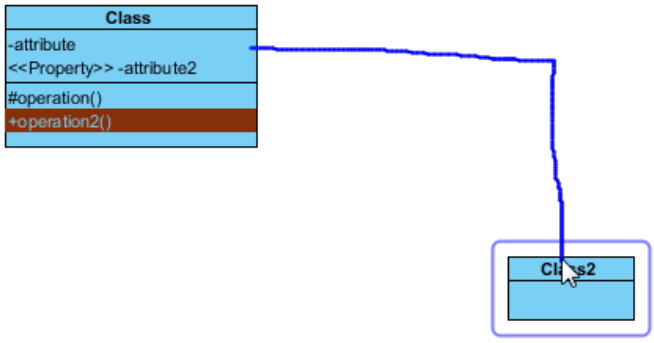
2. If draw the line until outside the class, a public operation will be created.



Public operation is created

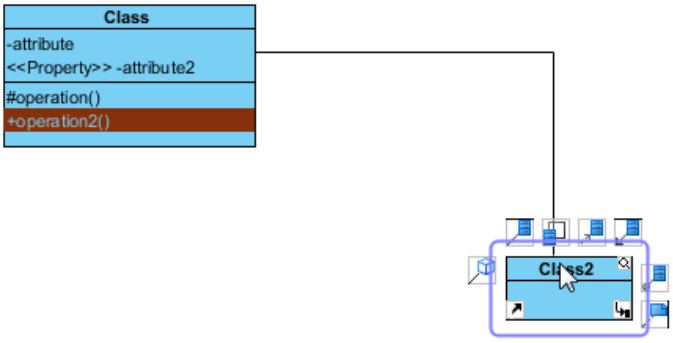
Connecting shapes

1. Draw a line from one shape to another.



Draw a line from a shape to another

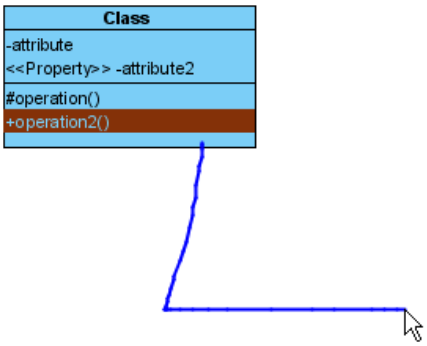
2. After the mouse is released, a connector is created between two shapes.



Association is created

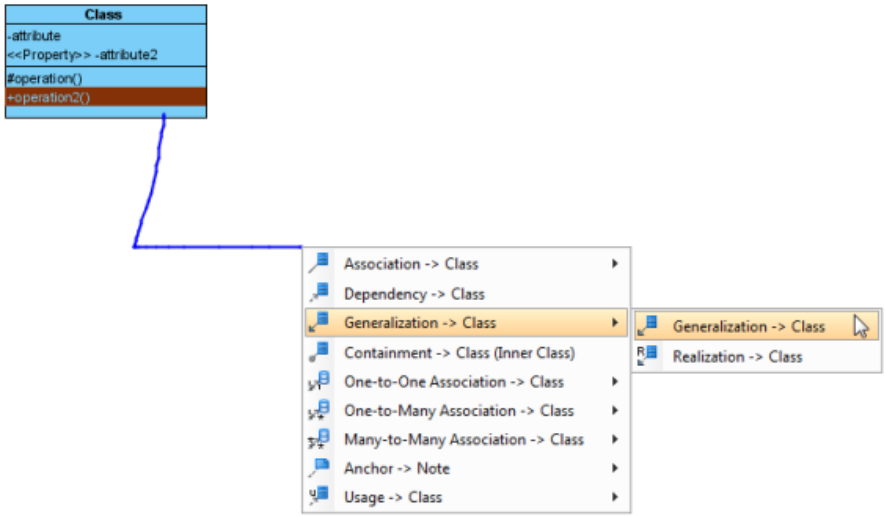
Creating a new shape

1. A new shape can also be created. To do so, draw a line from an existing shape to your preferred place.



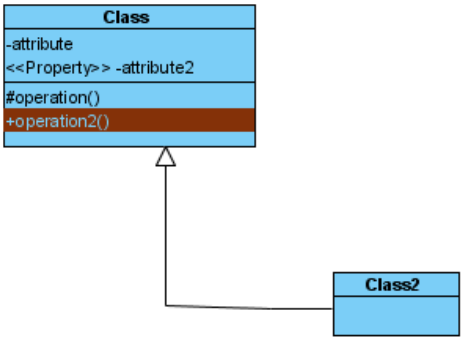
Draw a line to an empty area

2. After the mouse is released, a pop-up menu will be shown. You can select your preferred type of connector and shape from the pop-up menu.



Create generalization with class

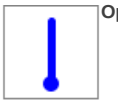
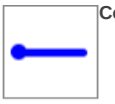
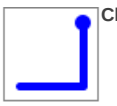
3. The two shapes with connector are created.



Class with generalization is created

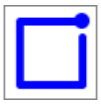
List of supported mouse gestures

General

Icon	Description
	Layout diagram
	**Open diagram specification**
	**Connect new shape**
	Connect existing shape
	**Close Diagram**
	Thumbnail view

The description of general mouse gestures

Activity diagram

Icon	Description
	Action
	Activity
	Decision Node
	Initial Node/Finial Node (If there is no Initial Node, an Initial Node will be created. Else if there is no Final Node, a Final Node will be created.)

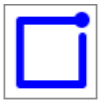
The description of mouse gestures for activity diagram

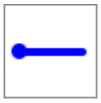
Activity diagram (UML 1.x)

Icon	Description
	Action State
	Sub-Activity
	Swimlane
	Horizontal Synchronization Bar
	Vertical Synchronization Bar
	Initial State/Final State (If there is no Initial State, an Initial State will be created. Else if there is no Final State, a Final State will be created.)

The description of mouse gestures for activity diagra

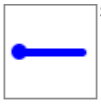
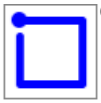
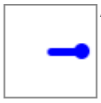
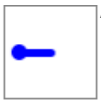
Business process diagram

Icon	Description
	Sub-Process
	Pool/Task

	Horizontal Lane
	Vertical Lane

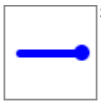
The description of mouse gestures for business process diagram

Class diagram

Icon	Description
	Sync. to ERD
	Class
	Package
	Add attribute (Add an attribute to class. If mouse released outside the class, getter and setter property will be set to true.)
	Add operation (Add an operation to class. If mouse released inside the class, visibility will be protected, otherwise it will be public.)

The description of mouse gestures for class diagram

Communication diagram

Icon	Description
	Sync. To Sequence Diagram
	Lifeline
	Actor
	Package

The description of mouse gestures for communication diagram

Component diagram

Icon	Description

Icon	
	Component
	Instance Specification
	Package

The description of mouse gestures for component diagram

Composite structure diagram

Icon	Description
	Class
	Interface
	Collaboration
	Collaboration Use

The description of mouse gestures for composite structure diagram

Data flow diagram

Icon	Description
	Process
	External Entity
	Data Store

The description of mouse gestures for data flow diagram

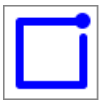
Deployment diagram

Icon	Description
	Node
	Component

	Instance Specification
	Package

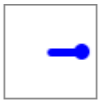
The description of mouse gesture for deployment diagram

EJB diagram

Icon	Description
	Entity Bean
	Message-Driven Bean
	Session Bean
	Package

The description of mouse gestures for EJB diagram

Entity relationship diagram

Icon	Description
	Sync. to Class Diagram
	Entity
	Add column

The description of mouse gestures for ERD

Interaction overview diagram

Icon	Description
	Interaction
	Decision Node
	Initial Node/Final Node (If there is no Initial Node, an Initial Node will be created. Else if there is no Final Node, a Final Node will be created.)

The description of mouse gestures for interaction overview diagram

Mind mapping diagram

Icon	Description
	Node

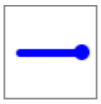
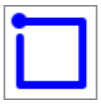
The description of mouse gesture for mind mapping diagram

Object diagram

Icon	Description
	Instance Specification
	Class
	Package

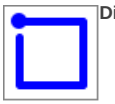
The description of mouse gestures for object diagram

ORM diagram

Icon	Description
	Sync. Classes -> Entities
	Sync. Entities -> Classes
	Class
	Entity
	Package

The description of mouse gestures for ORM diagram

Overview diagram

Icon	Description
	Diagram Overview

The description of mouse gesture for overview diagram

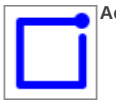
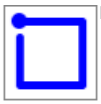
Package diagram

Icon	Description

	Package

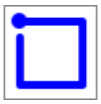
The description of mouse gesture for package diagram

Sequence diagram

Icon	Description
	Sync. to Communication Diagram
	Lifeline
	Actor
	Alt
	Loop

The description of mouse gestures for sequence diagram

State machine diagram

Icon	Description
	State
	Submachine State
	Initial Node/Final Node (If there is no Initial State, an Initial State will be created. Else if there is no Final State, a Final State will be created.)

The description of mouse gestures for state machine diagram

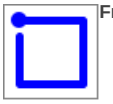
State machine diagram (UML 1.x)

Icon	Description
	State
	Concurrent State
	Submachine State

	Horizontal Synchronization Bar
	Vertical Synchronization Bar
	Initial State/Final State (If there is no Initial State, an Initial State will be created. Else if there is no Final State, a Final State will be created.)

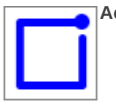
The description of mouse gestures for state machine diagram (UML 1.x)

Timing diagram

Icon	Description
	Frame

The description of mouse gesture for timing diagram

Use case diagram

Icon	Description
	Use Case
	Actor
	Package

The description of mouse gestures for use case diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

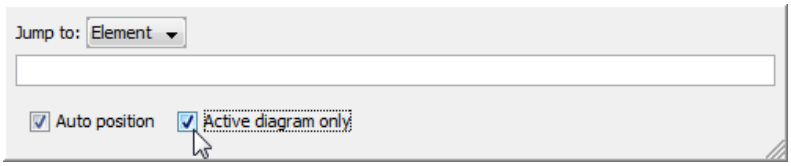
Jumping to shape

For searching shape/shapes in shortcut, the application of the jump to shape/shapes facility is introduced. You can select either jump to a model element in an active diagram, or jump to any model elements in the current project, or even jump to a diagram in current project.

Jumping to a diagram/model element in project

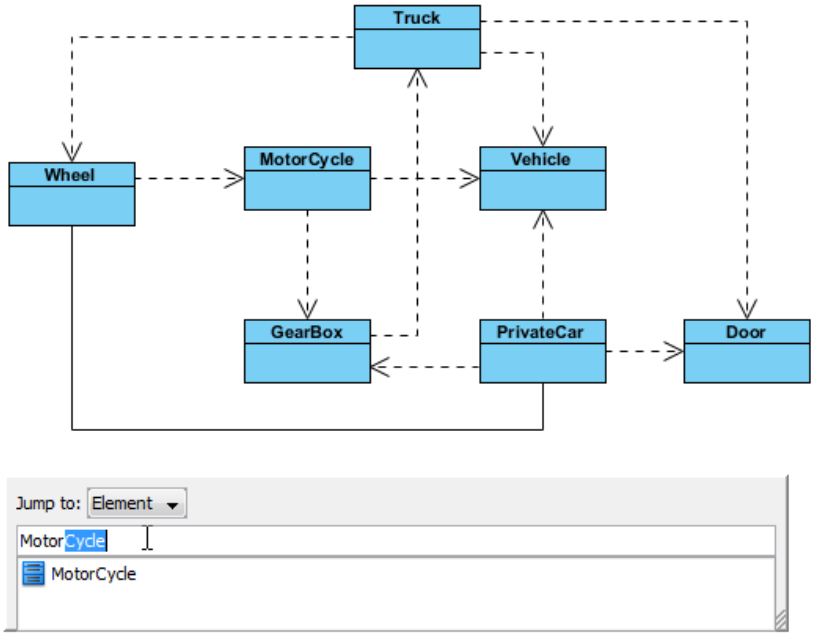
If there is an active diagram opened, you can jump to the model element in the active diagram.

1. You can select **Edit > Jump to Element in Active Diagram...** from the main menu or press **Ctrl+J** to unfold **Jump to** dialog box.
2. Apart from jumping to element in active diagram, you can also jump to any element within the project. You may select **Edit > Jump to Element...** from the main menu or press **Ctrl+Shift+J** to unfold **Jump to** dialog.
3. In **Jump to** dialog box, if you want all elements within project to be searched, uncheck the **Active diagram only** checkbox. In some cases, the checkbox is disabled because no diagram is opened.



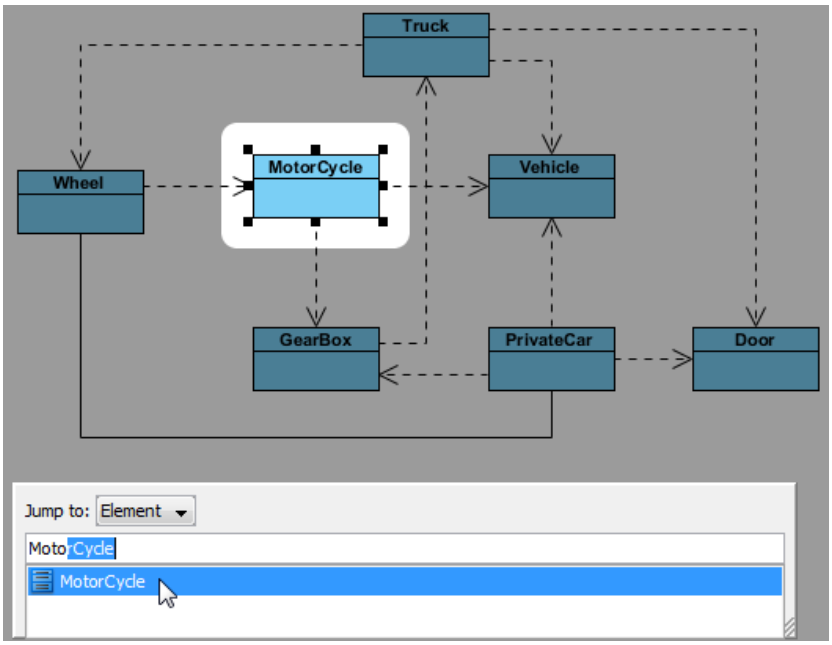
Jump to dialog box is shown

4. Enter a word in the text field, a list of model element's name that starts with the word you typed will be shown .



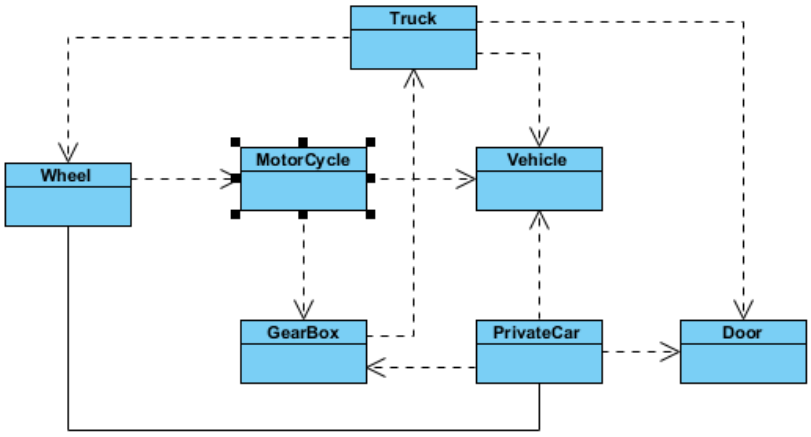
A list of possible shapes is shown

5. Press **Down** key to search for the model element's name if the list is too long. Click your target model element's name and it will be spot-lighted on the active diagram.



Shape is spot-lighted

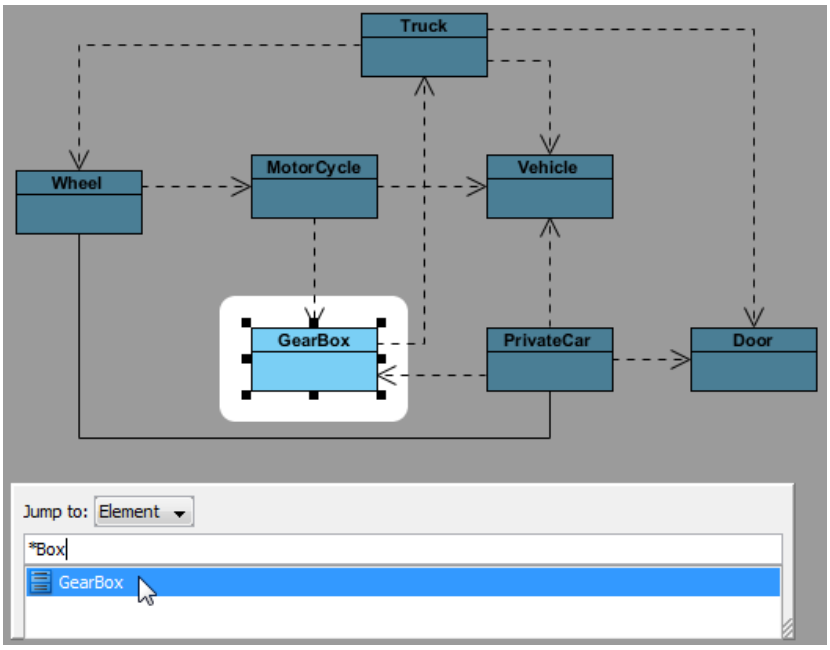
6. Press **Enter** to confirm jump to the model element. Finally, the **Jump to** dialog box will then be hidden and the model element will be selected on diagram.



Shape is selected after confirm jump to

Filtering with wild card character

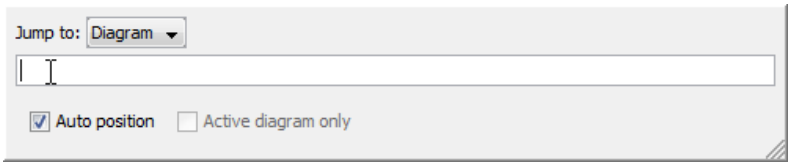
For quick search, you can type a word with * in the text field. The asterisk can substitute a character or a word when you don't remember the exact spelling. As a result, all names of shape that similar to the word you typed will be shown.



*Enter name with **

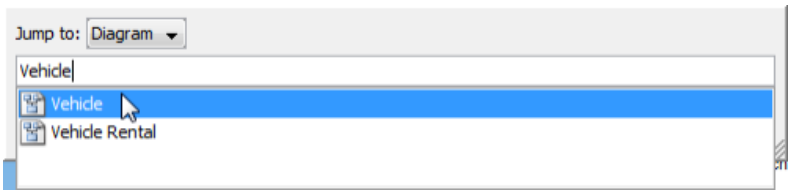
Jumping to diagram

1. To do so, select **Edit > Jump to Diagram...** from the main menu or press **Ctrl+Shift+D** to show **Jump to** dialog box.
2. **Jump to** dialog box is shown with selected **Diagram** in combo box. It means **Jump to** dialog box will search all diagrams within the project.



Jump to dialog will search all diagrams within the project

3. Enter a word out of the whole diagram's name will show a list of diagrams' names similar with the word you typed . Select the diagram and press **Enter** will open the diagram.



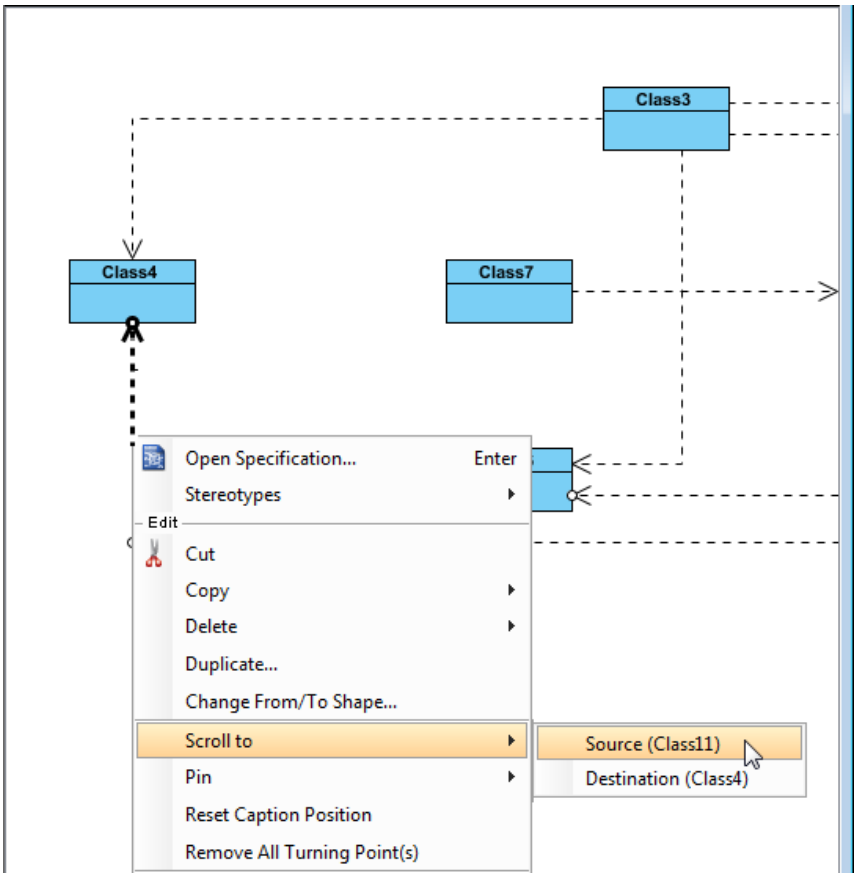
Select the diagram on list

Easy navigation to connected elements

If you have an oversized diagram that the from/to model elements of connector can't be shown on the screen, it is hard for you to know the from/to model elements. Agilian supports **Scroll to** function to scroll to from/to model element of a connector. If you want to know which model element is the from model element that is connected with a model element .

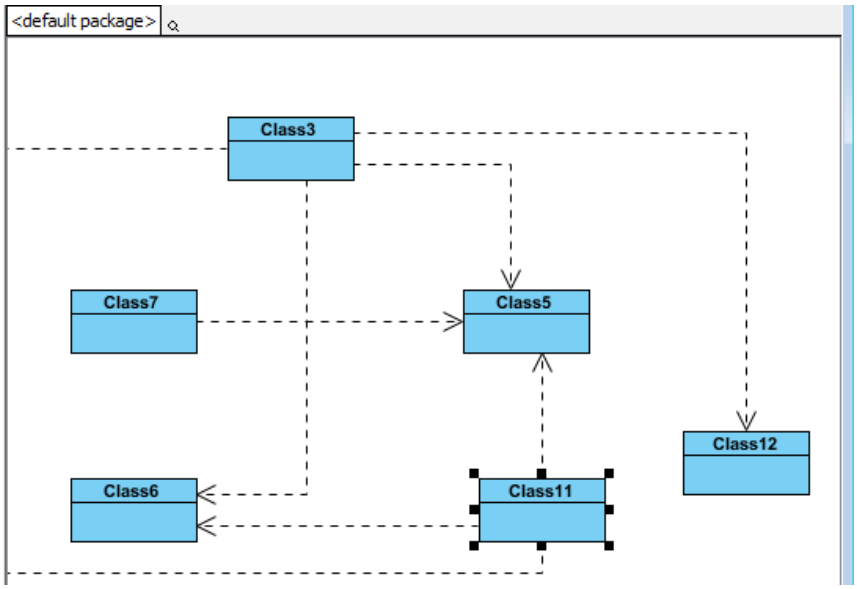
To achieve this:

1. Right click on the connector and select **Scroll to** and then select **Source** with parentheses from the pop-up menu. (The word in parentheses is the name of the from model element.)



Scroll to from model element

2. The from model element is subsequently selected on diagram.



From model element is selected

Related Resources

The following resources may help you learn more about the topic discussed in this page.

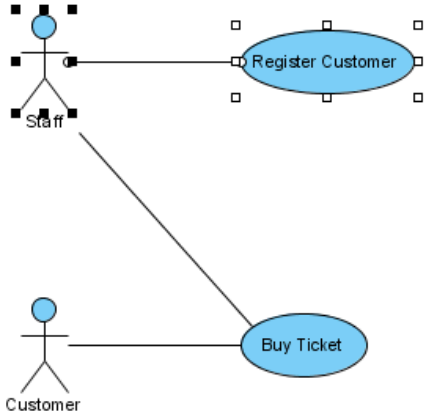
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Grouping diagram elements

When you have aligned shapes to a group of shapes, with grouping, you are able to move a number of shapes simultaneously, or let them share the same formatting properties, like background color and line formatting. By grouping shapes, shapes within the group will move together when moving any shape inside the group. If you edit formatting of one shape, all shapes will also be shared.

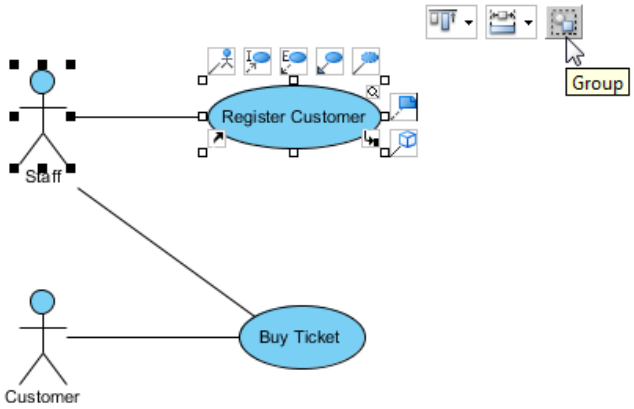
Grouping diagram elements

1. Select the target shapes for grouping.



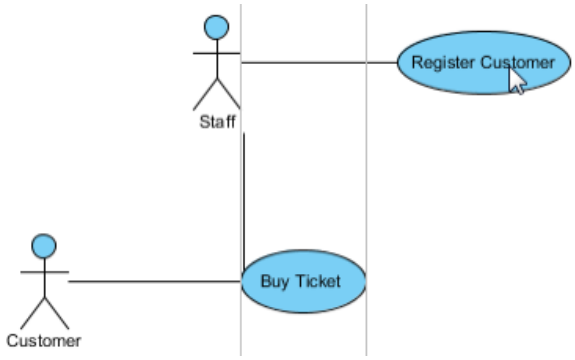
Select several shapes

2. When resource icon reveals, click resource icon **Group** to group the selected shapes.



*Group through resource icon **Group***

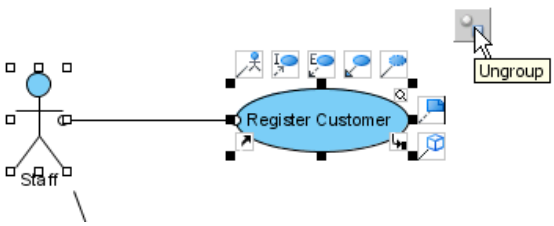
3. When you move a shape, other shapes within the same group will also be moved.



All shapes in group are moved together

Ungrouping diagram elements

To ungroup the shapes, click resource icon **Ungroup**.



*Click **Ungroup***

Related Resources

The following resources may help you learn more about the topic discussed in this page.

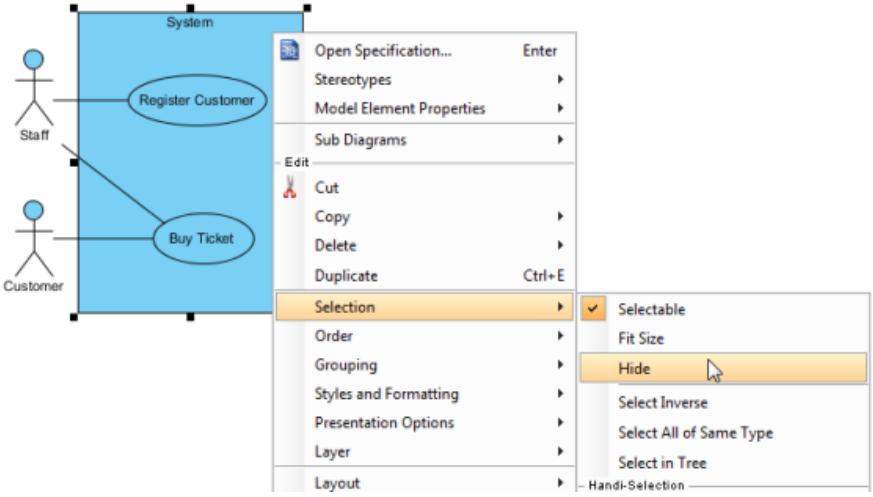
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Show/hide diagram elements

If some shapes on a compact diagram are needed to be hidden or shown , you can do it through pop-up menu. In some cases, you need to hide away diagram elements on diagram temporarily. For example, you have to hide away annotation shapes in which internal communications are recorded before you print the diagram out for customers' review. In Agilian , hiding of diagram elements can be done per shape, per shape type or by stereotype.

Hiding selected diagram element(s)

Right click on the target shape(s) for hiding and select **Selection > Hide** from the pop-up menu.



Select a shape for hiding

Apart from the selected shape, its related shape(s) (e.g. children and relationship) will also be hidden.

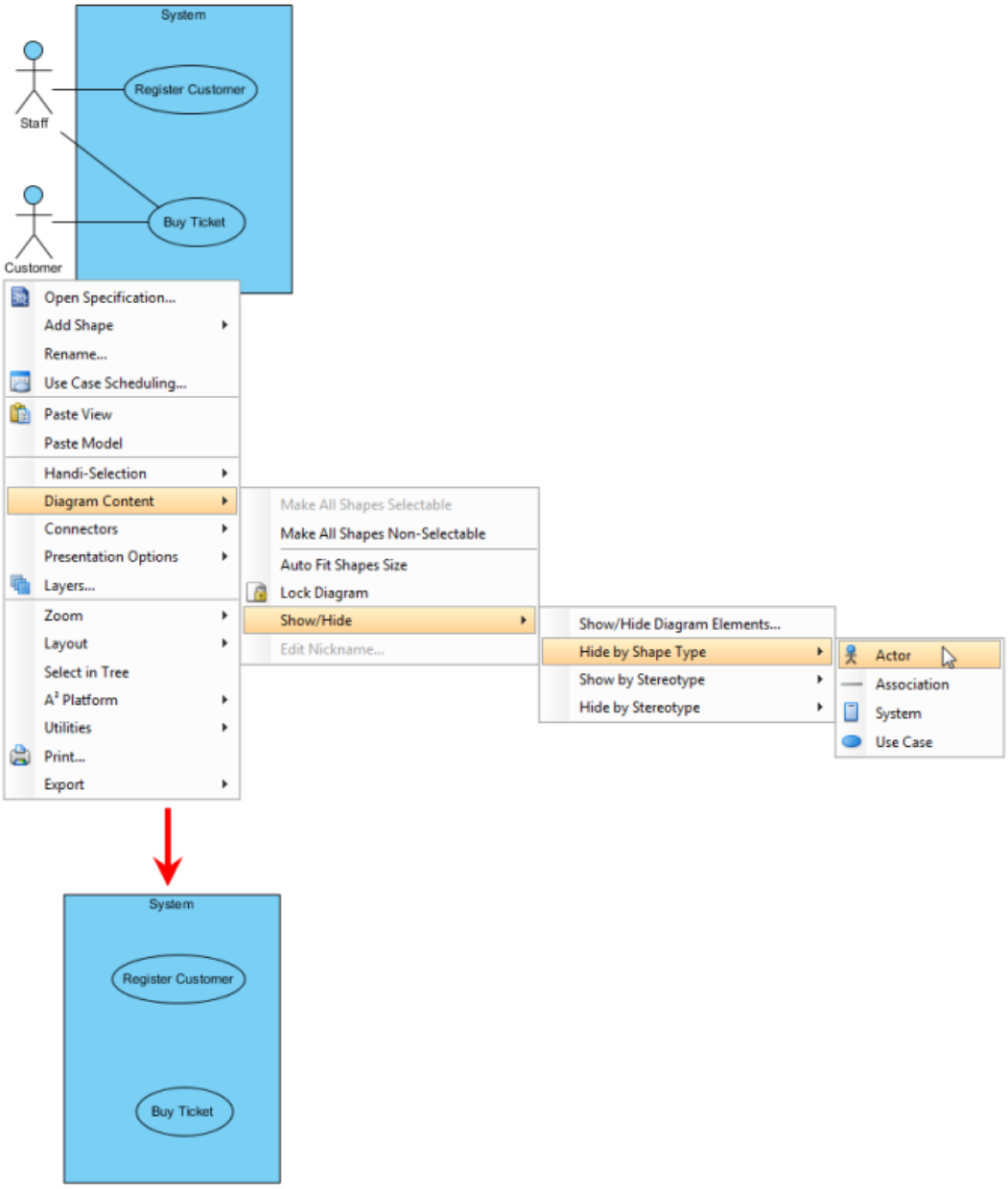
The selected shape and its related shapes are hidden

NOTE: To hide a shape will also make the connectors that attached to it hidden.

NOTE: To hide a container shape (e.g. package) will also make the contained shapes hidden.

Hiding diagram elements by shape type

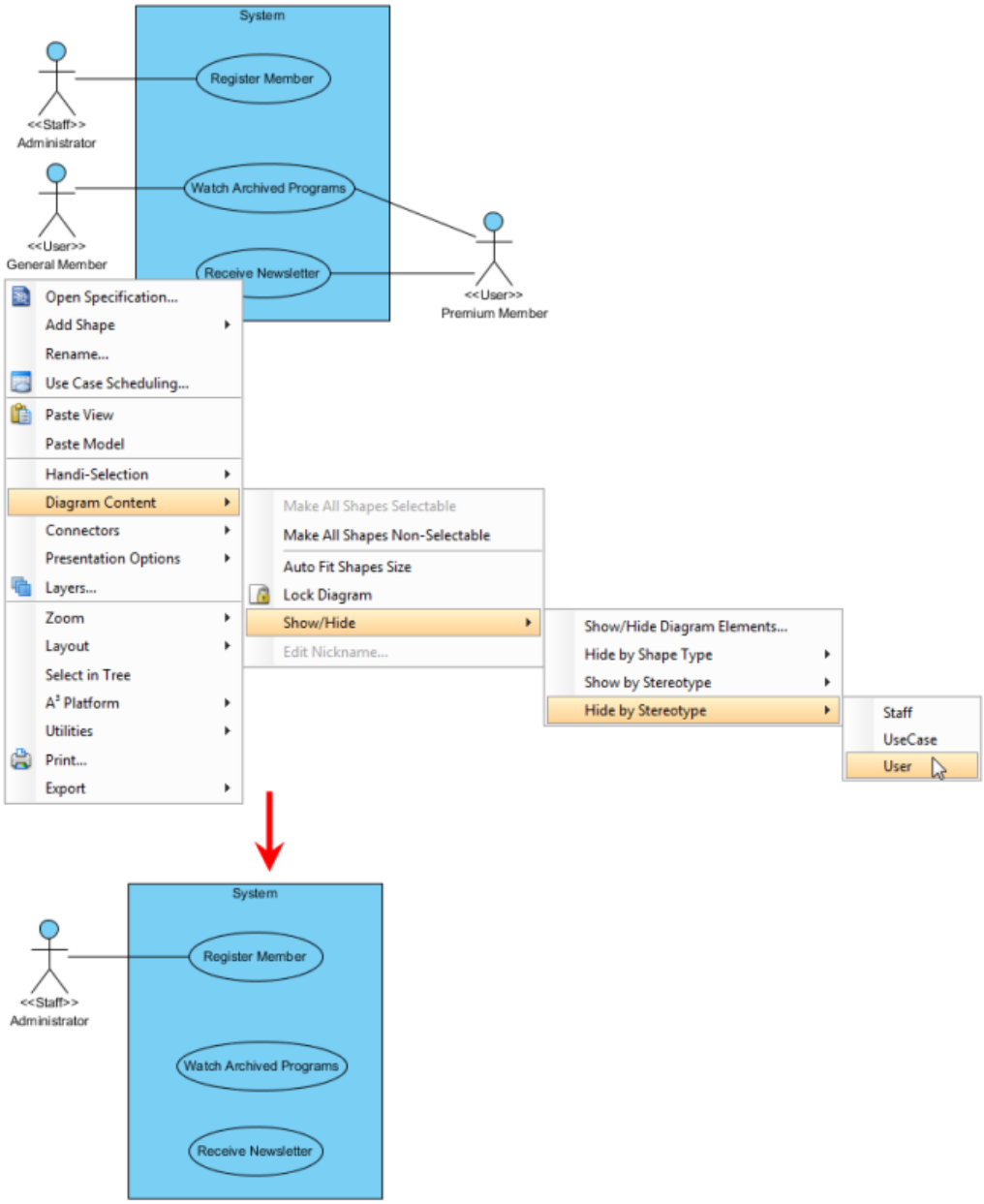
All the shapes with same type on the diagram can be hidden together. Right click on the diagram's background and select **Diagram Content > Show/ Hide > Hide by Shape Type,** and then select the shape you want to be hidden from the pop-up menu. As a result, all shapes with the same type you selected will be hidden.



Hide all shapes by shape type

Hiding diagram elements by stereotype

The stereotype of all diagram elements from the diagram can be hidden. Right click on the diagram's background and select **Diagram Content > Show/Hide > Hide by Stereotype**, and then select one stereotype out of the list from the pop-up menu. As a result, all shapes with same stereotype you selected will be hidden.



Hide all shapes by stereotype

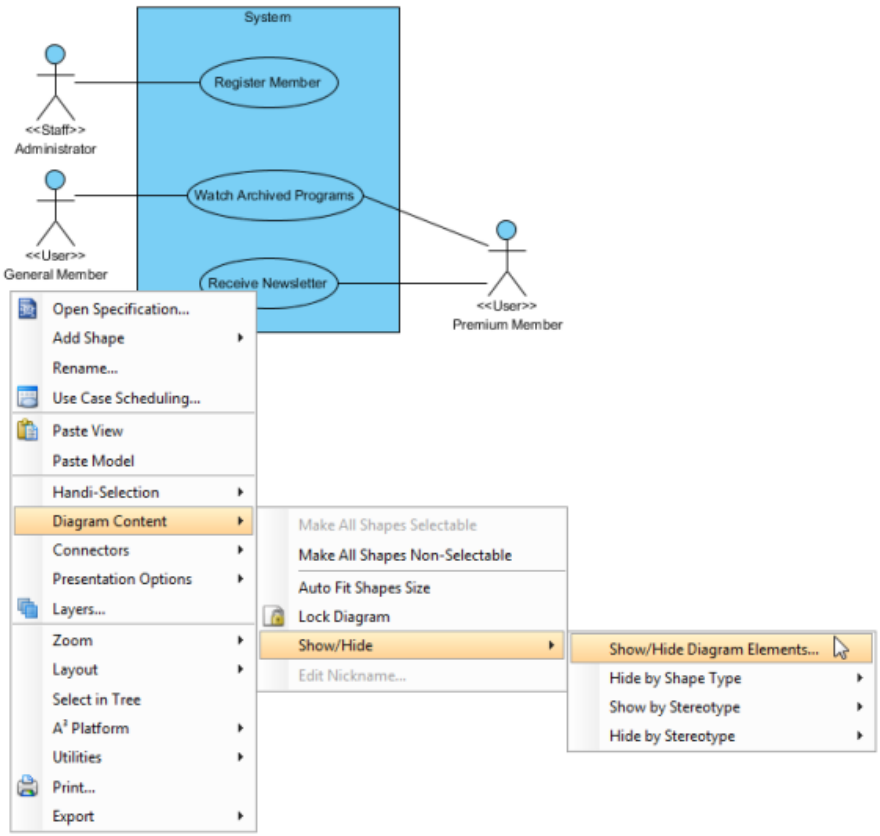
Showing all hidden diagram elements

All hidden shapes from the diagram can be shown again. Right click on the diagram's background and select **Diagram Content > Show/Hide > Show all Diagram Elements**. As a result, the hidden shapes will be shown on the diagram.

Managing show/Hide of specific diagram element(s)

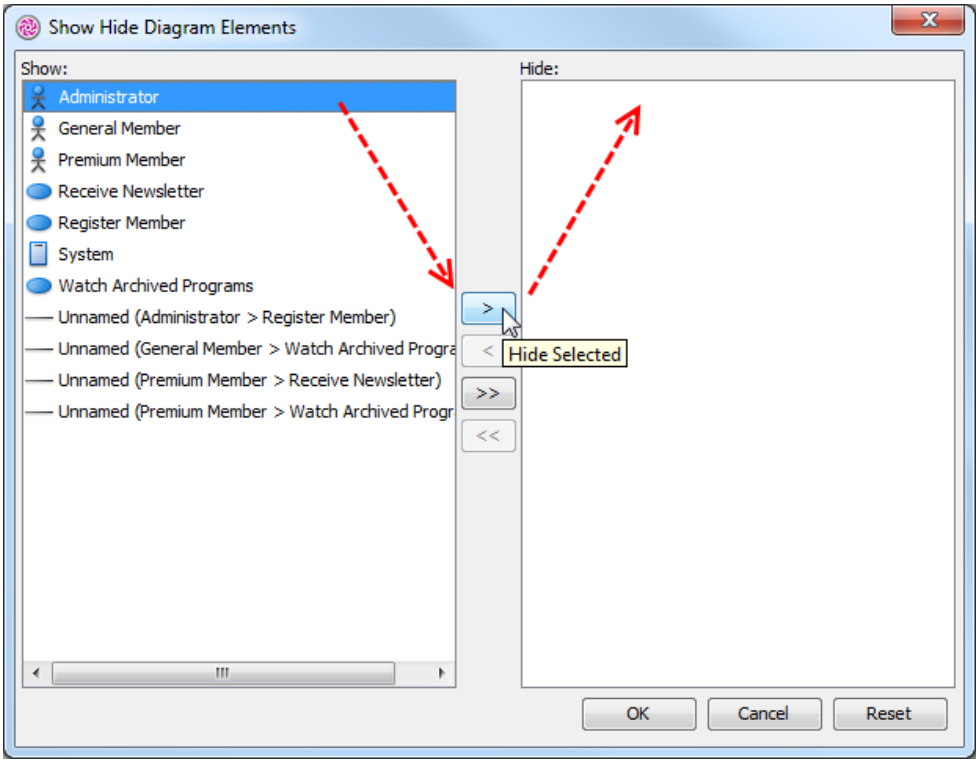
A better management of showing and hiding the shapes can be done through **Show Hide Diagram Elements** dialog box.

1. Right click on the diagram's background, select **Diagram Content > Show/Hide > Show/Hide Diagram Elements...** from the pop-up menu.



*Select **Show/Hide Diagram Elements...** from the pop-up menu*

2. When **Show Hide Diagram Elements** dialog box is unfolded, you may select the shape(s) you would like to be hidden or to be shown. Click the diagram element you would like to move to **Hide** list and press **Hide Selected** button; on the other hand, click the diagram element you would like to remove it back to **Show** list and press **Show Selected** button. For moving all diagram elements to **Hide** list, press **Hide All** button; press **Show All** button, vice verse. Finally, click **OK** to confirm.



*Select a diagram element to be moved to **Hide** list*

NOTE: If a shape is selected from **Show** list to **Hide** list, the related shape(s) of the selected shape will be also removed to **Hide** list.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

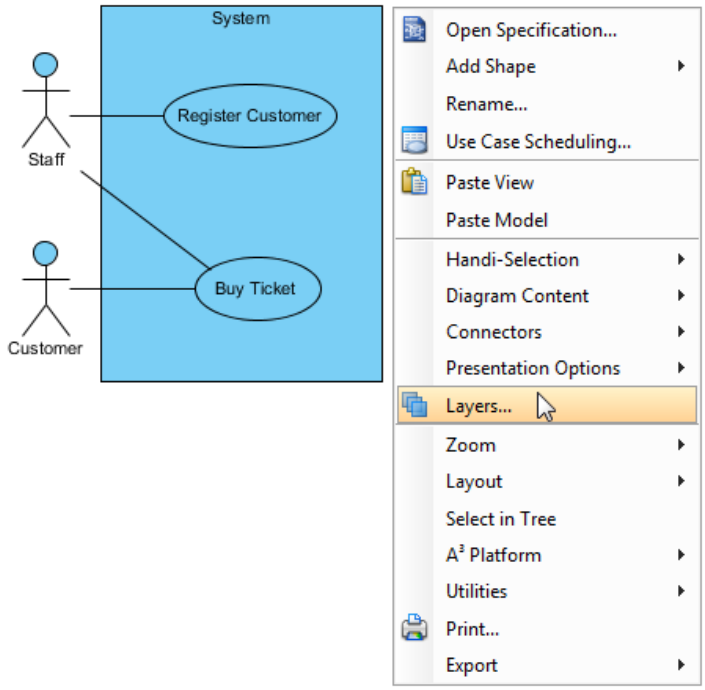
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Layer

If you have ever experienced how hard it is to deal with a number of shapes, try the application of multi-layers. In Agilian, you can manage different shapes efficiently by making using of multi-layers. The function of layer assists you in assigning different shapes into different layers, hiding unnecessary shapes, locking shapes and selecting shapes in shortcut.

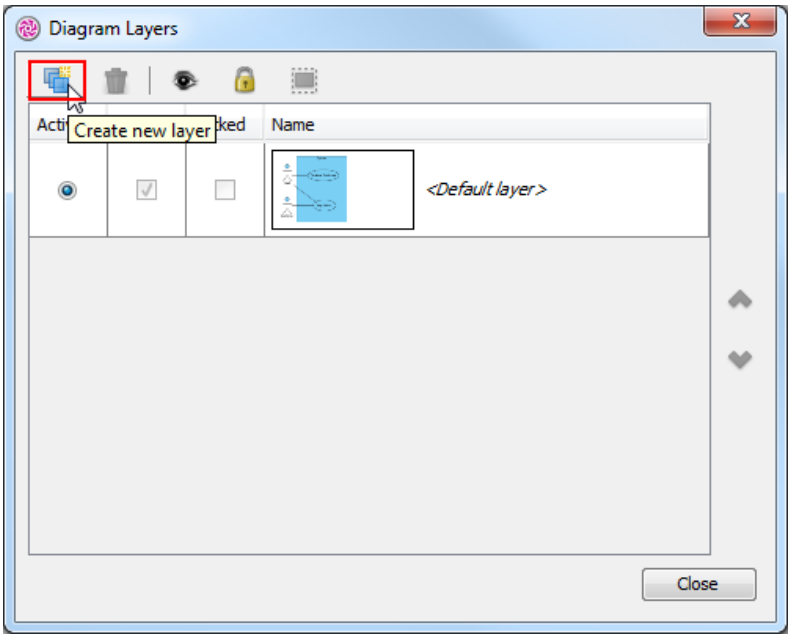
Creating a layer

1. Right click on the diagram background and select **Layers...** from the pop-up menu to open **Diagram Layers** dialog box.



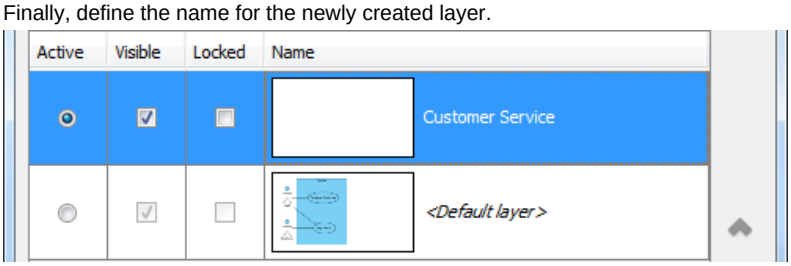
*Select **Layers...** from the pop-up menu*

2. In **Diagram Layers** dialog box, click **Create new layer** button to create a new layer.



*Create a new layer in **Diagram Layers** dialog box*

3. Finally, define the name for the newly created layer.

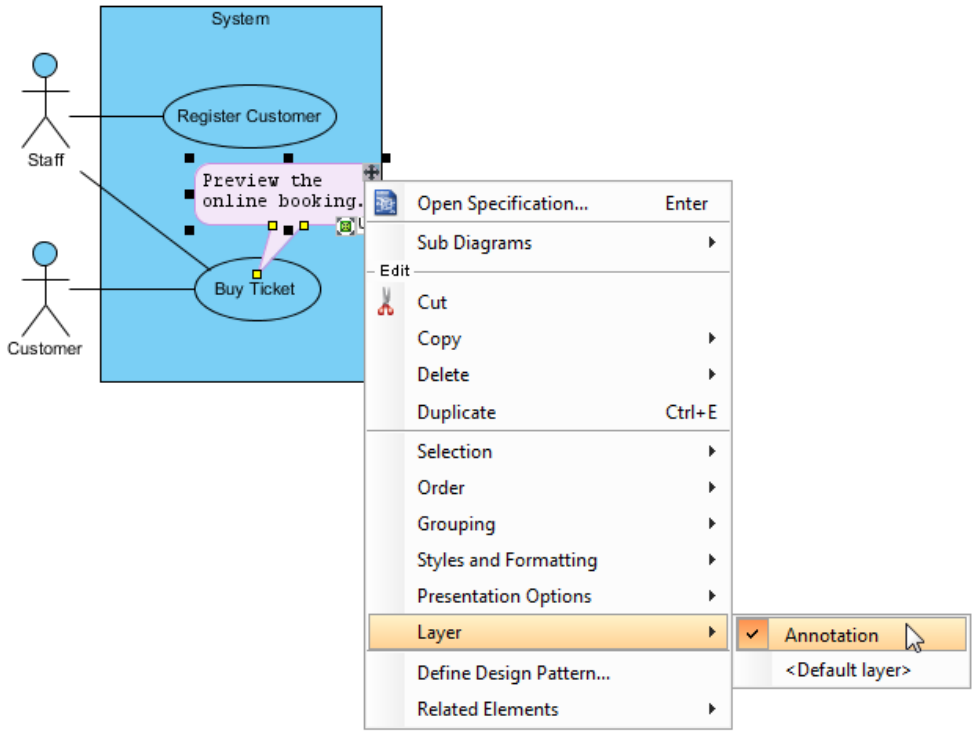


New layer will be an active layer

Sending shapes to a layer

The existing shapes are kept in default layer. However, both existing shapes and the newly created shapes can be sent to a new layer.

Create a new layer by following the steps of previous section. Right click on a shape and select **Layers**, and then select the new layer you create in advance.

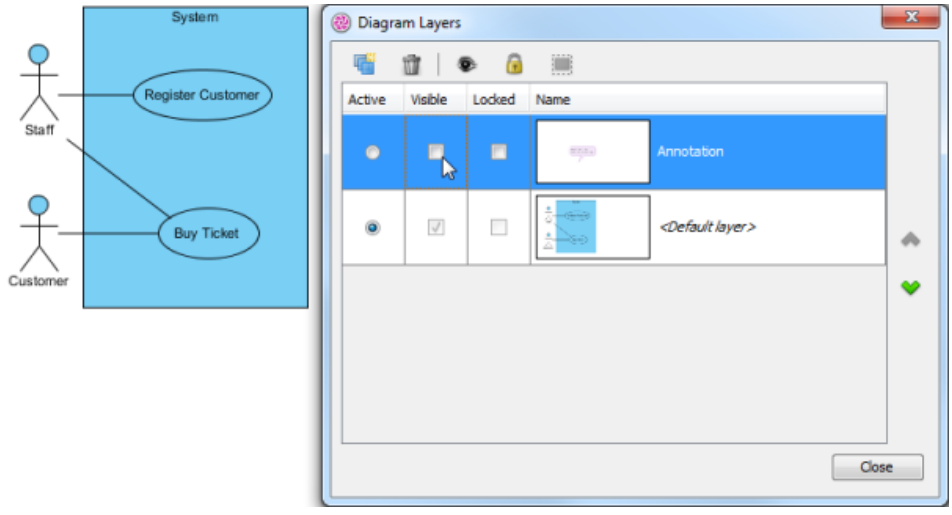


Send the shape to a new layer

As a result, the shape you selected will be sent to the new layer.

Hiding shapes on a layer

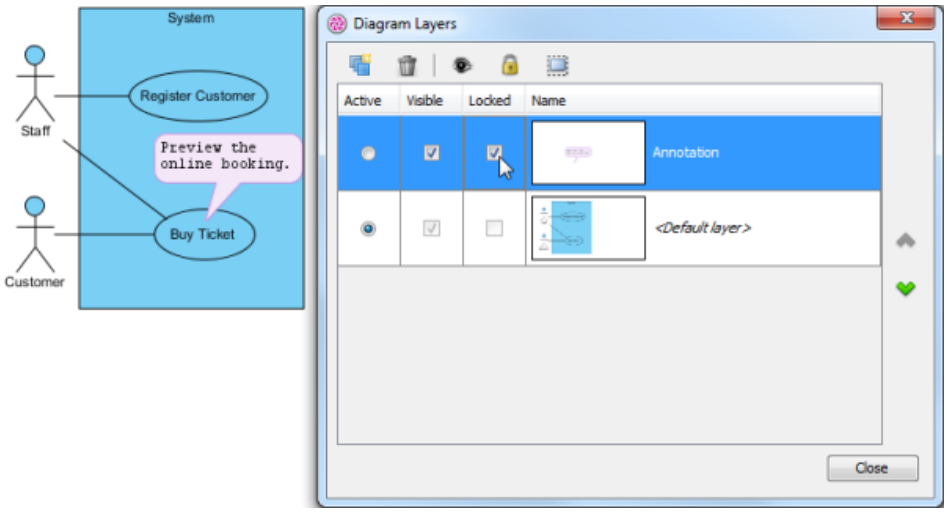
You can make a layer invisible on diagram through **Diagram Layers** dialog box. To do so, uncheck **Visible** of your target layer that you want to be hidden on diagram.



Shapes on the layers are invisible on diagram

Locking shapes on a layer

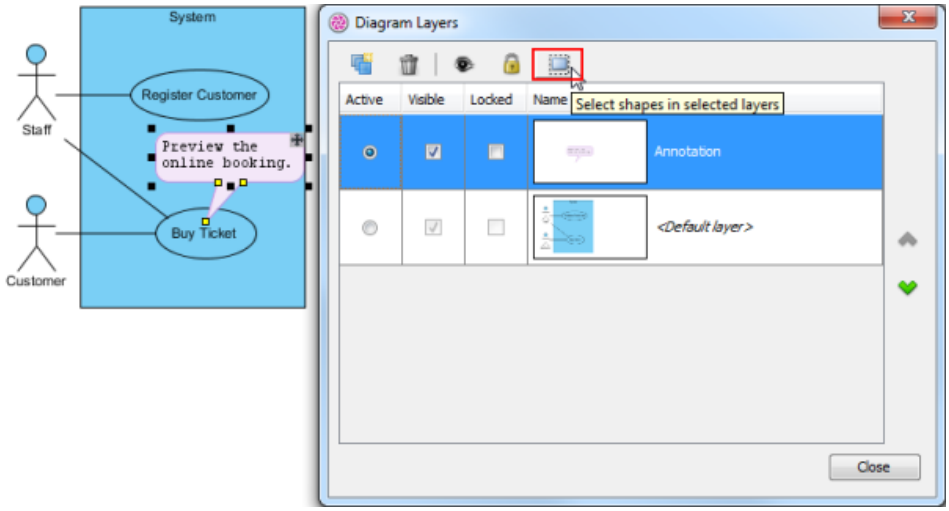
In **Diagram Layers** dialog box, check **Locked** of your target layer to make all shapes in that layer immovable and non-editable while uncheck **Locked** to make them movable and editable.



Shapes on the selected layer are locked

Selecting shapes on a layer

You can also select all shapes of the selected layer. To do so, click **Select shapes in selected layers** button in **Diagram Layers** dialog box.



Shapes of the layer are selected on diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

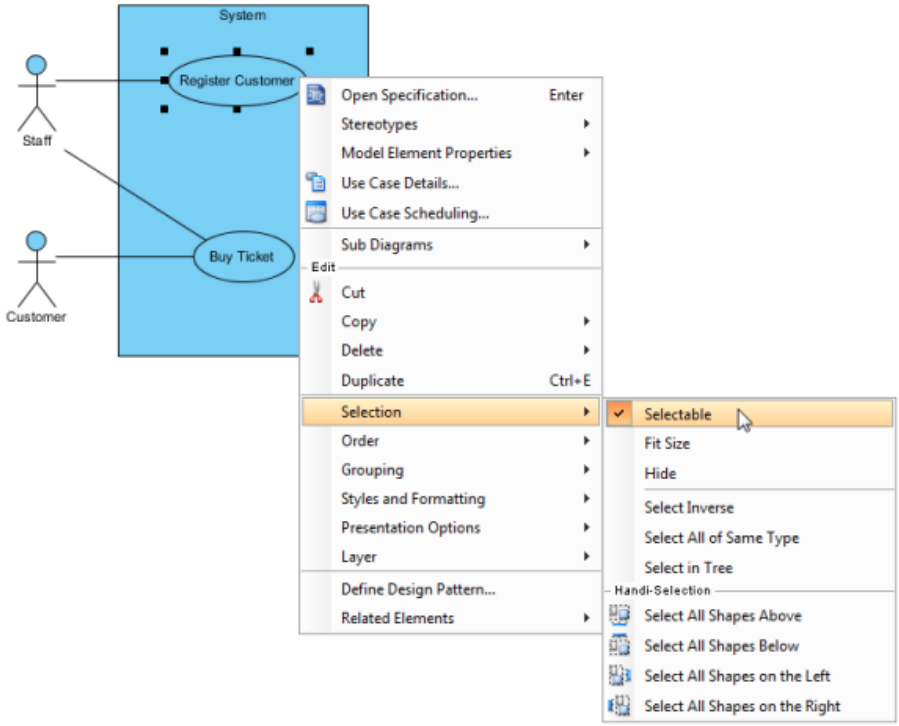
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Making shape non-selectable

If you find it annoying to move some shapes carelessly or select some shapes accidentally, the function of selectable/non-selectable would be your ideal trouble-shooter. This page is going to teach you how to make shapes non-selectable or even change them back to selectable with only few clicks.

Changing the shape to non-selectable

1. Right click on the selected shapes, uncheck **Selection > Selectable** from the pop-up menu.

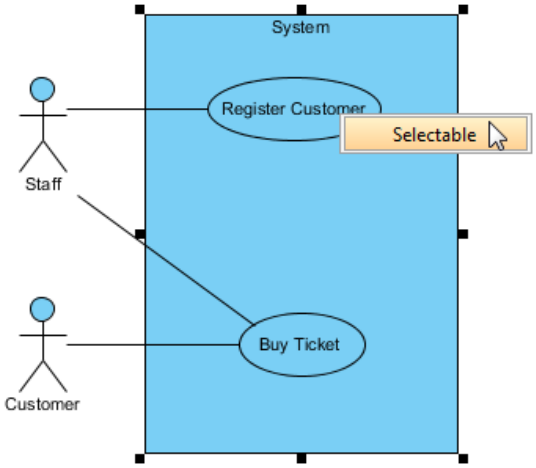


Selected shape to be non-selectable

2. After the shape is non-selectable, click the shape cannot make it to be selected. Neither will the shape be selected by mouse dragging on diagram.
3. Therefore, the non-selectable shape(s) will not be moved when other shapes are moved.

Changing the shape back to selectable

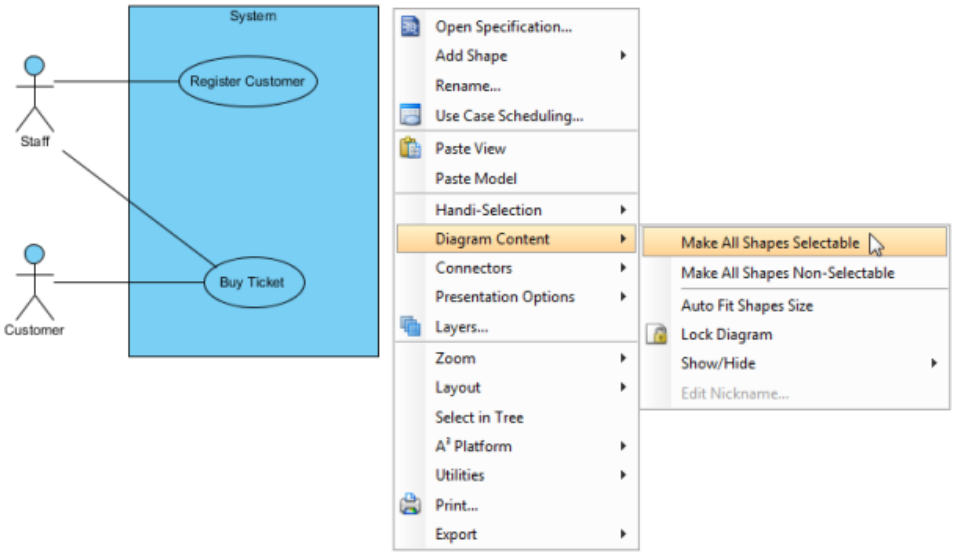
To retrieve the shape selectable, right click on the non-selectable shape and select **Selectable** from the pop-up menu.



Select Selectable

Setting all shapes to selectable or non-selectable

To makes all shapes on the diagram to be selectable or non-selectable, right click on the diagram background, select **Diagram Content** and then select either **Make All Shapes Selectable** or **Make All Shapes Non-Selectable** from the pop-up menu.



*Select **Make All Shapes Selectable***

Related Resources

The following resources may help you learn more about the topic discussed in this page.

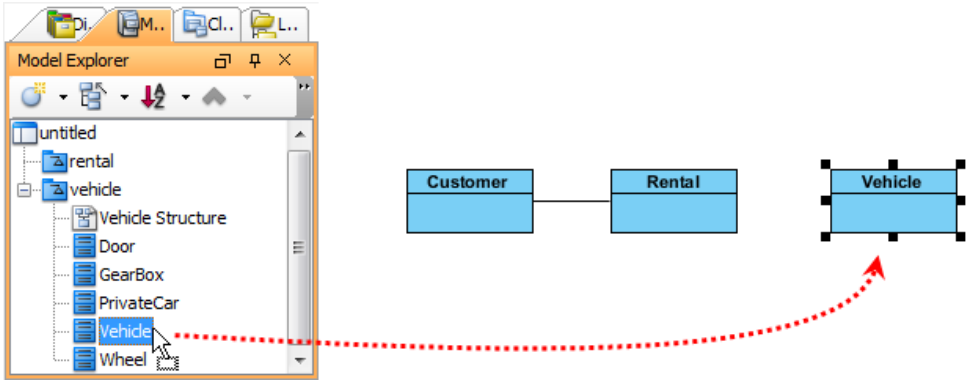
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Showing model element in multiple diagrams

You can show a model element in multiple diagrams to fit into different contexts. Two ways are suggested in Agilian to show model elements on different diagrams. Firstly, you may drag and drop the model element(s) from the tree to the diagram and secondly, you can copy and paste it as view.

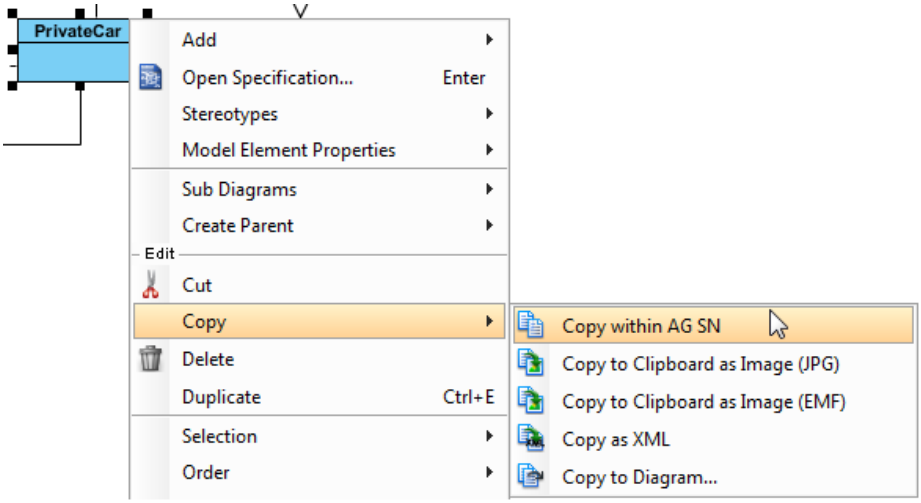
Reusing model elements

In **Model Explorer**, a model element or several model elements from the source diagram can be dragged from the tree and drop it/ them on the destination diagram.



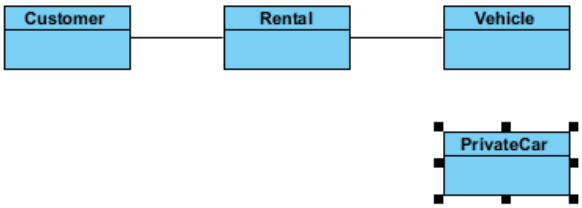
Drag a model element from Model Explorer

Alternatively, you may copy and paste the selected model element(s) as view in destination diagram. Right click on the selected model element(s) in the source diagram and select **Copy > Copy with AG SN** from the pop-up menu.



Copy a model element

When you switch to the destination diagram, right click on the preferred place that you want the copied model element(s) to be pasted on and select **Paste View** from the pop-up menu. As a result, the selected model element(s) will be pasted on destination diagram.



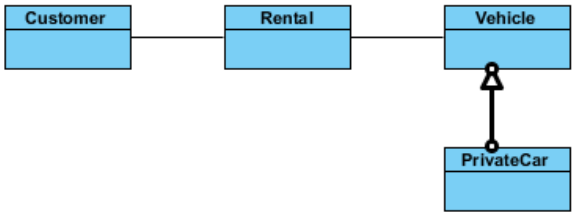
Class PrivateCar is pasted as a new view

NOTE: Choosing **Paste Model** means a new model element will be created and shown on the destination diagram.

The relationship between the model elements may not be shown on diagram. If you want their relationship to be revealed, right click on the model element, select **Related Elements > Visualize Related Model Element...** from the pop-up menu.

When the **Visualize Related Model Element** dialog box pops out, check a relationship for the model element(s) and then click the **Visualize** button to proceed.

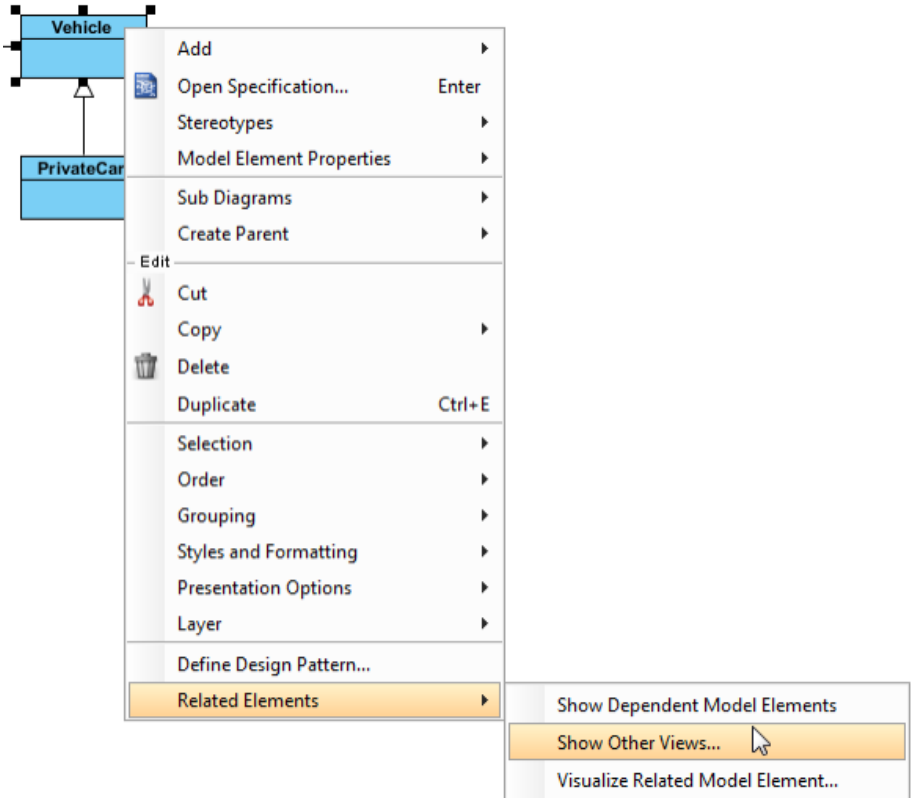
As a result, their relationship is shown on the diagram.



Relationship is visualized

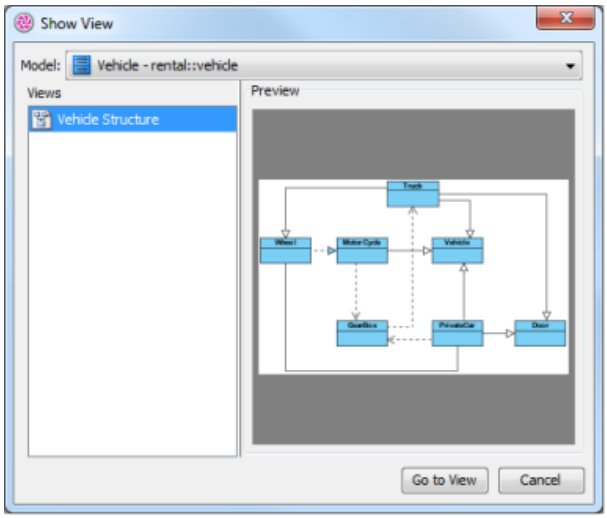
Opening other views

You can find out the source diagram of a model element by right clicking on it and select **Related Elements > Show Other Views...** from the pop-up menu.



*Click **Show Other Views...***

All diagrams, which contain the same model elements you selected, are listed and shown in the **Show View** dialog box. You can check a diagram under **Views** and preview it under **Preview**; furthermore, click **Go to View** button to jump to and view the actual diagram.



***Show View** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

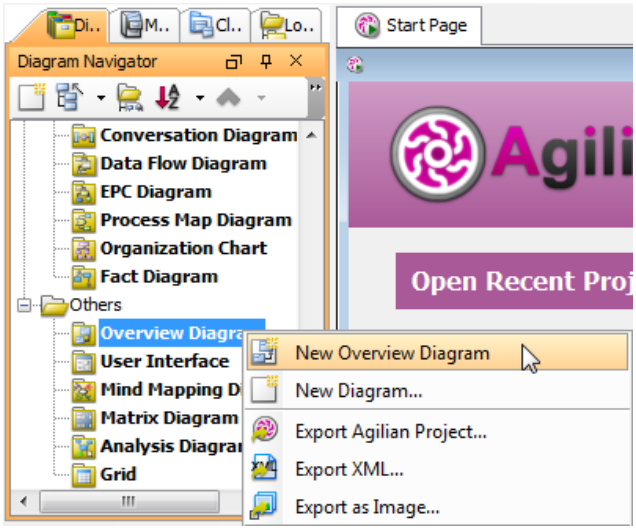
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Using overview diagram

When various diagrams with different contexts are modeled in your system, an overview diagram can be created to illustrate their relationships related each other. With the overview diagram, you can have an overview of your system, after all, you can study their relationships. Since the overview diagram is consist of thumbnail of diagrams that you have created previously, the overview of each diagram if find to be related to another diagram can be linked up with connectors.

Creating an overview diagram

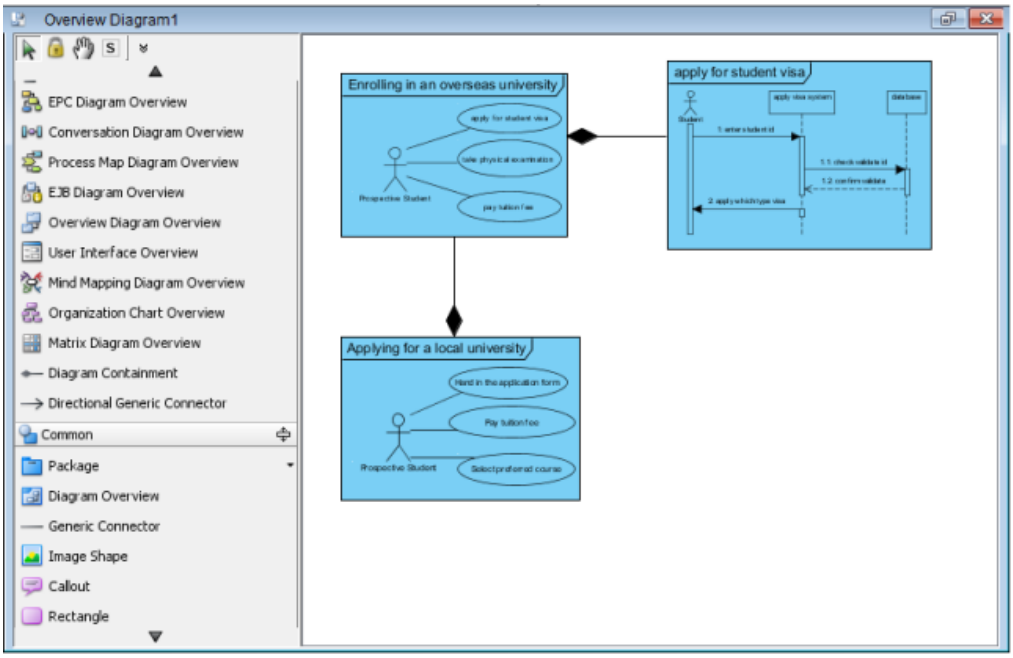
Right click **Overview Diagram** on **Diagram Navigator** and select **New Overview Diagram** from the pop-up menu to create an overview diagram.



Create a new overview diagram

Alternatively, select **File > New Diagram > Others > Overview Diagram** from main menu to create an overview diagram.

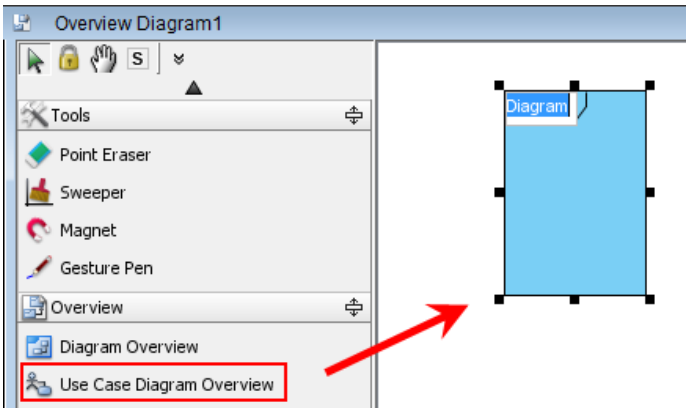
As a result, an empty overview diagram is created. Create overviews of diagrams and relate them to illustrate their relationship with each other.



The overview diagram

Creating an overview of new diagram

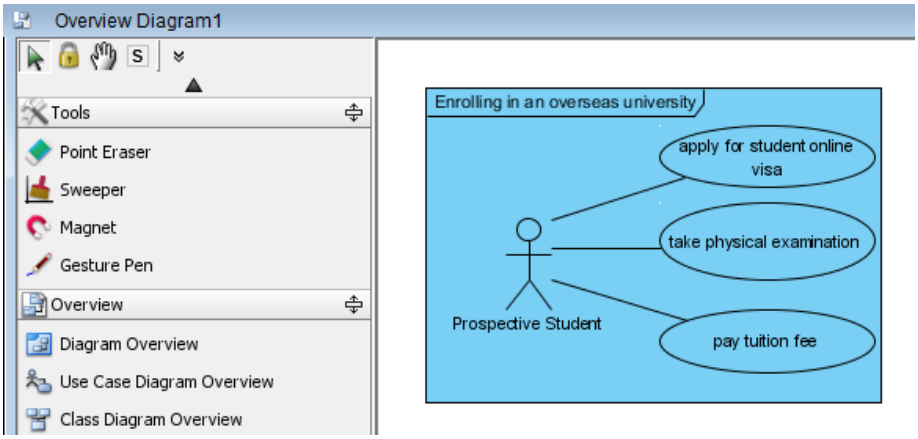
You can create a diagram from overview diagram. To create a new diagram from overview diagram, select your preferred diagram type from the diagram toolbar and drag it on the diagram pane.



Create a use case diagram overview

The new diagram will be opened subsequently. Enter a name for new diagram and start working on it.

If you go back to the overview diagram, the preview of newly created diagram can be seen. If you realize that the diagram in diagram overview is too small and vague for previewing, you may resize the diagram overview by dragging its bottom right corner.

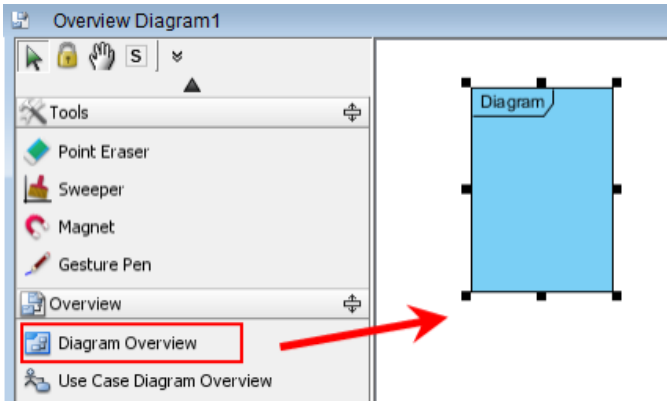


The newly created diagram is shown on overview diagram

Creating an overview of existing diagram

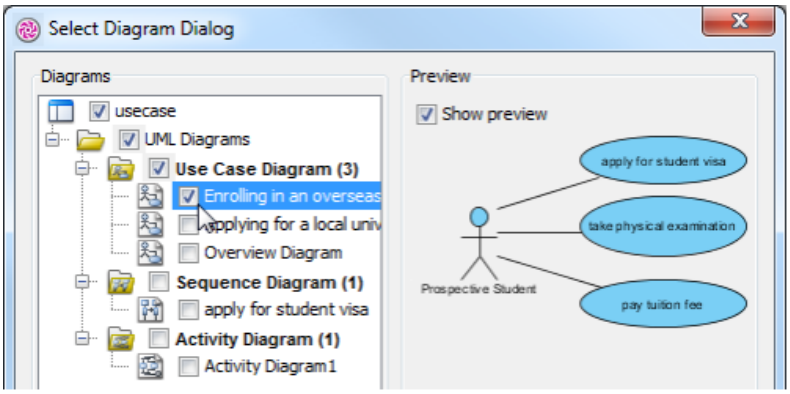
Besides creating an overview diagram of new diagram, you can also create an overview of existing diagram.

1. Create a diagram overview from the toolbar of overview diagram.



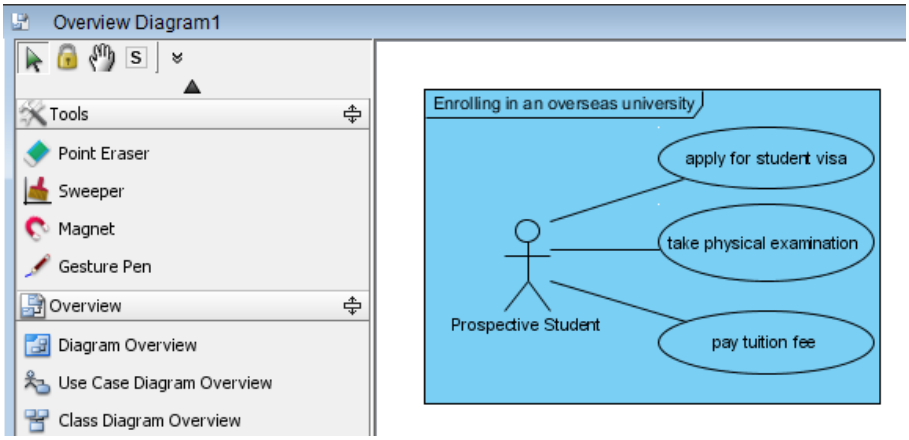
Create a diagram overview

2. Right click on the diagram overview and select **Associate to Diagram...** from the pop-up menu.
3. Select a diagram that you want to be shown on diagram overview and select **OK** button.



Select an existing diagram

4. As a result, the selected diagram will be shown on diagram overview. If you realize that the diagram in diagram overview is too small and vague for previewing, you may resize the diagram overview by dragging its bottom right corner.

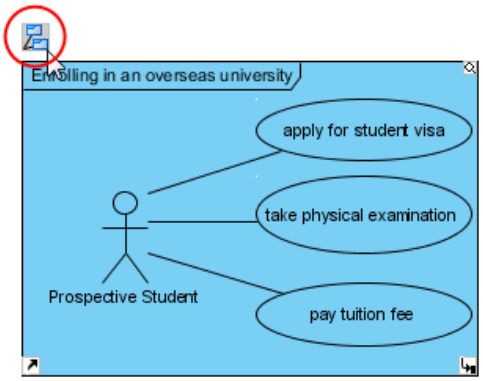


The selected diagram is shown

Visualizing sub-diagram

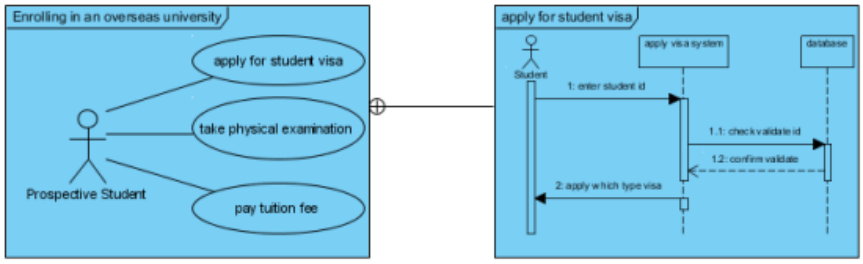
When a diagram overview you created has a sub-diagram, you can connect them together and make them shown on the current overview diagram.

Move the mouse over a diagram overview and click its resource icon **Visualize Related Diagrams**.



*Press **Visualize Related Diagrams***

As a result, the sub-diagram will be shown and both diagram overviews are connected.

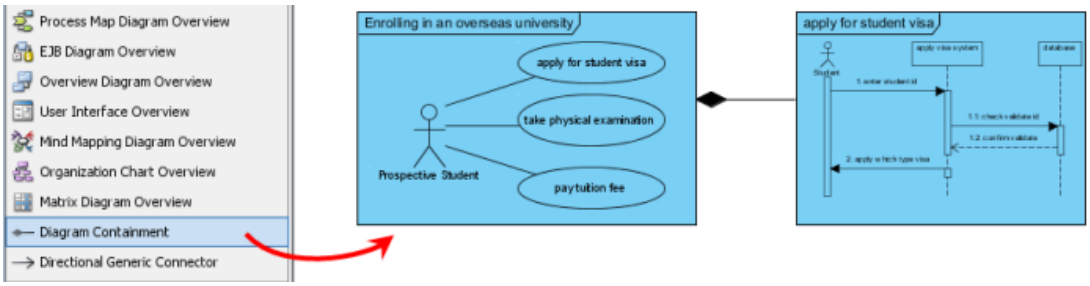


Two diagram overviews are connected

Selecting a connector for diagram overviews

In order to present a relationship between two diagrams, different connectors should be used. Diagram containment is used when a diagram contained is a part of the containing diagram overview while directional generic connector is used to model the sequence of diagrams, from advance to another.

Select a connector from the diagram toolbar and drag it from the source diagram overview to the destination diagram overview.



*Select **Diagram Containment***

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Changing Model Element Type

As your project develops, requirements will be set, issues will be found, old requirements may no longer be valid, etc. Inevitably, all these lead to changes in your existing design. When re-drawing could mean a bigger time investment, you would most likely refine your existing diagrams instead. In such case, the ability to convert a model element in the diagram from one type to another will be very useful.

When do You Convert a Model Element Type?

We support a rich set of modeling languages and notation. You can always find the best notation for thorough expression. When the type of an element is no longer appropriate or is not the best type to express an idea, you will want to replace it with a better notation. Here are some of the situations when type conversion is needed.

Design Evolvement

Take BPMN as an example. Task and sub-process are two notations introduced by BPMN. They both model activities to do within a business process. If you need to model an order processing system with BPMN, you may initially use a BPMN task to model the *Ship Order* activity. As the design evolves, let's just say that shipping order turns out to be much more complex than you thought and involves several sub-activities like arranging shipment, delivering goods, collecting payment, etc. To depict that, you need to convert the atomic task to a sub-process, not only for representing its complex nature but also for making it possible to drill down the process and model its details.

Correcting Mistakes

Things are never perfect and so are our work. Sometimes your knowledge in modeling language falls short. You may mistakenly apply a wrong type of model element in your design. For example, you could have incorrectly treated UML activity as UML action. When you realize it is a mistake, you want to correct it.

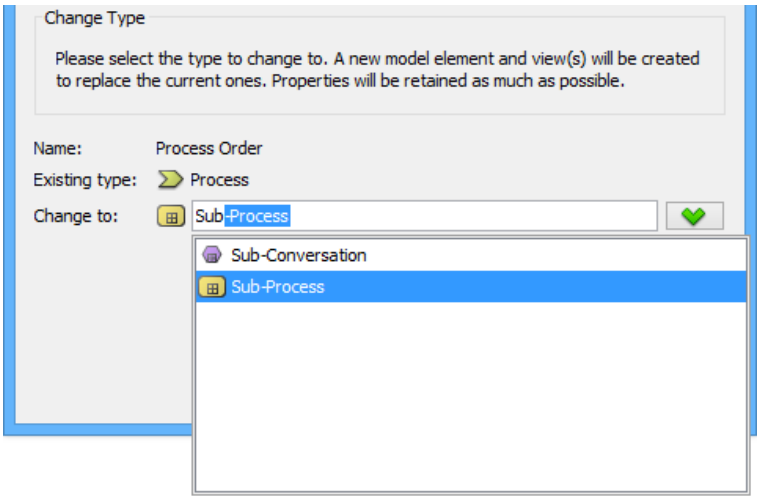
Describe More Precisely What You Want to Express

Some notations are sub-types of a more general one. For example, send-signal-action is a kind of UML action that creates a signal from its input and transmits it to the next object. While using UML action in a broad manner is not exactly wrong, using send-signal-action would reflect the truth more precisely.

Changing Model Element Type

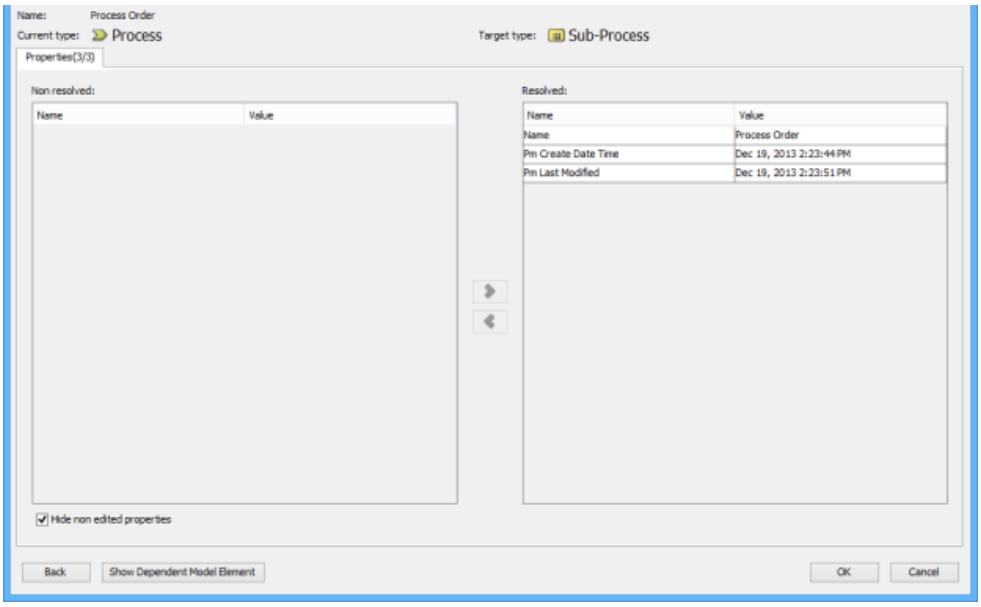
Method 1: Through diagram

1. Right click on the desired shape and select **Related Elements > Change Type...** from the popup menu.
2. In the **Change Model Element Type** window, enter the type to convert to.



Entering the type to change to

3. Click **Next** to confirm the conversion.
4. Due to the difference in the element type, some properties, relationships and tagged values may no longer be compatible after the conversion and are forced to be discarded. In the Change Model Element Type window, you will find the properties, relationships and tagged values that will be kept and discarded after the conversion. The data that will be discarded is put on the left hand side while the resolvable data is on the right. Click **OK** at bottom right.



Change Model Element Type

5. Note that the undo history will be cleared after the conversion, meaning that you cannot perform any undo action after it. Click **Yes** to confirm the change and to return to the diagram.

Method 2: Through Model Explorer

1. Right click on the desired model element in **Model Explorer** and select **Related Elements > Change Type...** from the popup menu.
2. In the **Change Model Element Type** window, enter the type to convert to.
3. Click **Next** to confirm the conversion.
4. Due to the difference in the element type, some properties, relationships and tagged values may no longer be compatible after the conversion and are forced to be discarded. In the Change Model Element Type window, you will find the properties, relationships and tagged values that will be kept and discarded after the conversion. The data that will be discarded is put on the left hand side while the resolvable data is on the right. Click **OK** at bottom right.
5. Note that the undo history will be cleared after the conversion, meaning that you cannot perform any undo action after it. Click **Yes** to confirm the change and to return to the diagram.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Selecting contained shapes with InstantFreeze

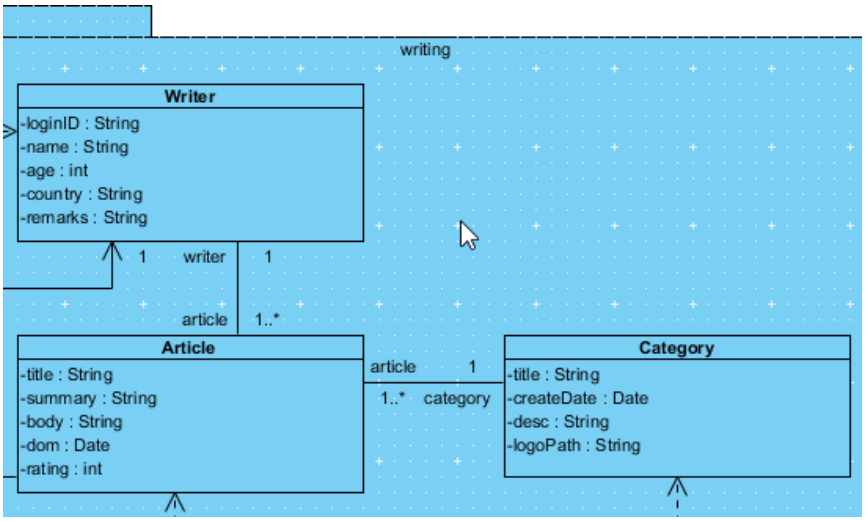
Imagine if there is a big, big container shape, such as a package shape, on a diagram. The shape is so big that spans over the viewing range. Inside the container shape there are few child shapes and now, you want to move these shapes to somewhere else. You attempted to perform a ranged shape selection first, but when you press on the container shape and start dragging, the package moves, following your mouse pointer movement. Here, in order to "tell" the application that you want not to move the shape in background, but to select shapes in foreground, use InstantFreeze.

InstantFreeze is a diagramming technique that allows you to temporarily freeze a container shape. By freezing a container shape, you cannot move and edit it until unfreeze. This allows you to select shapes inside the container shape as if the container shape does not exist.

Using InstantFreeze

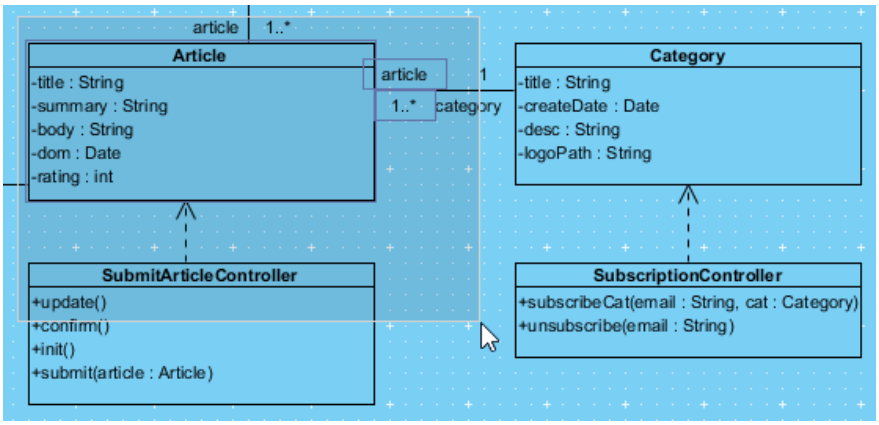
To temporarily freeze a container shape with InstantFreeze:

1. Move your mouse pointer over the container shape that you want to freeze.
2. Press twice. Note that it is NOT a double click. Instead, it is two separate presses, with a delay in between. When success, you can see the container shape painted with a grid background.



Using InstantFreeze

3. Now, you can select shapes freely inside the container shape, without worrying about moving the container shape in background.



Selecting shapes

4. When you finish, unfreeze the container shape either by pressing the **Esc** key or by clicking once outside the container shape.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Annotations and freehand shapes

There are some types of shape which let you annotate shapes on diagrams. They include text annotation, callout and freehand shape.

Text annotation

A standard BPMN notation which can be added to diagram, and connect to shapes with association for annotation purpose.

Callout shape

A speech-bubble like shape made for annotation purpose. It's pointer can be adjusted to point to specific position.

Freehand shape

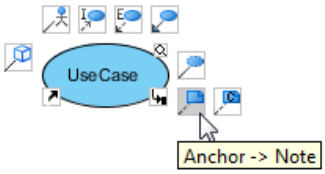
A set of shapes that can be bended to different styles to fit in different ways of annotation.

Note shape

In Agilian, notes can be created and attached to a shape via an anchor connector. The main advantage of using notes is annotating a specific model element on the diagram with normal text or HTML text, or to define OCL.

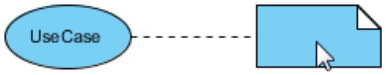
Creating a note through resource centric

1. Move the mouse over a shape and select a note resource icon.



Move the mouse over a shape

2. Drag it to your desired place on the diagram pane.



Drag the resource icon

3. Release the mouse. The shape is connected with the newly created note.

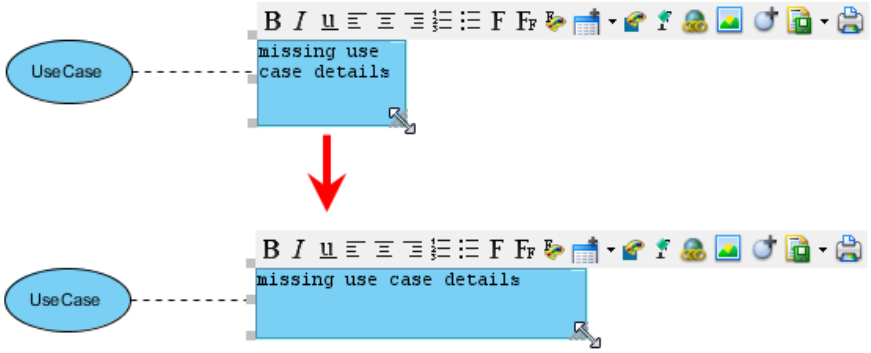
Note is created

Editing note content

1. Double click the note to start editing its content. Note that the toolbar above the note can be used to format its content.

Start editing

2. Drag the bottom right corner of note to resize it.



Resize the note

3. Click on the diagram background to confirm editing the note.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

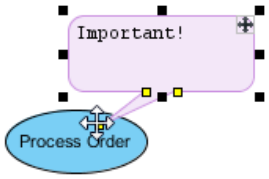
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Callout shape

Callout shape is a label that is used for explanation on your model elements. Inserting callout shape aims to draw others' attention and give them additional remarks. Basically, its function is similar to a photo caption or a comment. However, it does more than merely either a photo caption or a comment. In fact, it looks more likely to a dialog box that you can place it next to your model elements.

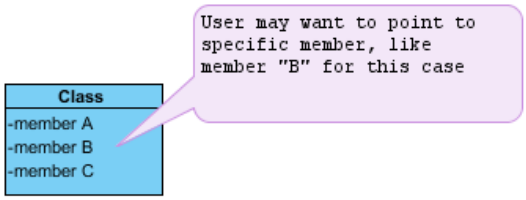
Adjusting the direction of callout shape pointer

1. A remark can be inserted by selecting **Callout** from the diagram toolbar and dragging it to the model elements directly on the diagram pane.
2. The direction of callout shape pointer can be adjusted by simply dragging the pointer's end.



Adjust the pointer

3. The pointer can be adjusted to point out a more specific position, such as pointing to the name of diagram element, or a specific class member (attribute/operation) out of a class.

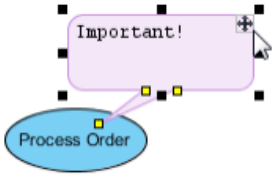


Adjust the pointer to a specific class member out of a class

> **NOTE:** You can adjust not only the pointer's direction, but also its length.

Repositioning the shape

The position of callout shape can be moved by dragging the **+** icon that is located on the top right corner of the callout. This icon is used to reposition the shape in a straight and simple way.



Reposition the shape

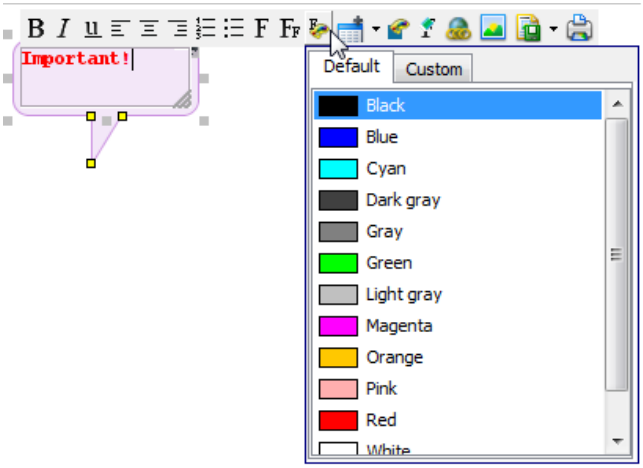
> **NOTE:** The callout shape will be moved but the position of its pointer won't be changed if you only drag the **+** icon. This helps you to remain the pointing and move the callout shape simultaneously.

Editing callout shape content

The content of callout shape can be edited by double clicking on the callout shape.

Emphasizing the content of callout shape

Formatting buttons can be used to insert formatting for the text in order emphasize your key points while you are editing the content, such as changing font color. This helps to enhance the visual effect of the text as well.



Format the text

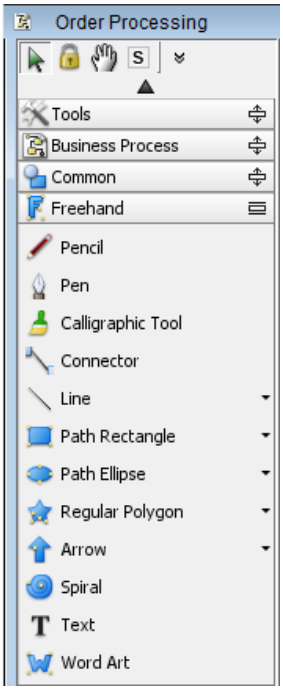
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

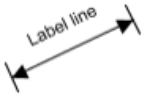
Freehand shapes

Freehand shapes are multi-functional features in Agilian. With freehand shapes, you can not only draw various kinds of shapes, but also insert an annotation. If you want to stress an important message with visual effect, you may use word art rather than using a note or a callout because you can reshape the text in word art, however, not in a note or a callout. Moreover, the shapes in freehand are flexible that you can twist them freely, but you can only resize a note or a callout.



Freehand toolbar interface

Summary of freehand shapes

Shape Type	Sample
Pencil	
Pen	
Calligraphic Tool	
Connector	
Line	
Label Line	
Path Rectangle	
Rectangle	

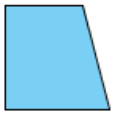
Round Rectangle

Round Rectangle 2

Diamond

Parallelogram

Trapezoid



Isosceles Trapezoid

Path Ellipse

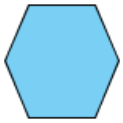
Ellipse

Arc

Chord

Pie

Regular Polygon

Isosceles Triangle	
Hexagon	
Arrow	
Two Head Arrow	
Spiral	
Text	
Word Art	

Summary of freehand shape

NOTE: Freehand can be switched on by clicking on diagram toolbar and select **Category > Freehand** from the pop-up menu.

Drawing free style path with pencil

1. Press on the empty space on diagram pane and drag to form the outline.

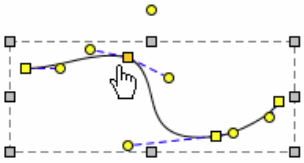
Draw with pencil

2. Release the mouse and new freehand shape will then be created.

Activating the fine editing selector

Fine editing selector shows a second later after the freehand shape is being selected. To show it immediately, press keyboard ' **N**' key.

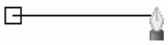
Press on a yellow selector for selecting and the selected selector will turn into orange. More fine editing selectors will appear for curve adjustment.



Selected fine editing selector

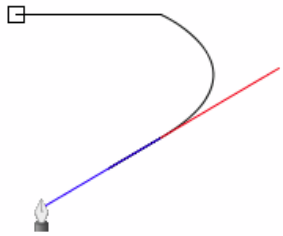
Drawing curve with pen

1. Click on empty space on diagram pane and drag it to create the first stroke.



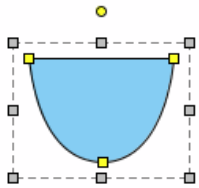
Straight line created

2. To create a curve, press and move the mouse. The indication line will appear. Release the mouse button when finishing editing. On the other hand, the last stroke can be cancelled by right clicking on the diagram.



Create curve

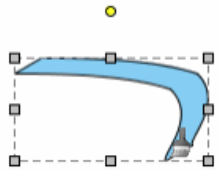
3. To confirm editing of the freehand shape, double click on diagram and a new freehand shape will be created. If the point returns to the starting point, it will form a closed path.



A close path

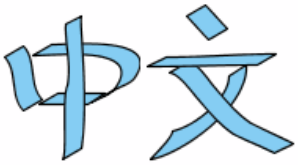
Drawing calligraphic path with calligraphic tool

1. Press on the diagram and drag to form the outline of shape. Release the mouse to create the shape.



Freehand shape created

2. By combining several other calligraphic shapes, you can create a complete diagram.



Calligraphy example

Draw straight and curve line with connector

1. Press on a source shape and drag it to the destination shape.

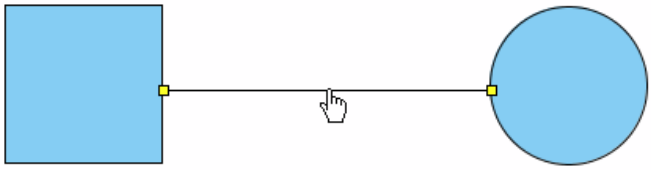
Connect shapes

2. Release the mouse and a new connector will be created between them.

A line is created

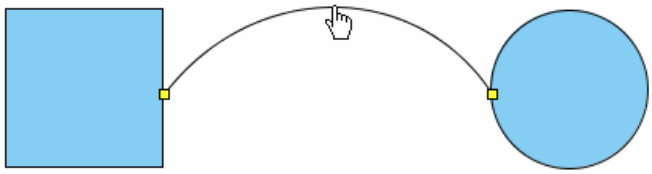
Bend a straight connector into a curve

1. Press on a straight connector.



Click on straight line

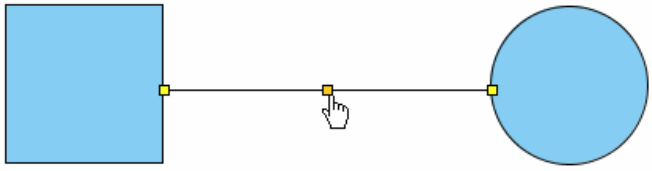
2. Bend it to your preferred direction and it will become a curve connector.



A curve connector

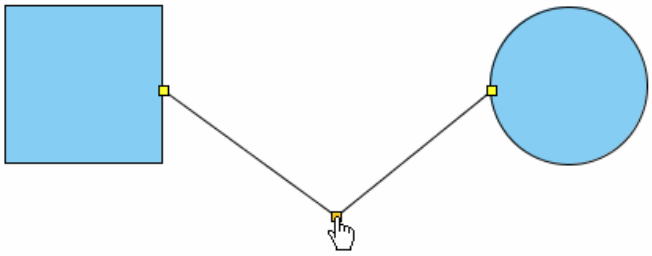
Split a straight connector

1. Press the **Ctrl** key.
2. Click on the specified location to split. A new point at where you have clicked will turn into orange.



Splitting line

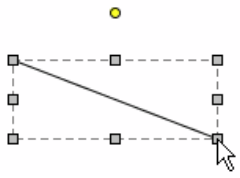
3. Drag on that point to split the line.



Move mid point

Drawing straight and curved line

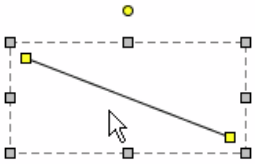
1. Press on the diagram pane and drag to form the outline.
2. Release the mouse button and a straight line will be created.



Line created

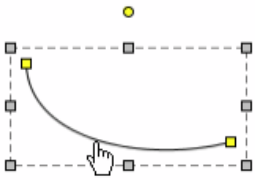
Bend a straight line into a curve

1. Select a straight line for a second to wait for the fine editing selectors popping out.



Show fine editing selector

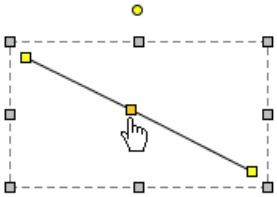
2. Press on the straight line. Drag it to bend into your preferred direction.



Drag line as curve

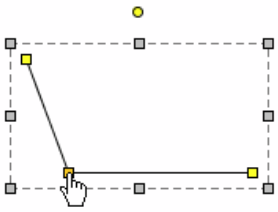
Split straight line

1. Press the **Ctrl** key.
2. Click on the specified location to split. A new point at where you have clicked will turn into orange.



Splitting line

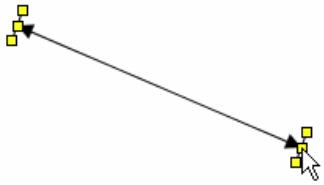
3. Drag on that point to split the line.



Move mid point

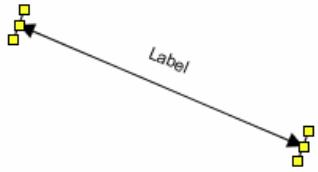
Drawing labelled line

1. Press on the diagram and starting dragging to form its outline.
2. Release the mouse button to create the labelled line.



Freehand shape created

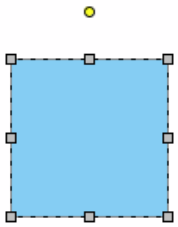
3. Double click on the line. Enter the name for the line.
4. Press **Enter** to confirm editing.



Label Line

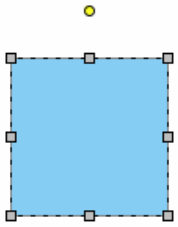
5. You may drag the yellow selector to modify the line's outline.

Drawing rectangle
Click on the diagram to create a rectangle.



Freehand shape created

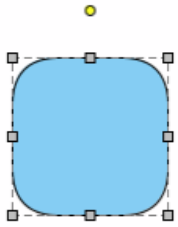
Drawing path rectangle
Click on the diagram to create a path rectangle.



Freehand shape created

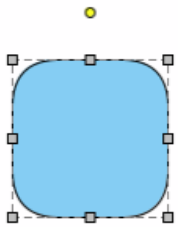
What's the difference between rectangle and path rectangle?
Path rectangle is formed by path, which enables you to freely reshape it, while rectangle always keeps shape as a rectangle.

Drawing rounded rectangle
Click on the diagram to create a rounded rectangle.



Freehand shape created

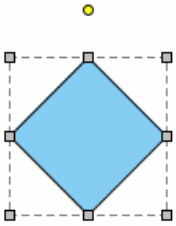
Drawing rounded rectangle 2
Click on the diagram to create a rounded rectangle 2.



Freehand shape created

What's the difference between rounded rectangle and rounded rectangle 2?
Rounded rectangle uses a single control point to control the deepness of corner, which ensures that the four corners remain consistent while rounded rectangle 2 uses two points to control the deepness of corner, which can produce irregular corners.

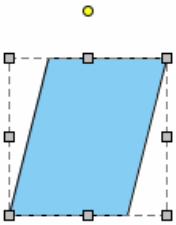
Drawing diamond
Click on the diagram to create a diamond shape.



Freehand shape created

Drawing parallelogram

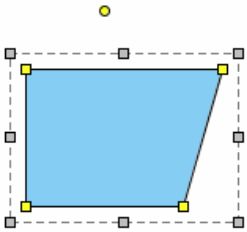
Click on the diagram to create a parallelogram shape.



Freehand shape created

Drawing trapezoid

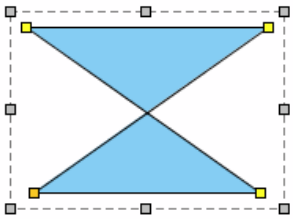
1. Click on the diagram to create a trapezoid shape.
2. You can adjust the slope by dragging the fine editing selectors in yellow.



Other trapezoid outline

Drawing isosceles trapezoid

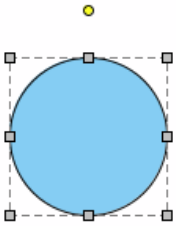
1. Click on the diagram to create an isosceles trapezoid shape.
2. You can reshape the Isosceles Trapezoid by dragging the fine editing selectors in yellow.



Other isosceles trapezoid outline

Drawing ellipse

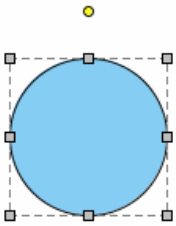
Click on the diagram to create an ellipse shape.



Freehand shape created

Drawing path ellipse

Click on the diagram to create a path ellipse shape.



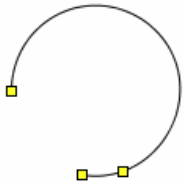
Freehand shape created

What's the difference between ellipse and path ellipse?

Path ellipse is formed by path, which enables you to freely reshape it while ellipse always keeps shape as an oval.

Drawing arc

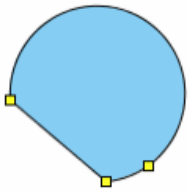
1. Click on the diagram to create an arc shape.
2. You can extend the line by dragging on the fine editing selectors in yellow.



Other arc outline

Drawing chord

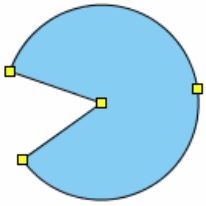
1. Click on the diagram to create a chord shape.
2. You can extend the arc of chord by dragging on the fine editing selectors in yellow.



Other Chord outline

Drawing pie

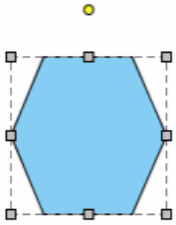
1. Click on the diagram to create a pie shape.
2. You can extend the arc of pie by dragging on the fine editing selectors in yellow.



Other pie outline

Drawing hexagon

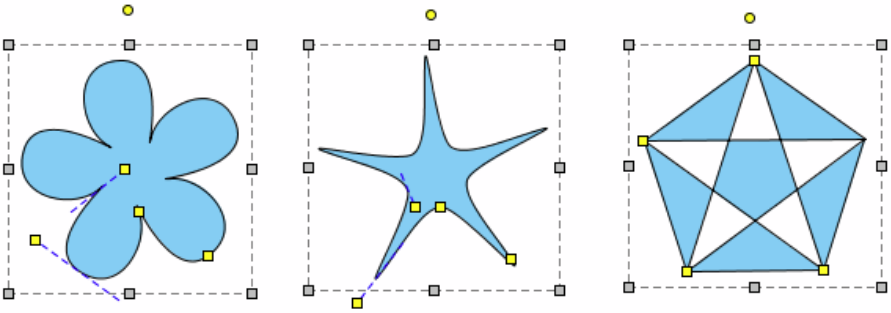
Click on the diagram to create a hexagon shape.



Freehand shape created

Drawing regular polygon

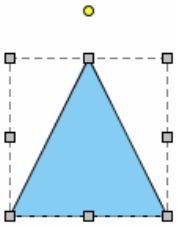
1. Click on the diagram to create a regular polygon shape.
2. You can modify the outline of shape by dragging the fine editor selectors in yellow.



Other regular polygon outline

Drawing isosceles triangle

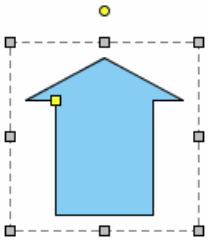
Click on the diagram to create an isosceles triangle shape.



Freehand shape created

Drawing single head arrow

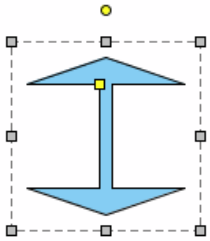
1. Click on the diagram to create an arrow shape.
2. You can reshape it by dragging the fine editing selectors in yellow.



Other Arrow Outline

Drawing two head arrow

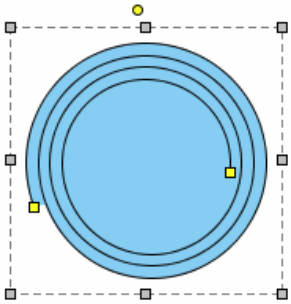
1. Click on the diagram to create a two head arrow shape.
2. You can reshape it by dragging the fine editing selectors in yellow.



Other two head arrow outline

Drawing spiral

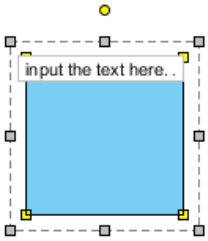
1. Click on the diagram to create a spiral shape.
2. You can reshape it by dragging the fine editing selectors in yellow.



Other Spiral outline

Inserting text

1. Click on the diagram to create a text shape, and input the text. You can press **Enter** to insert line break.



Input the text in text shape

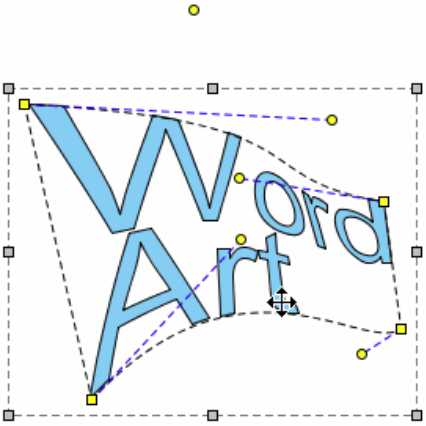
2. You can click **Ctrl** while pressing **Enter** to confirm editing.

Inserting word art

1. Click on the diagram to create a word art shape, and input the text. You can press **Enter** to insert line break.
2. You can click **Ctrl** while pressing **Enter** to confirm editor.

Freehand shape created

3. You can reshape it by dragging the fine editing selectors in yellow.



Edit word art

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Resource referencing

To include more information you can link to both internal and external material as reference. In this chapter, you will see how to refernece to shape, diagram, external file, folder and URL, as well as how to add sub-diagram.

Reference to external resources

File, folder and URL can be attached to a shape as references. This page will teach you how to do.

Reference to diagrams and shapes

In addition to external reference, internal reference can be added, too. You will learn how to refernece to another shape and diagram.

Elaborating model element with sub-diagram

Sub-diagram helps you describe a model element in detail by making use of a separate diagram.

Showing sub-diagrams and reference indicators

To indicate that a shape has sub-diagram or reference added, you can show the indicators on diagram. The indicators will show in exported image, too.

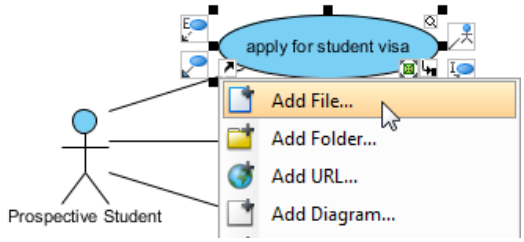
Reference to external resources

Additional references can be attached to a shape through resource icon **References**, such as inserting a file, a folder and a URL. After that, you can open and view the inserted references through resource icon **References**.

Adding external resources

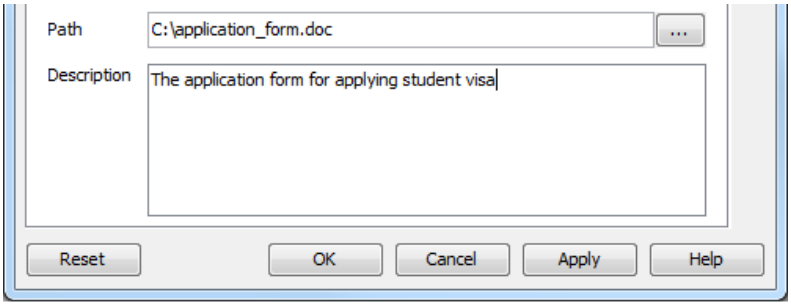
Adding a file reference

1. Move the mouse over a shape to add reference, click the resource icon **References** and select **Add File...** from the pop-up menu.



Add a file to shape

2. When the specification dialog box pops out, enter the referenced file path or select the file path by clicking **...** button. You may also enter the description for the file in **Description**.

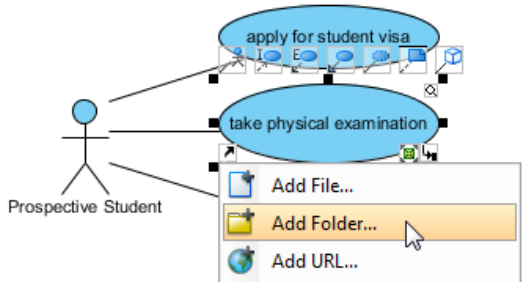


Enter file reference path

3. Click **OK** button to confirm the creation for file.

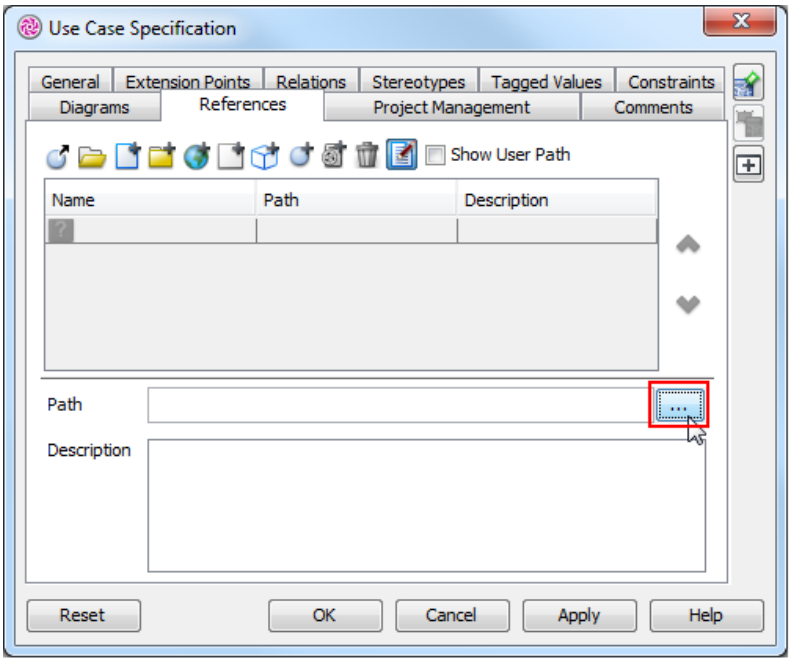
Adding a folder reference

1. Move the mouse over a shape to add reference, click the resource icon **References** and select **Add Folder...** from the pop-up menu.



Add a folder to shape

2. When the specification dialog box pops out, click the **...** button in **Path** to select a folder path.

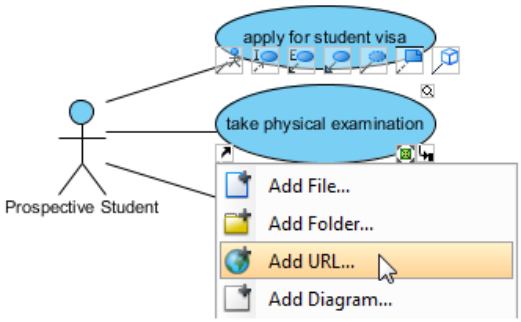


*Click the **...** button to select a folder*

3. When **Select Directory** dialog box pops out, select a folder to be referenced and click **OK** button.
4. You may enter the description for the folder. Finally, click **OK** button to confirm the creation for folder.

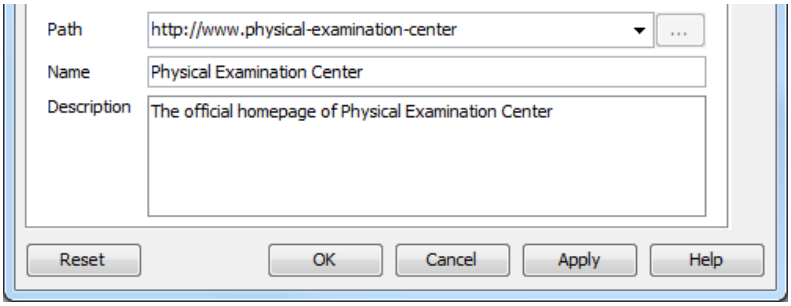
Adding a URL reference

1. Move the mouse over a shape to add reference, click the resource icon **References** and select **Add URL...** from the pop-up menu.



Add URL to shape

2. When the specification dialog box pops out, enter the URL path. You may also enter the URL name and the description for the URL.



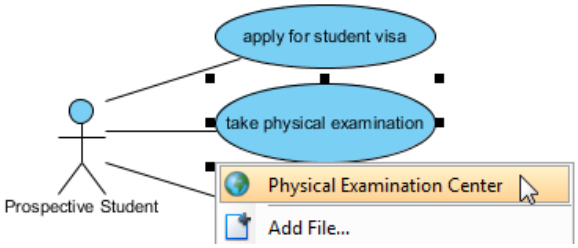
Enter URL path

3. Click **OK** button to confirm the creation for URL.

Opening external resources

Move the mouse over a shape to open reference, click the resource icon **References** and select an external resource from the pop-up menu.

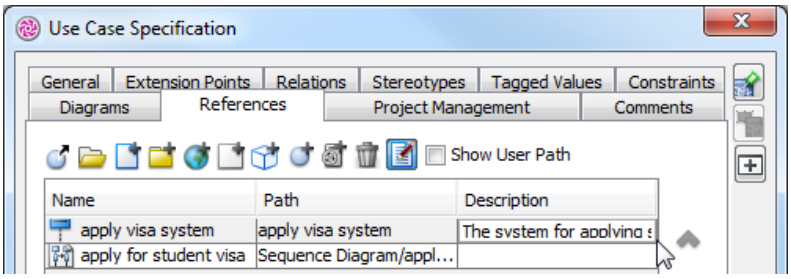
If you select a URL reference to open, it will be opened by default web browser. If you select a file reference to open, it will be opened by your system with the program used to open this kind of file. If you select a folder reference to open, it will be opened by your system automatically.



Open a URL reference

Editing references

1. Move the mouse over a shape which you want to edit its reference information and press it resource icon **References** and select **Edit References...** from the pop-up menu.
2. When the specification dialog box pops out, double click on the row of reference you want to enter its description or modify it.
3. Enter the description or modify it under **Description** column.

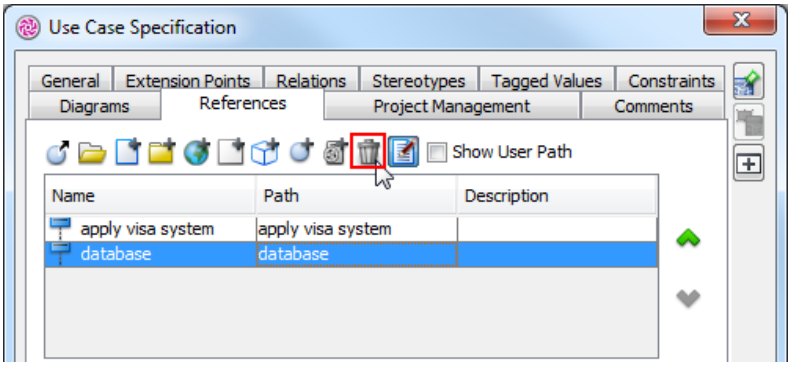


Enter the description

4. Finally, click **Enter** button to confirm editing.

Removing a reference

1. Move the mouse over a shape which you want to remove its reference, click its resource icon **References** and select **Edit References...** from the pop-up menu.
2. When the specification dialog box pops out, select a reference to be removed on the list and press **Remove** button to delete the selected reference.



Select a reference to remove

3. Finally, click **OK** button to confirm the reference removal.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

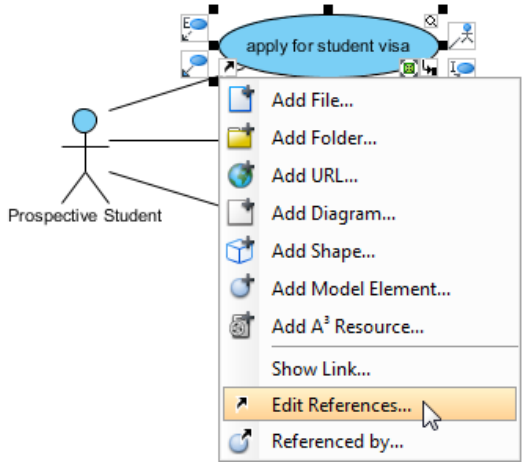
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Reference to diagrams and shapes

Additional references can be attached to a shape through resource icon **References**, for example, inserting a diagram and a shape. After that, you can open and view the inserted references through resource icon **References**.

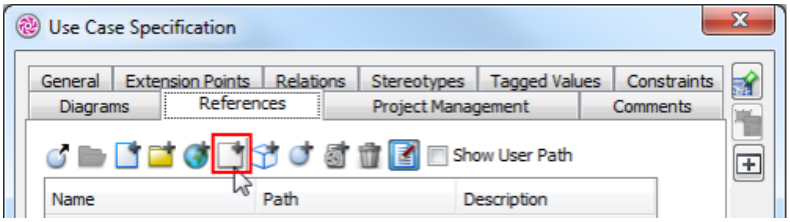
Reference to diagrams

1. Move the mouse over a shape, press its resource icon **References** and select **Edit References...** from the pop-up menu.



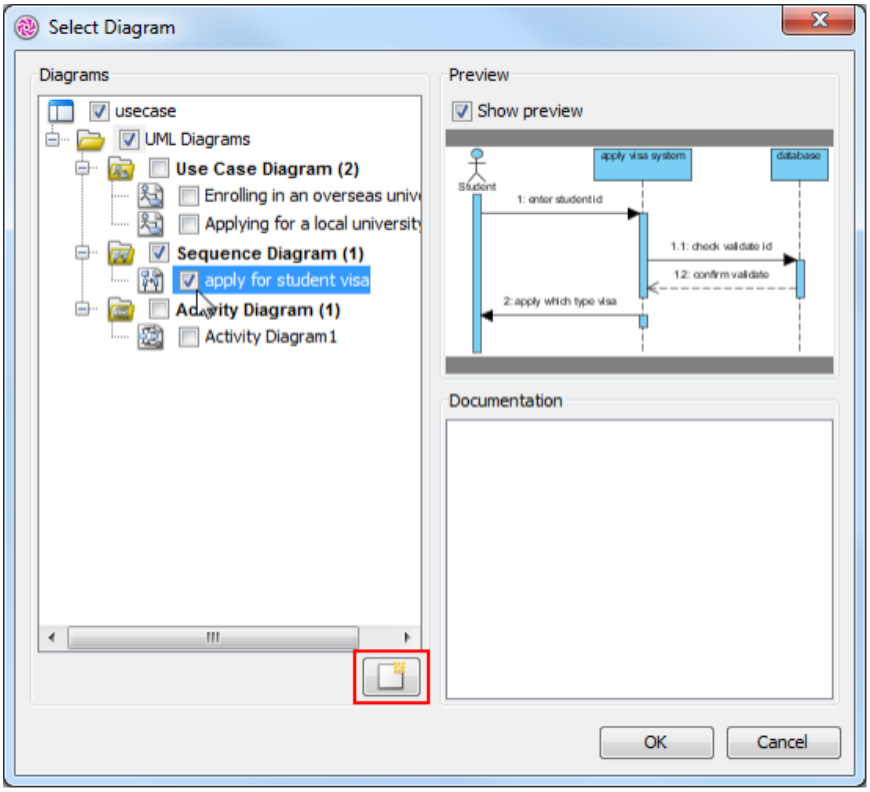
*Select **Edit References** to edit shape*

2. When the specification dialog box pops up, click **Add Diagram...** button.



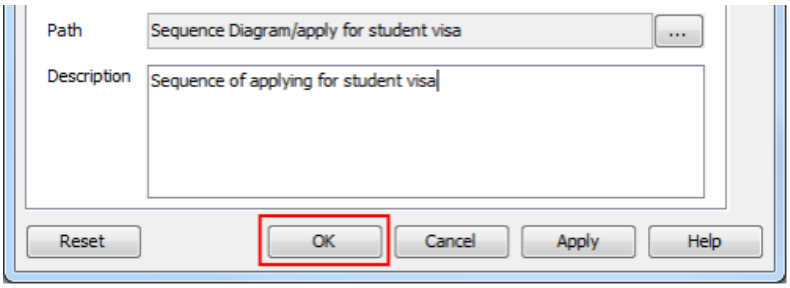
*Click **Add Diagram...** to add a diagram*

3. In **Select Diagram** dialog box, check an existing diagram or create a new diagram by clicking the icon under **Diagrams**.



Check an existing diagram

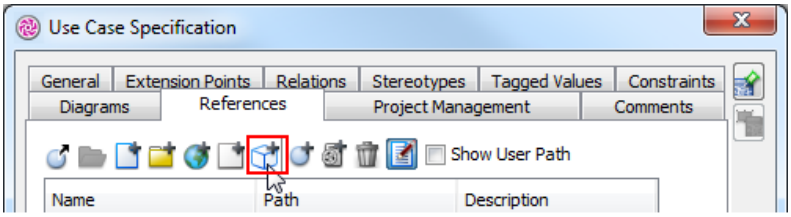
4. You may enter the description for the diagram in the specification dialog box. Finally, click **OK** button to confirm the reference creation.



Enter description

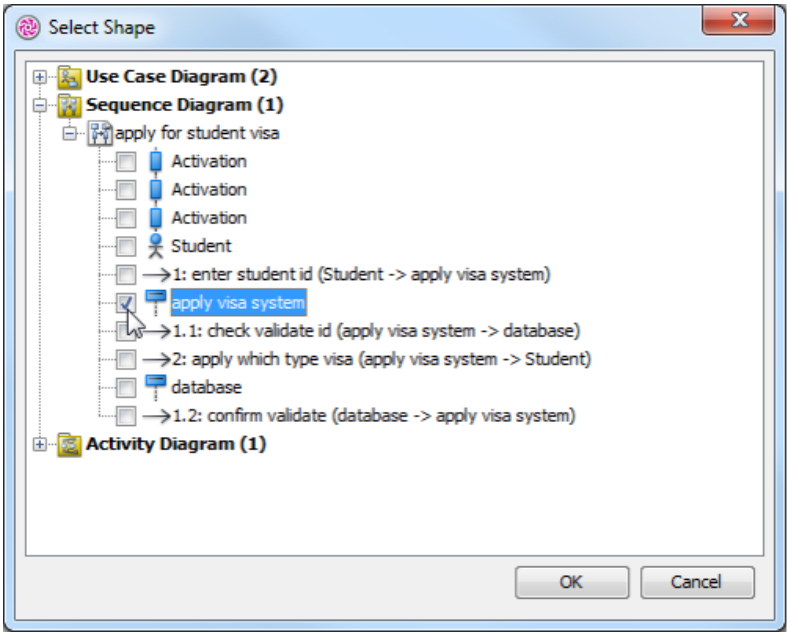
Reference to shapes

1. Move the mouse over a shape, press its resource icon **References** and select **Edit References...** from the pop-up menu.
2. When the specification dialog box pops up, click **Add Shape...** button.



*Click **Add Shape...** to add a shape*

3. In **Select Shape** dialog box, check a shape to be referenced and click **OK** button.

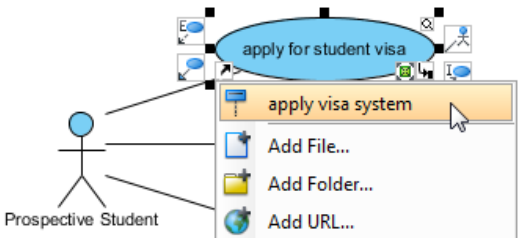


Check a shape

4. Enter the description for the shape in the specification dialog box. Finally, click **OK** button to confirm the creation of shape.

Opening a reference

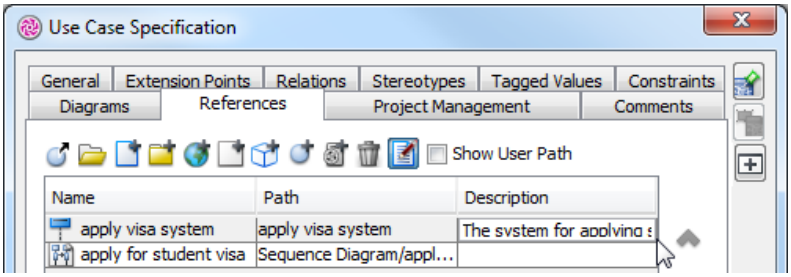
Move the mouse over a shape, press its resource icon **References** and select a shape/ a diagram to open from the pop-up menu. If you select a shape to open, it will switch to the diagram where the shape belongs to and the shape will be selected by filled-selector. If you select a diagram to open, it will switch to the selected diagram immediately.



Open a shape reference

Editing references

1. Move the mouse over a shape and press it resource icon **References** and select **Edit References...** from the pop-up menu.
2. When the specification dialog box pops out, double click on the row of reference you want to enter its description or modify it.
3. Enter the description or modify it under **Description** column.

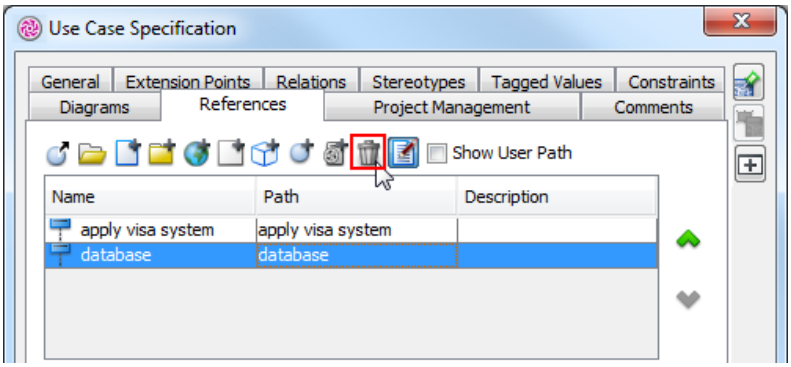


Enter the description

4. Finally, click Enter button to confirm editing.

Removing a reference

1. Move the mouse over a shape which has references, click its resource icon **References** and select **Edit References...** from the pop-up menu.
2. When the specification dialog box pops out, select a reference to be removed on the list and press **Remove** button to delete the selected reference.



Select a reference to remove

3. Finally, click **OK** button to confirm the reference removal.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

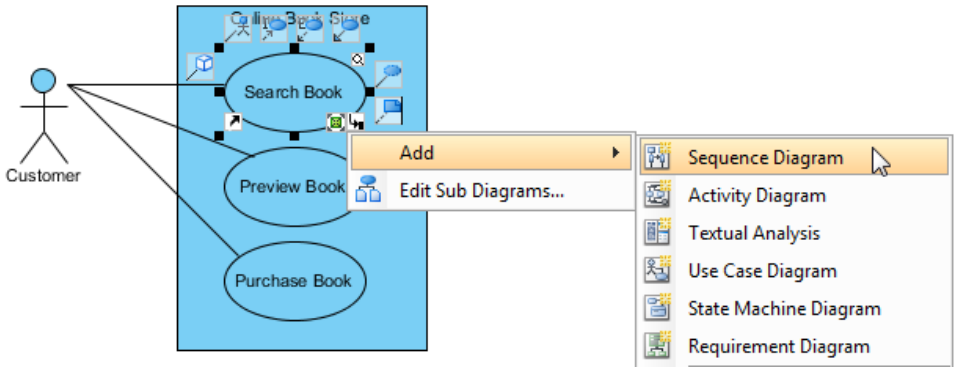
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Elaborating model element with sub diagram

Each notation has its own meaning and semantic. For example, you use a use case to present users' goals (system functions) but not how to achieve the goals. In order to model other aspects like the dynamic behavior of use case, you can elaborate a model element with a proper type of sub-diagram and contribute the details on the model element.

Creating a new sub diagram

To create a sub diagram to a shape, move the mouse over the target shape, press its resource icon **Sub Diagrams** when it reveals and select **Add** and then your preferred type of diagrams from the pop-up menu.

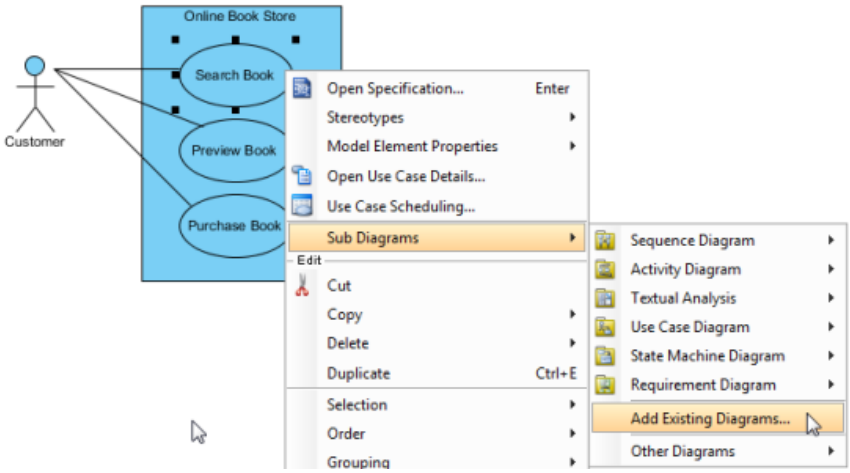


Add a sub diagram

NOTE: Inserting a sub diagram on a model element, all child model elements of the sub diagram will also be attached.

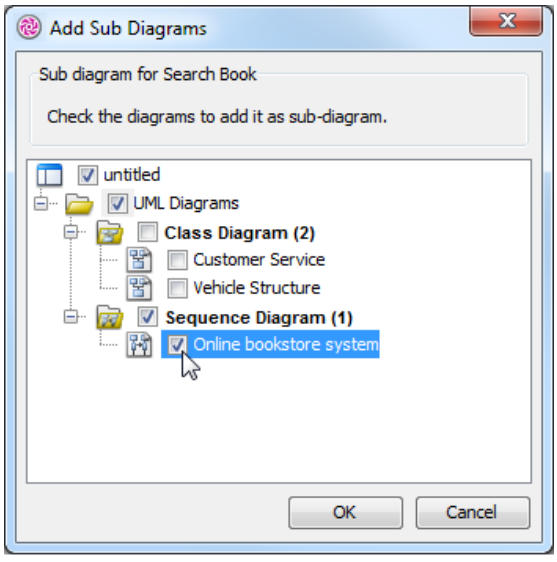
Adding an existing diagram as sub diagram

Right click on the target shape, select **Sub Diagrams > Add Existing Diagrams...** from the pop-up menu.



Add an existing diagram as sub diagram

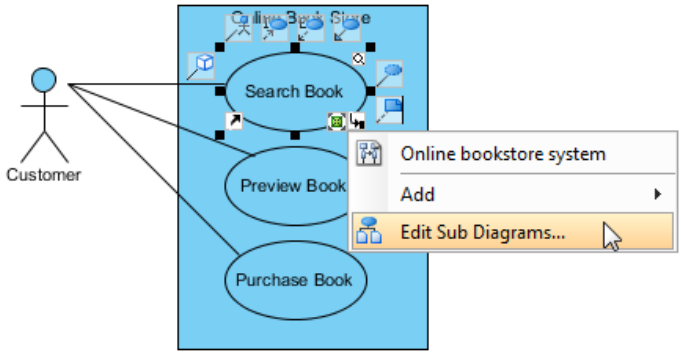
When the **Add Sub Diagrams** dialog box pops out, check a sub diagram and then click **OK** button.



Add Sub Diagrams dialog box

Removing sub diagram

1. Move the mouse over a shape, press its resource icon **Sub Diagrams** and select **Edit Sub Diagrams...** from the pop-up menu.



*Click **Edit Sub Diagrams...** from the pop-up menu*

2. In the specification dialog box, select a sub diagram that you want it to be removed from the list and click **Remove** button. When the **Confirm Remove** dialog box pops out, click **Yes** button to confirm the deletion.
3. Finally, click **OK** button to proceed.

Related Resources

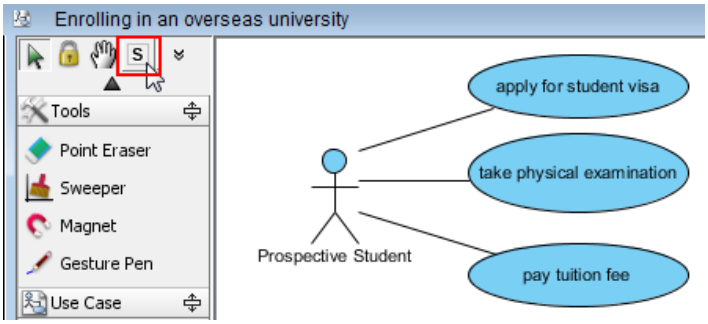
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Showing sub diagrams and reference indicators

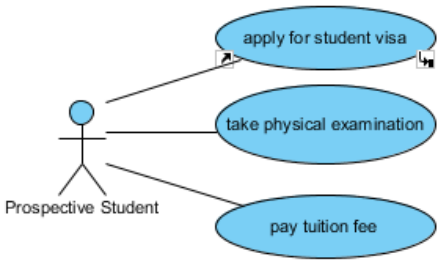
The indicator of sub-diagrams and references helps you identify which shapes in diagram have sub-diagrams and/ or attached with references. The indictor can be shown through the button on the top of diagram toolbar.

1. Click **Always Show Reference, Sub Diagram, Model Transitor and Documentation Resource** button on diagram toolbar.



*Click **Always Show Reference, Sub Diagram, Model Transitor and Documentation Resource** button*

2. As a result, references resource icon and sub-diagrams resource icon are shown.



References resource icon and sub-diagrams resource icon are shown

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Using shape editor

Shape editor is a diagramming tool for you to design your own notation (stencil). In this chapter, you will learn how to make use of shape editor to create your own shapes.

Creating shape in shape editor

Teaches you how to start shape editor and create a shape in it.

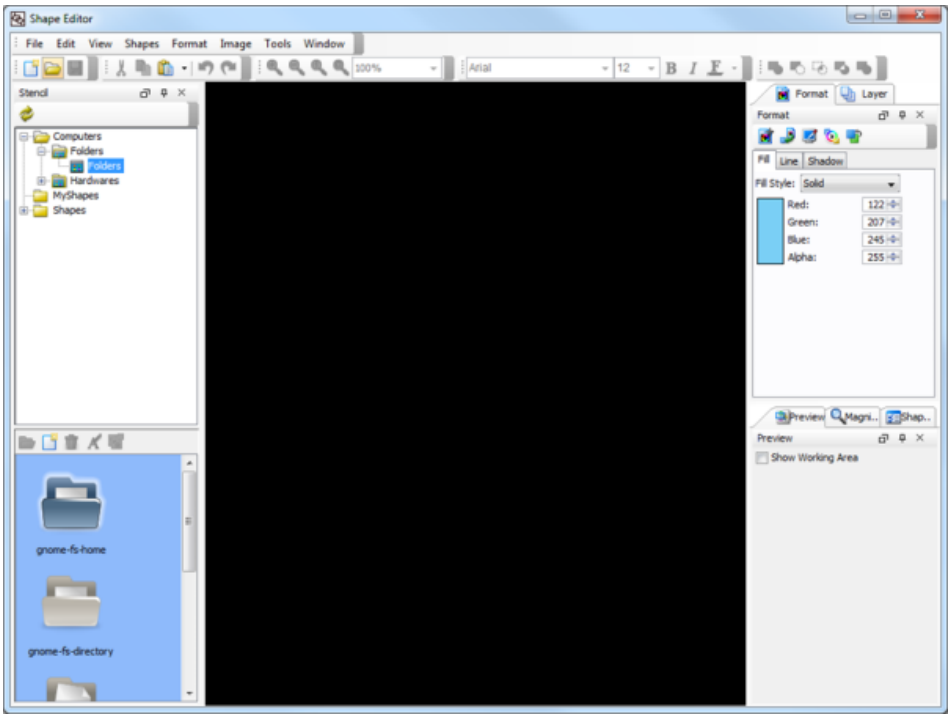
Creating shapes from stencil pane

Shows you how to create a shape from a stencil.

Creating shape in shape editor

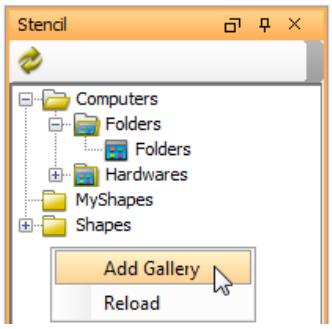
Although Agilian has a well-established notation, sometimes it is still not rich enough to express domain specific idea. Shape editor is a diagramming tool for you to design your own notation (stencil). Notations created can be incorporated into diagrams in Agilian.

1. To launch shape editor, select **Tools > Shape Editor...** from the main menu.
2. When shape editor unfolds, you can create a shape by creating a gallery in advance. It is because a shape needs to be created under a stencil, while a stencil is put under a category of a gallery.



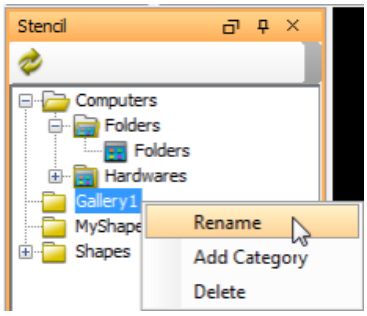
Shape Editor *dialog box*

3. A shape need to be created under a stencil, while a stencil is put under a category, under a gallery. To create a gallery, right click on the **Stencil** pane background and select **Add Gallery** from the pop-up menu.



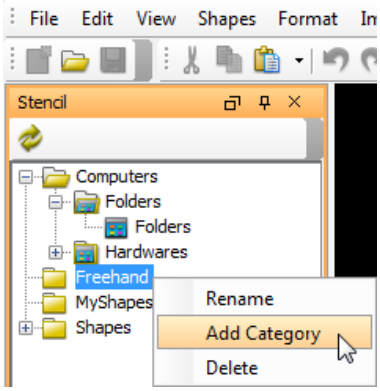
Add a gallery

4. To create a stencil, right click on a category and select **Add Stencil** from the pop-up menu. The newly created gallery is named as *Gallery1* by default. If you want to rename it, right click on it and select **Rename** from the pop-up menu. Enter your preferred gallery name in the pop-up **Input** dialog box and press **OK** button to confirm.



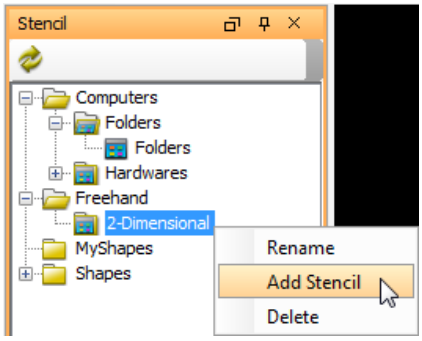
Rename a gallery

5. To create a category, right click on a gallery and select **Add Category** from the pop-up menu. Enter the category name in the pop-up **Input** dialog box and click **OK** button to confirm.



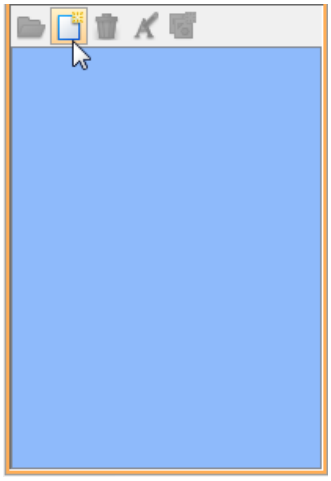
Add a category

6. To create a stencil, right-click on a category and select **Add Stencil** from the pop-up menu. The newly created stencil is named as *Stencil 1* by default. If you want to rename it, right click on it and select **Rename** from the pop-up menu. Enter your preferred stencil name in the pop-up **Input** dialog box and press **OK** button to confirm.



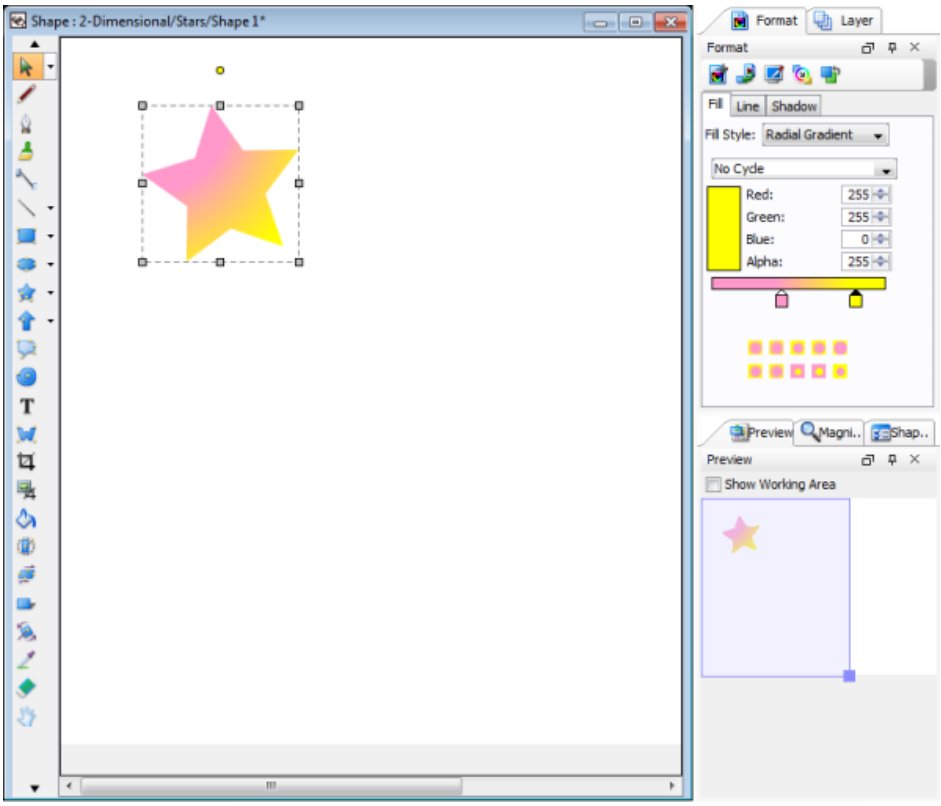
Add a stencil

7. To create a shape, click the **New Shape** button at the bottom part of the **Stencil** pane to create a blank drawing for drawing the shape.



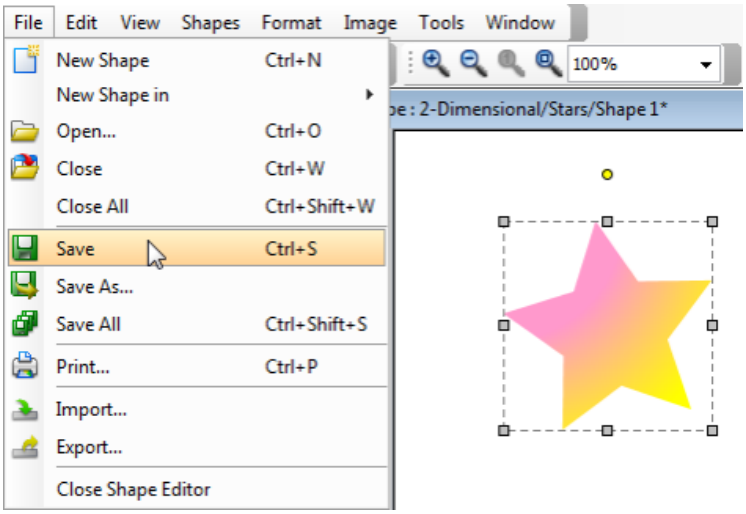
Add a shape

8. Create your preferred shape on the pane and you can also format its color in **Fill** tab of **Format** pane.



Create a shape on pane

9. To save the shape, select **File > Save** in the main menu.



Save the shape

Shapes created in shape editor can be used in Agilian. For details, please refer to the next page.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

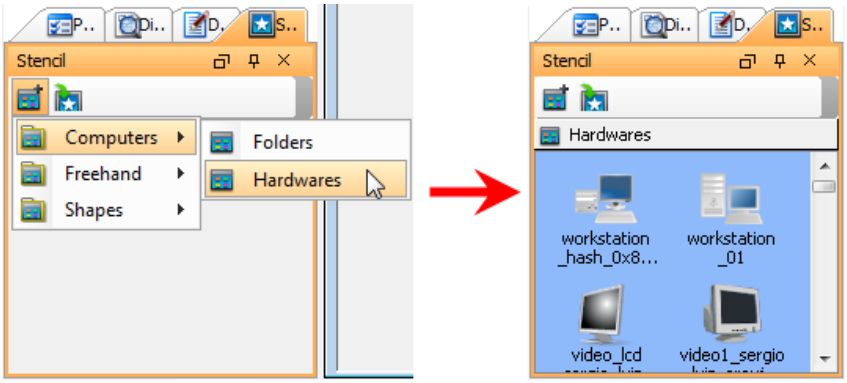
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Creating shapes from stencil pane

The Stencil pane is where user-created stencil shapes are stored. User can create a stencil shape on diagram by first displaying a stencil, dragging and dropping a shape from **Stencil** pane to diagram. Below are the steps in detail.

1. Let's create a shape from the **Stencil** pane which is resided at the panes at the bottom left of Agilian by default. If the **Stencil** pane is hidden, select **View > Panes > Stencil** from the main menu to reveal it.

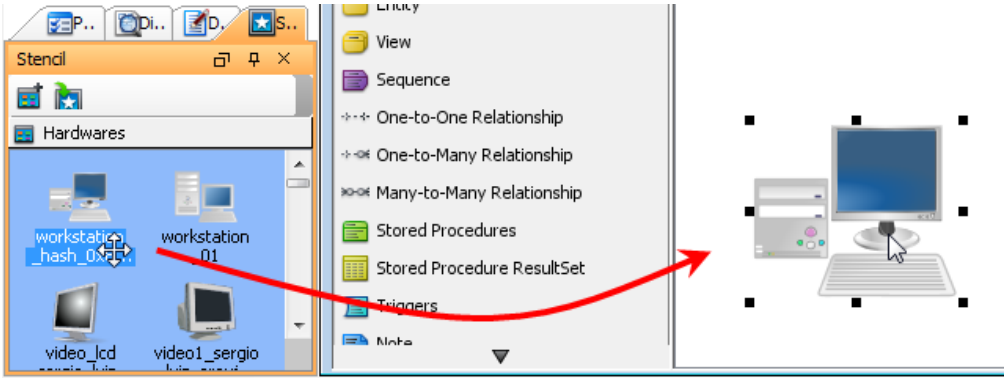
2. Click the **Add Stencil** button on the top of **Stencil** pane. Select a category from the pop-up list of gallery. Select the stencil to add. The stencil is then added to the **Stencil** pane.



*Select a category from **Stencil** pane*

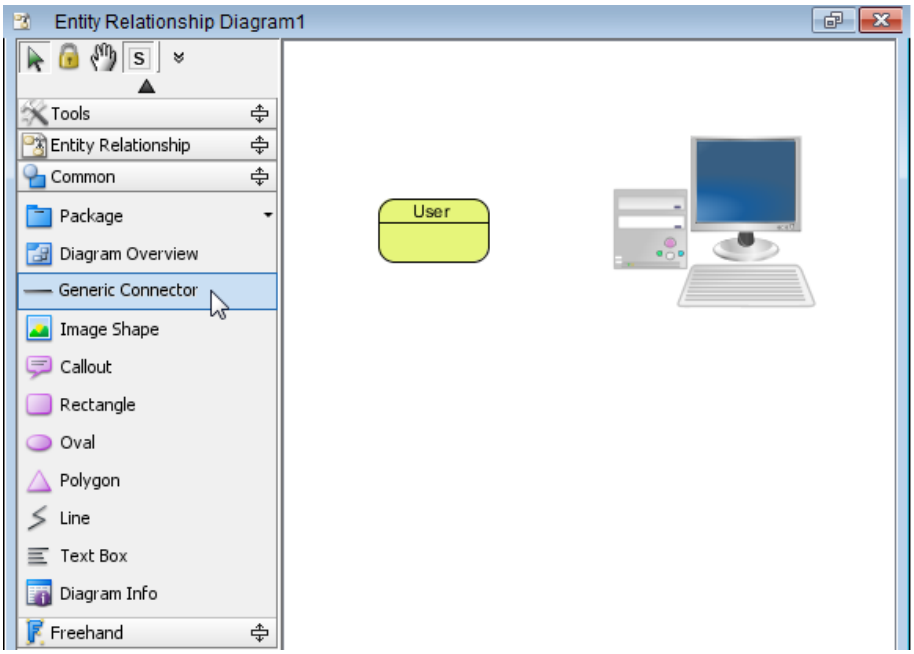
> **NOTE:** You can add multiple stencil by repeating this step.

3. When all shapes of the selected category is unfold on the **Stencil** pane, move the mouse over the target shape and drag it on the diagram pane.



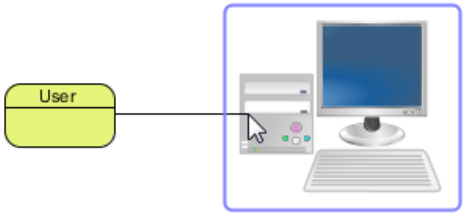
Drag a shape on the diagram pane

4. You can use generic connector to connect built-in notations shapes and stencil shapes. To do so, select **Generic Connector** from the diagram toolbar under the **Common** category.



*Select **Generic Connector** from diagram toolbar*

5. Move the mouse over the source shape, drag the connector from the source shape and release the mouse until it reaches on the target shape. As a result, two shapes are linked with connector.



Connect two shapes

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Customizing user interface

You can hide away certain menu/toolbar button, or to configure the font of user interface through user interface customization.

Hiding user interface components

Hide away certain menu/toolbar button to ignore functions that you are not interested to use.

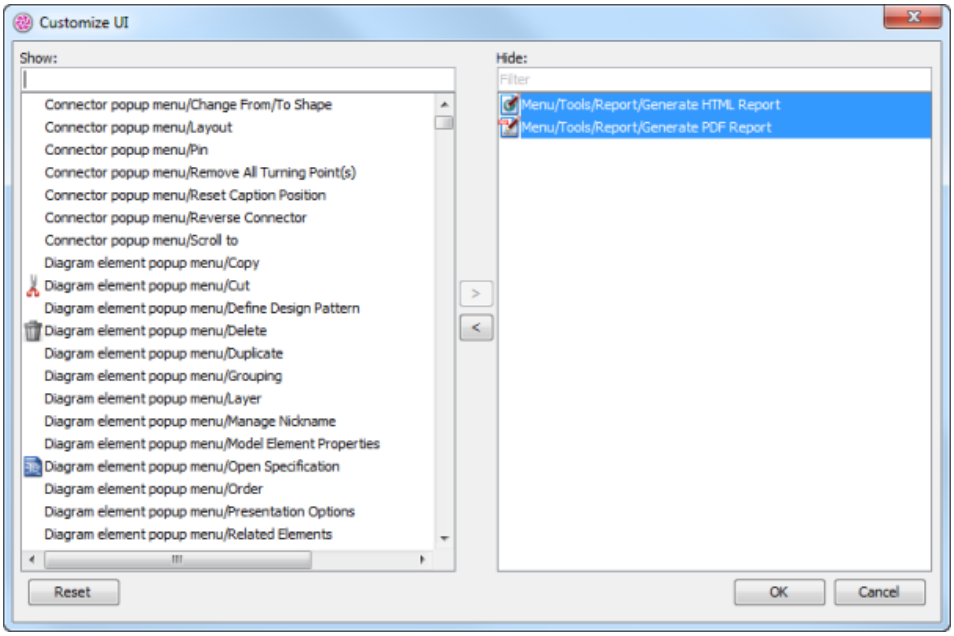
Adjusting user interface font

Adjust show the user interface with your favorite font.

Hiding user interface components

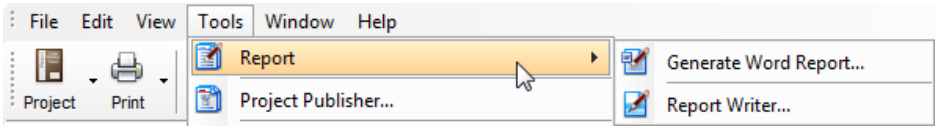
You may want to hide away some diagram types, menu items or toolbar items to avoid your team creating wrong types of model. This can be done by user interface (UI) customization.

1. Select **Help > Customize UI** from the main menu.
2. In the **Customize UI** dialog box, select the menu(s)/toolbar button(s)/popup menu(s) to hide, and click on the **>** button to hide them.



Select the menus to hide

3. Click **OK** to confirm. By restarting the application, the chosen user interface componenets will be hidden.



User interface components are hidden

Related Resources

The following resources may help you learn more about the topic discussed in this page.

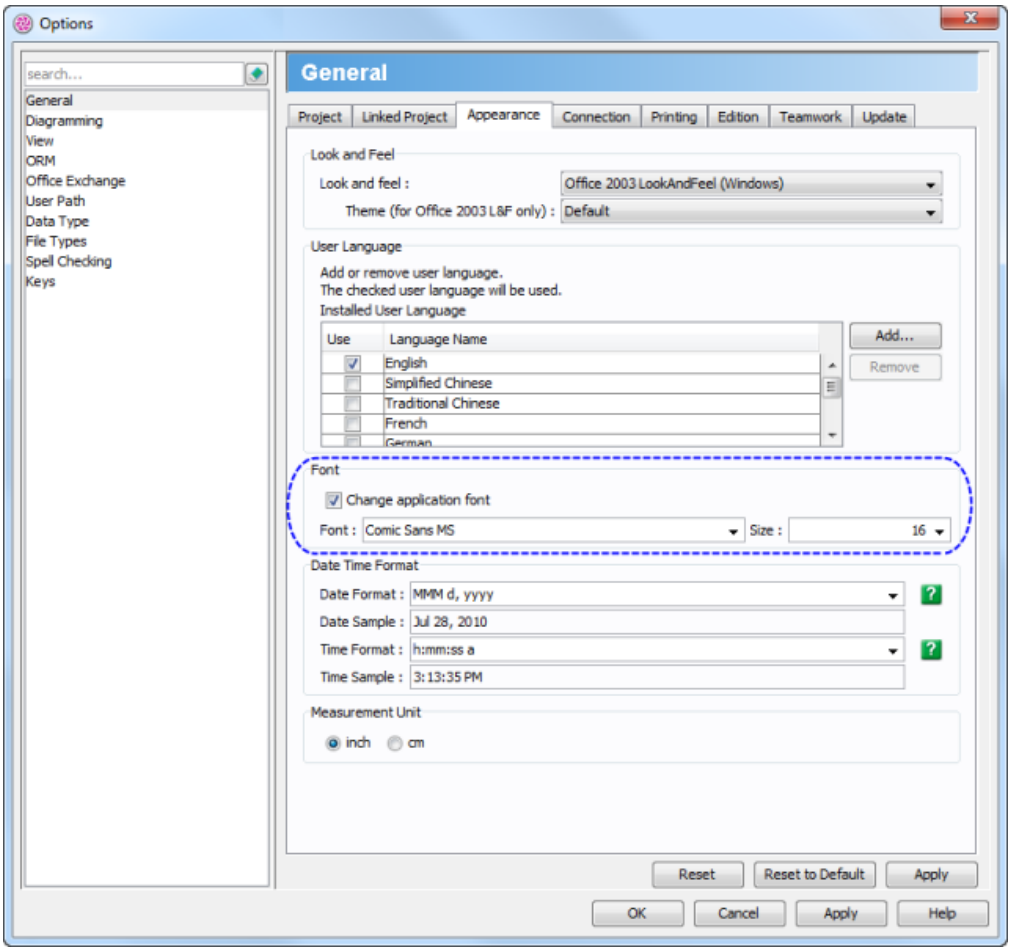
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Adjusting user interface font

Agilian runs with a screen design that is friendly enough for most users, so that you can customize it to make it suit your preference. One of the possible customization is to adjust the font settings for text appears on user interface, like the button caption for tools in toolbar, diagram editor tab's title, menus' captions, etc. The settings will be stored in workspace. Hence, you can keep the settings every time you run Agilian.

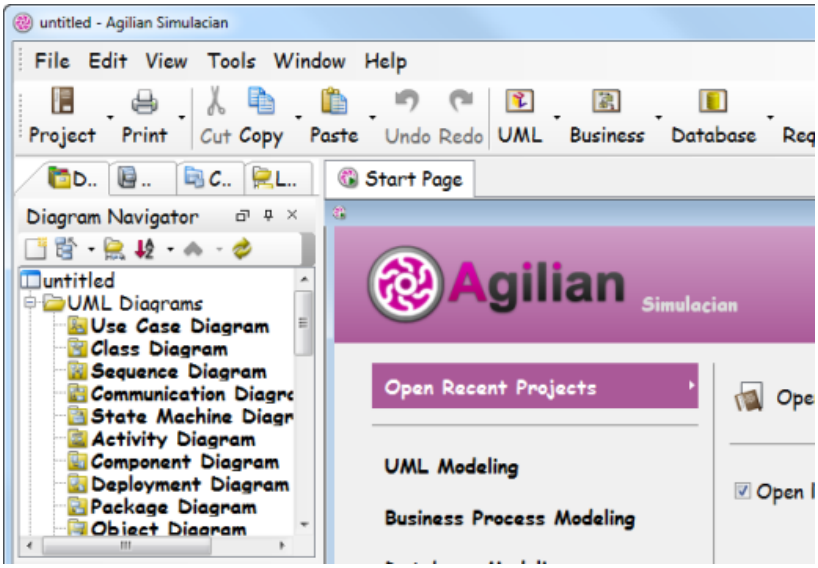
To adjust font settings :

1. Select **Tools > Application Options** from the main menu.
2. Select **General** from the list on the left hand side.
3. Open the **Appearance** page.
4. Check **Change application font** in the **Font** section. Adjust the font type and size.



Adjusting font settings

5. Click **OK**.
6. Restart the application to let the settings take effect.



Updated user interface

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Organizing works with model

A Model is a component that you can create in your project for organizing shapes and diagrams which acts like a folder. Modelers generally use models to differentiate stages or nature within project, such as an "as-is" model for storing diagrams and model elements about the current system, and a "to-be" model for recording blueprints of the system to be implemented. In this chapter, you will learn how to use model to organize your work.

Using model

Concepts about model will be discussed in this page.

Creating diagram under model

Shows to steps of creating model in Model Explorer.

Moving diagram to model

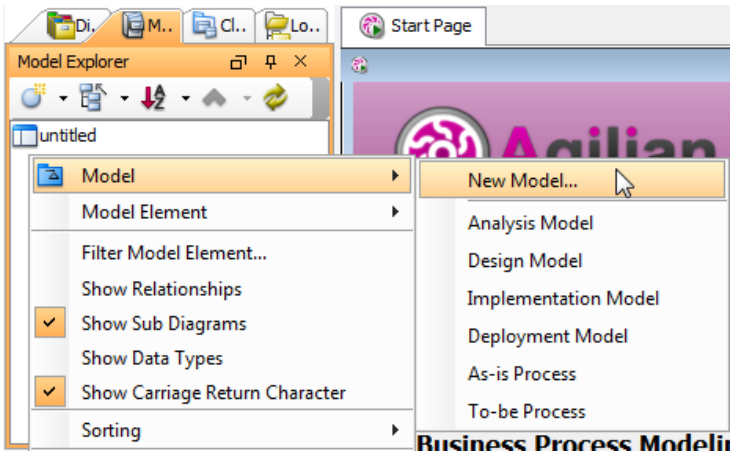
You can move a diagram to another move. This page shows you how to do.

Using model

A Model is a component that you can create in your project for organizing shapes and diagrams which acts like a folder. Modelers generally use models to differentiate stages or nature within project, such as an "as-is" model for storing diagrams and model elements about the current system, and a "to-be" model for recording blueprints of the system to be implemented. The use of model improves not only the structuring of work, but also the performance by reducing the number of root model elements to load. As a result, you can look up diagram or model element you needed easier.

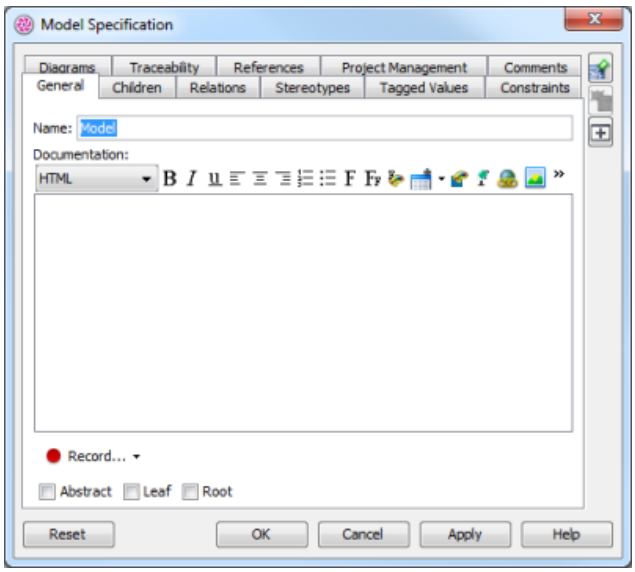
To create a model:

Right click on the **Model Explorer**'s background and select **Model** from the pop-up menu. You can either create a custom model by selecting **New Model...**, or create a pre-defined model (e.g. As-is Process model) by selecting it on the list.



Create a new model

Once the model is created under the project node on the Model Explorer, the **Model Specification** dialog box pops out. You may specify the model's details in the **Model Specification** dialog box. Note that if you create a pre-defined model, the **Model Specification** dialog box will not pop out.



***Model Specification** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

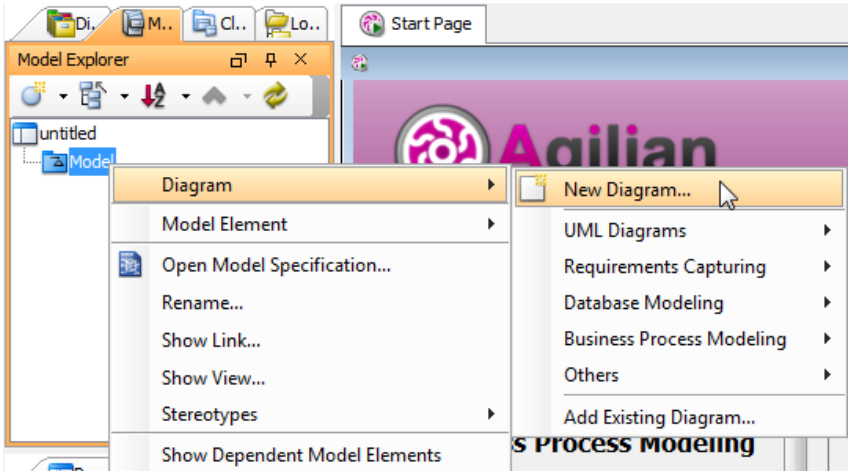
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Creating diagram under model

When a model has been created on the **Model Explorer**, you can start creating diagram(s) under the model. It is recommended to group diagrams using **Model** instead of laying them flat in the project. This can avoid accidentally loading diagrams and model elements that you never use, and thereby can speed up project loading and saving.

To create a diagram under model:

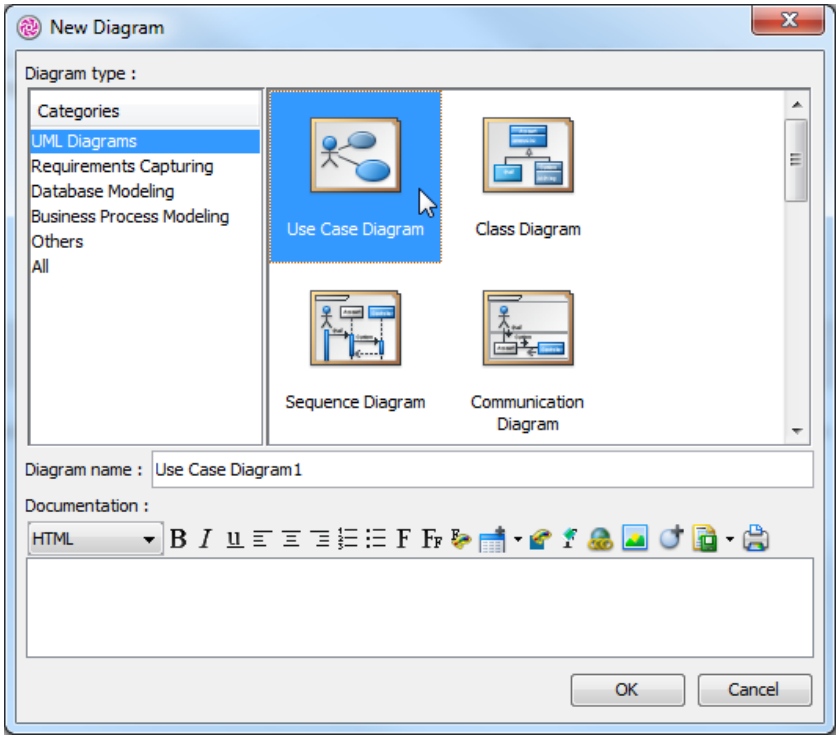
Right click on the target model and select **Diagram > New Diagram...** from the pop-up menu.



Create a new diagram under the model

NOTE: Alternatively, you can select a specific diagram under categories from the pop-up menu.

When the **New Diagram** dialog box pops out, select a specific category and the target diagram under the selected category. The new diagram name is *[Diagram Type]1* by default, you can rename it by entering in **Diagram name** text field. Click **OK** button to confirm and close the dialog box.



***New Diagram** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

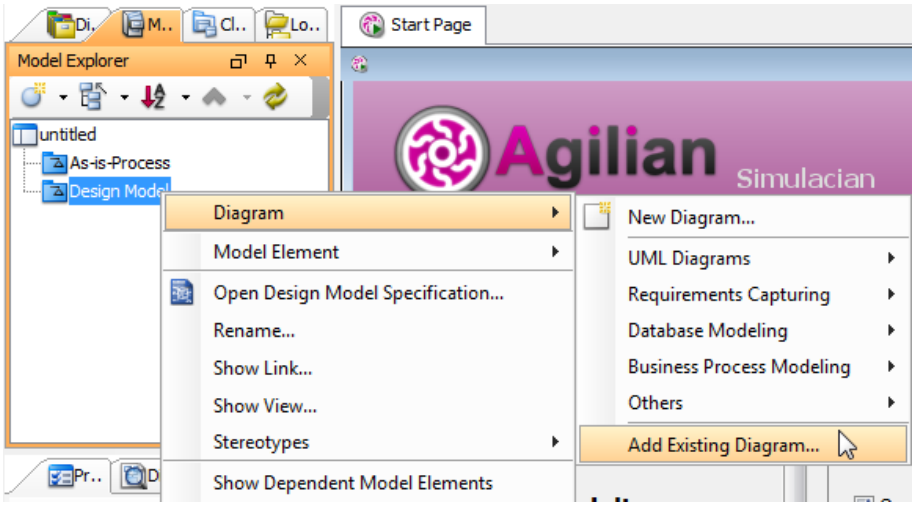
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Moving diagram to model

If you haven't organized project structure with model previously, but want to do it at this stage, you can move a diagram from root into a model, or transfer a diagram from one model to another.

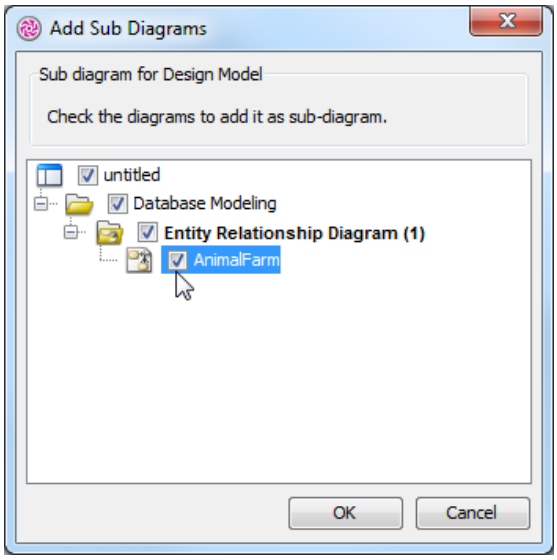
To move diagram from one model to another:

Right click on the target model in **Model Explorer** and select **Diagram** > **Add Existing Diagram...** from the pop-up menu.



Add existing diagram

When **Add Sub Diagrams** dialog box pops out, check the diagram(s) you want to move and then click **OK** button to proceed.



*Check a diagram in **Add Sub Diagrams** dialog box*

As a result, the selected diagram(s) will be moved to the target model.

NOTE: If you move a diagram which has the master view of model element(s), the model element(s) will be moved together with the diagram to the new model.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Animacian

Animacian is a tool that helps you make possible paths in an active diagram by presenting the paths in animation form. This chapter will describe Animacian in detail, and tells you how to animate a business process diagram and export the result to Adobe Flash.

What is Animacian?

Describe Animacian in detail.

Animate business process diagram

Shows you how to animate a BPD and describe the various parts of Animacian dialog box.

Exporting animation to Adobe Flash

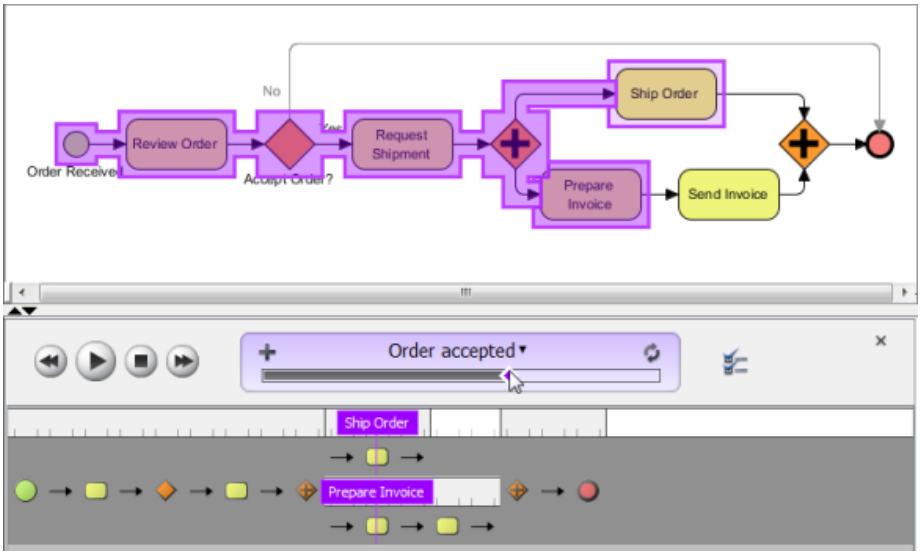
Shows you how to export the animation to Adobe Flash movie to play externally.

What is Animacian?

Animacian is a tool that helps you make possible paths in an active diagram by presenting the paths in animation form. This can make your design more attractive by animating it. Besides, you can control the flow of animation yourself to help demonstrating your work to client with your annotation. It also calculates all possible paths of the interaction, making the design more accurate.

Animating Paths in Diagram

Animation can be played directly on diagram. When the animation begins, a tiny black dot will be attached to the begining of the path selected to animate. During the animation, the black dot will traverse through the path, shapes that lie on the path will be painted in purple one by one, when being approached by the black dot, until the black dot reached the end of the path.



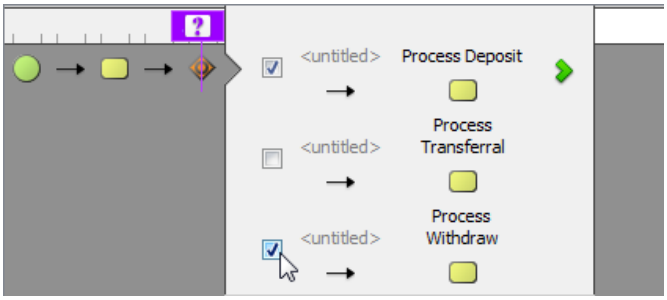
An animating path

Automatic Paths Identification

Interconnected shapes form a path. It is possible to have multiple paths on a diagram. Animacian helps finding out all possible paths in a diagram. When opening the Animacian dialog box, valid paths on the opening diagram will be identified and listed for selection. You can then select a path to animate. Unclosed paths or paths that does not obey the notation are classified as invalid, thus won't be available for playing animation.

Filtering Business Paths Base on Conditions

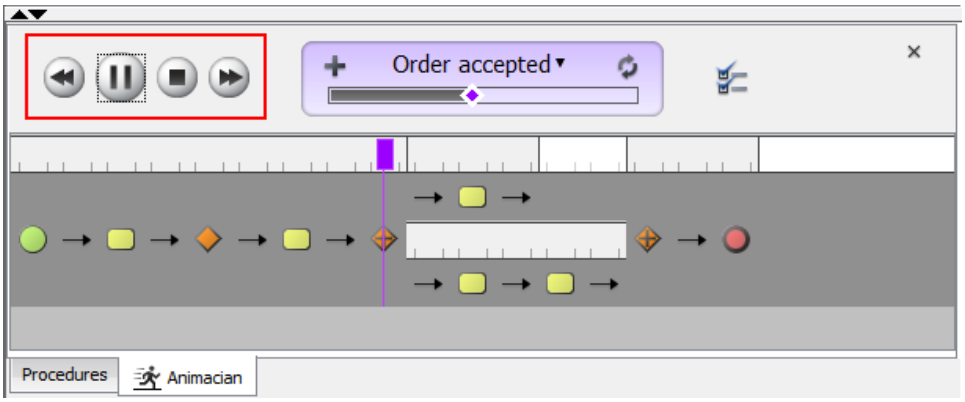
In BPD, BPMN gateway can be used to control how process flows. It can have multiple incoming and outgoing flows. There are several kinds of gateways that result in different flow behaviors. When you add an animation for process that involves gateway, you can select the outgoing flows for gateway(s).



Selecting outgoing flow for gateway

Walking through a Path Step-by-Step

Instead of auto animation, you can control it manually. The horizontal bar that appears at the bottom of Agilian when animating lets you control the animation. Besides pausing, playing and stopping the animation, you can also move a shape backward or forward. By making use of the forward and backward buttons, you can walk through a path shape by shape.



Walk though a path step by step

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

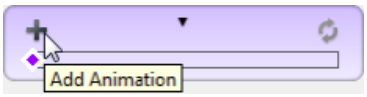
Animating Business Process Diagram

By animating a business process diagram with Animacian, you can see the flow of tasks within a process, from the beginning until the end. This does not only help to understand a process, but also trace the bottleneck and look for improvements.

Adding an Animation

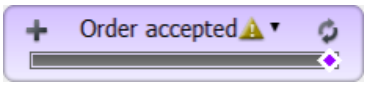
1. Select **Modeling > Animacian...** from the main menu. At the bottom of the diagram, you can see the **Animacian Panel** opened. If it is not there yet, open it manually by right clicking on the background of the diagram and selecting **Show Animacian Panel** from the popup menu.

2. Click **Add Animation** (graphically, a plus sign) at the top of the **Animacian Panel**.



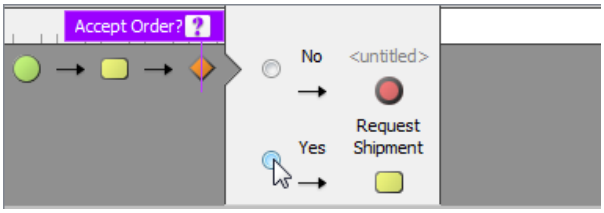
Add animation

3. Give a meaningful name to the animation based on the flow you want to animate. Press **Enter** to confirm editing.



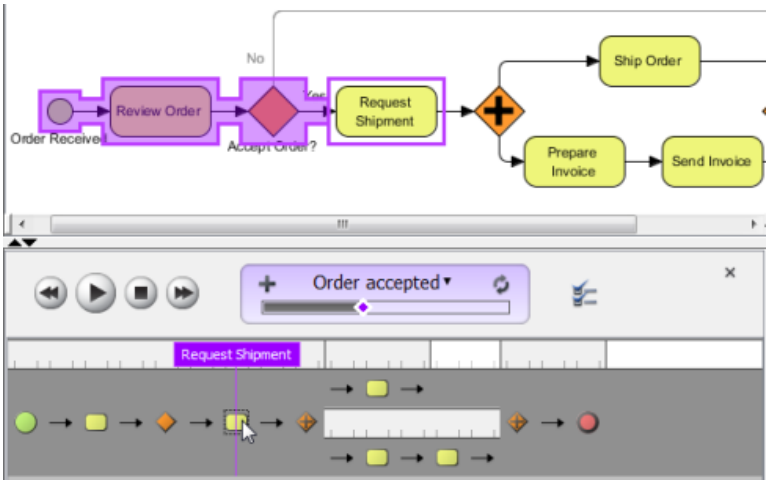
Named animation

4. When you add an animation for a process that involves gateway, you need to select the outgoing flows for gateway(s) in order to complete the path. To resolve an exclusive gateway requires the selection of the outgoing path. To resolve inclusive gateway requires the selection of zero to multiple outgoint paths. Make your selection and click the green arrow button to confirm.



Select the outgoing path for an exclusive gateway

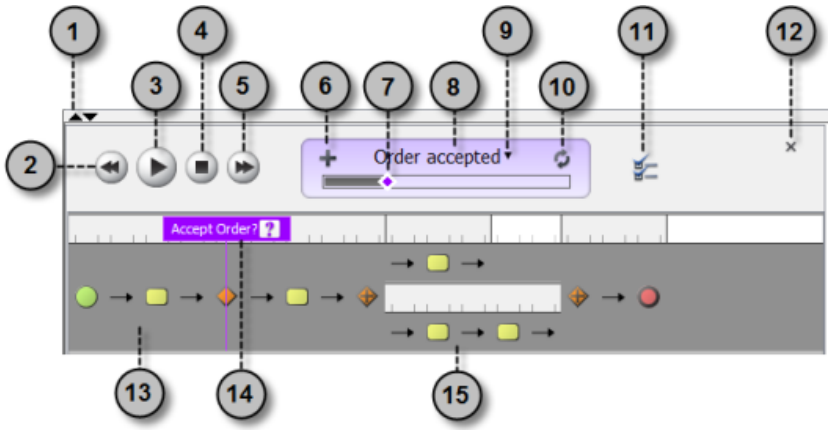
The path of the animation is determined automatically by evaluating the flow modeled in the diagram. Shapes that form the path are shown in the **Animacian Panel** as icons. If you click on any of them, it will jump right to the corresponding model element in the diagram.



Select a shape in Animacian Panel

Overview of Animacian Panel

The **Business Process Diagram Animacian** dialog box will pop out after clicking **Animacian...**. This dialog box is where you can select an execution path to play an animation.

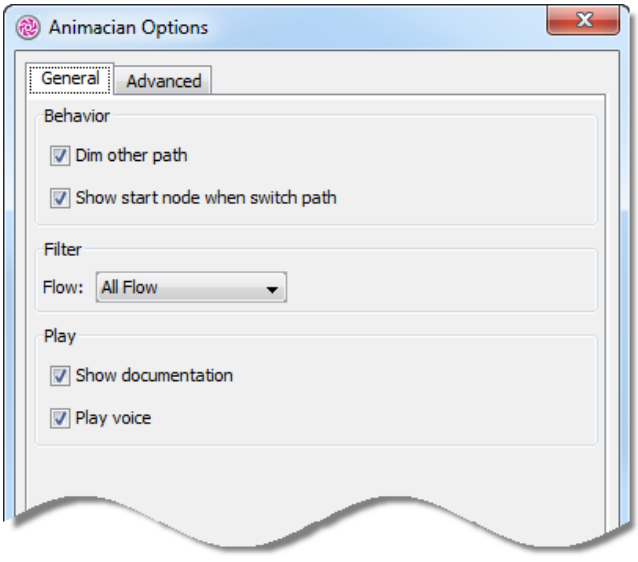


The Animacian Panel

No.	Name	Description
1	Expand/Collapse	Expand or collapse Animacian Panel
2	Backward	Move one shape backward in the flow.
3	Play	Play or continue to play the animation with Animacian minimized.
4	Stop	Terminate the animation.
5	Forward	Advance to the next shape in the flow.
6	Add Animation	Create a new animation.
7	Slider	It is used for controlling the flow of animation.
8	Current Animation	The name of the animation.
9	Animation selection	Click on this button to select another path to animate.
10	Refresh	It is used for re-identifying the flow of animations base on the diagram content.
11	Options	Click to configure Animacian.
12	Close	Click to close the **Animacian Panel**.
13	Flow of shapes	It displays all shapes of the current animation. Pressing on a shape here will highlight the corresponding shape in diagram.
14	Shape name	The name of the selected or animating shape. A question mark indicates that the shape is a decision shape. You can click on the question mark to re-select its outgoing flow.
15	Parallel flow	Parallel flow is presented as a branch.

*Description of **Animacian Panel***

Animacian Options - General

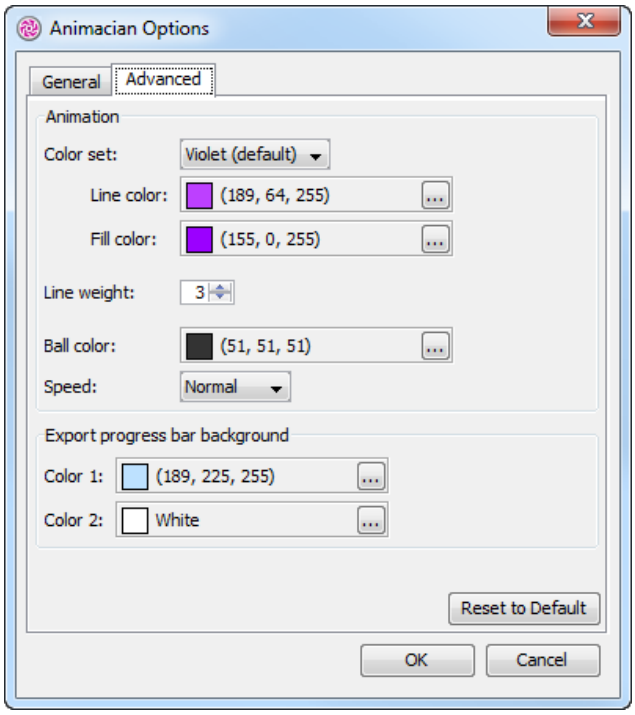


Animacian Options - General

Name	Description
Dim other path	It dims the components that are not a part of the selected path.
Show start node when switch path	Jump to the first node of the selected path, or keep staying at the current viewing field.
Flow	All Flow: Re-evaluate added animations to accept all available paths. Sequence Flow: Re-evaluate added animations to accept only paths that are joined by sequence flows. Message Flow: Re-evaluate added animations to accept only paths that are joined by message flows.
Show documentation	It shows documentation of shape when playing the animation in exported HTML.
Play voice	Voice can be recorded as documentation of model element. Check this if you want to play recorded voice when playing the animation in exported HTML.

The description of Animacian Options (General) window

Animacian Options - Advanced



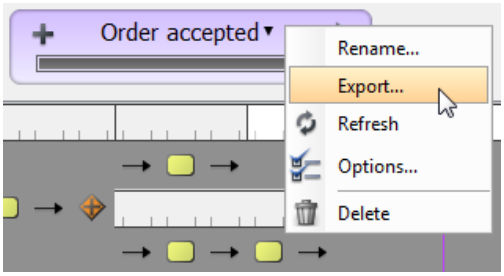
Animacian Options - Advanced

Name	Description
Color set	Select a color set to controls the line and fill color of visited shape.
Line color	The line color of visited shapes.
Fill color	The fill color of visited shapes.
Line weight	The thickness of rectangle that surround the visited shapes'.
Ball color	The color of ball that goes through a path during animation for indicating the progress of flow.
Speed	The pace of animation.
Export progress bar background - Color 1	The background color for the top of progress bar in exported HTML.
Export progress bar background - Color 2	The background color for the bottom of progress bar in exported HTML.

The description of Animacian Options (General) window

Exporting Animation

You can export the animation to Web contents so that you can play it externally in another computer just by playing in a Web browser. To export animation, right click beside the name of animation in **Animacian Panel** and select **Export...** from the popup menu. Then, fill in the file path and click **OK** in the **Export window**. You can add paths to be exported by clicking the plus button.



Export animation

Related Resources

The following resources may help you learn more about the topic discussed in this page.

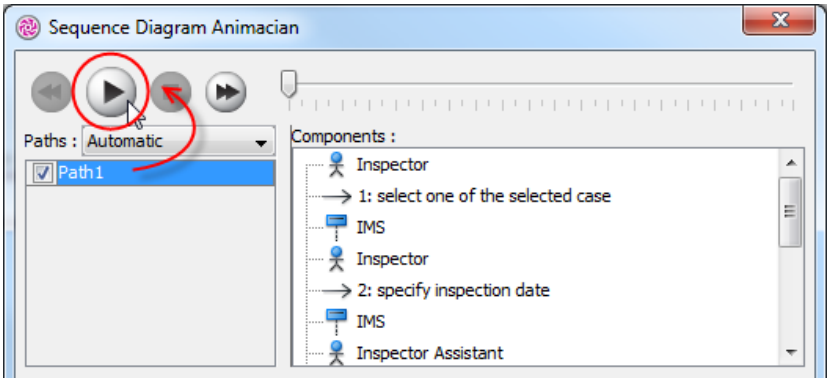
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Animating Sequence Diagram

By animating a sequence diagram with Animacian, you can see interaction between lifelines and the flow of message calls in active.

Launching an Animation

1. Select **Modeling > Animacian...** from the main menu.
2. In **Sequence Diagram Animacian** dialog box, select a path and then click **Play**.



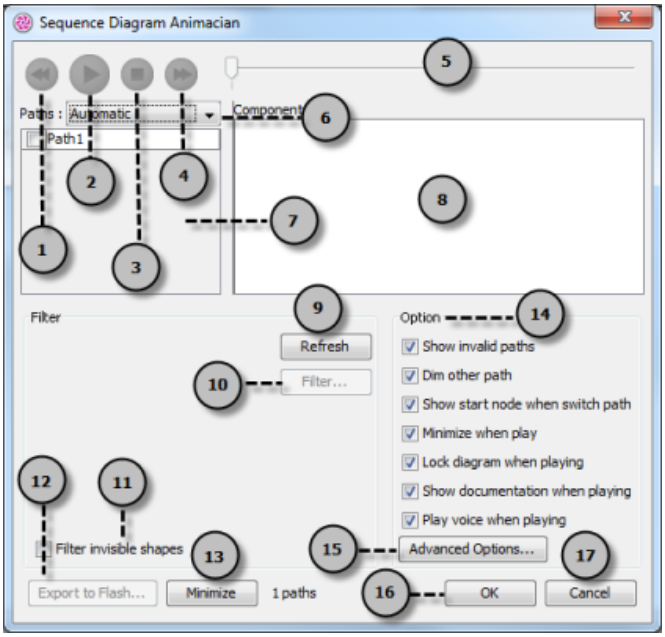
*Clicking **Play** in **Sequence Diagram Animacian** dialog box*

NOTE: Animacian can also be started by using any of the ways below:

- Right-click on the diagram background and select **Utilities > Animacian...** from the popup menu.
- Click the drop-down menu of **Modeling Tools** and select **Animacian...** on the toolbar.

Overview of Animacian

The **Sequence Diagram Animacian** dialog box will pop out after clicking **Animacian...**. This dialog box is where you can select an execution path to play an animation.



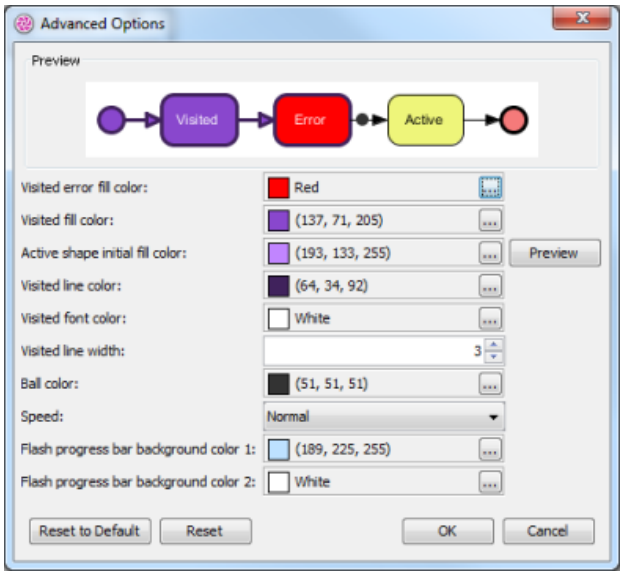
***Sequence Diagram Animacian** dialog box*

No.	Name	Description
1	Backward	Move one shape backward in the flow.
2	Play	Play or continue to play the animation with Animacian minimized.
3	Stop	Terminate the animation.
4	Forward	Advance to the next shape in the flow.
5	Slider	It is used for controlling the flow of animation.
6	Paths	It provides two ways of producing animation for the possible paths. **Automatic**: It is chosen by default. This helps you to detect all possible paths automatically. **Manual**: Choose when you want to select the possible path(s) manually.

7	Paths list	It lists all possible ways of executing a sequence. By default, paths are named as Path1, Path2, and so forth. You can rename them by double clicking on them and giving meaningful names.
8	Components list	It displays all components of the selected path. Pressing on a component will highlight the first shape of the chosen path until the chosen shape in the diagram.
9	Refresh	It is used for re-identifying the paths base on filter assignment and diagram content.
10	Filter...	It helps removing the non-selected paths by specifying the end result of fork nodes.
11	Filter invisible shapes	A shape can be set invisible on a diagram, or become invisible due to belonging to an invisible layer. By checking this option, invisible shapes will be ignored when calculating paths. By unchecking, invisible path will be included when calculating paths. By unchecking, you will see a black ball fly on diagram without attaching to the invisible shape(s) when executing a path.
12	Export to Flash...	Select an output path for exporting this diagram's animation to Adobe Flash.
13	Minimize	Click to minimize this dialog box.
14	Options pane	The Options pane helps you to configure animation. **Show invalid paths**: It lists not only the valid and selected path, but also the invalid and non-playable paths in the **Paths list**. **Dim other path**: It dims the components that are not a part of the selected path. **Show start node when switch path**: Jump to the first node of the selected path, or keep staying at the current viewing field. **Minimize when play**: It minimizes this dialog box when playing an animation. **Lock diagram when playing**: It locks the diagram when playing the animation to prevent accidental editing. **Show documentation when playing**: It shows documentation of shape at the bottom right of diagram when playing the animation. **Play voice when playing**: Voice can be recorded as documentation of model element. Check this if you want to play recorded voice when running animation.
15	Advanced Options...	It provides the color and speed options for animation.
16	OK	Click this button to confirm the settings and close Animacian.
17	Cancel	Click this button to close Animacian without saving the editing.

*Description of **Sequence Diagram Animacian** dialog box*

Advanced Options



***Advanced Options** dialog box*

Name	Description
Visited error fill color	The background color of visited shape that cause an error. An error means the flow object that causes a path invalid.
Visited fill color	The background color of visited shapes.
Active shape initial fill color	When playing an animation, a tiny black ball will traverse the chosen path, from one shape to another. When it reaches a shape, the shape will render with a transition effect that means transiting from an initial color to visited fill color. This option manages the initial background color for visiting shape.

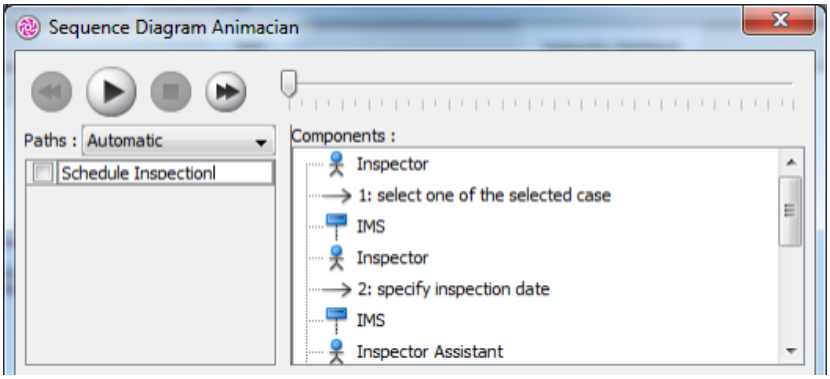
Visited line color	The line color of visited shapes.
Visited font color	The font color of visited shapes.
Visited line width	The thickness of visited shape's border.
Ball color	The color of ball that goes through a path during animation for indicating the progress of flow.
Speed	The pace of animation.
Flash progress bar background color 1	The background color for the top of progress bar in exported Flash movie.
Flash Progress Bar background color 2	The background color for the bottom of progress bar in exported Flash movie.

*The description of **Advanced Options** dialog box*

Naming a Path

The **Paths** list displays all possible animation paths of your diagram. Each path represents a possible way to go through the diagram. By default, paths are named as Path1, Path2, and so forth. It is recommended to name to the path(s) for better clarification.

1. To rename a path, move the mouse pointer on a path in the list and double click on it.
2. Enter the name of path.
3. Press **Enter** to confirm editing.

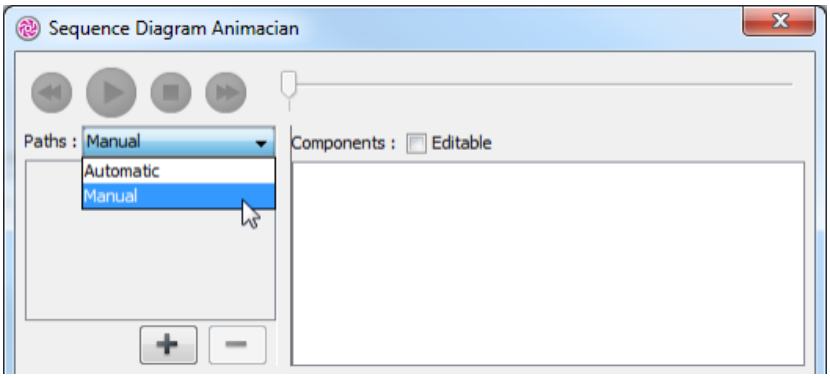


Naming the path

Creating a Manual Path

In **Sequence Diagram Animacian** dialog box, all paths are listed in **Paths list** by default. However, you can manage the flow of animation with your own choice. To create a manual path:

1. Select **Manual** in **Paths**.

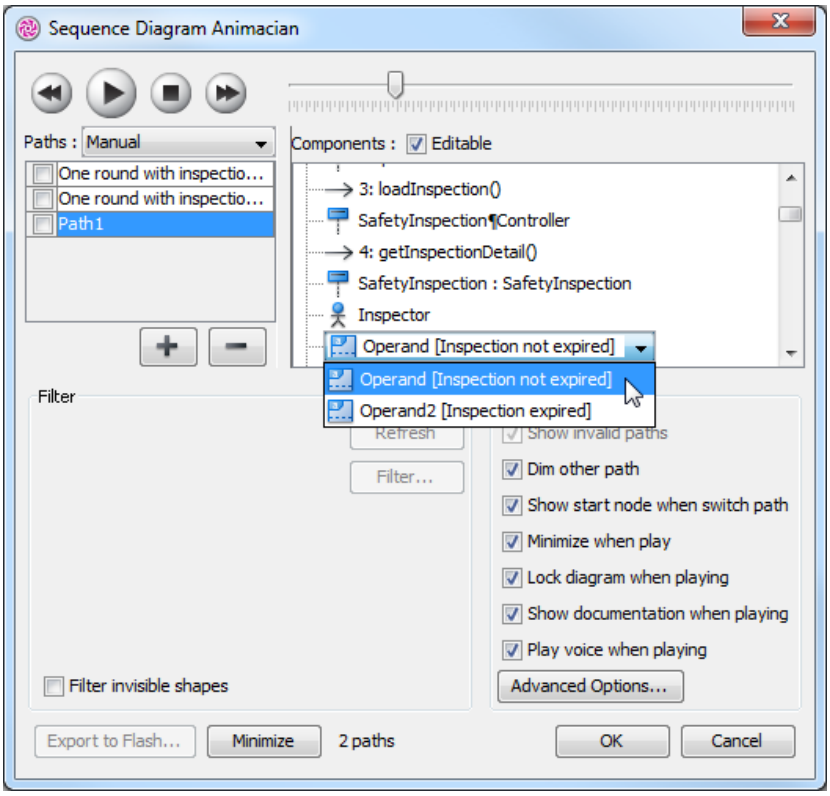


*Selecting **Manual** in **Paths***

2. Press **Add Path** to insert a new path.
3. Select the shapes that are shown on the **Components list** to direct the flow of animation.
4. Click **OK** to confirm editing.

Handling Decision

You should choose an operand when there is more than one option in the interaction. Different decisions will lead to different forks and make a different outcome for the flow of animation. Make either decision to view the outcome.



Making a decision for the flow of path

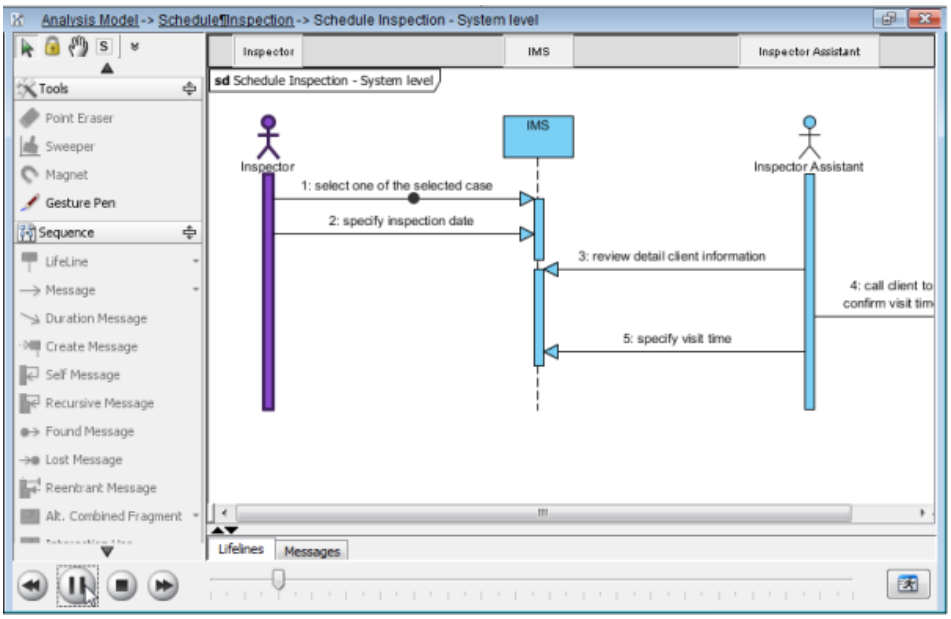
Reviewing an Animation

1. When everything is ready, click **Play** to start the animation of the selected path.
2. After click **Play**, **Sequence Diagram Animacian** dialog box will be minimized to the bottom of your diagram, with several buttons and a slider revealing on it.

Button	Name	Description
	Backward	Move one shape backward in the flow.
	Pause	Temporary stop playing the movie. Press **Play** to continue to play.
	Play	Play or continue to play the animation.
	Forward	Advance to the next shape in the flow.
	Stop	Terminate the animation.
	Maximize	Maximize **Animacian**.

Description of Animacian bar

3. When the animation starts, a black ball will appear at beginning of path and traverse through the path until the end.
4. When the black ball reaches a shape, the shape will turn into purple.

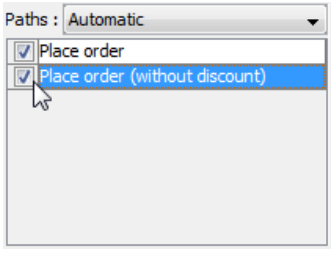


Reviewing the animation

Exporting an Animation

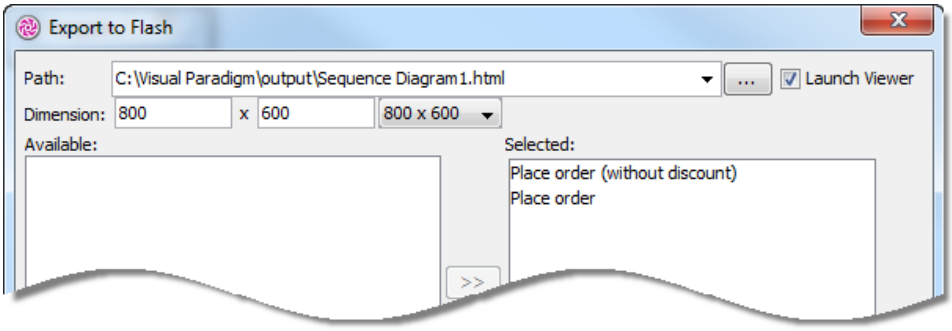
You can export the animation to Web contents so that you can play it externally in another computer just by playing in a Web browser.

1. From the **Paths** list in the **Animacian** window, select the execution paths to export as Flash movie.



Path selection

2. Click the **Export to Flash...** button at bottom left. This shows the **Export to Flash** dialog box. Here is a description of the **Export to Flash** dialog box.



The Export to Flash window

Here is a description of the **Export to Flash** dialog box.

Part	Description
Path	The path of the exported HTML file. Flash movie file (.swf) will also be exported to the same folder as the HTML file.
Launch Viewer	When checked, default web browser will automatically start and play the exported Flash movie.
Dimension	The width and height of viewing region of Flash.
Available	Available paths that can be selected to export to Flash movie for animation.
Selected	Selected paths to export to Flash movie for animation.

Description of the Export to Flash dialog box

3. An HTML web page will be exported. Specify the path of the HTML file. Note that the Flash movie files (.swf) will be exported to the same folder as the HTML file.
4. Choose or enter the dimension of movie if necessary. Note that the dimension determins the size of viewable region instead of the size of diagram.
5. Click **Export**. Open the HTML file in the web browser to play the movie. If there are more then one path being selected, you can click on the drop down menu at top right corner and select another path to play with.

Related Resources

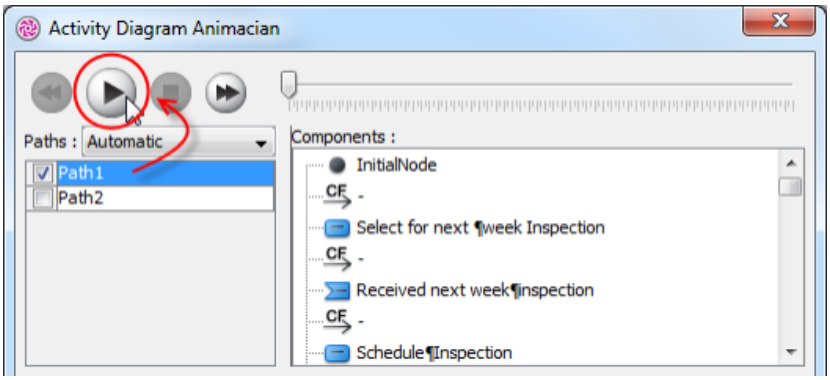
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Animating Activity Diagram

By animating an activity diagram with Animacian, you can see the flow of actions in active.

1. Select **Modeling > Animacian...** from the main menu.
2. In **Activity Diagram Animacian** dialog box, select a path and then click **Play**.



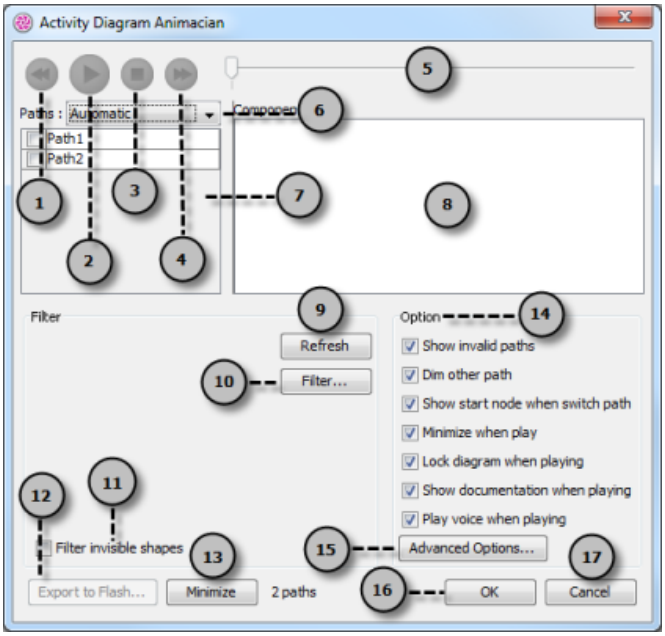
*Clicking **Play** in **Activity Diagram Animacian** dialog box*

NOTE: Animacian can also be started by using any of the ways below:

- Right-click on the diagram background and select **Utilities > Animacian...** from the popup menu.
- Click the drop-down menu of **Modeling Tools** and select **Animacian...** on the toolbar.

Overview of Animacian

The **Activity Diagram Animacian** dialog box will pop out after clicking **Animacian...**. This dialog box is where you can select an execution path to play an animation.



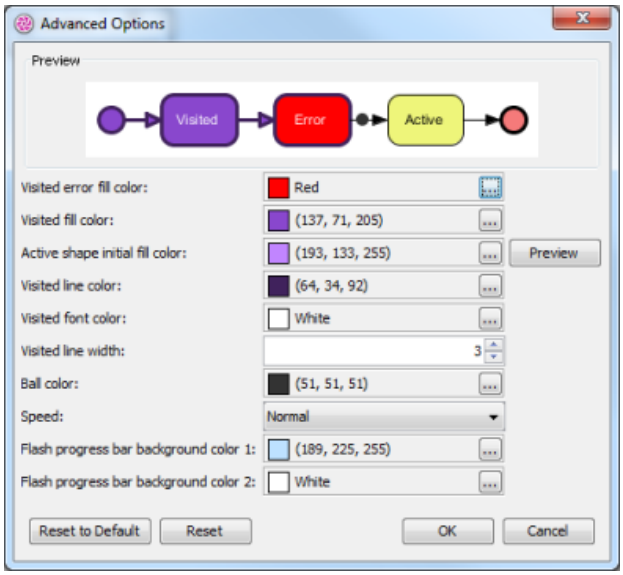
***Activity Diagram Animacian** dialog box*

No.	Name	Description
1	Backward	Move one shape backward in the flow.
2	Play	Play or continue to play the animation with Animacian minimized.
3	Stop	Terminate the animation.
4	Forward	Advance to the next shape in the flow.
5	Slider	It is used for controlling the flow of animation.
6	Paths	It provides two ways of producing animation for the possible paths. **Automatic**: It is chosen by default. This helps you to detect all possible paths automatically. **Manual**: Choose when you want to select the possible path(s) manually.

7	Paths list	It lists all possible ways of executing an Activity. By default, paths are named as Path1, Path2, and so forth. You can rename them by double clicking on them and giving meaningful names.
8	Components list	It displays all components of the selected path. Pressing on a component will highlight the first shape of the chosen path until the chosen shape in the diagram.
9	Refresh	It is used for re-identifying the paths base on filter assignment and diagram content.
10	Filter...	It helps removing the non-selected paths by specifying the end result of fork nodes.
11	Filter invisible shapes	A shape can be set invisible on a diagram, or become invisible due to belonging to an invisible layer. By checking this option, invisible shapes will be ignored when calculating paths. By unchecking, invisible path will be included when calculating paths. By unchecking, you will see a black ball fly on diagram without attaching to the invisible shape(s) when executing a path.
12	Export to Flash...	Select an output path for exporting this diagram's animation to Adobe Flash.
13	Minimize	Click to minimize this dialog box.
14	Options pane	The Options pane helps you to configure animation. **Show invalid paths**: It lists not only the valid and selected path, but also the invalid and non-playable paths in the **Paths list**. **Dim other path**: It dims the components that are not a part of the selected path. **Show start node when switch path**: Jump to the first node of the selected path, or keep staying at the current viewing field. **Minimize when play**: It minimizes this dialog box when playing an animation. **Lock diagram when playing**: It locks the diagram when playing the animation to prevent accidental editing. **Show documentation when playing**: It shows documentation of shape at the bottom right of diagram when playing the animation. **Play voice when playing**: Voice can be recorded as documentation of model element. Check this if you want to play recorded voice when running animation.
15	Advanced Options...	It provides the color and speed options for animation.
16	OK	Click this button to confirm the settings and close Animacian.
17	Cancel	Click this button to close Animacian without saving the editing.

*Description of **Activity Diagram Animacian** dialog box*

Advanced Options



***Advanced Options** dialog box*

Name	Description
Visited error fill color	The background color of visited shape that cause an error. An error means the flow object that causes a path invalid.
Visited fill color	The background color of visited shapes.
Active shape initial fill color	When playing an animation, a tiny black ball will traverse the chosen path, from one shape to another. When it reaches a shape, the shape will render with a transition effect that means transiting from an initial color to visited fill color. This option manages the initial background color for visiting shape.

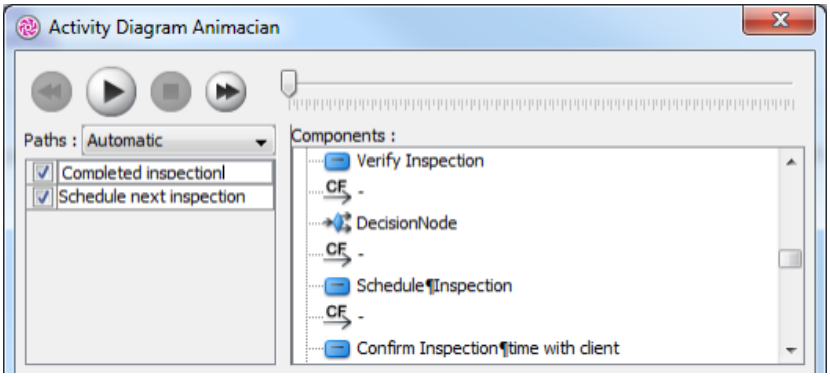
Visited line color	The line color of visited shapes.
Visited font color	The font color of visited shapes.
Visited line width	The thickness of visited shape's border.
Ball color	The color of ball that goes through a path during animation for indicating the progress of flow.
Speed	The pace of animation.
Flash progress bar background color 1	The background color for the top of progress bar in exported Flash movie.
Flash progress bar background color 2	The background color for the bottom of progress bar in exported Flash movie.

*The description of **Advanced Options** dialog box*

Naming a Path

The **Paths** list displays all possible animation paths of your diagram. Each path represents a possible way to go through the diagram. By default, paths are named as Path1, Path2, and so forth. It is recommended to name to the path(s) for better clarification.

1. To rename a path, move the mouse pointer on a path in the list and double click on it.
2. Enter the name of path.
3. Press **Enter** to confirm editing.

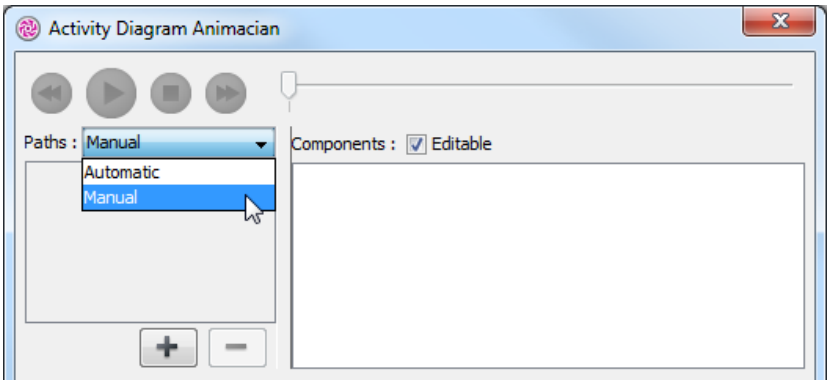


Naming the paths

Creating a Manual Path

In **Activity Diagram Animacian** dialog box, all paths are listed in **Paths list** by default. However, you can manage the flow of animation with your own choice. To create a manual path:

1. Select **Manual** in **Paths**.

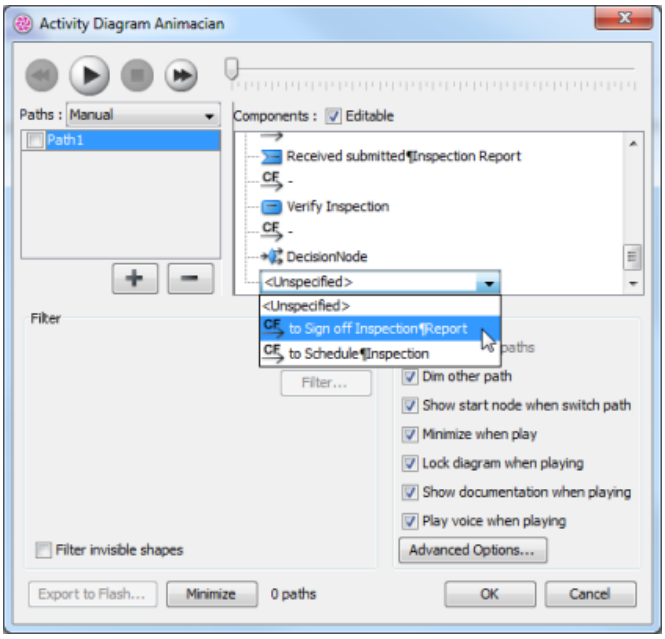


*Selecting **Manual** in **Paths***

2. Press **Add Path** to insert a new path.
3. Select the shapes that are shown on the **Components list** to direct the flow of animation.
4. Click **OK** to confirm editing.

Handling Decision

You should choose an outgoing flow when there is more than one option in the flow. Different decisions will lead to different forks and make a different outcome for the flow of animation. Make either decision to view the outcome.



Making a decision for the flow of path

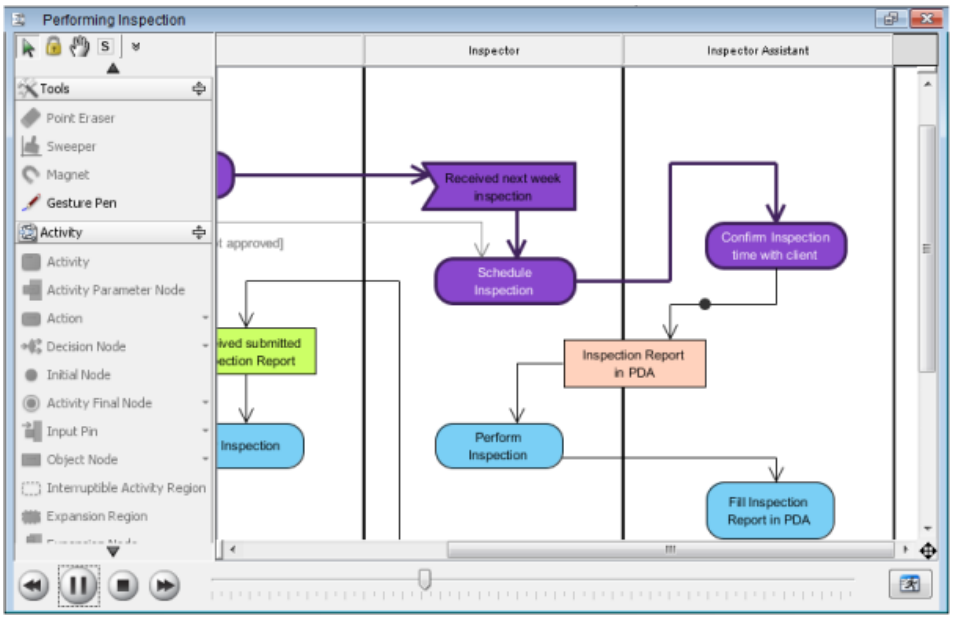
Reviewing an Animation

1. When everything is ready, click **Play** to start the animation of the selected path.
2. After click **Play**, **Activity Diagram Animacian** dialog box will be minimized to the bottom of your diagram, with several buttons and a slider revealing on it.

Button	Name	Description
	Backward	Move one shape backward in the flow.
	Pause	Temporary stop playing the movie. Press **Play** to continue to play.
	Play	Play or continue to play the animation.
	Forward	Advance to the next shape in the flow.
	Stop	Terminate the animation.
	Maximize	Maximize **Animacian**.

Description of Animacian bar

3. When the animation starts, a black ball will appear at beginning of path and traverse through the path until the end.
4. When the black ball reaches a shape, the shape will turn into purple.

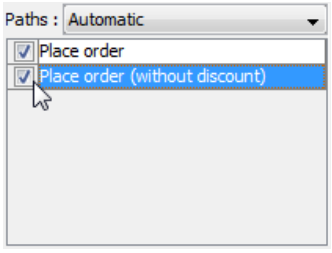


Reviewing the animation

Exporting an Animation

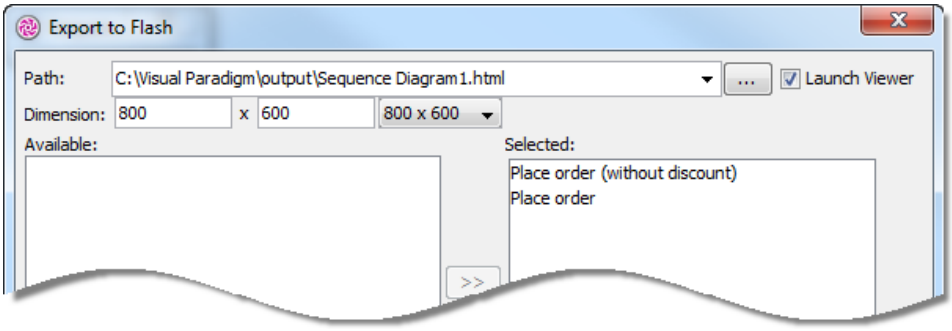
You can export the animation to Web contents so that you can play it externally in another computer just by playing in a Web browser.

1. From the **Paths** list in the **Animacian** window, select the execution paths to export as Flash movie.



Path selection

2. Click the **Export to Flash...** button at bottom left. This shows the **Export to Flash** dialog box. Here is a description of the **Export to Flash** dialog box.



The Export to Flash window

Here is a description of the **Export to Flash** dialog box.

Part	Description
Path	The path of the exported HTML file. Flash movie file (.swf) will also be exported to the same folder as the HTML file.
Launch Viewer	When checked, default web browser will automatically start and play the exported Flash movie.
Dimension	The width and height of viewing region of Flash.
Available	Available paths that can be selected to export to Flash movie for animation.
Selected	Selected paths to export to Flash movie for animation.

Description of the Export to Flash dialog box

3. An HTML web page will be exported. Specify the path of the HTML file. Note that the Flash movie files (.swf) will be exported to the same folder as the HTML file.
4. Choose or enter the dimension of movie if necessary. Note that the dimension determins the size of viewable region instead of the size of diagram.
5. Click **Export**. Open the HTML file in the web browser to play the movie. If there are more then one path being selected, you can click on the drop down menu at top right corner and select another path to play with.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Maintaining project reference

To reference another project enables you to link to another project, and use its model elements in developing your own project. In this chapter you can see how to add and maintain referenced project.

Referencing another project

Shows you how to add a referenced project.

Referencing other projects' model elements

Having added a referenced project, you can make use of referenced elements in current project.

Mirroring model element

Mirror is a technique to create a partial copy of referenced element in current project. This make it possible to create shapes in a mirroed container shape like package.

Viewing referenced diagrams

You can read the diagrams in referenced project without actually opening it. Click on the diagram in Diagram Navigator to open it.

Duplicating element from linked project

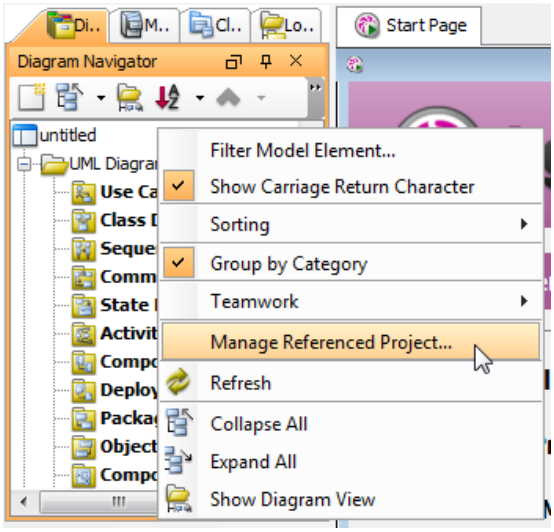
You can clone an element from referenced project through the duplicate function.

Reference another project

To reference another project enables you to links to another project, and use its model elements in developing your own project. You can organize your model elements in a more disciplined approach by having one project per library project. This also helps you to "slim up" projects through breaking down a project into smaller pieces. Moreover, you can reference to other projects, and create an overview project for them.

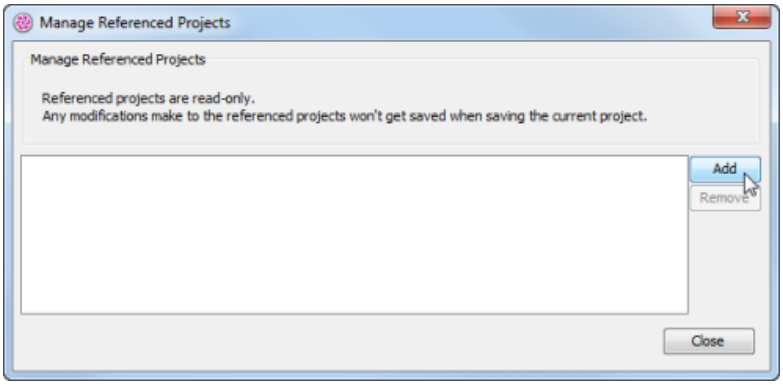
To reference to another project:

1. Right click on the background of any of the following panes and select **Manage Referenced Project...** from the pop-up menu.
   - Diagram Navigator
   - Model Explorer
   - Logical View



Manage referenced project

2. In the **Manage Referenced Projects** dialog box, click **Add** button.



Add a referenced project

3. In the **Open** dialog box, choose the Agilian project file (*.vpp) to reference to and click **Open**.
4. When you return the **Manage Referenced Projects** dialog box, click **Close** button.

Related Resources

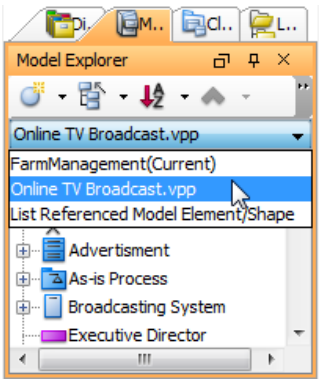
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Referencing other projects' model elements

Once a referenced project has been established, you can develop your model using model elements in referenced project. To do this:

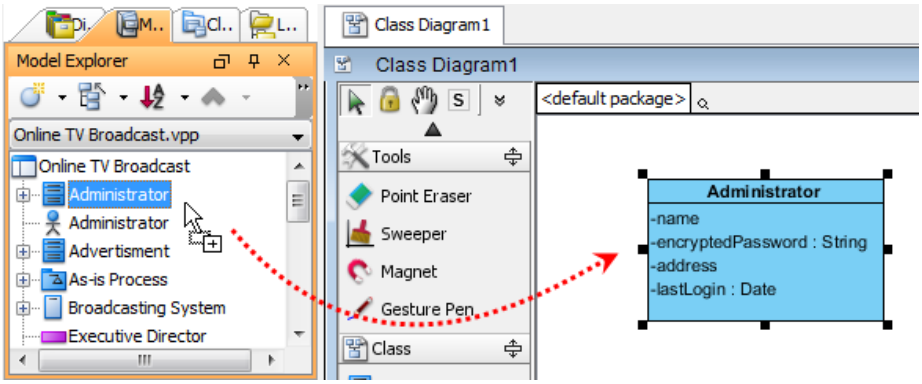
1. In **Model Explorer**, click on the drop-down menu at the top of the pane and select the project that contains the model elements you want to use.



Select a referenced project

> **NOTE:** By selecting the first selection **(Current)**, model elements in the current project will be listed.

2. In the model element list, drag the target model element(s) and drop it/ them on diagram. This creates views from them. In addition, you may continue modeling with the referenced elements. Note that Views for referenced project are read-only (i.e. non edit-able).

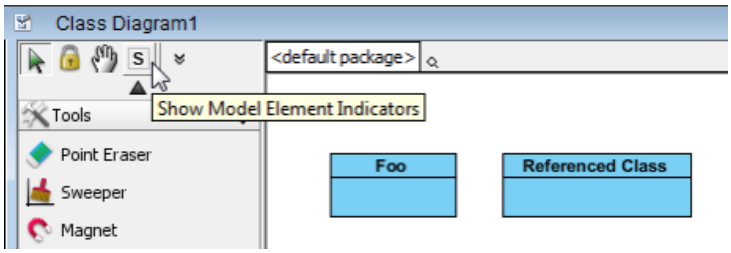


Drag a model element from referenced project

> **NOTE:** Alternatively, you can right click on the model elements and select **Create View in Active Diagram** from the pop-up menu.

Indicating referencing elements on diagram

In order to know which shape(s) on a diagram comes from a referenced project, you can enable to model element indicators. Click on the **Show Model Element Indicators** button at the top of diagram toolbar.



To show model element indicators

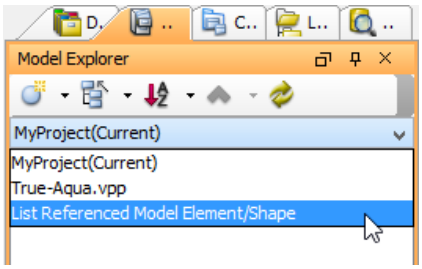
After that, you can see the indicator, which is a small arrow, appear at the shapes that are referencing a referenced project.

Model element indicator appears

Locating referenced element

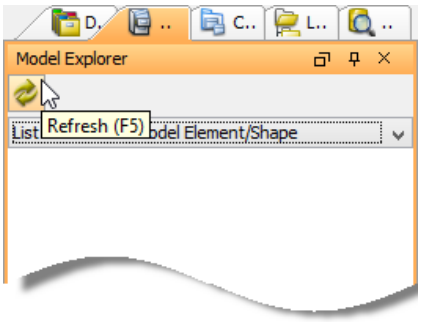
You can go to a shape that references a model element from referenced project, by walking through the steps below:

1. Open the **Model Explorer**.
2. From the drop down menu at the top of the Model Explorer, select **List Referenced Model Element/Shape**.



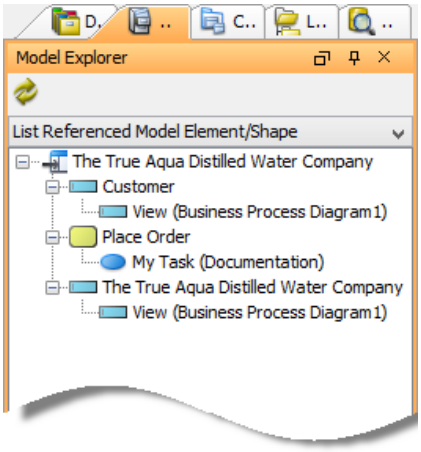
List referenced model elements and shapes

3. Click the **Refresh** button to load the content.



Refresh Model Explorer

4. A tree of model elements are listed. Those are came from the referenced project(s) and are used by the currently opening project. The first level of the tree lists the model elements in used, while the second level represent the usage. For example, **View ([DIAGRAM NAME])** means that the model element has been visualized in the diagram specified. **[MODEL ELEMENT NAME] (Documentation)** means that the model element has been added as a link in the model element specified.



Referenced model elements listed

Related Resources

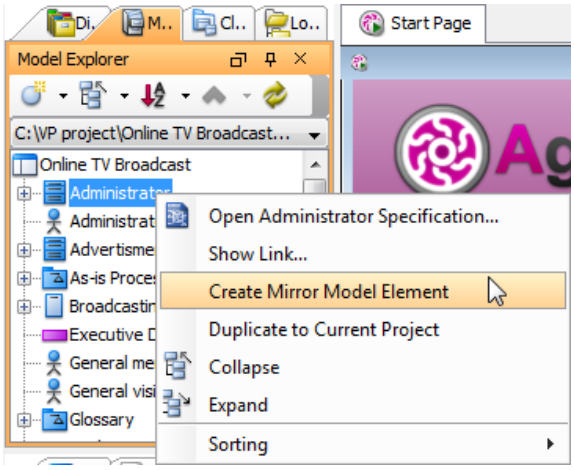
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Mirroring model element

Due to views of referenced model elements are read-only, you cannot add shapes in it. This may be a problem when you want to use referenced packages in your project, and to add model elements such as classes in it. To overcome this problem, you can create a mirror of container-typed referenced model element. By mirroring, a referenced element is localized partially by keeping a mirrored copy in your project which echoes the element in referenced project. The mirrored copy can be accessed in **Model Explorer** that lists model elements in current project, but not editable.

To create a mirror from a container-typed referenced model element (e.g. package, pools/lanes), right click on the element in **Model Explorer** and select **Create Mirror Model Element** from pop-up menu.



Mirror a referenced package

By doing so, you can use it in your project and add shapes in it. If you have already created a view for a non-mirrored element and you want to create a mirror, you may right click on the shape and select **Convert to Duplicate Model Element** from pop-up menu.

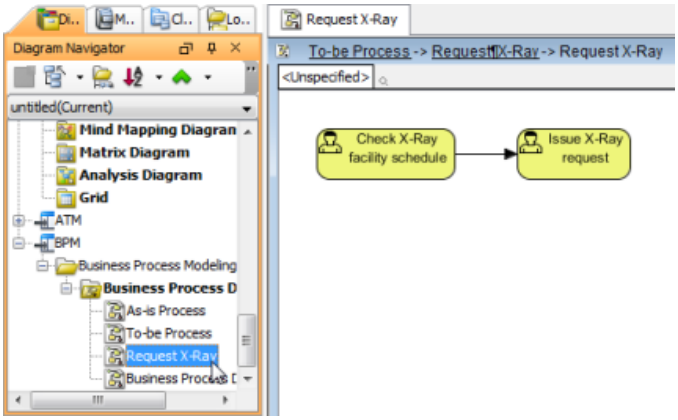
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Viewing referenced diagrams

Sometimes, you may want to take a look at the design created in referenced project to make yourself familiar with it. Instead of the traditional way of opening a project, you can read diagrams of referenced diagrams by scrolling down to the bottom of diagram navigator, expanding the referenced project node and double clicking on a diagram node to open it. Note that the diagram is read-only.



Read a diagram in referenced project

Related Resources

The following resources may help you learn more about the topic discussed in this page.

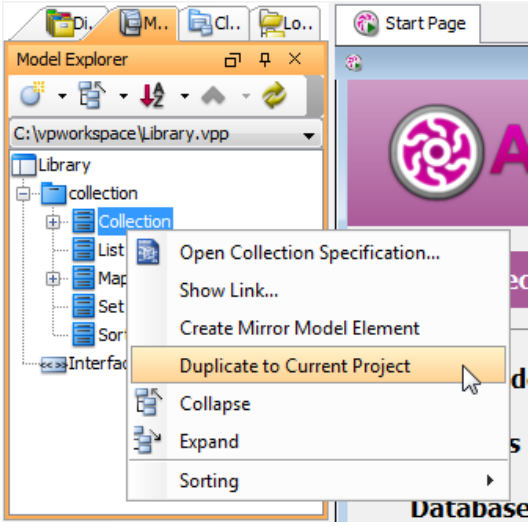
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Duplicating element from linked project

When we have added a referenced project, we say we are referencing that project. By reference, it’s like a linkage that points to the actual data stored in referenced project. Those elements borrowed from referenced project are not editable by current project.

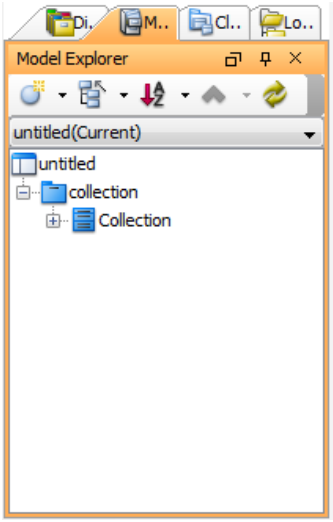
Sometimes, you need the elements created in another project, not to appear as form of references, but to make it become your own data. You will need to duplicate element from linked project. To duplicate a model element:

1. Open the **Model Explorer**.
2. Right click on the element you want to duplicate ands elect **Duplicate to Current Project** from the pop-up menu.



Duplicate a model element

The duplicated element can be found in **Model Explorer**, under current project.



Duplicated element appears in Model Explorer

Related Resources

The following resources may help you learn more about the topic discussed in this page.

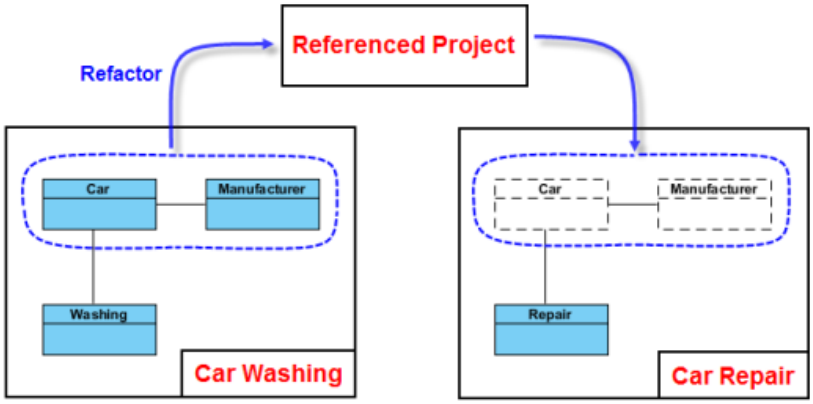
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Refactoring

Refactoring refers to the action to move project data like diagrams and/or model elements between projects that have a reference relationship in between.

Refactoring diagrams and/or model elements to another project helps in the re-structuring of project and re-organizing project data. Through refactoring, generic project data is moved to a project file, generally known as a library project, readily for other project files to access through project referencing, such that they can get the common project data included without the need of re-definition. Another benefit of refactoring is that it guarantees the correctness of model definition by enforcing common project data to be defined just once. This is also, in fact, the benefit of using project reference.

Let's say, for example, you are modeling a vehicle maintenance company. You have created two projects for the distinct parts of the business - Car washing and car repair. When modeling car washing with class diagram, you have found that classes like Car and Manufacturer are also needed by the car repair model. You then refactor them to a referenced project so that the car repair model can link with it and have the classes included.



What is refactoring?

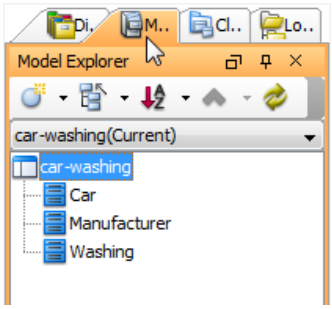
Refactoring also works in the opposite way, that is, to move diagram/model elements from the generic library project back to the 'consumer' project that use the data. While it is not so common, you may need it when you discover that certain diagram or model element in the generic project is not really that general, and you want to move the diagram/model element back to the only, specific project that use it.

How to refactor?

Keep in mind that refactoring only works with projects that have a reference in between. If you want to refactor diagram or model element to a library project but such project is either not existing or not a referenced project of your source project, get ready for the project and the source project reference to it. If you are unclear about project referencing, please read the previous chapters.

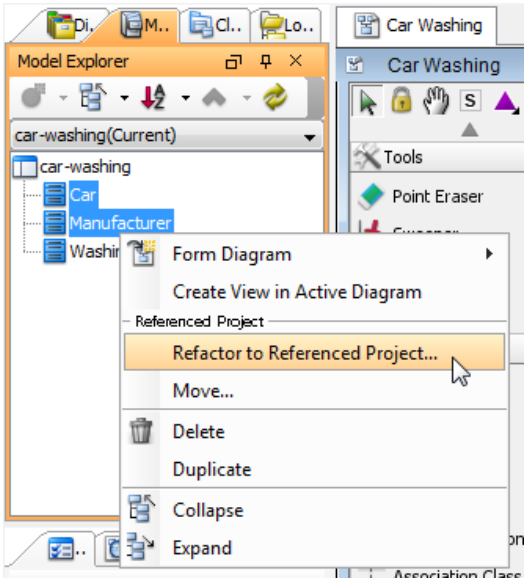
Refactor model elements to reference project

1. Open the **Model Explorer**.



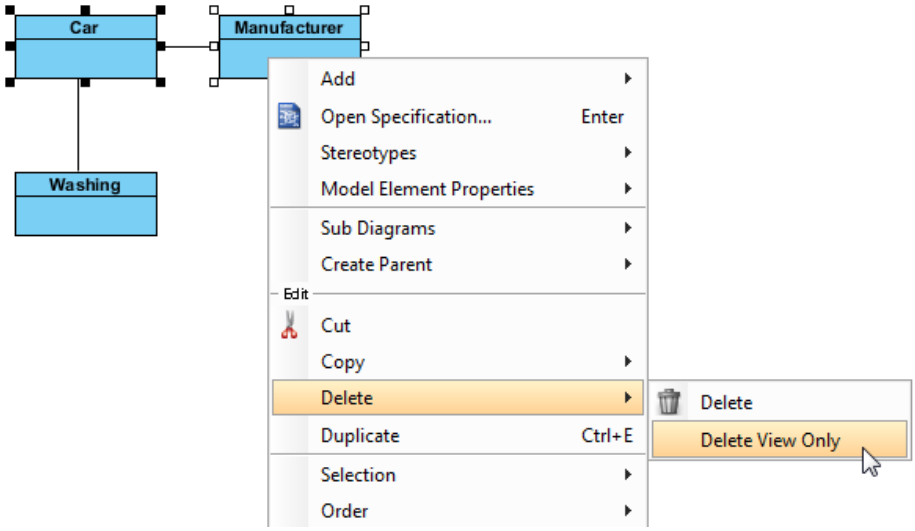
Open Model Explorer

2. Select and right click on the model elements you want to refactor to referenced project. Select **Refactor to Referenced Project...** from the popup menu.



Refactor classes

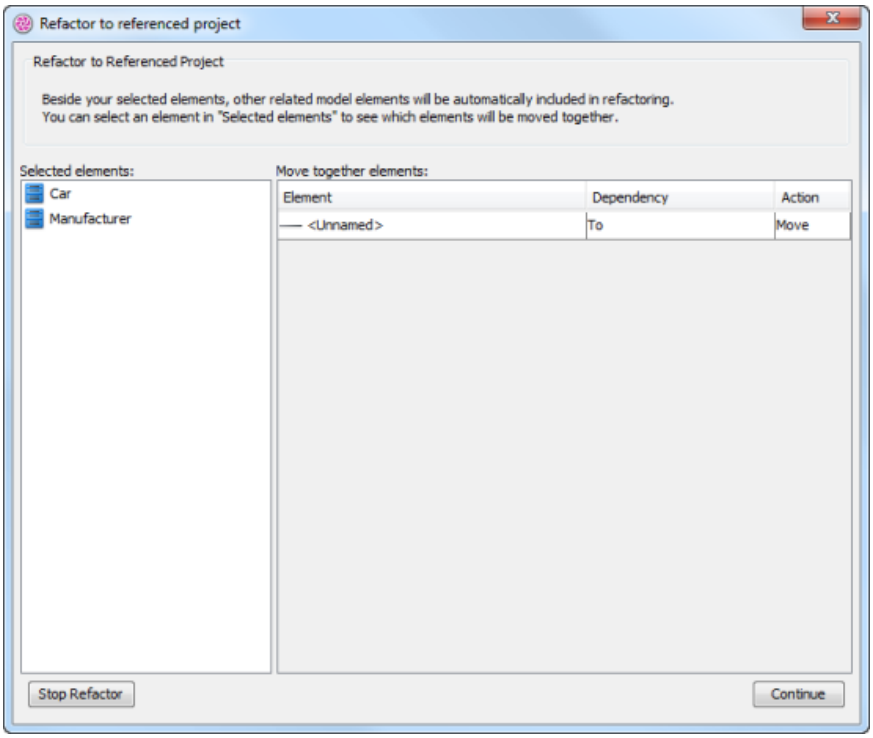
3. You are prompted for saving project. Click **Yes**. If you click **No**, refactor will stop.
4. If the **Include Related Diagrams** window appears, this means that one or more of the selected elements has been visualized in at least one diagram (as listed on the left of window) with master view created.
   Here you may take any of the following actions to continue.
   - **Include the diagrams** - Click **Include Diagram(s)** at the bottom right corner. This will move not only the model elements, but also the diagrams listed on the left hand side of window to the referenced project. Since the action is not undoable, think carefully before continuing.
   - **Remove the master views** - Click **Stop Refactor** at the bottom left corner. Open the diagrams where the master views of the selected model elements exist. Delete them. When delete, make sure you are deleting the **VIEW** instead of model element. Once all master views are removed, re-run step 1.



Delete only views of element, not model elements

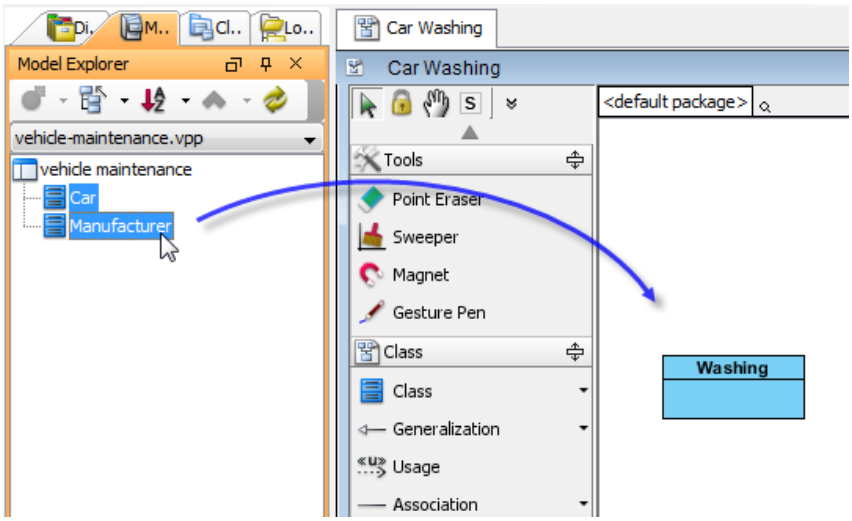
5. Refactor not only moves the selected elements to referenced project, but also those related elements. Here are two examples:
   - Connectors that connect between the selected elements.
   - Class being selected as type of attribute of selected class.

   If such related elements exist, you will be listed with those elements. Check them. If you accept moving them to referenced project, click **Continue** at the bottom right corner to continue. Otherwise, click **Stop Refactor** at bottom left corner to terminate the refactoring.



Elements to refactor (including those related elements)

6. Once the refactoring is complete, the project will reopen itself. From now on, you can access and use the refactored elements through the **Model Explorer**. Select the referenced project under the drop down menu. Drag to diagram the element(s) to use.



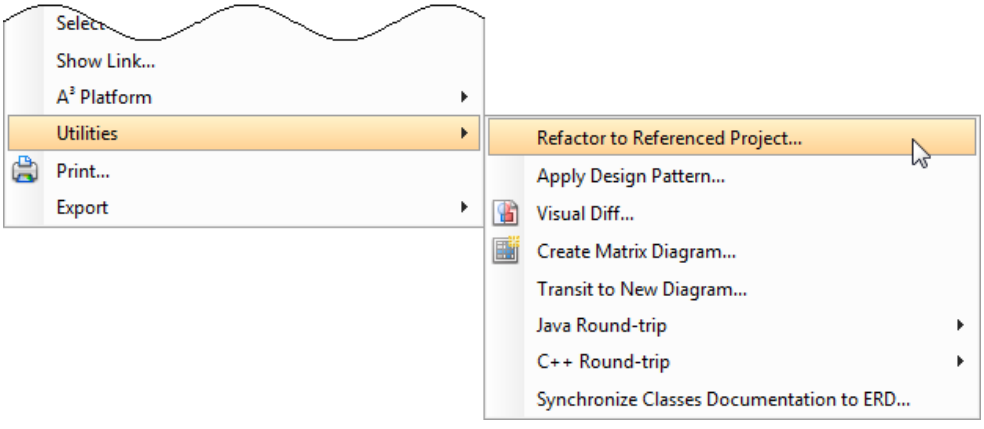
To re-use referenced classes on current project

Note that auxiliary views of refactored elements are now referencing the elements in referenced project.

Refactor diagram to reference project

To refactor diagram means to refactor the diagram as well as the elements on the diagram. As the steps are pretty close to refactoring model element, as described above, please read refactor model elements before reading this section.

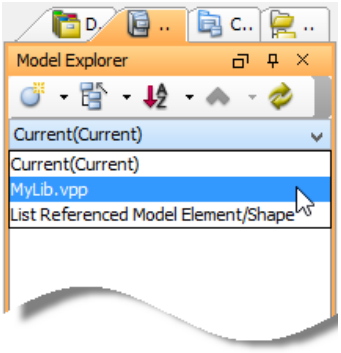
1. Right click on the diagram that you want to refactor and select **Utilities > Refactor to Referenced Project...** from the popup menu.



2. You are prompted for saving project. Click **Yes**. If you click **No**, refactor will stop.
3. If the **Include Related Diagrams** window appears, this means that one or more of the elements in the selected diagram has been visualized in another diagram (as listed on the left of window) with master view created. Read step 4 in the previous section to learn how to resolve the relationships.
4. Refactor not only moves the selected elements to referenced project, but also those related elements. Read step 5 in the previous section to learn how to carry on.
5. Once the refactoring is complete, the project will reopen itself. From now on, you can access and use the refactored elements through the **Model Explorer**. Select the referenced project under the drop down menu. Drag to diagram the element(s) to use. Note that auxiliary views of refactored elements are now referencing the elements in referenced project.

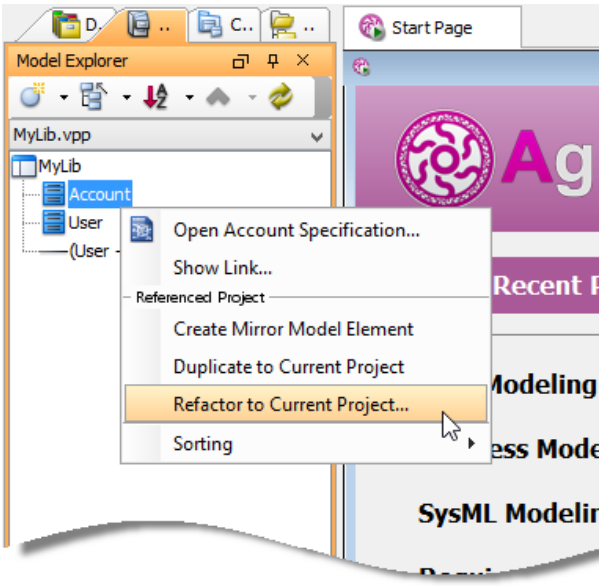
Refactor model elements from reference project

1. Open the **Model Explorer**.
2. Click on the drop down menu at the top of the **Model Explorer** and choose the reference project that contains the model element(s) you want to get.



Browsing model elements in reference project

3. Select and right click on the model elements you want to get from the reference project. Select **Refactor to Current Project...** from the popup menu.

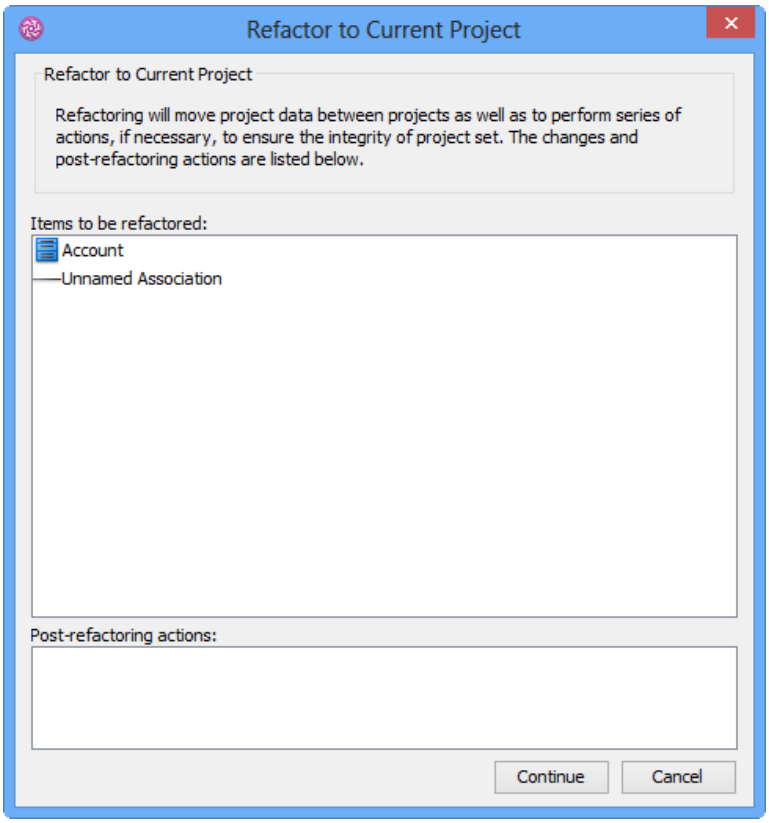


Refactoring a class

4. You are prompted a note with some points you should pay attention to before continue. Let's revise them in brief here:
   - Refactoring cannot be undone.
   - Project will be saved before refactoring.
   - Make sure all the 'related project files' are in your workspace folder, or any sub-directory under the workspace folder. 'Related projects' means the project files that are referencing the current or reference project, either with a direct reference (project A references B) or a transitive reference (project A references A1, A1 references B). Refactoring will scan through the project files in (and only in) your workspace. The reason is that the model element(s) you attempt to refactor might be in-used by another project file. Refactoring may need to modify such project by adding/updating the project and element reference, or to stop you from continuing if a potential error may occur.
   - If you work as a team, and is using a version control system (VPository/Teamwork Server/Other repository types), make sure the 'related projects' are all checked out. Refactoring will not and cannot detect the project files in your server.
5. Click **Yes** to continue.

6. You are prompted for previewing the changes. Click **Yes** if you want to preview changes. If you click **No**, refactoring will continue. If you click **Cancel**, refactoring will be cancelled.

7. If you clicked **Yes**, you see the window below, listing out the items to be refactored and the actions that will be performed after the refactoring.



Previewing changes

Refactor not only moves the selected elements to referenced project, but also those related elements. Here are several typical cases:

- Connectors that connect from/to the selected element.
- Model elements that takes the selected element as type (e.g. If class 'X' has an attribute 'Xa', which has the selected element as type, class 'X' will be refactored).
- Member-typed elements (e.g. Refactoring a class will cause its attributes and operations to be refactored. Refactoring an entity will cause its columns to be refactored).

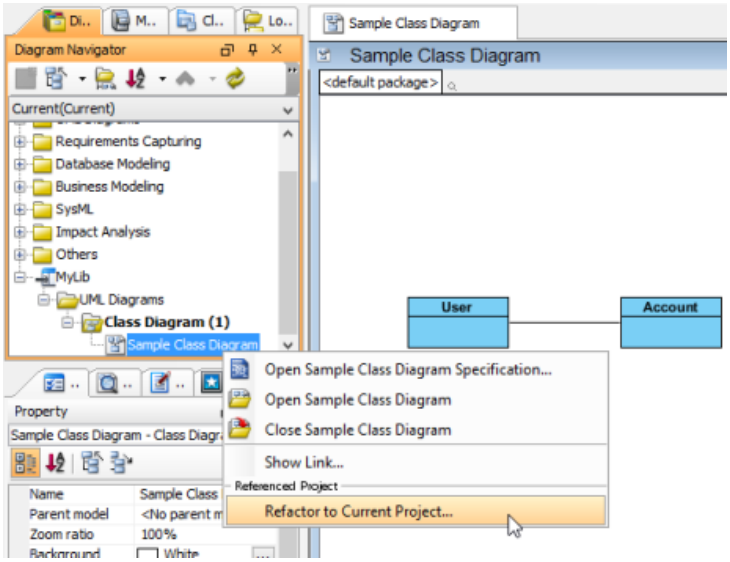
Notice that the searching of relevant project data is done recursively, meaning that whenever a model element is found relevant to any element selected to be refactored, that element will undergo the same checking performed to look for the other relevant elements, until the end. If you accept the changes, click **Continue** at the bottom right corner to continue. Otherwise, click **Cancel** at bottom right to terminate the refactoring.

8. Once the refactoring is complete, the project will reopen itself. The refactored element is now in your project. You can find them in the **Model Explorer**.

Refactor diagram from reference project

To refactor diagram means to refactor the diagram as well as the shapes on the diagram. As the steps are pretty close to refactoring model element, as described above, please read refactor model elements before reading this section.

1. In **Diagram Navigator** or **Model Explorer**, locate the diagram you want to get from reference project.
2. Right click on the diagram node and select **Refactor to Current Project...** from the popup menu.



Refactor class diagram

3. Click **Yes** when you are prompted the note.
4. You are prompted for previewing the changes. Click **Yes** if you want to preview changes. If you click **No**, refactoring will continue. If you click **Cancel**, refactoring will be cancelled.
5. Refactor not only moves the selected elements to referenced project, but also those related elements. If you clicked **Yes**, read step 6 in the previous section to learn how to carry on.
6. Once the refactoring is complete, the project will reopen itself. The refactored diagram is now in your project. Notice that the shapes in diagram should be referencing the model elements in reference project.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Model element nicknaming

Nickname enables you to creat multiple name and documentation set for your model. You can keep multiple language sets of your model by using the translation feature. In this chapter, both the nickname and translation feature will be covered.

What is nickname?

Introduce element nickname support.

Configure nickname

Shows you how to create a nickname or a language.

Using nickname

Shows you how to work with a new nickname and how to switch between nicknames.

Translation

Shows you how to make use of the translation feature to maintain model with multiple language.

Export and import word document of nickname

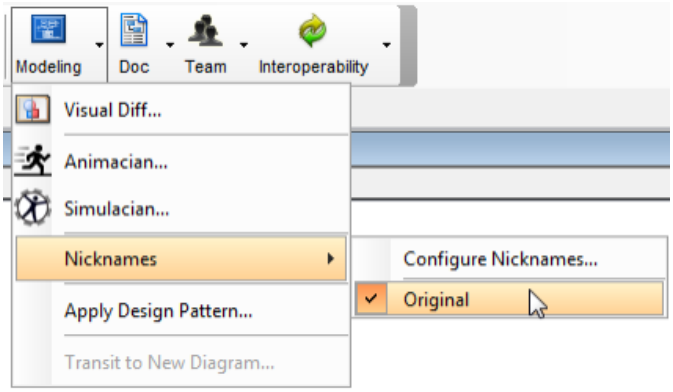
Export a Word document of nickname and nick-documentation, give it to a translator to perform translation, and import the work back to the tool.

What is nickname?

We all have a name, and we may have multiple names such as nicknames or names in other languages. This is the same for your Agilian project content. While we have applied certain language in naming and documenting model elements, we may have the need to model with another language to satisfy the readers of model. The nickname feature is designed to let you define multiple language sets for a model. Further to the definition of nickname, you also can make use of the translate function to translate your work into another language.

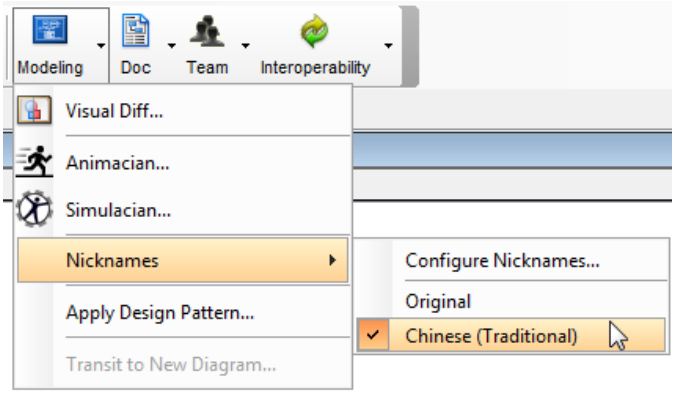
One model element can have one **Original** name and multiple nicknames, and the same for documentation. With nickname, you can define and view different names without affecting the original name of model elements. You can disable the effect of nickname anytime by switching to **Original** nickname. Features that related to code generation will always use **Original** name, i.e. changing Class's name in other nicknames will not affect the generated code.

The following screenshot shows the diagram in original name:



Original name

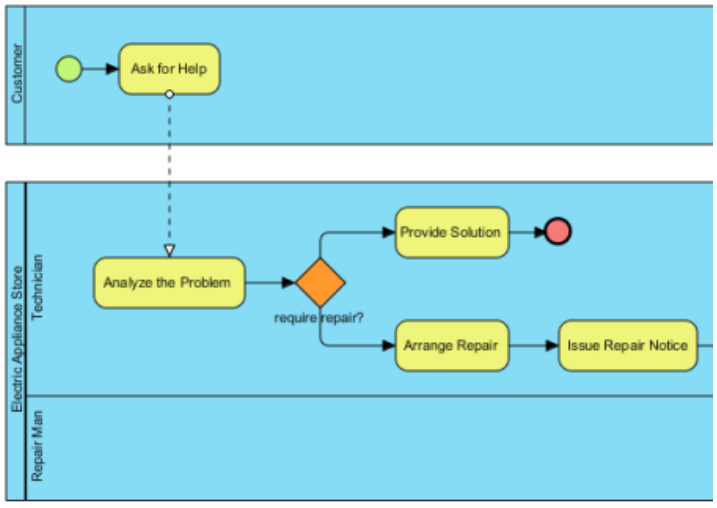
After you define a nickname and rename model element, the diagram will be switched to the selected nickname.



Select a defined nickname

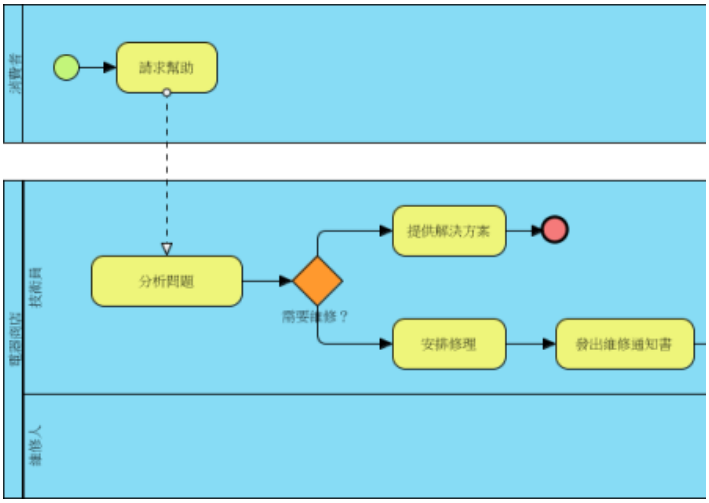
Multi-national team

If you are working in a team and your members using different languages, you can define model elements name and documentation in multiple languages. Each member can select their own language for modeling or view diagrams. The following example demonstrates the **Business Process Diagram** in English and Traditional Chinese respectively:



Original version of a business process

You can create a *Traditional Chinese* nickname and rename the model elements:



Traditional Chinese version of a business process

From now onwards, you can switch between English (Original) and Traditional Chinese anytime, or even create more nicknames.

Related Resources

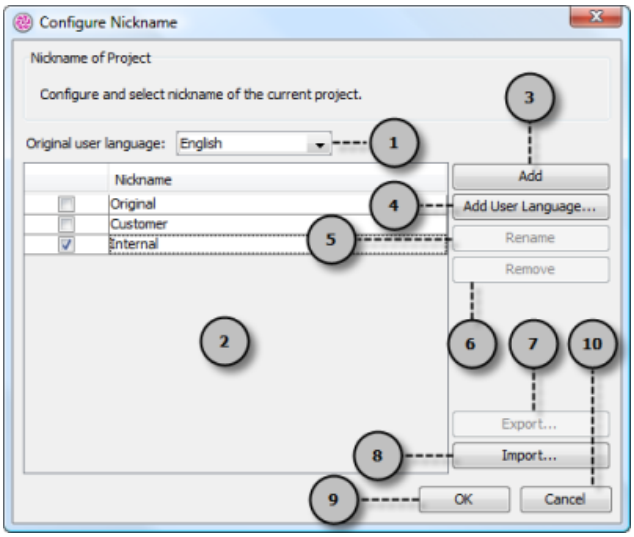
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Configure nickname

You can add a nickname through **Modeling > Nicknames > Configure Nicknames...** on the toolbar, or through the **View** menu from the main menu. By adding a nickname, you can start editing the names and documentations of model elements under the new nickname.

Overview of Configure Nickname dialog box



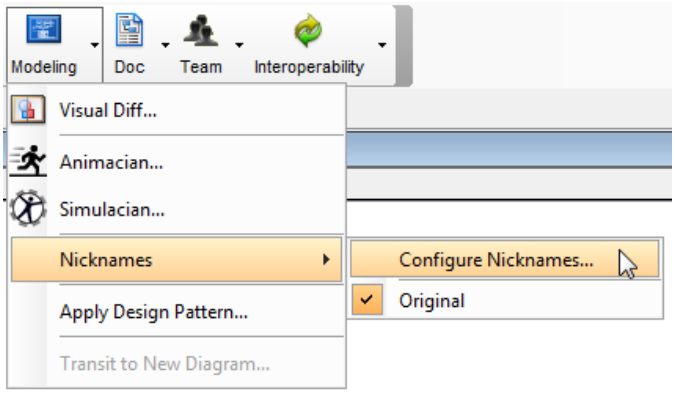
An overview of Configure Nickname dialog box

No.	Name	Description
1	Original user language	The language (e.g. English) of the Original nickname The selection only affects the outcome of translation.
2	Nickname list	List all the nicknames.
3	Add	Click to add a nickname with a name.
4	Add user language	Click to add a nickname with selected language.
5	Rename	Click to rename a chosen nickname.
6	Remove	Click to delete a chosen nickname.
7	Export	Click to export an XML file that contains information about the original name and nickname of the name and documentation of model elements that are named differently in nickname.
8	Import	Click to import the XML exported from **Configure Nickname** dialog box.
9	OK	Click to apply the nickname configuration and close this dialog box.
10	Cancel	Click to close the dialog box without applying the changes.

Description of Overview dialog box

Adding nickname

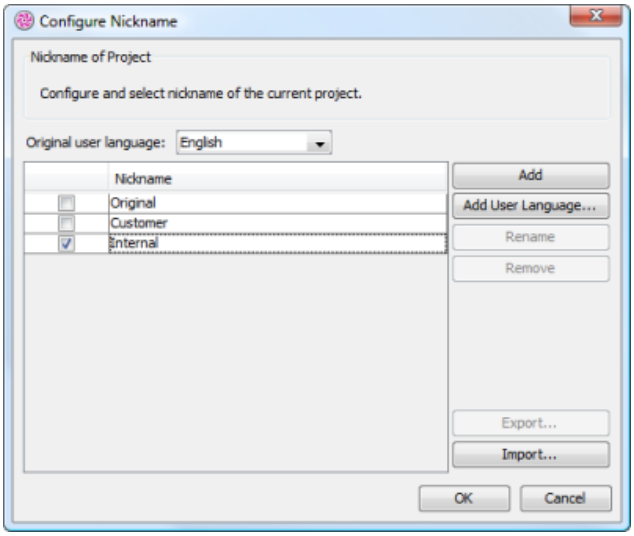
1. Select **Modeling > Nicknames > Configure Nicknames...** from the toolbar.



Add a nickname

NOTE: Alternatively, you can access the same function by selecting **View > Nicknames > Configure Nicknames...** from the main menu.

2. The current working copy is by default in **Original** nickname, with English as user language. Click **Add** button in the **Configure Nickname** dialog box.
3. In the **Input** dialog box, enter the name of the nickname set and click **OK** to confirm. Click **OK** to close the **Configure Nickname** dialog box.



Two nicknames are added

Renaming nickname

1. Select **Modeling > Nicknames > Configure Nicknames...** from the toolbar.

NOTE: Alternatively, you can select **View > Nicknames > Configure Nicknames...** from the main menu.

2. In the **Configure Nickname** dialog box, select the nickname to rename. Click **Rename**.

NOTE: You are not allowed to rename the original nickname.

3. In the **Input** dialog box, enter the name of the nickname set and click **OK** button to confirm. Click **OK** to close the **Configure Nickname** dialog box.

Removing nickname

1. Select **Modeling > Nicknames > Configure Nicknames...** from the toolbar.

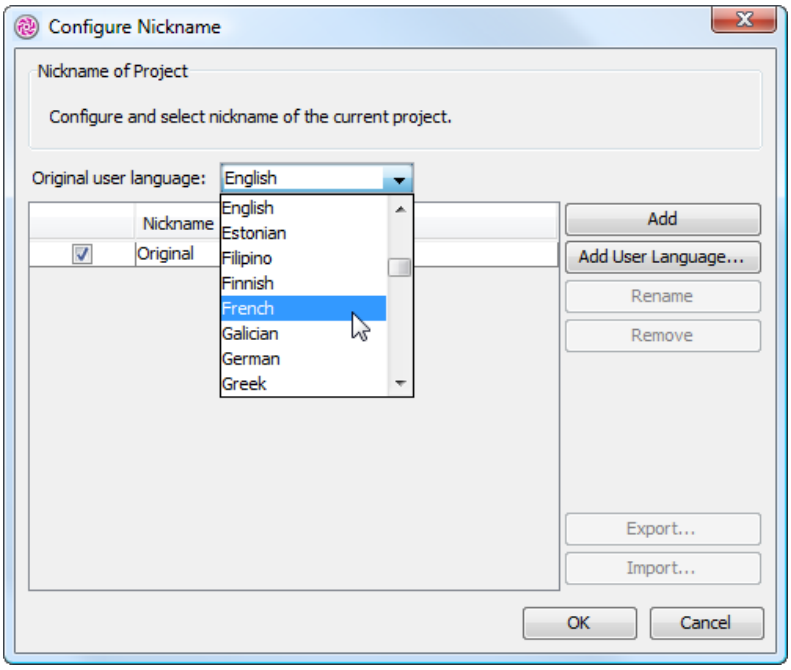
NOTE: Alternatively, you can select **View > Nicknames > Configure Nicknames...** from the main menu.

2. In the **Configure Nickname** dialog box, select the nickname to remove. Click **Remove** button. Click **Yes** when you are asked for confirmation.

NOTE: You are not allowed to remove the original nickname.

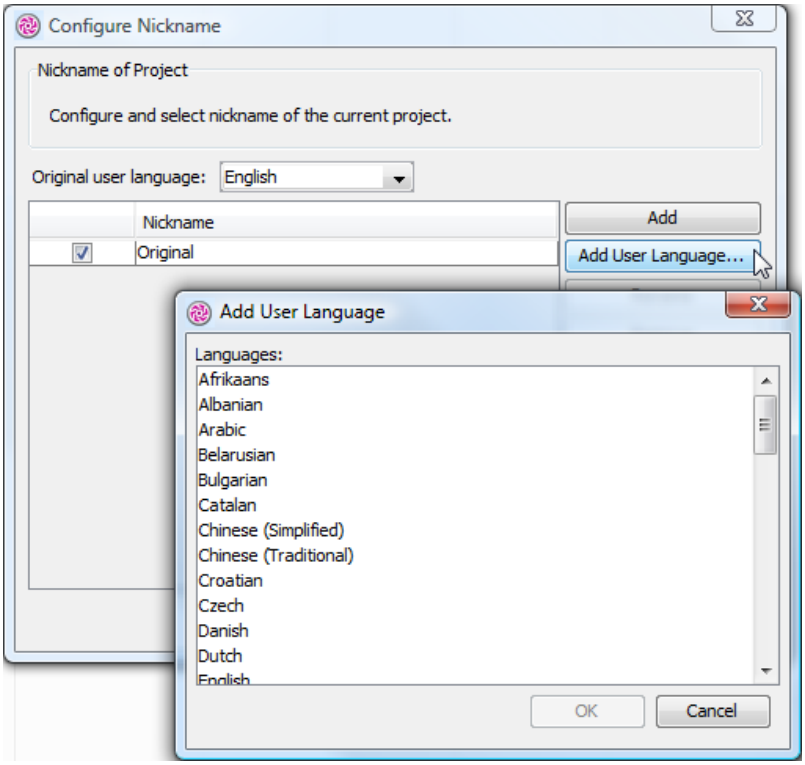
Specifying user language for the nickname

There must be an original nickname in every project, namely *Original*, representing the most standard version of language of project. You can specify the language for the original nickname, such as German. The language you have set affects the outcome of translation.



Select original user language

To add a nickname in specific language, click on the **Add User Language...** button and select a language.



Add a user language

Related Resources

The following resources may help you learn more about the topic discussed in this page.

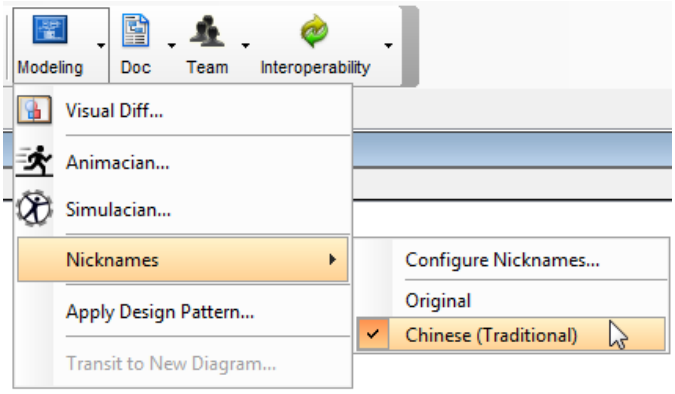
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Using nickname

Once you have defined a nickname, you can start updating your model by entering the new names and documentations of model elements.

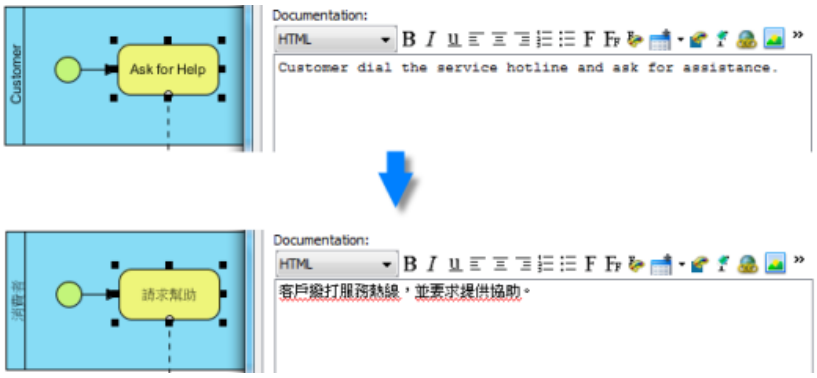
Start updating elements' nickname

1. Select **Modeling > Nicknames** and then select a nickname to work with from the toolbar.



Select a nickname to work with

2. Start renaming model elements and updating their documentation. The changes you made will only apply to the selected nickname.

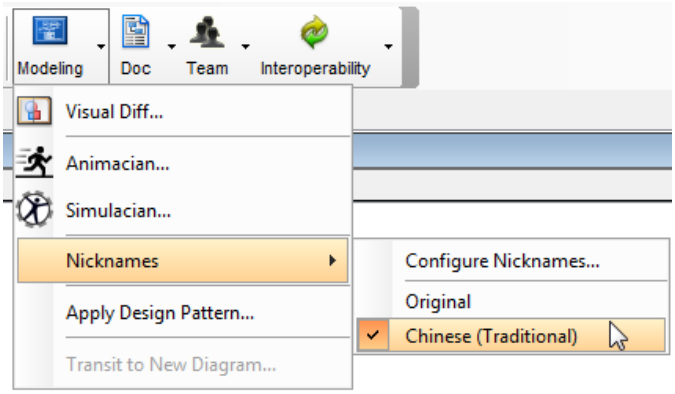


Update the name and documentation of elements under a nickname

Switching between nicknames

The names and documentations of model elements are language specific. This means that, the change you made applies only to a specific nickname. Once you have switched to another nickname, the names and documentations of model elements will be updated to show the definition under the new nickname.

To switch between nicknames, select **Modeling > Nicknames** and then select a nickname to switch to from the toolbar.



Switch nickname

Related Resources

The following resources may help you learn more about the topic discussed in this page.

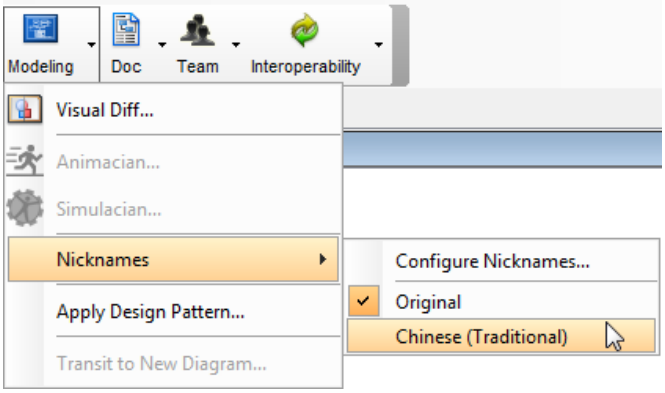
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Export and import word document of nickname

Exporting and importing the nickname of your diagrams to word document enables you to manage translation with ease. After all, you can update the nickname of exported Word document and import it back to synchronize the changes to the nickname of model elements, especially when you ask your team member (or your translator) to translate the project without providing the project to him/her.

In the following steps, let's try to export your project to word document and import the translated word document back to Agilian:

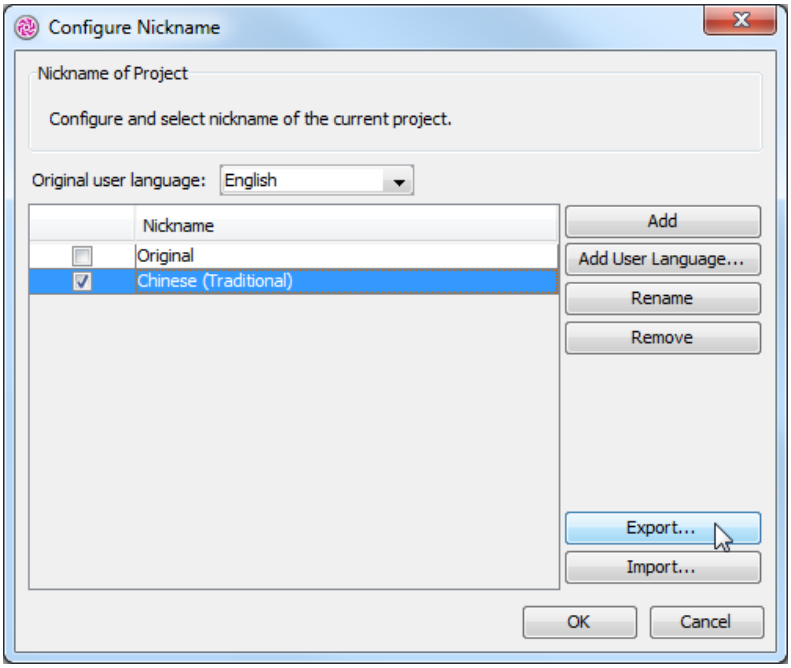
1. Switch your project to your target nickname. Select **Modeling > Nicknames > [your target nickname]** from the toolbar.



Switch your project into Chinese (traditional)

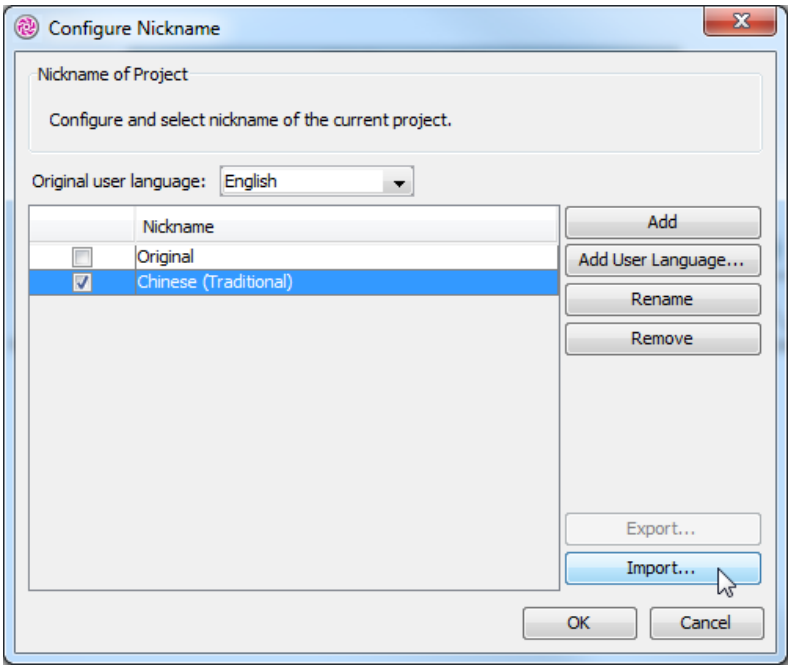
> **NOTE:** To create a nickname, select **Modeling > Nicknames > Configure Nicknames...** on the toolbar.
> Click **Add** to enter a nickname or click **Add User Language...** button to select one.

2. To export your project's model elements, select **Modeling > Nicknames > Configure Nicknames...** from the toolbar.
3. In **Configure Nickname** dialog box, click **Export** button.



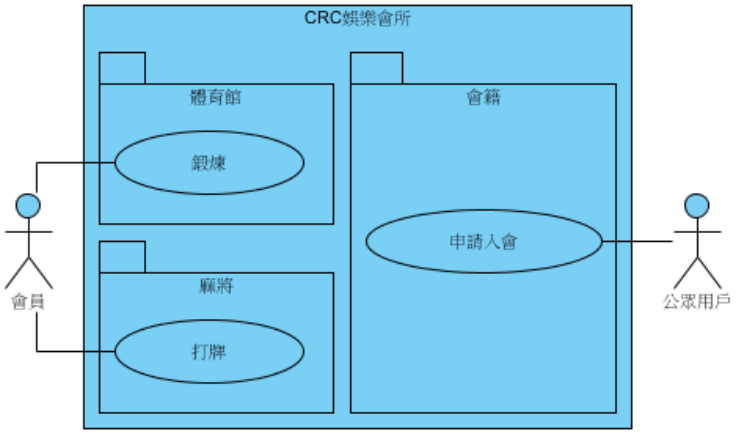
Export your project's model elements

4. In **Save** dialog box, select a directory for saving it as word document in your computer. Click **Save** button after you name the file.
5. Next, you send the word file to your team member (or your translator).
6. After s/he has translated the word file, you get it back from him/her and save it in your computer. Now, you can import the translated word document back to Agilian. Open your project with Agilian. Switch your project to your target nickname, and then open **Configure Nickname** dialog box.
7. In **Configure Nickname** dialog box, click **Import** button.



Import the translated word document back to Agilian

8. In **Open** dialog box, select the directory where you save your translated word document and click **Open** button. As a result, you can see your project is updated.



The name of your project's model elements has been updated to show the selected nickname

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Visual diff

You can use visual diff for comparing two diagrams, and to show their differences. The comparison can be made between diagrams in the same or different. In this chapter, you will see how to compare as-is and to-be BPD, logical and physical ERD with visual diff.

What is visual diff?

You will learn what is visual diff, and have a look at its configuration options.

Comparing as-is and to-be business process diagram

An example of comparing as-is and to-be BPD will be presented to help you understand how to use visual diff in practice.

Comparing logical and physical ERD

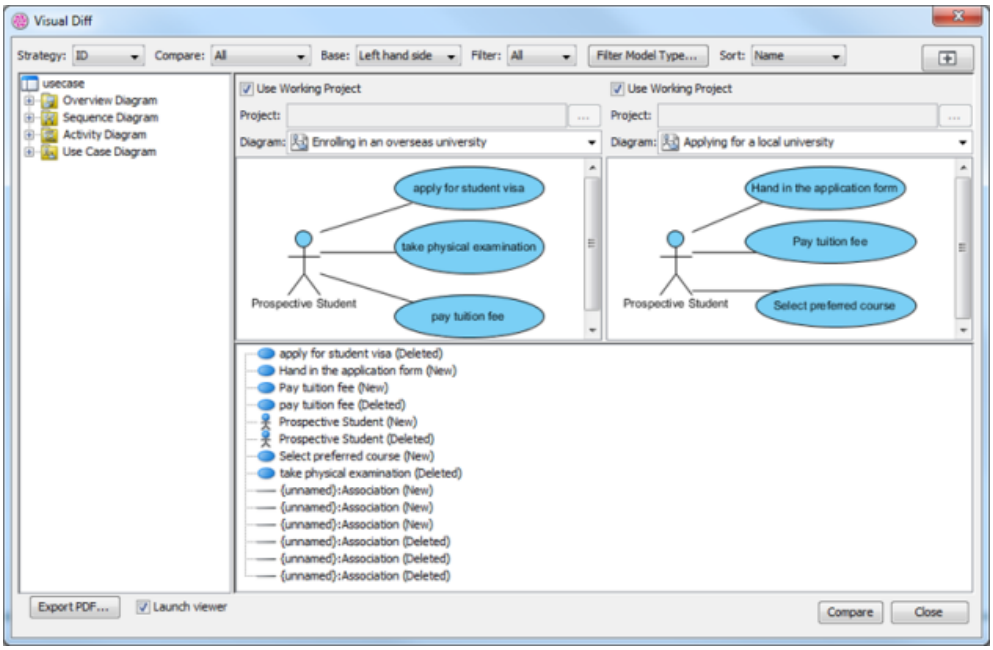
An example of comparing logical and physical ERD will be presented to help you understand how to use visual diff in practice.

What is visual diff?

The situation of comparing two diagrams is common. For example, comparing an ERD of conceptual model with an ERD of physical model and comparing a domain class diagram with a class diagram ready for implementation. Agilian allows you to compare the differences between diagrams in order to trace the changes between them.

Diagram comparison

With the feature of **Visual Diff**, two diagrams can be compared to recognize their differences. Changes, such as modification of properties (e.g. name) and addition/removal of containing models, thereby can be found easily.



The overview of Visual Diff dialog box

Various comparison strategies

A comparison strategy determines how two diagrams will be compared. Each strategy is used for its own specific purpose. You can select the appropriate strategy that suits your case most. The following is the description of strategies:

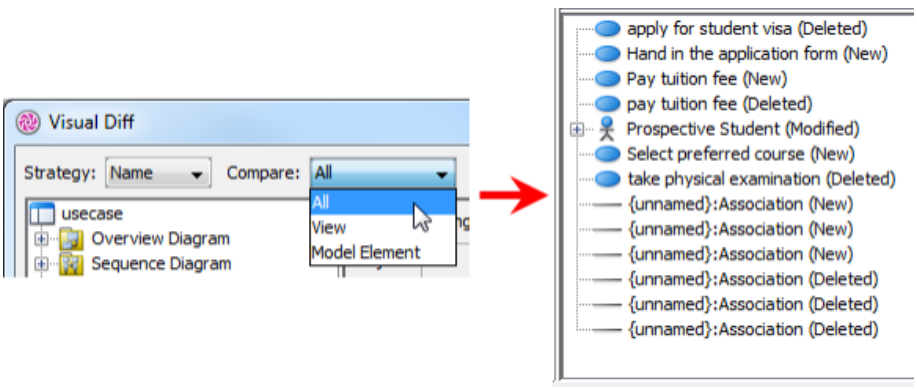
Strategy	Description
ID	Shapes will be matched base on their internal ID. Differences between shapes that have same ID will be displayed in the result pane. This strategy is usually used to visualize the changes of same shapes in two projects.
Name	Shapes will be matched base on their names. This strategy is useful when visualizing differences for external works. One of typical examples is to compare databases and class models.
Transitor	Shapes will be matched base on their transition established by Model Transitor. This way of comparison is usually used to visualize the differences between model elements.

Description of comparison strategies

Comparing view only, model element only, or both

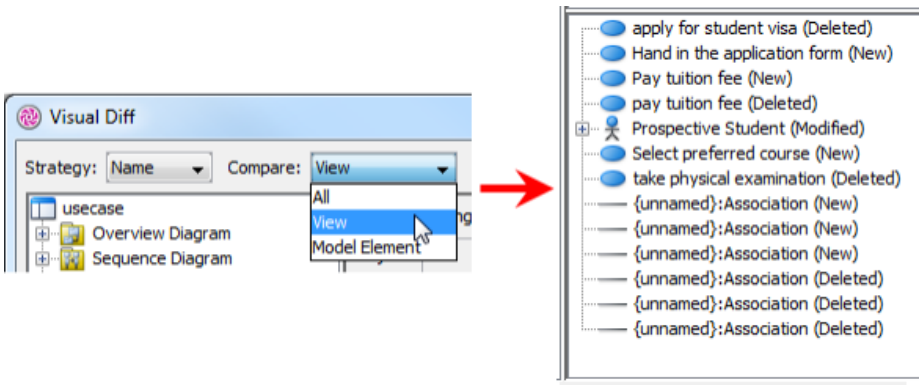
Comparison is divided into view, model element and both of them. The selected option from the drop-down menu of **Compare** determines the display for comparison. By selecting **View**, the differences in view settings, such as the coordination of shapes, will be considered as changes. By selecting **Model Element**, the differences in model element level, such as their names, will be considered as changes. By selecting **All**, the differences in both view settings and model element level are displayed.

The following screenshot displays both view and model element differences:



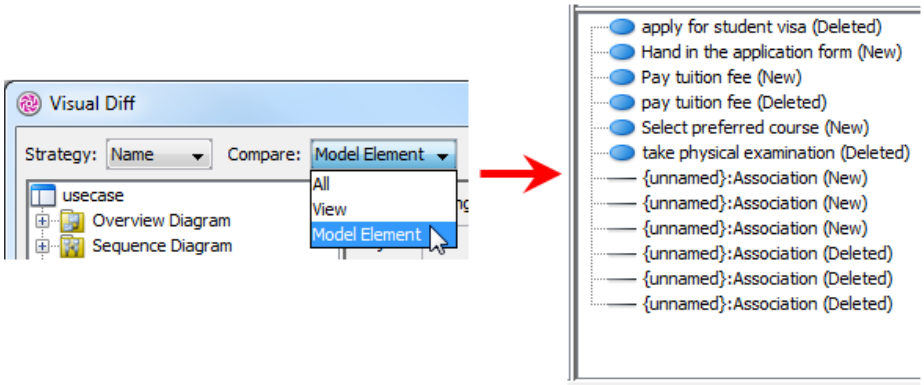
Compare All

The result of selecting **View** is shown as follows:



Compare View

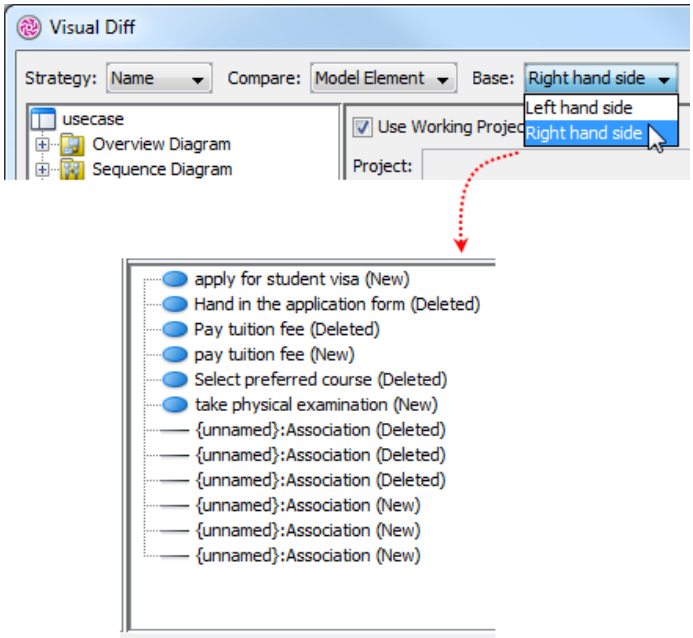
The result of selecting **Model Element** is shown as follows:



Compare Model Element

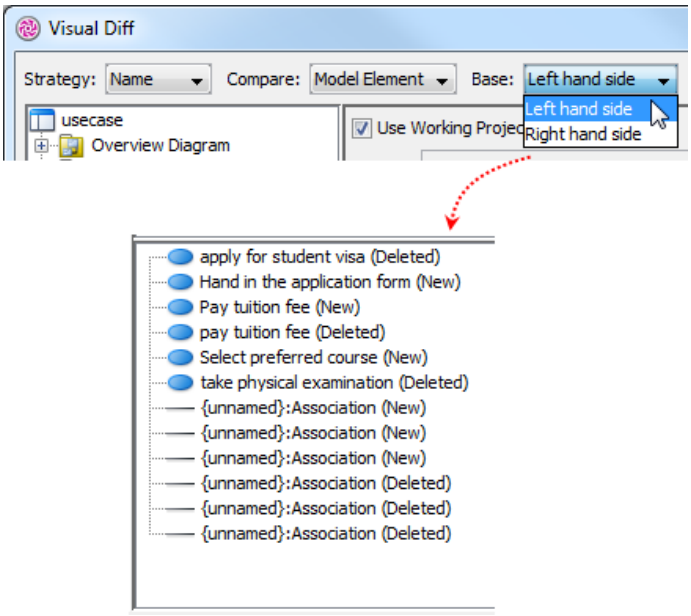
Comparing from the left to the right and vice versa

Comparisons are made between two diagrams, which are put on the left and the right hand side in the **Visual Diff** dialog box respectively. By default, comparison is based on left hand side, which means, if a shape does not exist on the left hand side but on the right hand side, the shape will be considered as newly added in the result pane. The base can be swapped from the right to the left, and vice versa. By doing so, the absence of shape on the left hand side will result in a report of deleted shape.



Compare diagrams with right hand side as base

The result of left hand side as base is shown below. The deleted model element is regarded as an addition (it indicates as **New**).

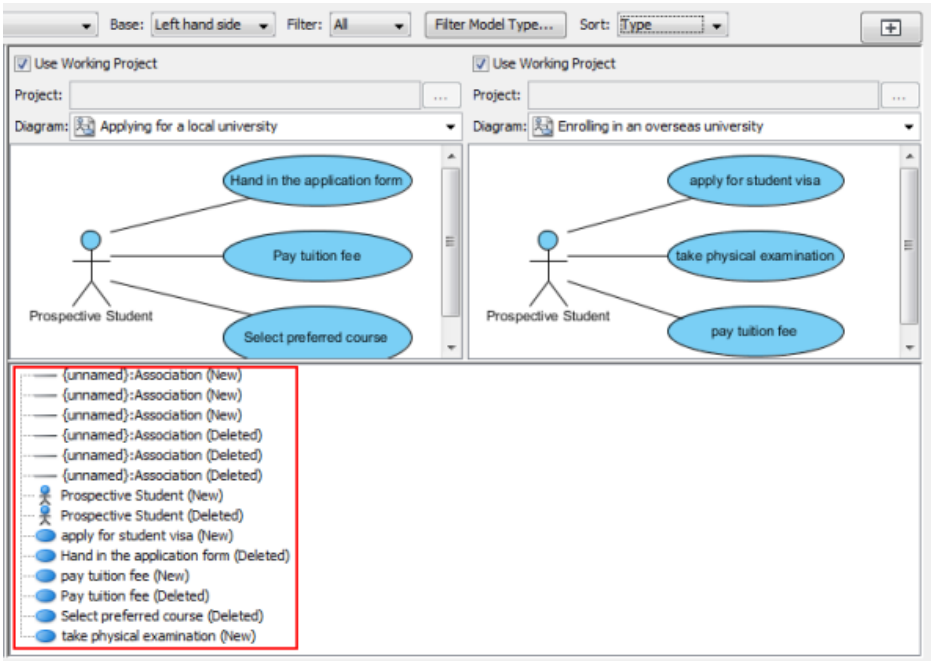


Compare diagrams with left hand side as base

Sorting the data on result pane

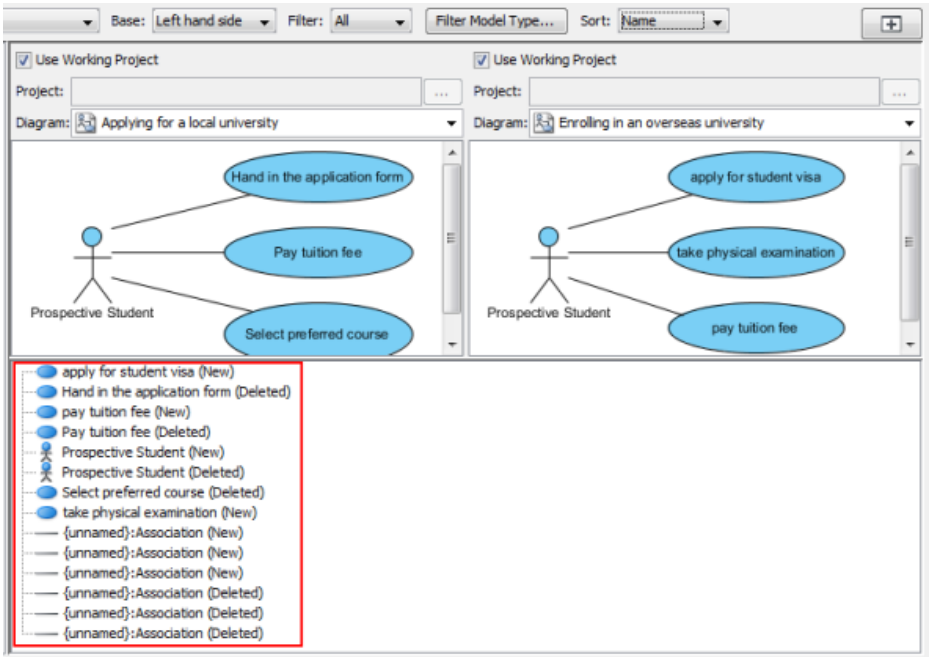
The selected option from the drop-down menu of **Sort** determines the order of data on the result pane. You can select sort by type, by name or by change type.

The result of selecting **Type** is shown as follows:



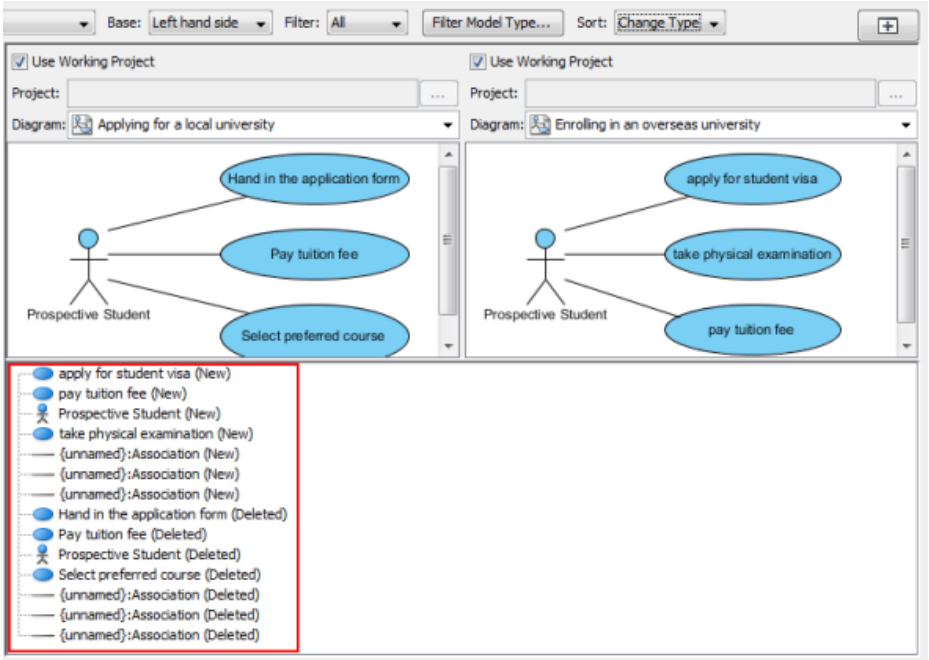
Sort by type

The result of selecting **Name** is shown as follows:



Sort by name

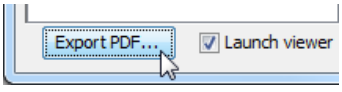
The result of selecting **Change Type** is shown as follows:



Sort by change type

Exporting a report of differences

A report of the current comparing diagrams can be saved in your computer as PDF. Click **Export PDF...** button on the bottom left corner of the dialog box. Select a directory for storing in **Export PDF** dialog box and then click **Save** button.



*Click **Export PDF...** button*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

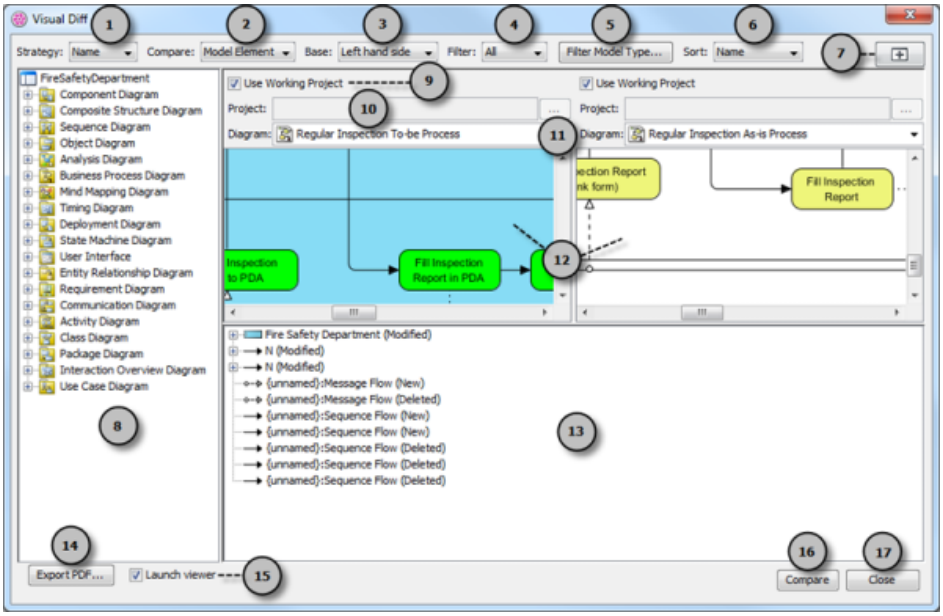
Comparing as-is and to-be business process diagram

In this page, two Business Process Diagrams are compared: one for modeling the As-is Process and another for modeling the To-be Process. The features of **Visual Diff** are applied in order to find the differences between them.

1. In *As-Is Process* diagram, select **Modeling > Visual Diff...** from the main menu.

> **NOTE:** Alternatively, you may start **Visual Diff** as follows:
> - Right click on diagram background and select **Utilities > Visual Diff...** from the pop-up menu.
> - Select **Modeling** > **Visual Diff** on the toolbar.

The overview of **Visual Diff** dialog box:



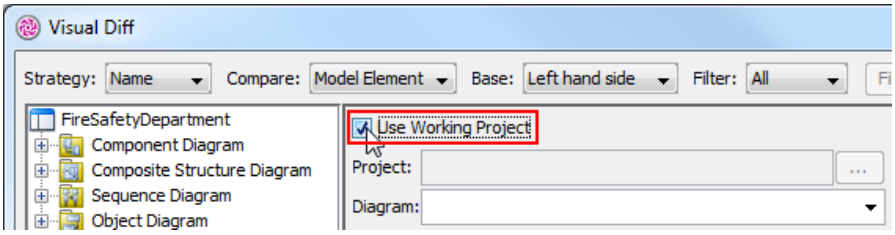
*The **Visual Diff** dialog box*

No.	Name	Description
1	Strategy	It determines how two diagrams will be compared. Select the appropriate strategy that suits your application most. **ID** : Shapes will be matched base on their internal ID. Differences between shapes that have same ID will be displayed in the result pane. This strategy is usually used to visualize the changes of same shapes in two projects. **Name** : Shapes will be matched base on their names. This strategy is useful when visualizing differences for external works. One of typical examples is to compare databases and class models. **Transitor** : Shapes will be matched base on their transition established by Model Transitor. This way of comparison is usually used to visualize the differences between model elements.
2	Compare	It determines how two diagrams will be displayed for comparison. **All**: Both view and model elements are displayed. **View** : Differences, such as coordinates, width, height and color of shapes, are displayed. **Model Element** : Differences, such as model name, are displayed.
3	Base	It determines which diagram is used as a base for comparison. **Left hand side**: Comparison is based on the diagram on the right hand side. For example, if there is a shape absent on the left hand side, but appear on the right hand side, the shape is said to be a new shape. **Right hand side** : Comparison is based on the diagram on the right hand side. For example, if there is a shape absent on the left hand side, but appear on the right hand side, the shape is said to be a removed shape.
4	Filter	It determines what kind of comparing data will be displayed on the result pane. **All**: Display all kinds of differences including the addition, modification and removal of shapes. **New**: Display only results about the addition of shape and then hide the rest. **Modified**: Display only results about the modification of shapes and then hide the rest. **Deleted**: Display only results about the removal of shapes and then hide the rest.
5	Filter Model Type...	It determines what type of model elements will be shown on the result pane. When Filter Model Type dialog box pops out, uncheck the type of model elements if you don't want it to be shown; otherwise, check it if you want it to be shown.
6	Sort	It determines how the result of comparison be displayed on Result pane. You can select sort by Type, by Name or by Change Type. **Name**: All model elements are sorted by their name. **Change Type**: All model elements are changed their type.

		Type: All model elements are sorted by their type.
7	Maximum	Click it to enlarge the **Visual Diff** dialog box.
8	Diagram list	It lists all diagrams on projects selected in the left hand side and the right hand side respectively.
9	Use Working Project	Check it to select a diagram for comparison within the current working project.
10	Project	Select and open a project directory for comparison.
11	Diagram	Select and open a specific diagram for comparison after selecting a project.
12	Display pane	A pane that has two sides. Each side displays a diagram out of a project.
13	Result pane	The differences of the two projects are shown here. When the word *New* is shown behind a task that means it is newly added, the word *Deleted* is shown behind a task that means it is deleted.
14	Export PDF...	Click it to save a report of the two compared diagrams in your computer as PDF.
15	Launch viewer	Open the exported file after you have export a PDF file. Check it to open the PDF file. If you uncheck it, nothing will happen even after you have exported a PDF file.
16	Compare	Click it to obtain the result of comparison.
17	Close	Close **Visual Diff** dialog box.

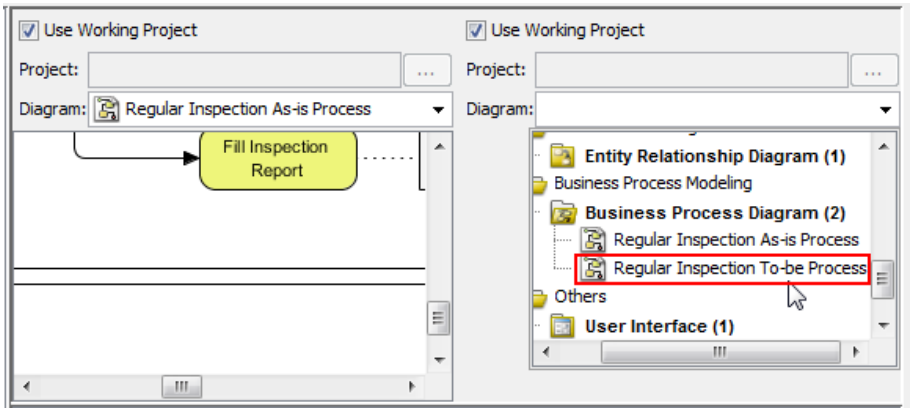
*The description of features in **Visual Diff** dialog box*

2. In **Visual Diff** dialog box, the left hand side window shows the currently opened diagram by default. You may remain it unchanged; otherwise uncheck **Use Working Project** on the left hand side if you want to select a diagram in other projects to compare with. Click **...** button in **Project** to select the directory of other projects. Similarly, check/ uncheck **Use Working Project** on the right hand side window.



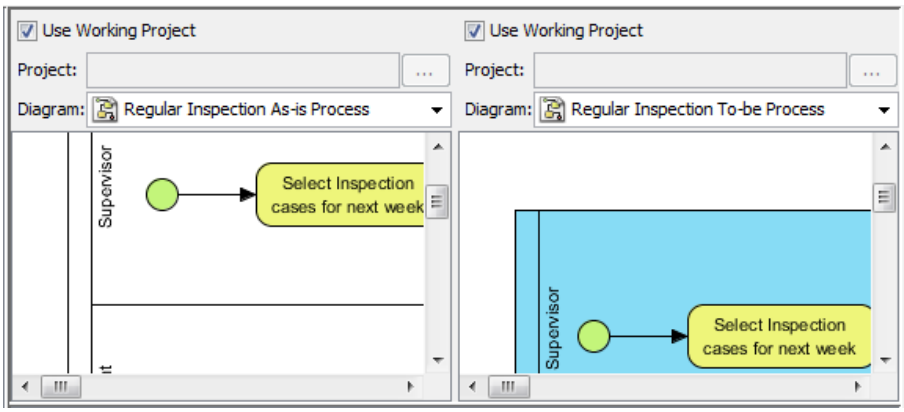
*Check **Use Working Project***

3. Select the *To-be Process* diagram to compare with from the drop-down menu of **Diagram**.



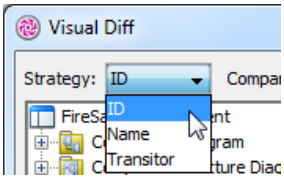
Select a diagram for comparison

The two diagrams are shown side by side on display pane. However, the ways of comparison has not yet been configured. Let's configure them one by one in the following steps.



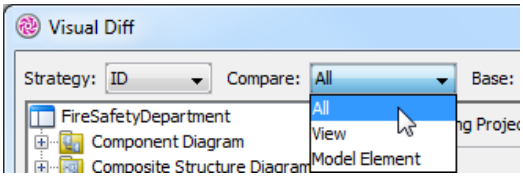
Two diagrams are selected

4. Select an option in **Strategy**.



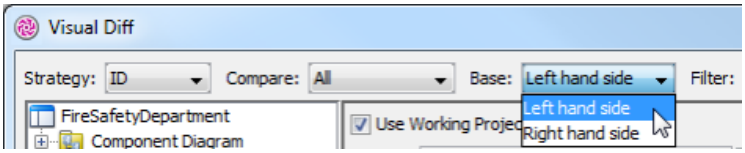
Select a strategy

5. Select an option in **Compare**.



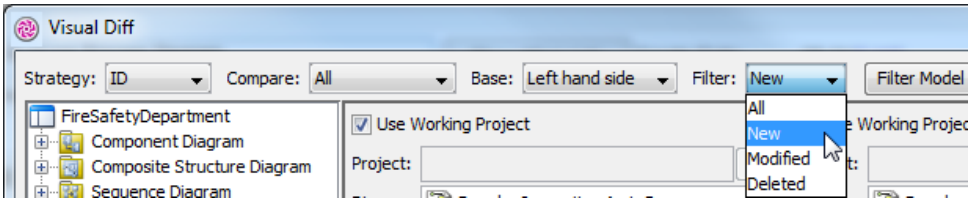
Select a compare

6. Choose an option in **Base**.



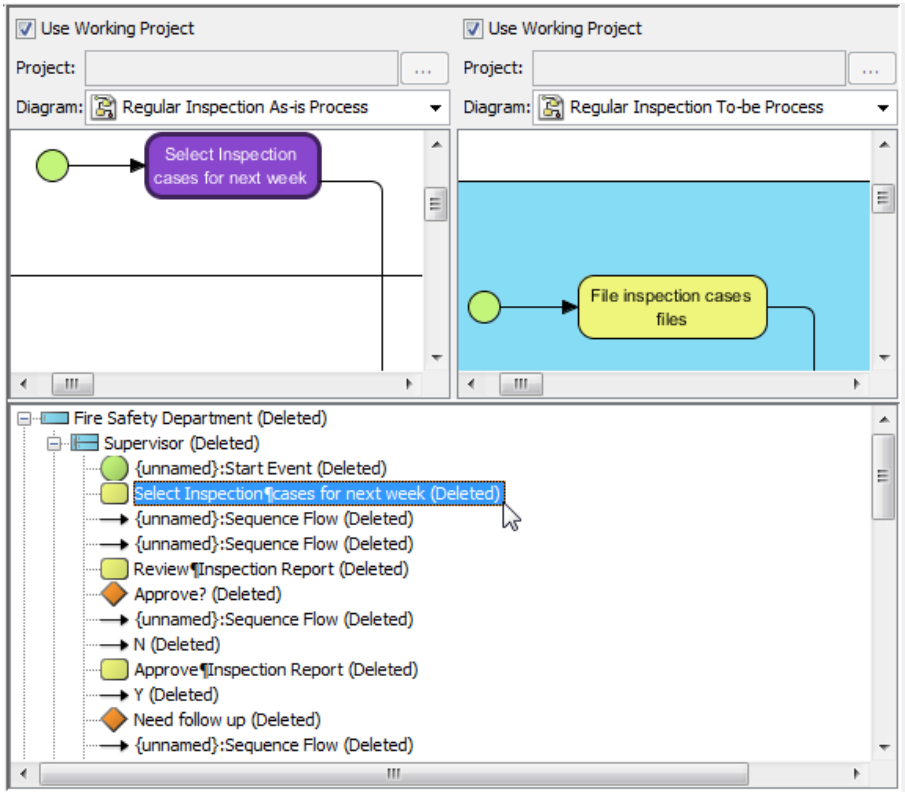
Choose a base

7. Select an option in **Filter**.



Select a filter

8. Once everything is set, the differences of the two diagrams will be shown on the result pane.
9. If you want to view a specific shape, you may click its node on the result pane, and then the selected shape will be painted in dark purple on the display pane.



Select a node on the result pane

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Comparing logical and physical ERD

Entity relationship diagram (ERD) represents a detailed picture of the entities needed for a business. In forward engineering, ERD will be transformed into a relational database eventually. There are at least two types of ERD – Logical and Physical. They are used in different stages of development, and are inter-related.

Logical ERD models information gathered from business requirements. Entities and relationships modeled in such ERD are defined around the business's need. The need of satisfying the database design is not considered yet.

Physical ERD represents the actual design of database. It deals with conversion from logical design into a schema level design that will be transformed into relational database. When modeling a physical ERD, Logical ERD is treated as base, refinement occurs by defining primary keys, foreign keys and constraints. Sometimes, relationships need to be resolved by introducing additional tables, like a Linked table for a many to many relationship.

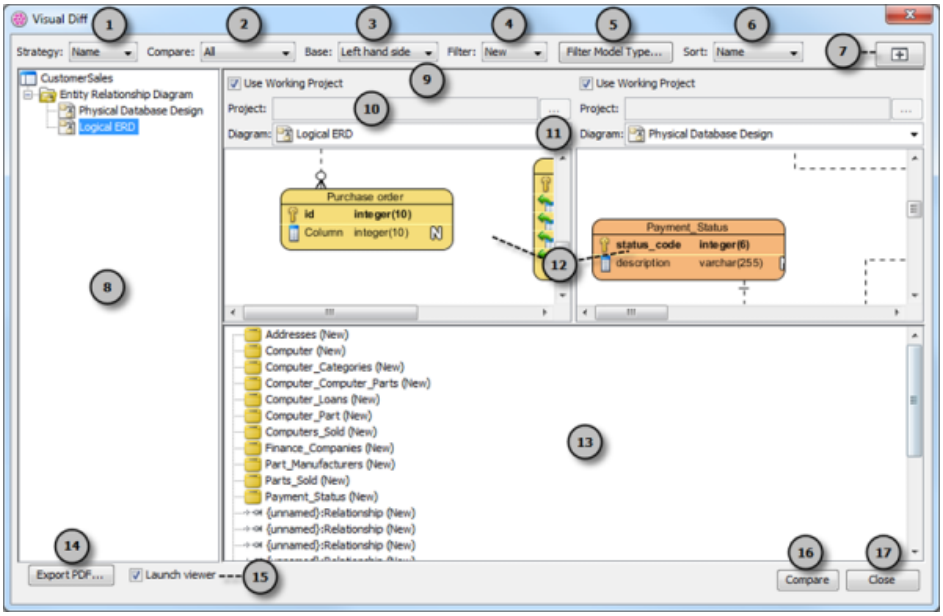
Since physical ERD and logical ERD represent the business requirement and database schema respectively, comparing physical and logical ERD helps to find out the differences between them in order to confirm the database is exactly following the initial business requirements regardless of the changes.

Two ERDs are compared: one for modeling the Logical Model and another one for modeling the Physical Model. With the features of **Visual Diff**, the differences between Logical and Physical ERD can be found easily.

1. In *Logical ERD* diagram, select **Modeling > Visual Diff...** from the main menu.

> **NOTE:** Alternatively, you can start **Visual Diff** as follows:
> - Right click on diagram background and select **Utilities > Visual Diff...** from the pop-up menu.
> - Select **Modeling > Visual Diff** on the toolbar.

The overview of **Visual Diff** dialog box:



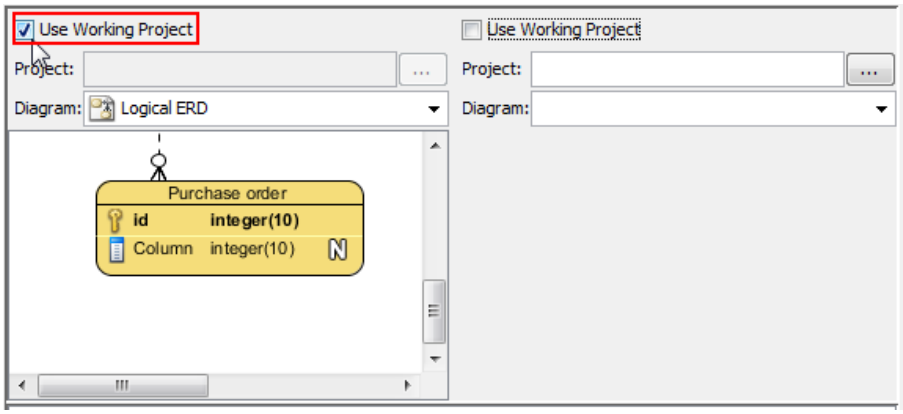
*The **Visual Diff** dialog box*

No.	Name	Description
1	Strategy	It determines how two diagrams will be compared. Select the appropriate strategy that suits your application most. **ID**: Shapes will be matched base on their internal ID. Differences between shapes that have same ID will be displayed in the result pane. This strategy is usually used to visualize the changes of same shapes in two projects. **Name**: Shapes will be matched base on their names. This strategy is useful when visualizing differences for external works. One of typical examples is to compare databases and class models. **Transitor**: Shapes will be matched base on their transition established by Model Transitor. This way of comparison is usually used to visualize the differences between model elements.
2	Compare	It determines how two diagrams will be displayed for comparison. **All**: Both view and model elements are displayed. **View**: Differences, such as coordinates, width, height and color of shapes, are displayed. **Model Element**: Differences, such as model name, are displayed.
3	Base	It determines which diagram is used as a base for comparison. **Left hand side**: Comparison is based on the diagram on the right hand side. For example, if there is a shape absent on the left hand side, but appear on the right hand side, the shape is said to be a new shape. **Right hand side**: Comparison is based on the diagram on the right hand side. For example, if there is a shape absent on the left hand side, but appear on the right hand side, the shape is said to be a removed shape.

4	Filter	It determines what kind of comparing data will be displayed on the result pane. **All**: Display all kinds of differences including the addition, modification and removal of shapes. **New**: Display only results about the addition of shape and then hide the rest. **Modified**: Display only results about the modification of shapes and then hide the rest. **Deleted**: Display only results about the removal of shapes and then hide the rest.
5	Filter Model Type...	It determines what type of model elements will be shown on the result pane. Uncheck the type of model elements if you don't want it to be shown; otherwise, check it if you want it to be shown.
6	Sort	It determines how the result of comparison be displayed on Result pane. You can select sort by Type, by Name or by Change Type. **Name**: All model elements are sorted by their name. **Change Type**: All model elements are changed their type. **Type**: All model elements are sorted by their type.
7	Maximum	Click it to enlarge the **Visual Diff** dialog box.
8	Diagram list	It lists all diagrams on projects selected in the left hand side and the right hand side respectively.
9	Use Working Project	Check it if you want to select a diagram for comparison within the current working project.
10	Project	Select and open a project directory for comparison.
11	Diagram	Select and open a specific diagram for comparison after selecting a project.
12	Display pane	A pane that has two sides. Each side displays a diagram out of a project.
13	Result pane	The differences of the two projects are shown here. When the word *New* is shown behind a task that means it is newly added, the word *Deleted* is shown behind a task that means it is deleted.
14	Export PDF...	Click it to save a report of the two compared diagrams in your computer as PDF.
15	Launch viewer	Open the exported file after you have export a PDF file. Check it to open the PDF file. If you uncheck it, nothing will happen even after you have exported a PDF file.
16	Compare	Click it to obtain the result of comparison.
17	Close	Close **Visual Diff** dialog box.

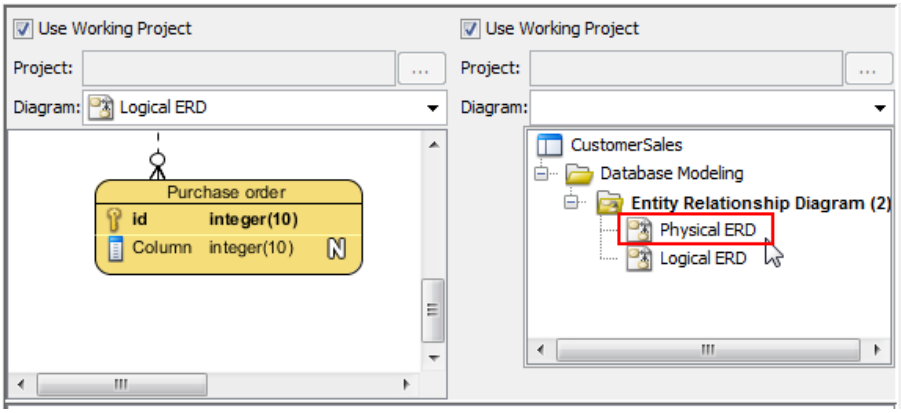
Description of various parts of Visual Diff dialog box

2. In **Visual Diff** dialog box, the left hand side window shows the currently opened diagram by default. You may remain it unchanged; otherwise uncheck **Use Working Project** on the left hand side if you want to select a diagram in other projects to compare with. Click **...** button in Project to select the directory of other projects. Similarly, check/ uncheck **Use Working Project** on the right hand side window.



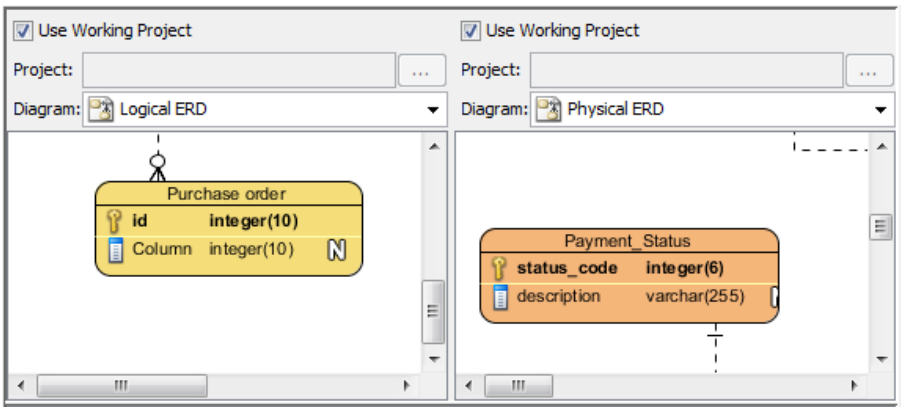
*Check **Use Working Project***

3. Select the *Physical ERD* to compare with from the drop-down menu of **Diagram**.



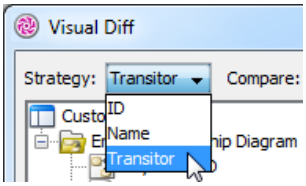
Select a diagram

The two diagrams are shown side by side on display pane. However, the ways of comparison has not yet been configured. Let's configure them one by one in the following steps.



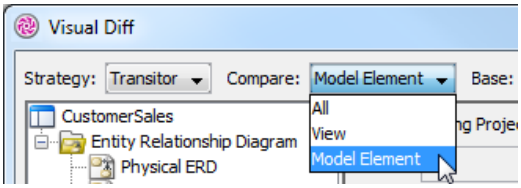
Two diagrams are selected

4. Select an option in **Strategy**.



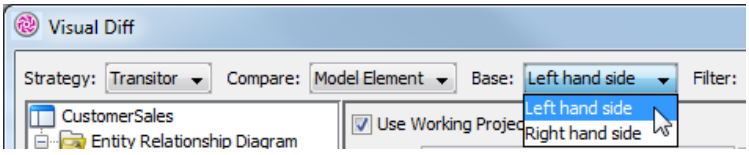
Select a strategy

5. Select an option in **Compare**.



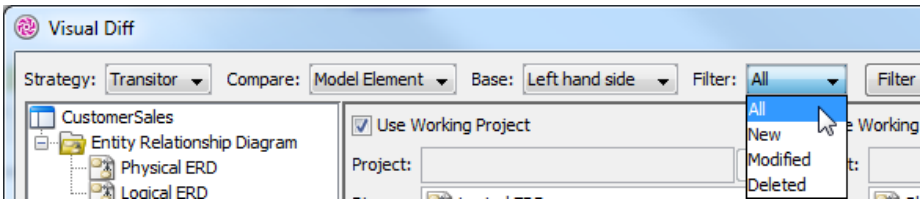
Select a compare

6. Choose an option in **Base**.



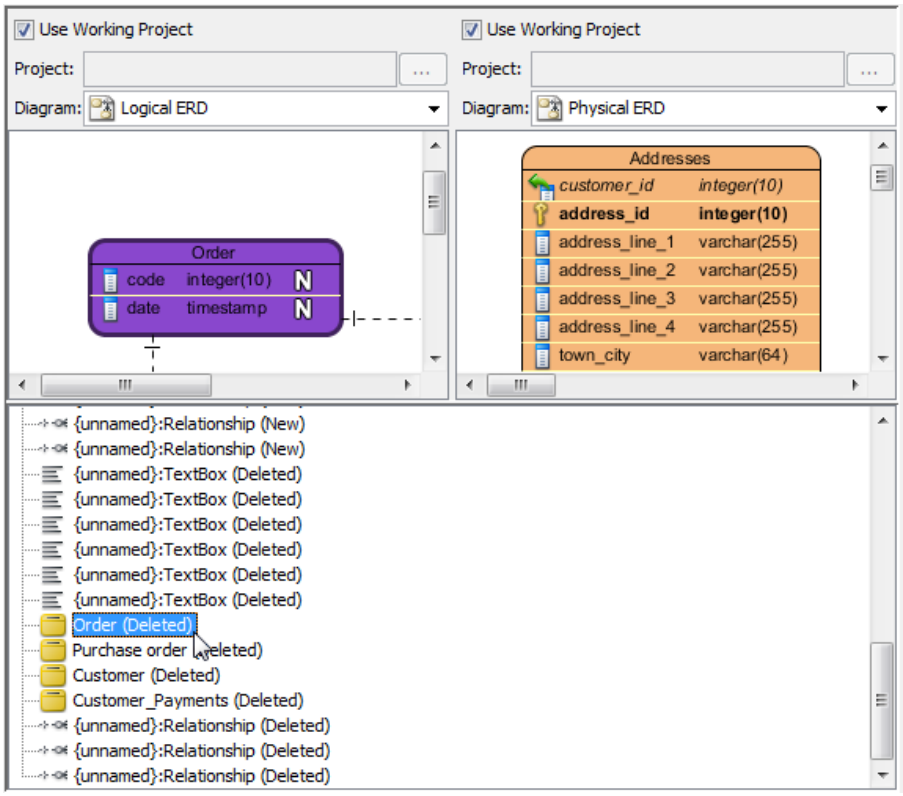
Select a base

7. Select an option in **Filter**.



Select a filter

8. Once everything is set, the differences of the two diagrams will be shown on the result pane.
9. If you want to view a specific shape, you may click its node on the result pane, and then the selected shape will be painted in dark purple on the display pane.



Select a node in the result pane

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Using design pattern

Design pattern is a part of diagram that can be used in many different diagrams. In this chapter, you will learn what design pattern is, and how to apply it in your design.

Defining design pattern

You need to draw the pattern, and define it afterwards.

Applying design pattern

Defined patterns can be applied to your project, or another project (through an export and import of pattern).

Synchronize design pattern with teamwork server

Patterns can be shared among team members through the team collaboration support.

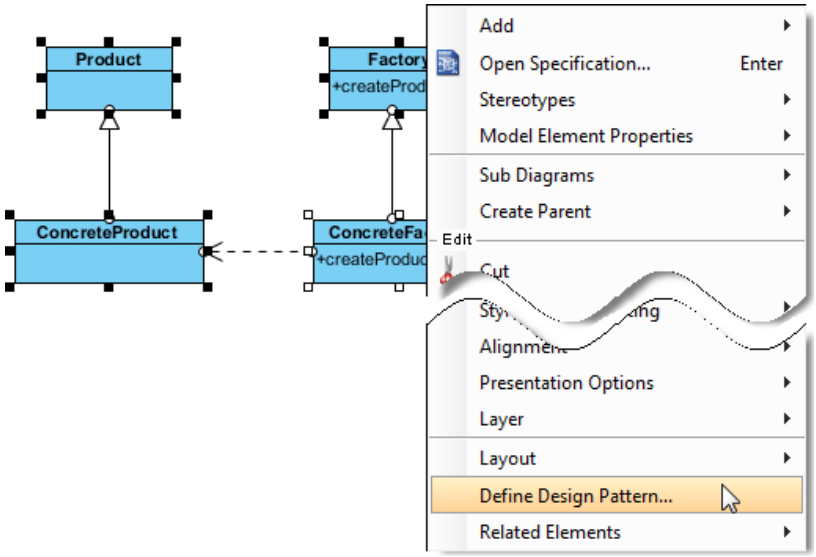
Defining design pattern

In Agilian, design pattern is a part of diagram that can be used in many different diagrams, thus form a pattern. Design pattern typically shows the shapes and more importantly, the relationships between the shapes. You can define and reuse design pattern in your project, across projects, or share with your team members. You can define and apply design patterns on any kinds of diagram.

In order to apply a pattern, you need to define it first, and save it as a pattern file ready for being used. To define a pattern, draw the pattern on diagram. After that, you can save the pattern, which is the diagram content as a pattern file.

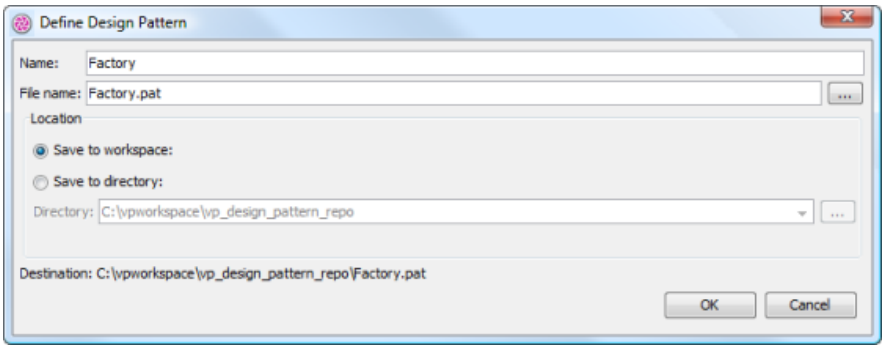
Defining design pattern

1. In the diagram, select the target shapes to be involved in pattern.
2. Right click on any selected shapes, select **Define Design Pattern...** from the popup menu.



Define design pattern

3. A design pattern is needed to save as a file. In the **Define Design Pattern** dialog, specify the name and file name for the pattern, with **.pat** as extension. You can save the pattern file to workspace for ease of sharing with other projects that will be opened in current workspace. Besides, you can save to another directory and share the **.pat** file with your team member for reusing. Click **OK** button to finish defining design pattern.



Name design pattern

Related Resources

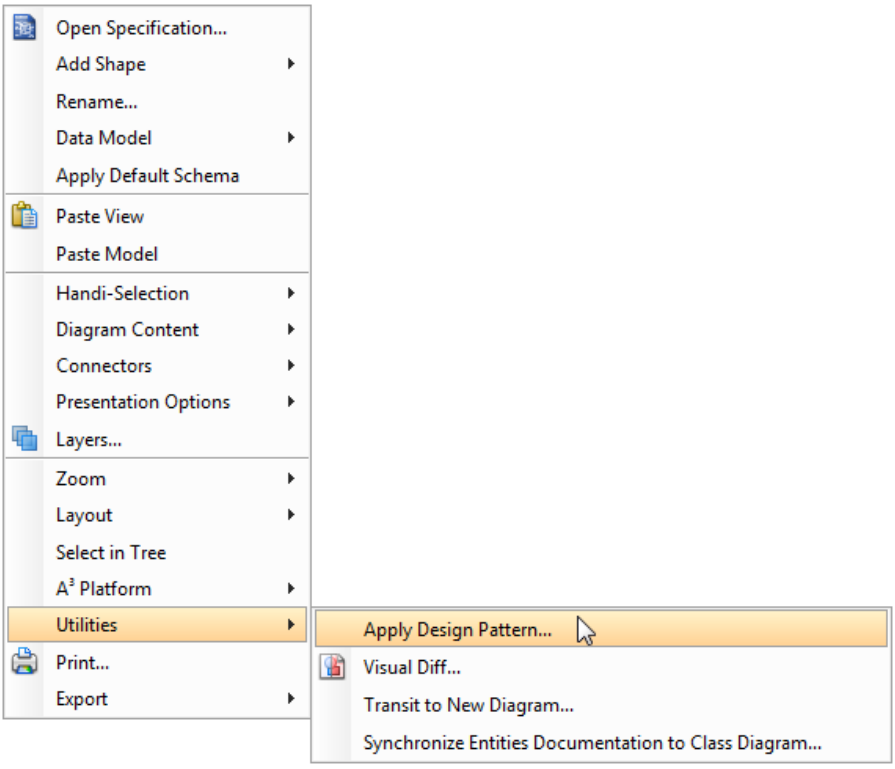
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Applying design pattern

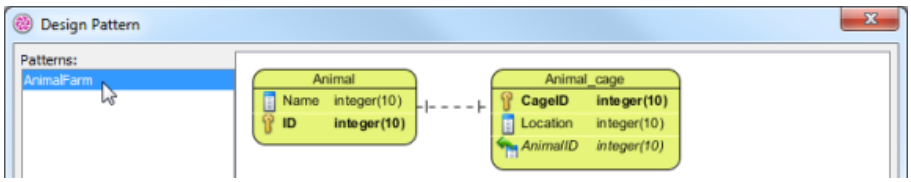
After you have previously defined a design pattern into your diagram, you can modify it to fit into your design. To apply a design pattern:

Open an existing diagram where you want to apply a design pattern or create a new diagram. Right click on the diagram background and select **Utilities > Apply Design Pattern...** from the pop-up menu.



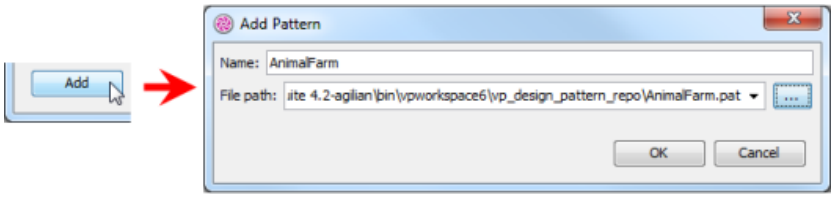
Apply design pattern

In **Design Pattern** dialog box, a list of defined patterns is listed on the left hand side of dialog box, select your target design pattern from the list and then the selected pattern will reveal on the right hand side of dialog box at once.



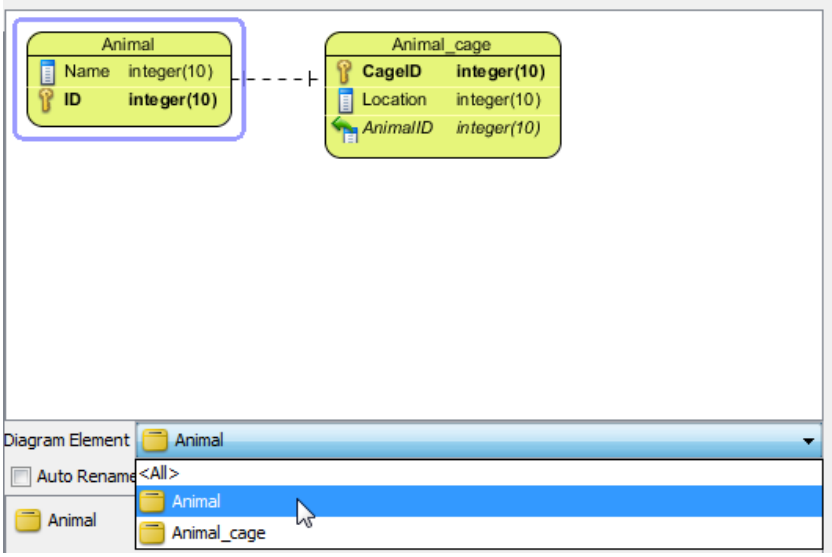
*Select a pattern in **Design Pattern** dialog box*

Alternatively, you can click **Add** button to import a .pat file into the list. Enter name and specify file path for the new pattern in the pop-up **Add Pattern** dialog box, and then click **OK** button to confirm.



*Click **Add** button to unfold **Add Pattern** dialog box*

For searching a specify shape on the defined pattern, select the shape from the drop-down menu of **Diagram Element** to filter the list.



*Filter pattern element from the drop-down menu of **Diagram Element***

Finally, click **OK** button in **Design Pattern** dialog box to confirm applying the selected pattern.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

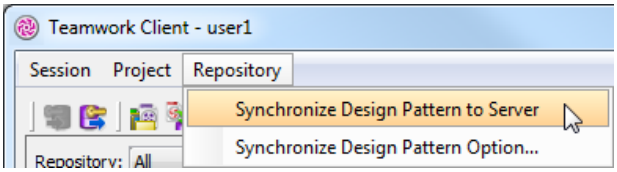
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Synchronize design pattern with teamwork server

Teamwork server can synchronize design patterns defined and saved to workspace, you can then share the design patterns with your team members. This feature is available to Visual Paradigm Teamwork Server, Subversion, CVS and Perforce.

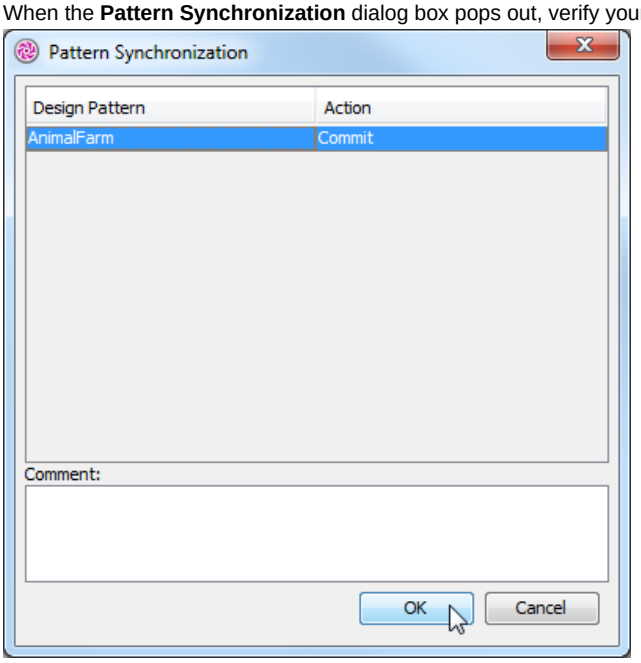
Synchronizing local design pattern to server

1. After you have defined a design pattern, click **Team Collaboration > Open Teamwork Client** on toolbar, or select **Teamwork > Open Teamwork Client...** from the main menu.
2. Login to the teamwork server.
3. In the **Teamwork Client** dialog box, select **Repository > Synchronize Design Pattern to Server** from its menu.



*Click **Synchronize Design Pattern to Server***

4. When the **Pattern Synchronization** dialog box pops out, verify your desired design pattern and action and click **OK** button to continue.

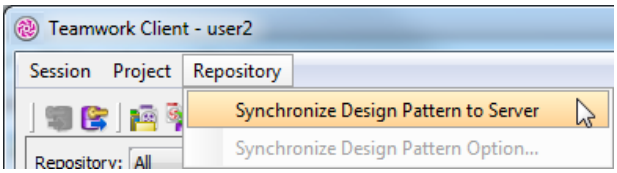


***Pattern Synchronization** dialog box*

As a result, the pattern will be committed to teamwork server.

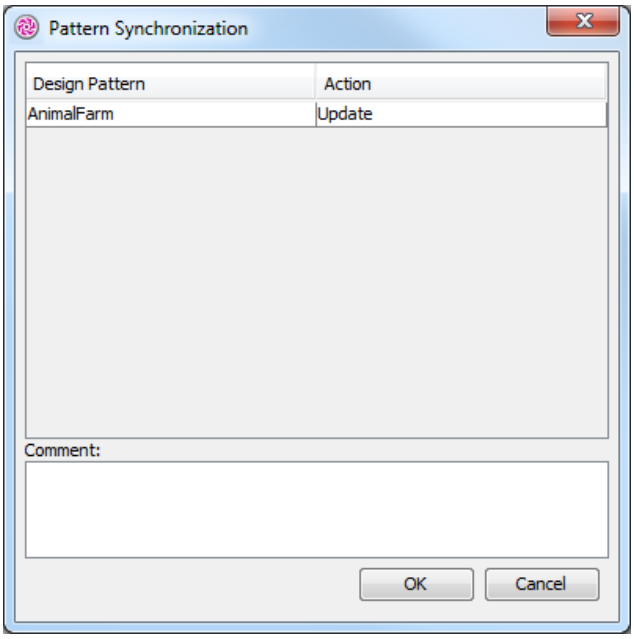
Synchronizing design pattern from server

1. After you open the **Teamwork Client** dialog box, select **Repository > Synchronize Design Pattern to Server**.



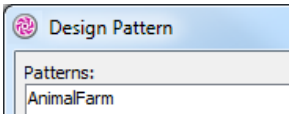
*Click **Synchronize Design Pattern to Server***

2. When the **Pattern Synchronization** dialog box pops out, it displays that your pattern is available for update from teamwork server. Click **OK** button and the pattern will be updated from teamwork server to workspace.



***Pattern Synchronization** dialog box*

3. Open an existing diagram where you want to apply a design pattern or create a new diagram. Right click on the diagram background and select **Utilities > Apply Design Pattern...** from the pop-up menu. The design pattern is available on the list of **Design Pattern** dialog box. You can now select the pattern and apply to your project.



*Design pattern is available in **Design Pattern** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Model transitor

Model transitor helps define the transition of phrasees of work and maintain the traceability in between. Both the use of model transitor for shape and diagram will be covered in this chapter.

Model transitor for shape

Shows you the steps of adding and maintaining transition between shapes.

Model transitor for diagram (diagram transitor)

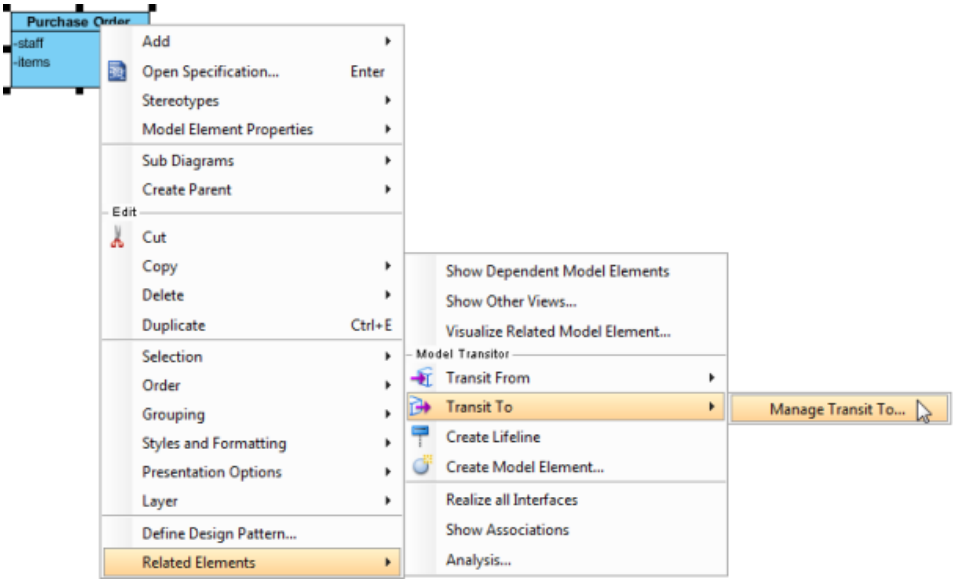
Shows you the steps of adding and maintaining transition between diagrams.

Model transitor for shape

You can use different diagrams for modeling different phrases of development, such as a diagram for modeling the current system, and another diagram for modeling to system to be implemented. In order to trace the evolution of model elements across diagrams, you can make use of model transitor. Model transitor enables you to establish transit relationship between shapes. With the transition relationship, you can trace between shapes across diagrams.

Adding a transition between shapes

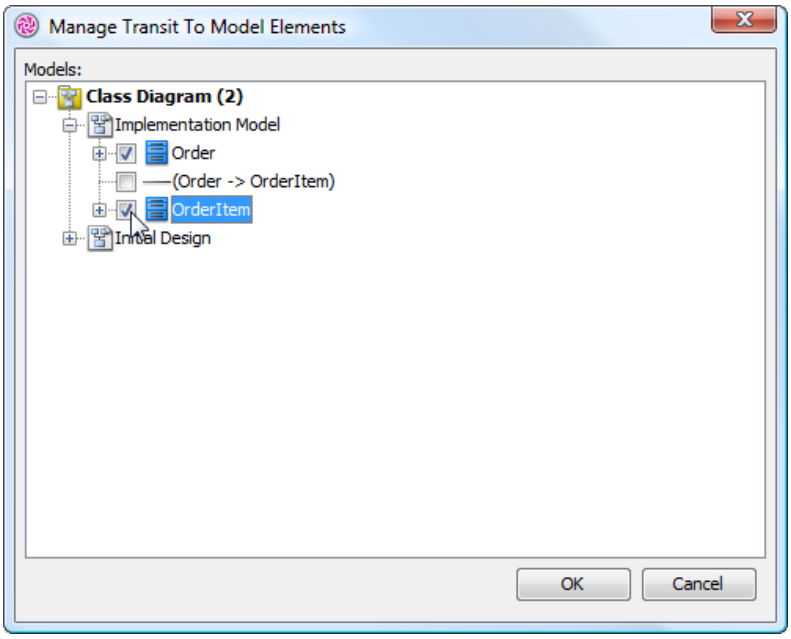
1. Right click on the shape you want to add a transition. It can be the source or target shape within the transition to add. If you right click on the source shape, select **Related Elements > Transit To > Manage Transit To...** from the pop-up menu.



Right click on a class

If you are right click on the target shape, select **Related Elements > Transit To > Manage Transit From...** from the pop-up menu.

2. In the **Manage Transit To/From Model Elements** dialog box, select the shape(s) you want to transit to/from. You can select multiple shapes to transit to/from. For example, an initial *Purchase Order* class will be transited to an *Order* class and an *OrderItem* class.



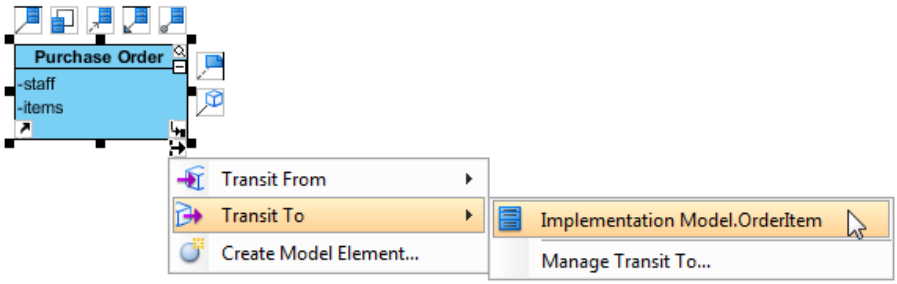
Select shapes to transit to

Navigating transited shape

Once a transition is added between two shapes, you can navigate between them. There are two methods to navigate to a transited shape. The first way is to go through the transitor popup menu of a shape. Right click on a shape and select **Related Elements > Transit From/To** from the popup menu, then the shape to navigate to.

Alternatively, make use of resource centric interface by following the steps below:

1. Move the mouse over the shape that you want to navigate to its transited shape.
2. Press its **Model Transitor** resource icon. Select **Transit From/To**, then the shape to navigate to.

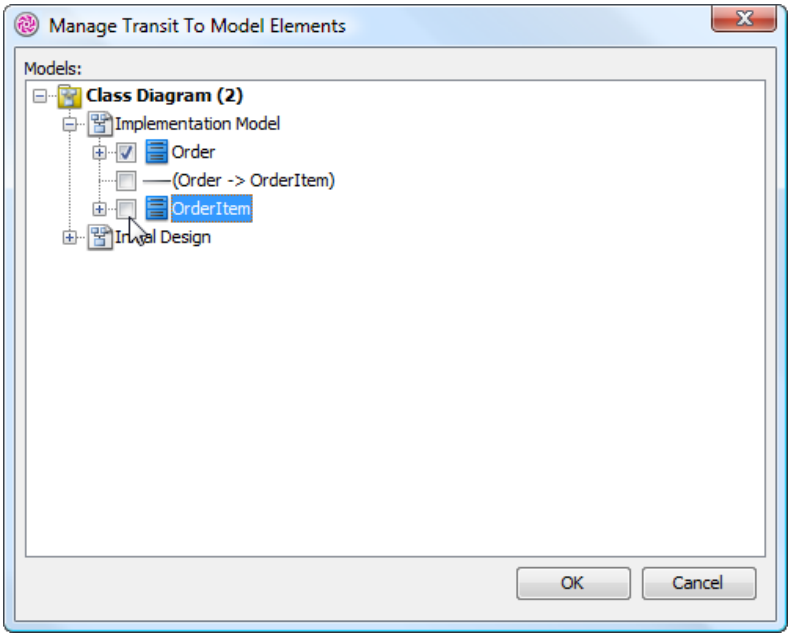


Navigate to a transited shape through the resource centric interface

Removing a transition

To remove a transition between shapes:

1. Right click on a shape and select **Related Elements > Transit From/To > Manage Transit From/To** from the pop-up menu.
2. In the **Manage Transit To/From Model Elements** dialog box, deselect the shapes that you do not want to transit with. Click **OK** button to confirm.



Deselect shapes to withdraw from transition

Related Resources

The following resources may help you learn more about the topic discussed in this page.

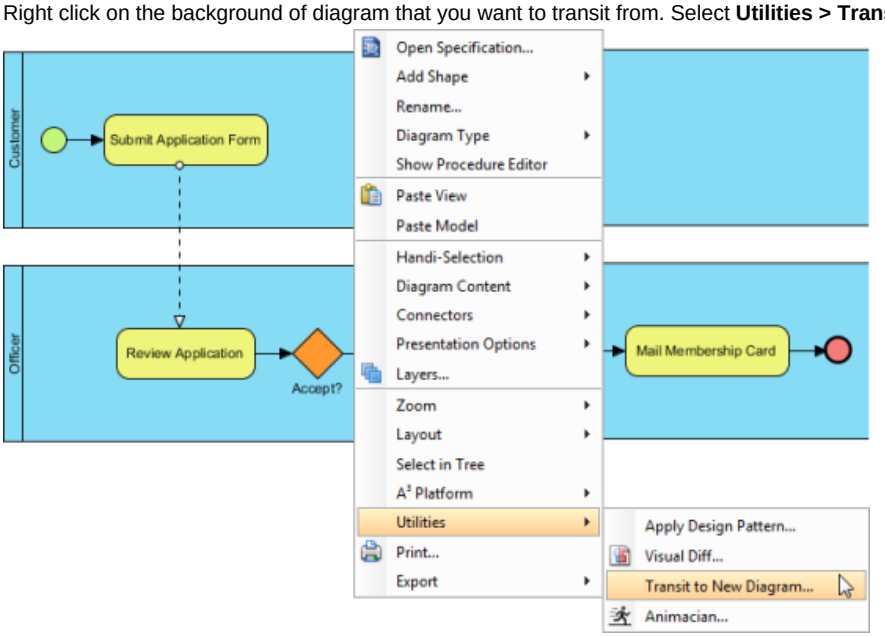
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Model transitor for diagram

You can use different diagrams for modeling different phrases of development, such as a diagram for modeling the current system, and another diagram for modeling to system to be implemented. Sometimes, diagrams across phrases are similar, but little variation. Model transitor enables you to duplicate a diagram with transition added in between. You can then continue modeling in the new diagram by using the original diagram's content as base.

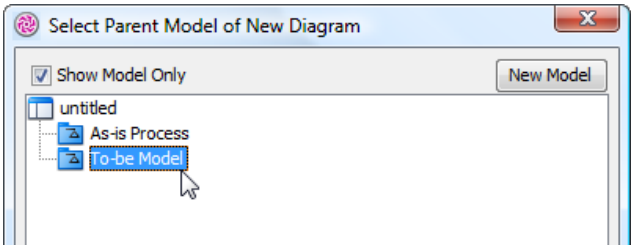
Transiting to a new diagram

1. Right click on the background of diagram that you want to transit from. Select **Utilities > Transit to New Diagram...** from the pop-up menu.



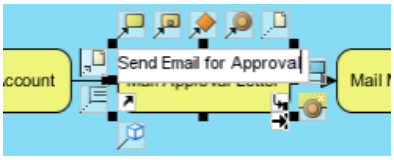
Transit to a new diagram

2. The **Select Parent Model of New Diagram** dialog box appears, enabling you to select a model for storing diagram. Visual Paradigm encourages structuring project with model for easier accessing of model elements and increasing application performance. If you want to place the new diagram in a model, select one or click **New Model** button at the top right corner to create one and select it. If you do not want to store diagram inside any model, do not make any model selection. Click **OK** button to continue.



Select a model for storing the new diagram

3. You can then start editing the new diagram.



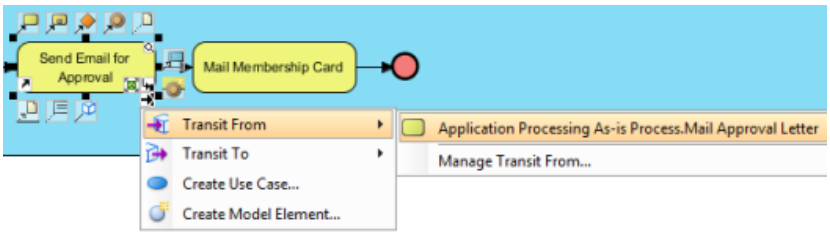
Edit transited diagram

Navigating transited shape

By adding a transition between diagrams, transitions are automatically added between shapes in the two diagrams. With the transition between shapes, you can navigate between them. There are two ways to navigate to a transited shape. The first way is to go through the transitor pop-up menu of a shape. Right click on a shape and select **Related Elements > Transit From/To** from the pop-up menu, then the shape to navigate to.

Alternatively, you can make use of the resource centric interface by following the steps below:

1. Move the mouse over the shape that you want to navigate to its transited shape.
2. Press its **Model Transitor** resource icon and select **Transit From/To**, then the shape to navigate to.



Navigate to a transited shape through the resource centric interface

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Using stereotypes

A stereotype defines how a model element may be extended, and enables the use of platform or domain specific terminology or notation in place of, or in addition to, the ones used for the extended metaclass. In this chapter, you will learn things about stereotypes and see how to apply stereotypes in your model.

Applying stereotype to model element

Tells you what stereotype is and how to apply to a model element.

Configure stereotypes

Shows you how to configure a stereotype like to define its color and add tagged values.

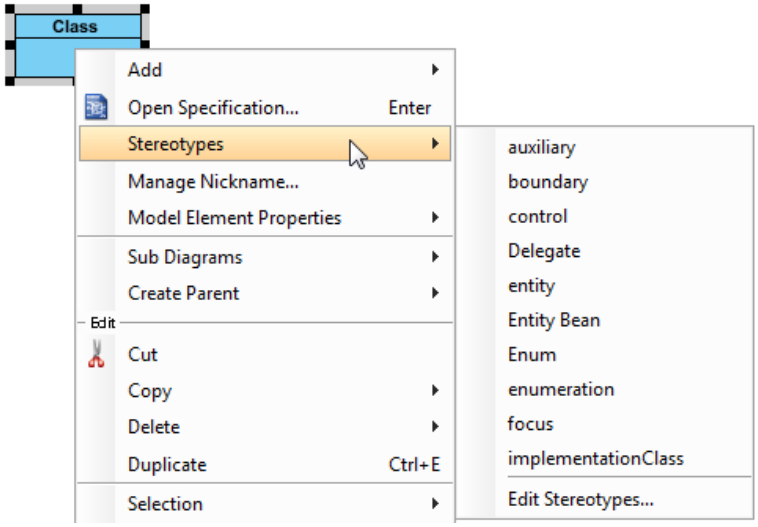
Shortcut of creating stereotyped model element

You can create a stereotyped element type easily through the diagram toolbar.

Applying stereotype to model element

A stereotype defines how a model element may be extended, and enables the use of platform or domain specific terminology or notation in place of, or in addition to, the ones used for the extended metaclass. In Agilian, you can apply one or more stereotypes to model elements, and decide whether or not to visualize the stereotype or tagged values in views. To apply stereotype to model element:

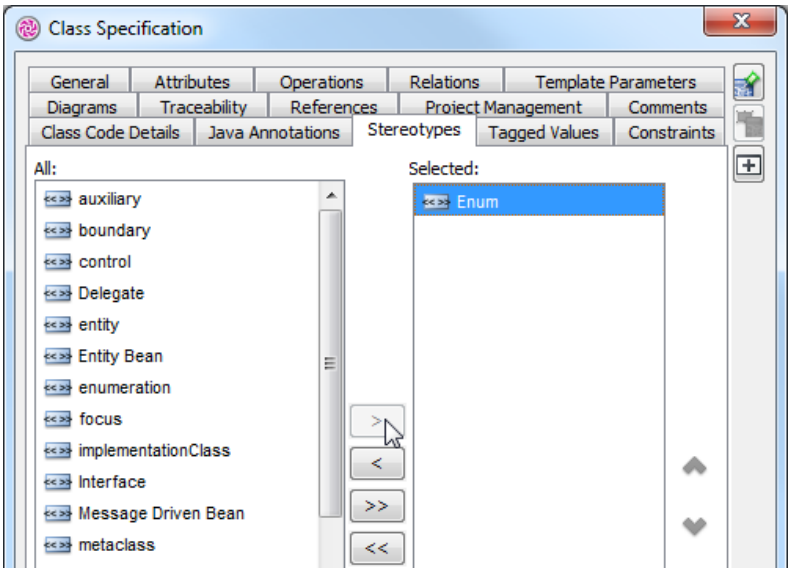
1. Right click on the model element, or the view of the model element that you want to apply stereotype to. Select **Stereotypes** from the pop-up menu.



Select a stereotype

Depending on the type of model element you are selecting, there may be a list of suggested stereotypes listing in the menu popped up. It consists of both the recently used stereotypes and stereotypes that place at the top of stereotype list. If you see the stereotype you want to apply, select it. Otherwise, select **Edit Stereotypes...** at the bottom of the menu to look for others.

2. In the **Stereotypes** page of the element specification, select the stereotype you want to apply, then click **>** to assign it to the **Selected** list.

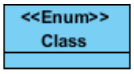


Stereotype Enum is selected

> **NOTE:** You can also double click on a stereotype to apply it.

> **NOTE:** While clicking on **>** applies the selected stereotype to model element, you can click **<** to remove a stereotype selected in **Selected** list.
>
> If you want to apply all available stereotypes to model element, click **>>**, and likewise, clicking on **<<** removes all the applied stereotypes.

3. Click **OK** to confirm. The stereotype will then be shown within a pair of guillemets above the name of the model element. If multiple stereotypes are applied, the names of the applied stereotypes are shown as a comma-separated list with a pair of guillemets.



Stereotype Enum is applied to a class

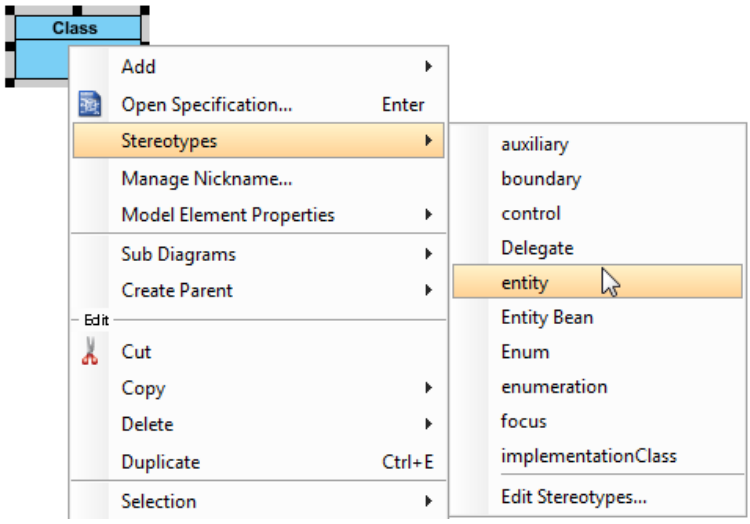
Robustness analysis icon

Robustness analysis helps to find out the relationships between actor, boundary, control and entity objects.

Robustness analysis icons

To draw a robustness analysis diagram with robust analysis symbols:

1. Create a class in diagram.
2. Depending on the type of robustness analysis symbol you want to create, apply either boundary, control and entity stereotype to the class.

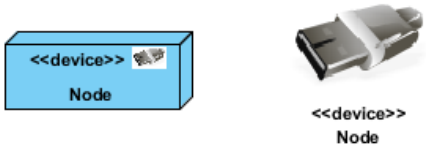


Apply entity stereotype to User class

NOTE: If you want to a class displays as traditional class shape instead of robustness analysis icons, right click on the class and de-select **Presentation Options > Display as Robustness Icon** from the pop-up menu.

Presenting a shape as stereotype icon

You can specify icon for a stereotype (Read the next chapter for details). When a stereotype is applied to a model element, you can let the stereotype icon show above the name of model element, which is the default presentation, or to make the model element show as the icon. To present a shape as stereotype icon, right click on the shape and select **Presentation Options > Stereotype Icon** from the popup menu.

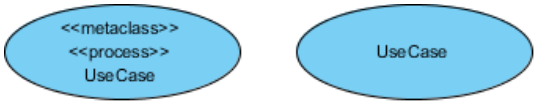


Different presentations of a model element with a stereotype that has icon defined

Showing or hiding stereotype

By default, applied stereotypes are shown within a shape. Yet, it is up to you whether to show or hide them. Furthermore, you can choose not to display the stereotypes, but to display only their tagged values.

To update the visibility of stereotypes, right click on the background of diagram where the shapes exist. Select/De-select **Presentation Options > Show Stereotypes** from the pop-up menu.



A use case with stereotype names shown and hidden

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Configure stereotypes

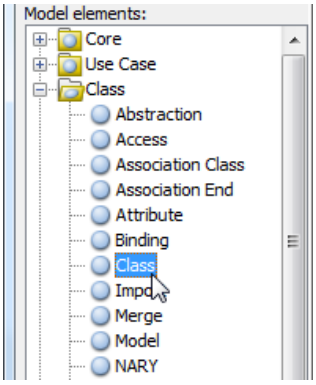
You can configure stereotypes, not just to create and name stereotypes for specific model element types, but also to format stereotypes like to set their colors, line formatting and font, and to define their tagged values. By configuring stereotypes, domain specific stereotype set can be built.

To configure stereotypes:

1. Select **Modeling > Configure Stereotypes...** from the main menu.
2. Click on the drop-down menu **Scope** at the top left corner of the **Configure Stereotypes** window, select whether to configure stereotypes in workspace or in the opening project.

> **NOTE:** Initially, stereotypes exist in workspace rather than in project. When you apply a stereotype to any model element, a copy of that stereotype will be made from workspace to project.
> By modifying stereotype in workspace, changes will not be applied to current project nor any project that has used the stereotype. To configure stereotype only in current project, you must select **Project** as scope.
> Alternatively, select **Workspace** but let the option **Apply changes to stereotypes in current project** on.

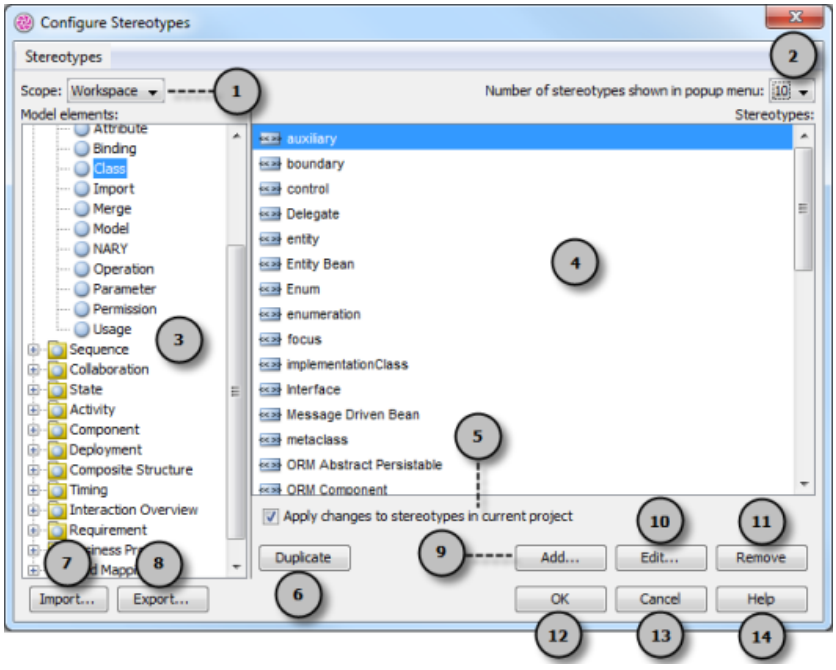
3. Select the type of model element that you want to add stereotype or edit its existing stereotypes.



Select class to edit its stereotypes

4. You may now perform any of the following action:
   - If you want to edit an existing stereotype, select the stereotype and click **Edit...**.
   - If you want to add a stereotype, click **Add...**.
   - If you want to remove a stereotype, select the stereotype and click **Remove**.
5. If you are adding or editing a stereotype, update its specification and click **OK** to confirm editing. For details about editing a stereotype, read the coming section.
6. Click **OK** to confirm.

An overview of Configure Stereotypes dialog box



*An overview of **Configure Stereotypes** dialog box*

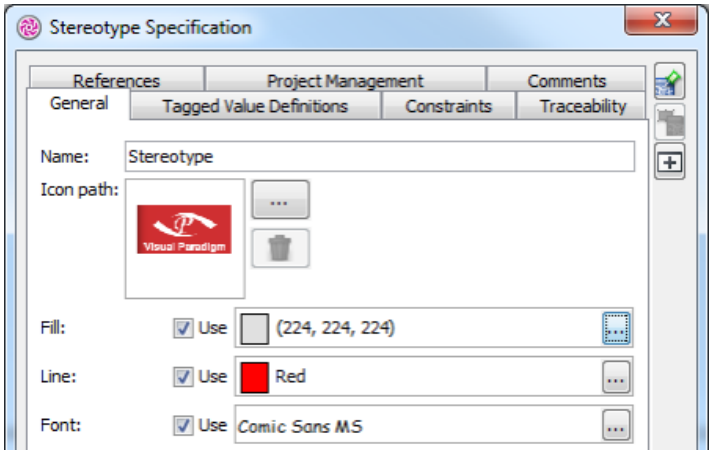
No.	Name	Description

1	Scope	Initially, stereotypes exist in workspace rather than in project. When you apply a stereotype to any model element, a copy of that stereotype will be made from workspace to project. By modifying stereotype in workspace, changes will be applied to neither current project nor any project that has used the stereotype because the stereotype copied to project is being followed. If you want to configure stereotype only in current project, you must select **Project** as scope, or select **Workspace** but let the option **Apply changes to stereotypes in current project** on to make changes apply on both workspace and project.
2	Number of stereotypes shown in popup menu	Select the number of stereotypes shown in **Stereotypes** list.
3	Model element list	A list of categorized model element types. You can select a type to configure its stereotypes.
4	Stereotypes	A list of stereotypes of the selected model element type.
5	Apply changes to stereotypes in current project	Available only when scope is Workspace, this option cause the stereotype configuration applies to both stereotypes in workspace and project, when pressing **OK**.
6	Duplicate	Click to duplicate the stereotype selected in Stereotype pane.
7	Import	Click this to import stereotype configuration (an XML) produced by others. Once clicked, the **Import Stereotypes** dialog box will pop out. You need to choose the XML file to import. At the bottom of the dialog box, there is an option **Add and update only (do not delete stereotypes)**. When checked, VP-UML will only add and update stereotypes from XML. When unchecked, VP-UML will add and update stereotypes, and additionally delete stereotypes that are not defined within the XML.
8	Export	Click to export stereotype configuration to an XML file.
9	Add	Click to add a stereotype for the selected type of model element.
10	Edit	Click to edit the selected stereotype.
11	Remove	Click to delete the selected stereotype.
12	OK	Click to apply the configuration and close the dialog.
13	Cancel	Click to discard the changes (if any) and close the dialog box.
14	Help	Click to open the Help contents.

*Description of **Configure Stereotypes** dialog box*

Editing stereotype

By adding or editing a stereotype, you can specify its icon and adjust its fill, line and font style in the **General** page within the **Stereotype Specification**.



Edit stereotype

By applying a stereotype that has icon defined to a model element, the icon above the name of model element, near the stereotype. You can optionally make the model element show as the icon. For details, read the previous chapter. To specify icon, click on the **...** button near the preview of Icon. Then, select the image file of icon.

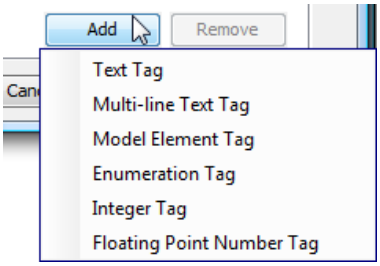
Fill, line and font styles will be applied automatically to model elements that have the stereotype applied. To adjust fill/line/font, check the corresponding Use button. Then, click on the **...** button to edit the settings.

Defining tagged values for stereotypes

A stereotype may have properties, which may be referred to as tag definitions. When a stereotype is applied to a model element, the values of the properties may be referred to as tagged values.

You can define tagged values for stereotypes. By doing so, when you apply the stereotype with tagged values defined to a model element, you can fill in the values for the model element.

1. Select **Modeling > Configure Stereotypes...** from the main menu.
2. In the **Configure Stereotypes** dialog box, select the stereotype that you want to define tagged value and click **Edit**. If you want to add a new stereotype, select the base model type and click **Add...**.
3. In the **Stereotype Specification** dialog box, open the **Tagged Value Definitions** tab.
4. Click **Add**. Select the type of tagged value to define. The type of tagged value limits the type of content user can enter for a tag.

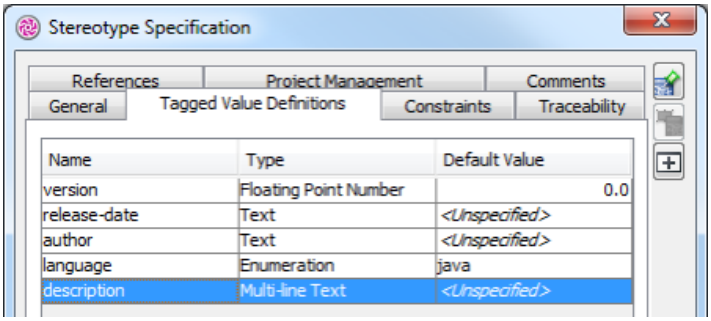


Add a tag

Tag type	Description
Text	The most common and general type of tagged value that consists of words.
Multi-line Text	The value of tag is a text in multiple lines.
Model element	The value of tag is a model element in project.
Enumeration	The value of tag can be chosen from a list of possible values. For example, to select "red" out of values red, green and blue.
Integer	The value of tag must be a real number.
Floating point number	The value of tag must be a number that consists of one or more digits.

Type of tags

5. Double click the name cell and enter the name of tag. Repeat step 4 and 5 to add all tagged values for this stereotype.



Tags defined

6. You can assign a default value to a tag by editing the **Default Value** cell. Usually, you give a tag a default value when the value is true in most cases. For example, a tag "in-door-temperature" can have "25" as default value.

Related Resources

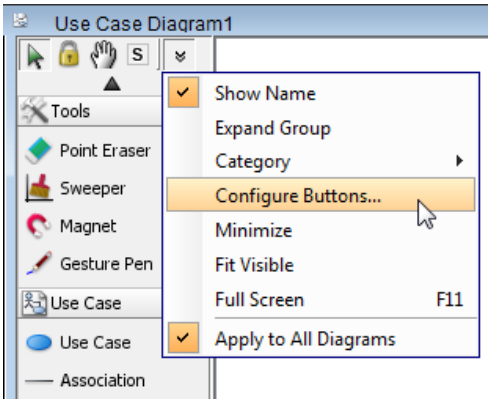
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Shortcut of creating stereotyped model element

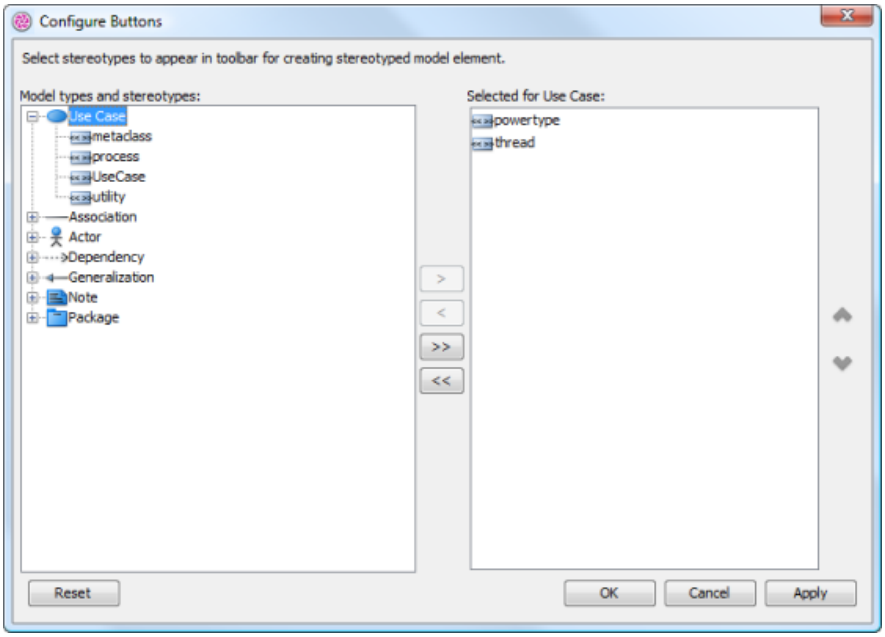
One of the ways of creating a shape is by selecting the element type in diagram toolbar, and clicking on diagram to create a shape of that type. If then you want to apply a stereotype to that model element, you will open the specification dialog box and make a stereotype selection. These steps can be simplified by adding a shortcut in diagram toolbar, for creating a type of model element with specific stereotype pre-set. To do so:

1. In any diagram, click on the double down arrow button at the top of diagram toolbar, then select **Configure Buttons...** from the pop-up menu.



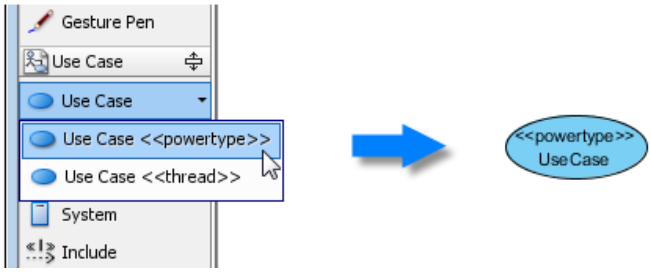
Configure buttons

2. In the **Configure Buttons** dialog box, expand the node(s) of model element type(s) that you want to add shortcut for. Select the stereotype that you need to apply to that type of model element in future. Click **>** or double click on it to assign. Click **OK** to confirm.



Configure Buttons dialog box

3. After all, you can find the shortcuts in diagram toolbar, under the selected type(s) of model elements. By selecting a stereotyped model element type and click on diagram, you can create a shape with stereotype applied.



Create a stereotyped shape

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Customizing elements with profile

A profile provides a generic extension mechanism for customizing model elements for different purposes. It is closely related to stereotype and tagged value. Details will be covered in this chapter.

Creating a profile

You will see how to create a profile in Model Explorer.

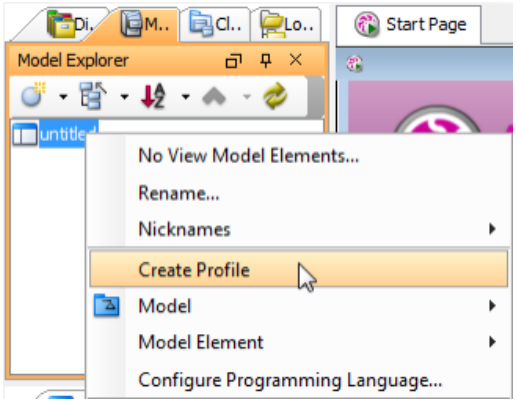
Drawing a profile diagram

A profile can be modeled through a profile diagram.

Creating a profile

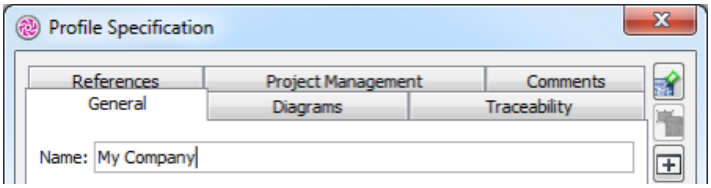
A profile provides a generic extension mechanism for customizing model elements for different purposes. Profiles are defined using stereotypes and tagged values definition for specific types of model elements. You can tailor meta model for different domains and platforms by creating and developing a profile.

1. In **Model Explorer**, right click on the project root node and select **Create Profile** from the pop-up menu.



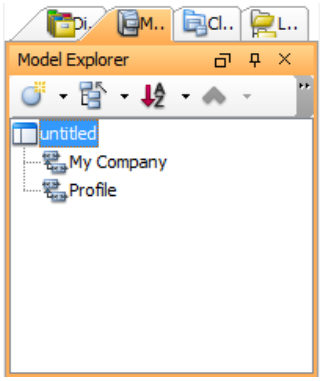
Create a profile

2. When the **Profile Specification** dialog box pops out, enter name for the profile and click **OK** button to confirm. A clear naming is needed to identify the part of the domain you want to create a profile for. For example, a Vehicle profile for managing stereotypes around different car types; a Transaction profile for managing stereotypes related to trading and loaning of cars.



Name the profile

3. You may create as many profiles as you want, just repeat step 1 and 2.



Profiles are created

A profile consists of domain-specific stereotypes and tagged values definition. To develop a profile, you need to create a profile diagram. For details about how to create and draw a profile diagram, read the next chapter.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

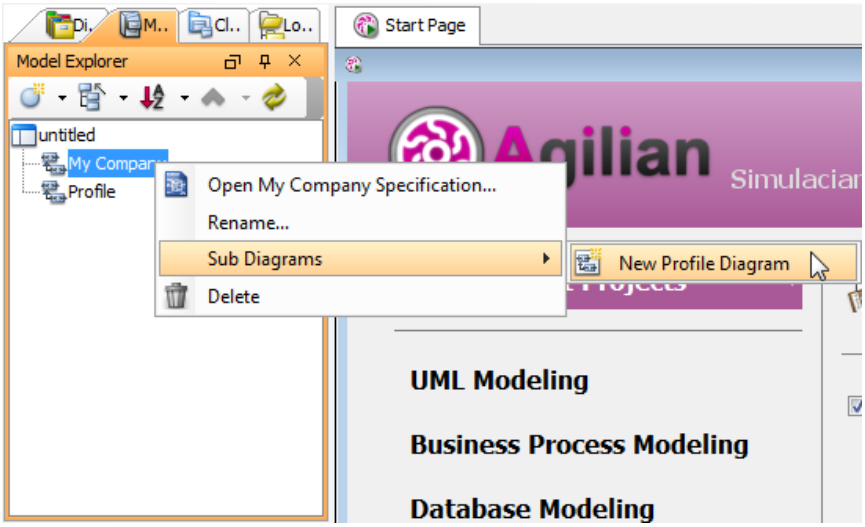
Drawing a profile diagram

A profile diagram enables you to create domain and platform specific stereotypes and define the relationships between them. You can create stereotypes by drawing stereotype shapes, and relate them with composition or generalization through the resource-centric interface. You can also define and visualize tagged values of stereotypes.

Creating a profile diagram

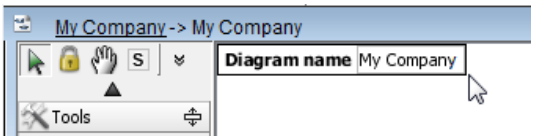
To create a profile diagram:

1. Right click on a profile in **Model Explorer** and select **Sub Diagrams > New Profile Diagram** from the pop-up menu.



Create a new profile diagram

2. Name the diagram and press **Enter** to confirm. By default, the name of profile is applied as the name of diagram. If you attempt to create only one profile diagram for a profile, and if your profile was well-named, you can keep using the profile name as diagram name.

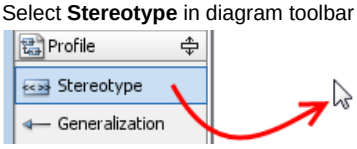


Name a profile diagram

Drawing a stereotype

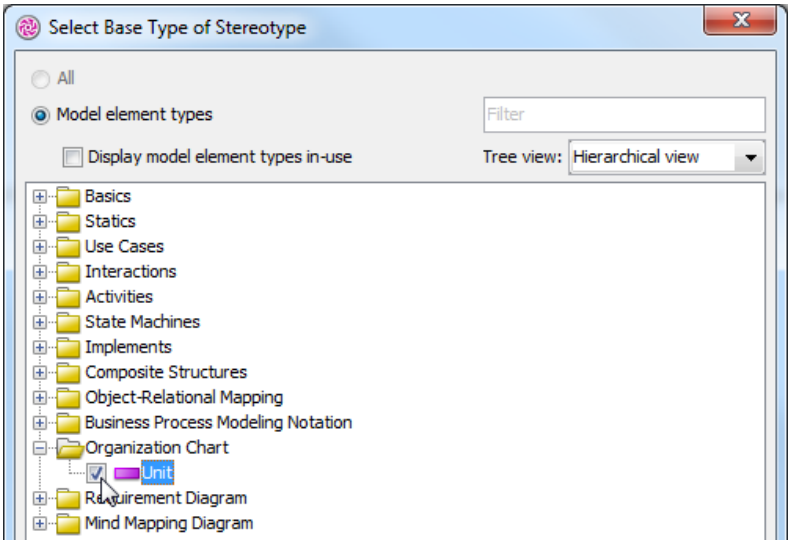
To draw a stereotype in profile diagram:

1. Select **Stereotype** in diagram toolbar and click on the diagram to create the stereotype.



*Select **Stereotype** on diagram toolbar*

2. In the **Select Base Type of Stereotype** dialog box, check the base type of stereotype from the specific model type tree. A base type defines the type of model element that the stereotype will extend. Click **OK** button to proceed.



Check a base type

> **NOTE:** You can check **Display model element types in-use** to list only types of model elements used in project.
> The text box **Filter** enables you to filter model element type base on the type name (e.g. enter class to list only class).

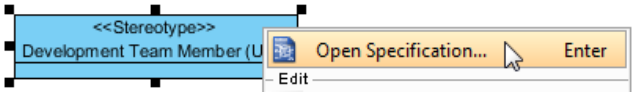
3. As a result, the stereotype is created. Name it and press **Enter** to confirm creation.

Defining tagged values for stereotypes

A stereotype may have properties, which may be referred to as tag definitions. When a stereotype is applied to a model element, the values of the properties may be referred to as tagged values.

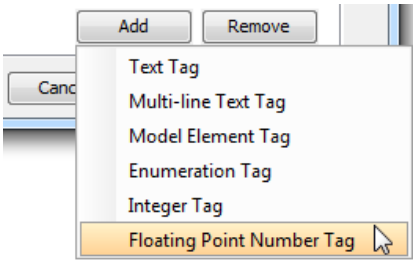
You can define tagged values for stereotypes. By doing so, when you apply the stereotype with tagged values defined to a model element, you can fill in the values for the model element.

1. Right click on a stereotype shape and select **Open Specification...** from the pop-up menu.



Open specification

2. In the **Stereotype Specification** dialog box, open the **Tagged Value Definitions** tab.
3. Click **Add** button to select the type of tagged value to define. The type of tagged value limits the type of content user can enter for a tag.

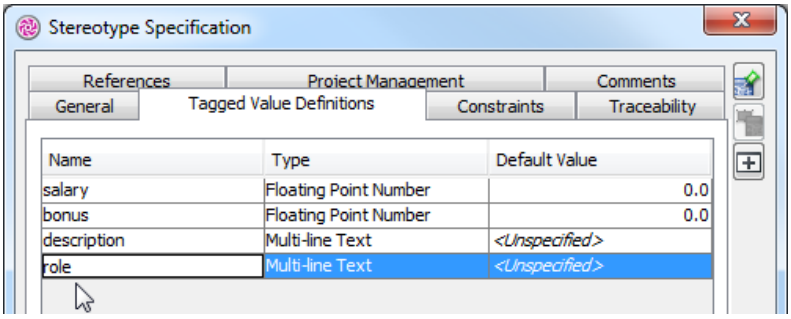


Add a tag

Tag type	Description
Text	The most common and general type of tagged value that consists of words.
Multi-line Text	The value of tag is a text in multiple lines.
Model element	The value of tag is a model element in project.
Enumeration	The value of tag can be chosen from a list of possible values. For example, to select "red" out of values red, green and blue.
Integer	The value of tag must be a real number.
Floating point number	The value of tag must be a number that consists of one or more digits.

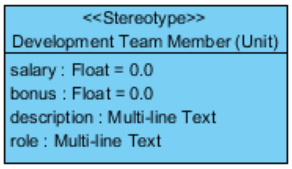
Type of tags

4. Double click the name cell and enter the name of tag. You may create as many tagged values as you want, just repeat step 3 and 4.



Tags defined for stereotype

5. You can assign a default value to a tag by editing the **Default Value** cell. Usually, you give a tag a default value when the value is true in most cases. For example, a tag "in-door-temperature" can have "25" as default value. By confirming changes, you can see the stereotype show on diagram, with tagged values shown below the stereotype name.



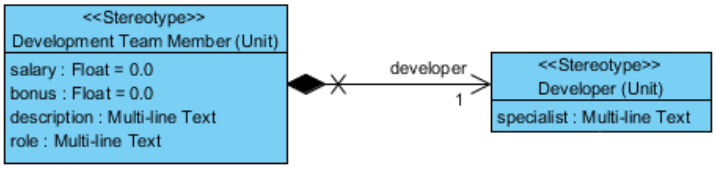
Stereotypes with tagged values defined

Relating stereotypes

Stereotypes can be related with each other by composition or generalization. Relating stereotypes not just affect the modeling in profile, but also the model elements that the stereotypes will be applied to.

Composition

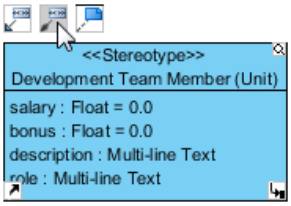
A composition relationship shows a "part of" relationship between stereotypes. The composite stereotype has responsibility for the existence and storage of the composed stereotype.



Composition between stereotypes

To create a composed stereotype:

1. Move the mouse over a stereotype and press its resource icon **Composition > Stereotype**.



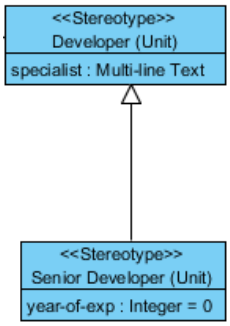
Create a composition

2. Drag it and release it until it reaches the position you want to create the composed stereotype. Select base class, name the stereotype and press **Enter** to confirm creation.

Since composition models are a "part of" relationship, when you apply a composite stereotype to a model element, you can add tagged value defined in composed stereotype in the model element. For example, stereotype *Development Team Member* is composed of stereotype *Developer*. If you apply stereotype *Development Team Member* to a unit, you can specify the properties of tagged values as defined by both stereotype *Development Team Member* and *Developer*.

Generalization

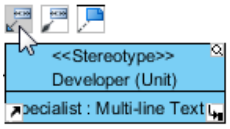
A generalization relationship shows a "kind of" relationship between stereotypes.



A generalization relationship

To create a specific stereotype from a general stereotype:

1. Move the mouse over a stereotype. Press on the resource icon **Generalization > Stereotype** from the pop-up menu.



Create a generalization relationship

2. Drag it out. Release it until it reaches the position you want to create the specialized stereotype. Select base class, name the stereotype and press **Enter** to confirm creation.

Since generalization models are a "kind of" relationship, when you apply a specialized stereotype to a model element, you can add tagged value defined in general stereotype in the model element. For example, stereotype *Developer* is a generalized stereotype of *Senior Developer*. If you apply stereotype *Senior Developer* to a unit, you can specify the properties of tagged values as defined by both stereotype *Developer* and *Senior Developer*.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Introduction of impact analysis

Impact analysis is the technique to find out what the potential influences are when updating a design blueprint. This chapter provides you with general understanding about impact analysis.

Introduction of impact analysis

Describe the several ways of impact analysis supported by Agilian.

Introduction of Impact Analysis

What is impact analysis?

Impact analysis is the technique to find out the potential influences that may happen when updating a design blueprint. For example, when we want to update the use case model, we may also want to update the sequence diagrams which model how to achieve the user goals. There are two ways of performing impact analysis in Visual Paradigm, analysis diagram and matrix.

How does impact analysis improves your work?

Impact analysis helps avoid unexpected consequences resulted by updating your design blueprint. Contrary to this, it lets you find out everything you need to update along when a change is to be made.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Matrix diagram

A matrix is a table, which shows the relationships among model elements of particular types. By reading a matrix, you can tell easily whether two model elements are related or not, and what kind of relationship do they have. This chapter not only tells you how to create a matrix but also how to read it, to get the information you need.

Creating a matrix

Shows you how to create a matrix.

Reading a matrix

Explains each part of a matrix in detail.

Showing the Use of Terms with Matrix

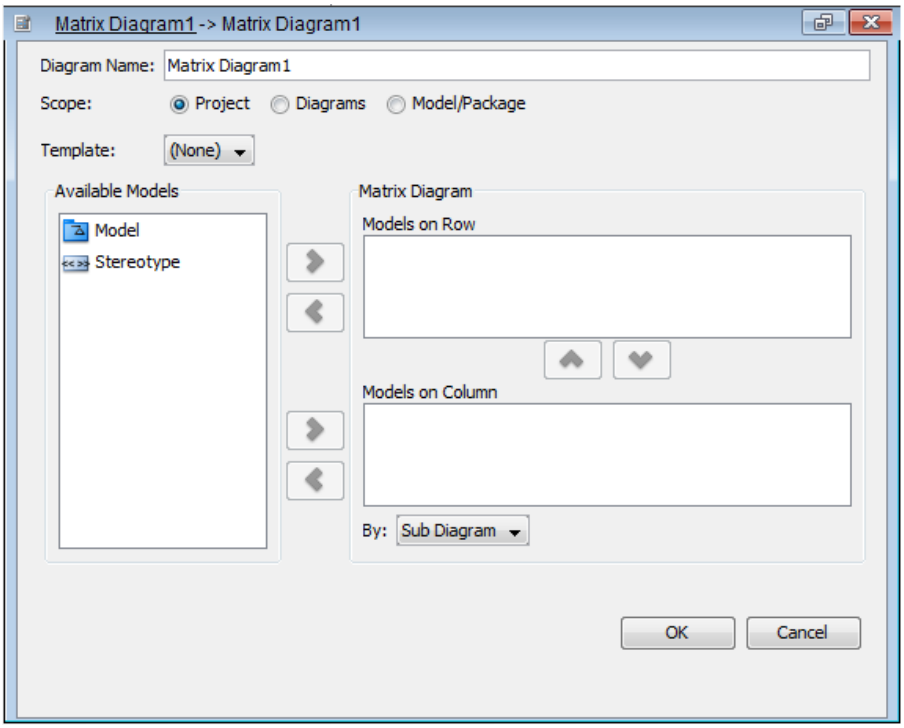
Shows you how to present the the use of terms by forming a matrix.

Creating a matrix

A matrix is a table, which shows the existence of relationships among model elements of particular types. By reading a matrix, you can tell easily whether two model elements are related or not, and what kind of relationship they have.

1. Right click on **Matrix Diagram**, under the category **Others** in **Diagram Navigator**, and select **New Matrix Diagram** from the pop-up menu.
   - Click on **Diagrams** on toolbar and select **Matrix Diagram** from the drop down menu **.**
   - Right click on **Matrix Diagram** in **Diagram Navigator**, under the **Impact Analysis** category, and select **New Matrix Diagram** from the popup menu.
   - Select **File > New Diagram > Impact Analysis > Matrix Diagram** from the main menu.

2. Configure the matrix.



Configure matrix

Field	Description
Diagram Name	The name of diagram which is also the name of matrix.
Scope	The source of model elements to compare in matrix, in **Project** (all model elements), **Diagram** (only model elements in specific diagrams which are selected by users) or under **Model/Package**.
Template	Template offers a default setup to **Models on Column**, **Models on Row** and **By**. It is available according to the project content. For example, template "Use Case <-> Requirement" appears for selection when a project has use case and requirement.
Available Models	All available models in your selected scope are listed here. You can select a model to add it in the target Models on Row and/or Models on Column.
Models on Column	The type of model element to list at the column side of matrix. In order to list multiple types of model element, select it/them on **Available Models** and click button to insert it/them in here.
Models on Row	The type of model element to list at the row side of matrix. In order to list multiple types of model element, select it/them on **Available Models** and click button to insert it/them in here.
By	The way how matrix will match against rows and columns. **Sub Diagram** - The column/row model element is placed in a sub diagram of the matching model element. **Child** - The column/row model element is a child of the matching model element. **Relationship** - The column/row model element is related with the matching model element. **Reference** - The column/row model element is referencing the matching model element. **As Classifier** - The column/row model element takes the matching model element as classifier. **Transitor** - The column/row model element is transited from/to the matching model element. **Dependent** - The column/row model element has properties that depend on the matching model element. "Depend" means any of the following: (1) With any kind of relationship such as association, dependency, etc. (2) As a contained element (3) As a link in documentation content (4) As a selected value for any property. **Using Term** - The column/row model element is a term (element) and is indirectly used by the matching model element, either or both as part of its name or in its documentation.

3. Click **OK** button to create the matrix.

Related Resources

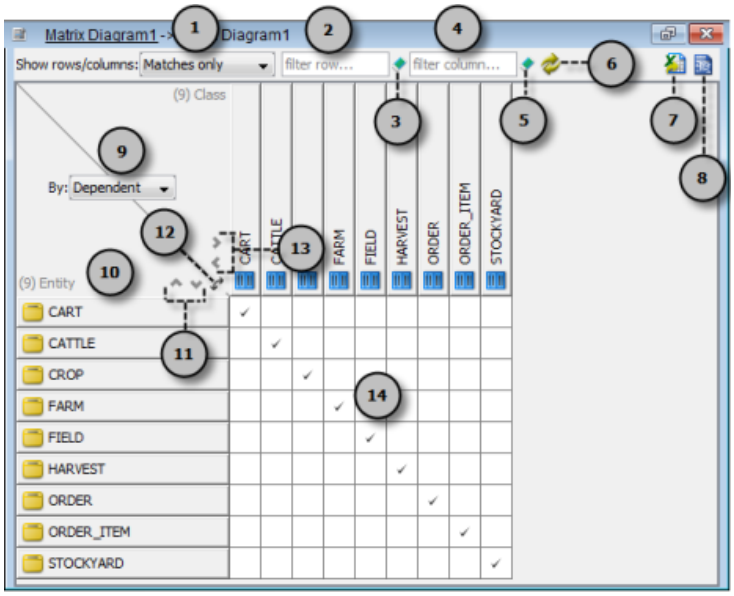
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Reading a matrix

To know how to read a matrix helps you to understand your model better, and to perform further modifications more efficiently. In this chapter, we will see how to read a matrix, and how to refine the matrix content with the help of functionalities like showing columns and rows with matches and filter.

A matrix is a table, with rows and columns, which represents sets of model elements of specific types. A cell in table is an intersection of a row and a column that reflects the relationship of the row and column. If the cell is filled either by a tick, or by a kind of relationship (when a matrix was set to match things by Relationship), this means that the model elements of the row and column are related. You can select the type of relationship in the drop-down menu of **By**.



Overview of a matrix diagram

No.	Name	Description
1	Show rows/columns	You can determine how model elements are shown in rows and columns. Select **Matches only** from the drop-down menu if you want to view model elements with matches while select **Non-matches only** if you want to view model elements without matches. Otherwise, select **All** to view all entries.
2	Filter row	Type the full name of a model element or just keyword that you are looking for to narrow down the searching field in rows when too much data is displayed.
3	Clear row filter	Delete all you have typed in **Filter row**.
4	Filter column	Type the full name of a model element or just keyword that you are looking for to narrow down the searching field in columns when too much data is displayed.
5	Clear column filter	Delete all you have typed in **Filter column**.
6	Refresh	You can update the content of matrix by clicking this button manually, for reflecting the changes you have made in models.
7	Export to Excel	Click this button to export the opening matrix to Excel.
8	Configure	Click this button to configure the content to display in matrix.
9	By	It shows how matrix will match against rows and columns.
10	Rows	The model elements have been chosen will be displayed in rows.
11	Move up and move down	Rows and columns are ordered alphabetically by default. They help to re-order rows and columns in moving vertically.
12	Swap rows and columns	It helps to change the presentation between rows and columns, but not switch the actual relationships between model elements being represented by them.
13	Move left and move right	Rows and columns are ordered alphabetically by default. They help to re-order rows and columns in moving horizontally.
14	Columns	The model elements have been chosen will be displayed in columns.

Description of matrix diagram

Exporting Excel

You can export Excel file from matrix, and analyze relationships between model elements in worksheet in Excel. To export Excel:

1. Click **Export to Excel** above the matrix, near the **Configure** button.

*Click **Export to Excel***

2. In the **Export Excel** dialog box, specify the output destination and click **Save** button.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Showing the Use of Terms with Matrix

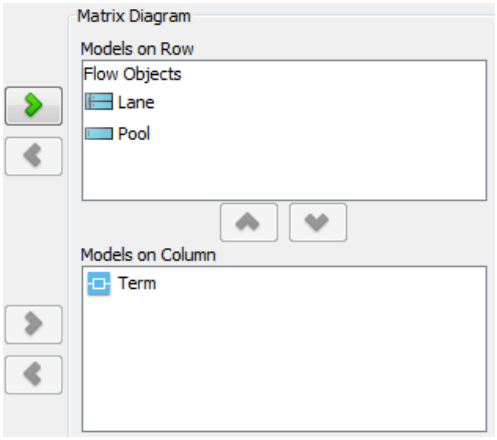
Glossary is a function to let you identify important terms from name and documentation of model elements, so that you can build a glossary with terms, and describe them in detail. For details about the use of glossary, please read the page Identify glossary term.

When you want to access the elements whose names or documentations have terms included, you may form a matrix to illustrate the relationships between elements and terms. You can even see how frequent the terms are referred to in different model elements in a matrix view.

To create such matrix:

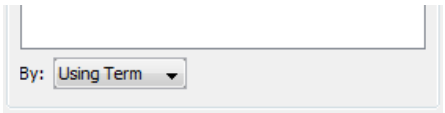
1. Take any of the following approach to create a matrix:
   - **Diagram Navigator** - Right click on **Matrix Diagram** in **Diagram Navigator** and select **New Matrix Diagram** from the popup menu.
   - **Toolbar** - Click on the **Diagrams** button in toolbar and select **Matrix Diagram** from the drop-down menu.
   - **Menu** - Select **File > New Diagram > Impact Analysis > Matrix Diagram** from the main menu.

2. Configure the matrix. If you want to place the terms at the top of matrix, add **Term** to **Models on Column**. If you want to place the terms on the left hand side of matrix, add **Term** to **Models on Row**.



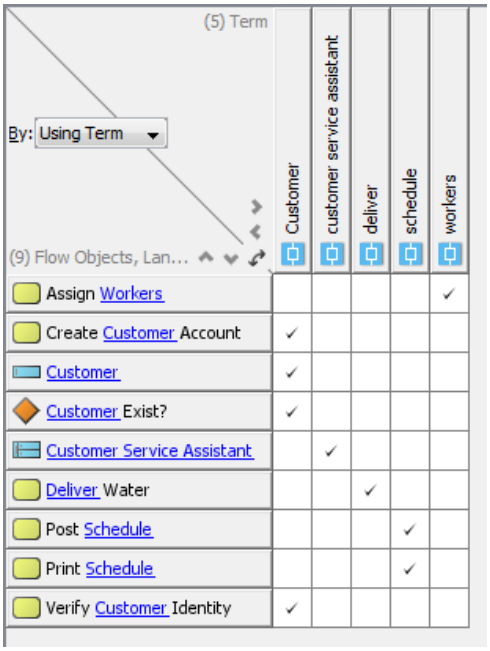
Elements are selected for row and column

3. Select **Using Term** under the drop down menu **By**.



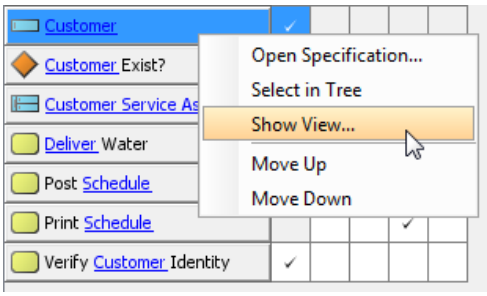
Select Using Term

4. Click **OK**. The ticks show in the matrix indicate the use of term for specific element.



Matrix formed

5. If you want to open a view of a model element, right click on the element and select **Show View...** from the popup menu. in the **Show View** window, select the diagram to open and click **Go to View**.



Show view of pool

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Analysis diagram

By analyzing a model element, you can know its relationships with other elements. This chapter shows you how to analyze things by forming and reading an analysis diagram.

Analyzing a model element

Shows you how to analyze a shape.

Updating analyzed result

When the model keep growing, you may need to update the analysis diagram to reflect the latest project content.

Grouping of nodes

Instead of having many many nodes show on analysis diagram, you may want to group nodes of same type to make the diagram tidier.

Opening view from node

Shows you how to open a view from a view node.

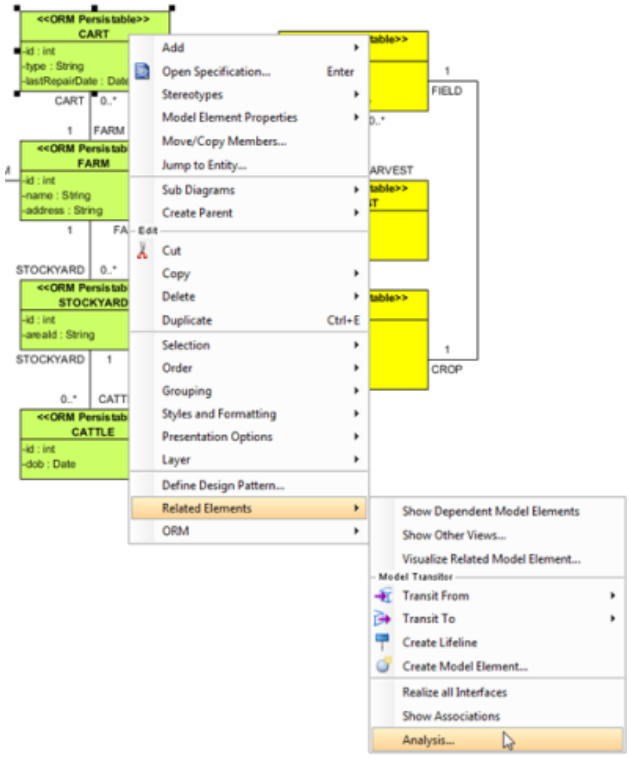
Analyzing a model element

We analyze a model element when we want to identify its related elements so as to foresee the potential impact that may cause on the model resulted by modifying the model element. The term "related" here represents any kinds of connection between two elements, such as a general to-and-from relationship, a parent-child relationship, transitor, or even a sub-diagram relationship with a diagram.

Analyzing a model element

By analyzing a model element, you can know its relationships with other elements. To analyze:

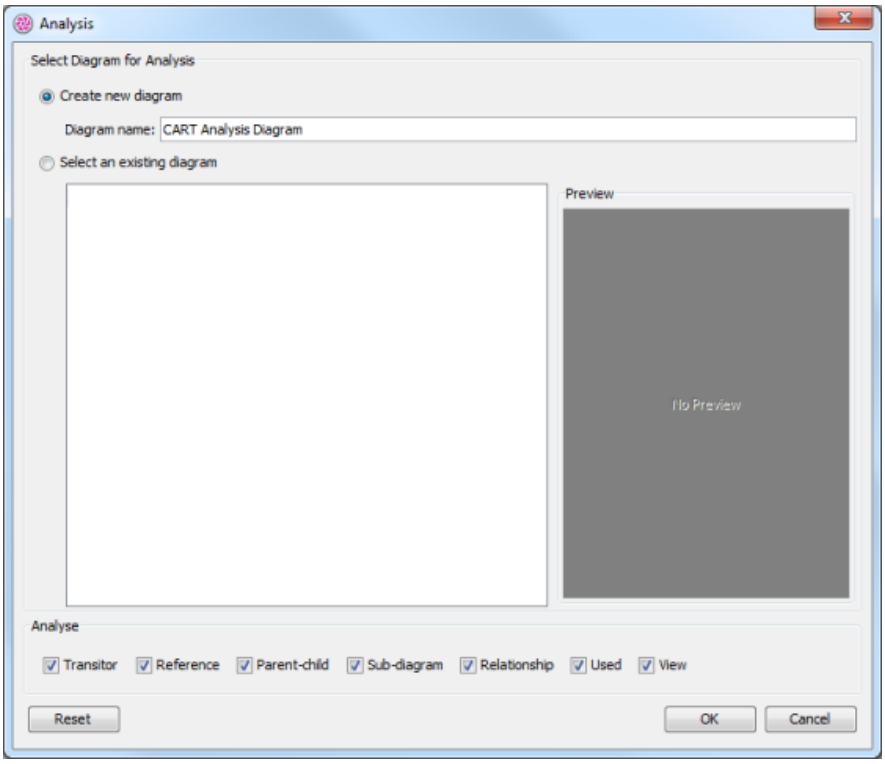
1. Right click on the diagram element you want to analyze, and select **Related Elements > Analysis...** from the pop-up menu.



Analyze a diagram element

> **NOTE:** To analyze a model element in Model Explorer, right click on the desired element node in Model Explorer and select **Analysis...** from the pop-up menu.

2. The result of analysis will be presented in analysis diagram. In the **Analysis** dialog box, check either **Create new diagram** to present the result in a new analysis diagram, or **Select an existing diagram** to present in an existing analysis diagram. The check boxes under the **Analyse** section manage the type(s) of relationship to be analyzed. Click **OK** button when ready.



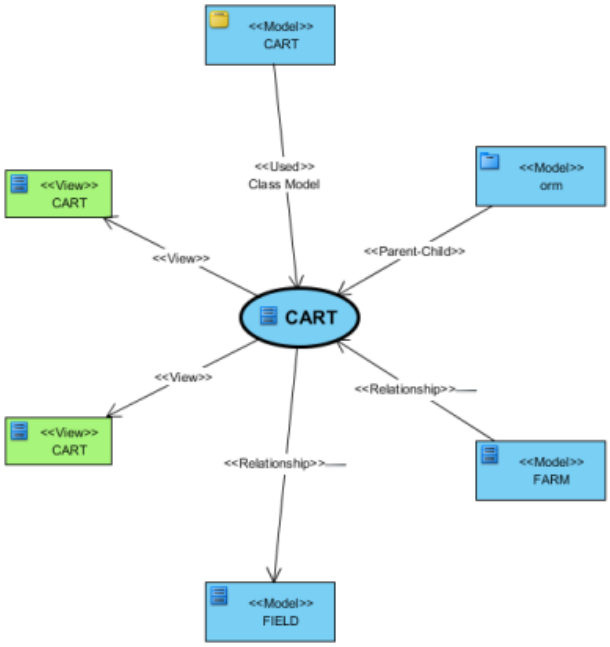
The Analysis dialog box

Type	Description
Transitor	The transited element of the chosen element, or the element where the chosen element was transited from.
Reference	The shape or diagram references of the chosen element.
Parent-Child	The parent (e.g. package) or the child of the chosen element.
Sub-diagram	The sub-diagram(s) of the chosen element.
Relationship	The relationship(s) of the chosen element, such as association, dependencies, sequence flow, etc
Used	The connection with the chosen element, other than any other kinds of relationship types. For example, requirement owned by use case added through use case description is a kind of Used relationship.
View	The view(s) of a model element, which can be seen as the shapes of a model element.

Kinds of relationships that can analyze

Reading analysis diagram

The result of analysis is shown in an analysis diagram.



An analysis diagram

The oval node at the center of diagram represents the element you have chosen to analyze, while the connectors are the relationships between the analyzing element and its related elements.

Inside a node of a related element, you can see a tag (e.g. <<View>>), which represents the type of that related element. At the bottom part of a node box is the name of the related element.

With analysis diagram, you can identify the relationship of a model element and determine the impact that may act upon the model when modifying the model element.

Related Resources

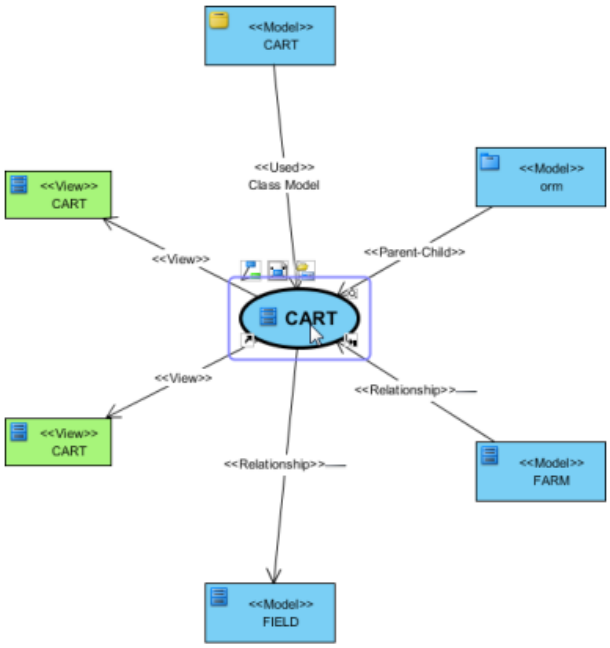
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Updating analyzed result

If a model is refined, the previous analysis result may no longer reflect the latest changes. You, therefore, need to update the analysis result in order to perform impact analysis for the chosen element with the latest model.

1. Move the mouse over the node that you want to update its relationships with others.



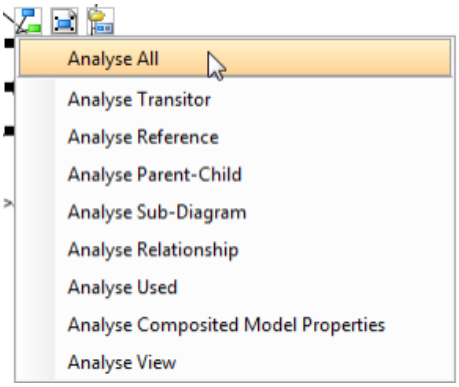
Analyze a diagram element

> **NOTE:** Besides the central node, you may update/show the relationships of other Model nodes on diagram too.

2. Click its resource icon **Analyze**.

*Click resource icon **Analyse***

3. Select a type of relationship to analyze.



Analyze all kinds of relationships

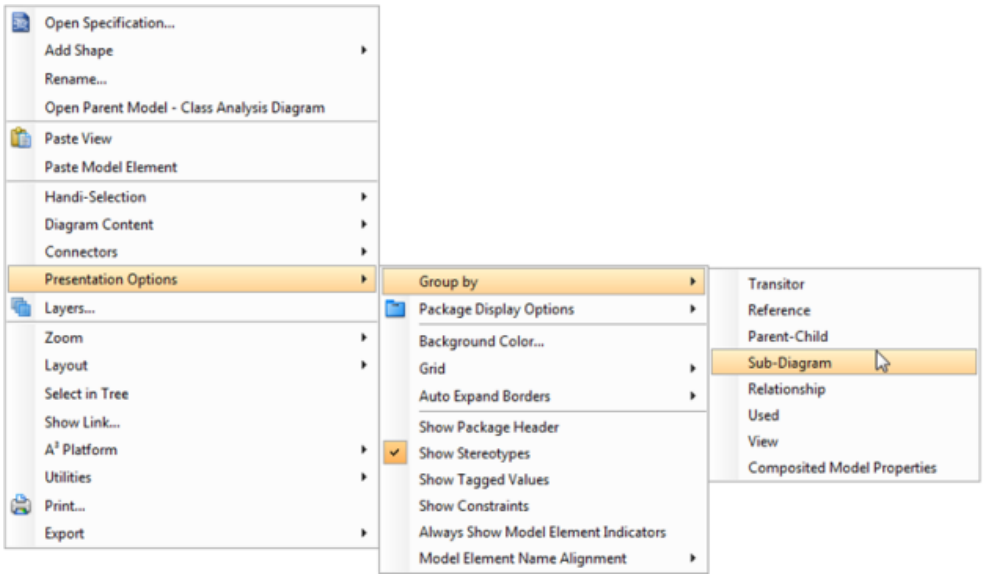
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

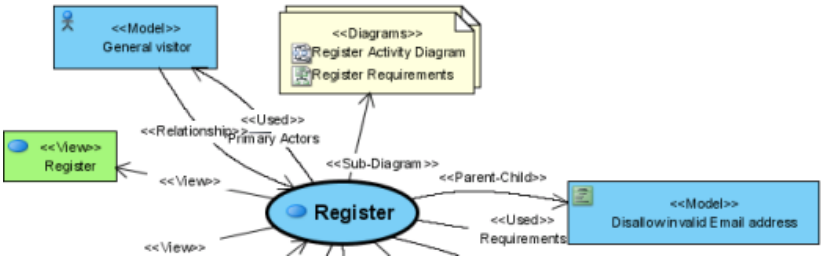
Grouping of nodes

To make it easier to identify the relationships of a chosen element, you can group all relationships of a particular kind into a single node, eliminating the connectors being shown on diagram. To group, right click on the background of analysis diagram, and select **Presentation Options > Group By**, and then the type of node from the pop-up menu.



Select a kind of relationship to group by

Nodes of same type are grouped.



Diagrams are grouped

To ungroup, simply deselect the node type by walking through the same pop-up menu path.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

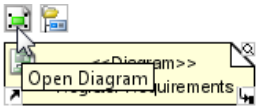
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Opening view from node

When reading an analysis diagram, you may find the existence of related model element, or related diagrams, such as sub-diagram, of the chosen model element. You can open the view of such related elements through the resource icons appear on top of nodes.

Opening a diagram of diagram node

1. Move the mouse pointer over a **Diagram** node.
2. Click on the **Open Diagram** resource icon.

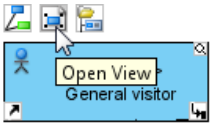


Open diagram of diagram node

This opens the diagram.

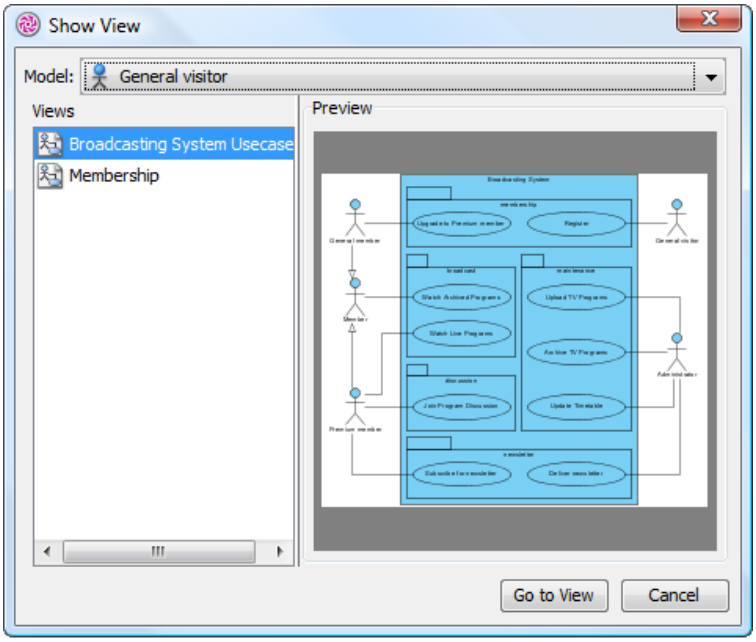
Opening a view of model node

1. Move the mouse pointer over a **Model** node.
2. Click on the **Open View** resource icon.



Open view of a model node

3. If the target model has only one view, the view will be automatically opened. If there a multiple views, the **Show View** dialog box is presented. Select a view to open, and click **Go to View** at the bottom right of dialog box.



*The **Show View** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Chart diagram

Chart diagram enables you to specify and show the relationship between model elements by mean of chart. This chapter shows you how to create and configure chart.

Developing a chart

Shows you how to create and develop chart.

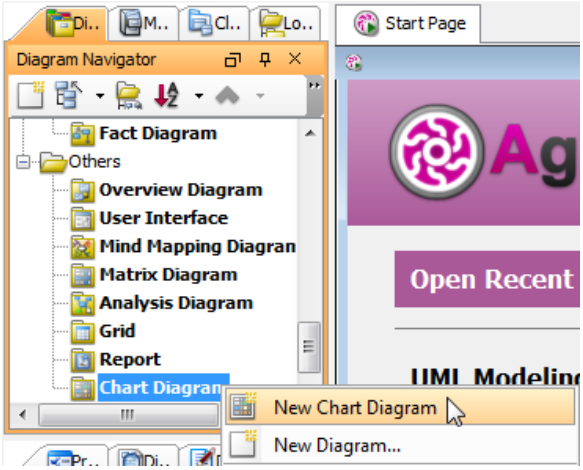
Configuring a chart

Shows you how to configure chart.

Developing a chart

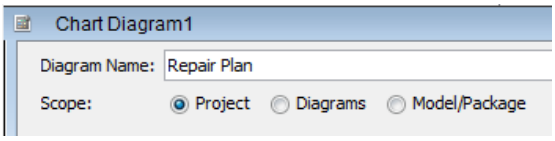
Agilian enables you to build a chart. In addition to the built-in RACI chart, you can define your own code type for your own specific purposes. In this page, you can learn how to develop a chart and apply various functions in a chart.

1. Right click on **Chart Diagram** on **Diagram Navigator** and select **New Chart Diagram** from the pop-up menu.



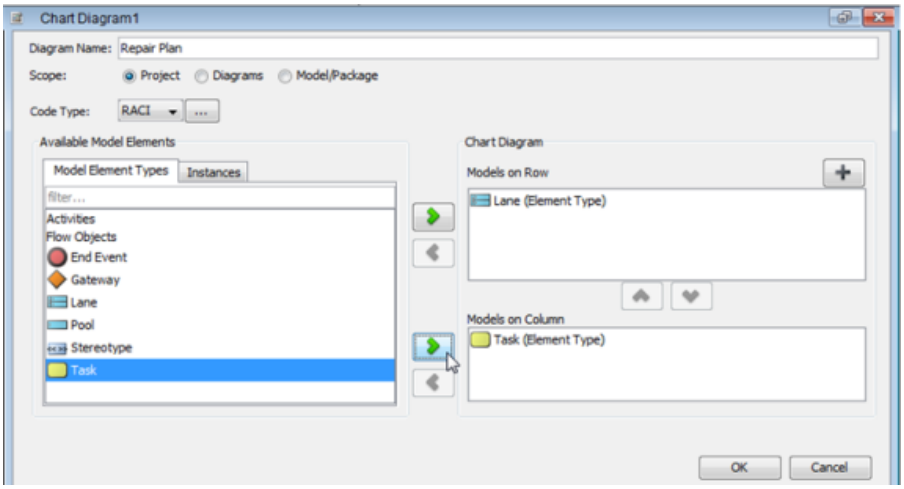
Create new chart diagram

2. Name the chart.



Name the chart diagram

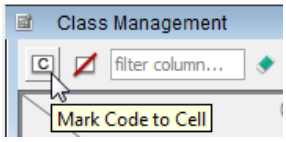
3. Specify the type of model element to list in row and column respectively.



Select a model element type for row and column

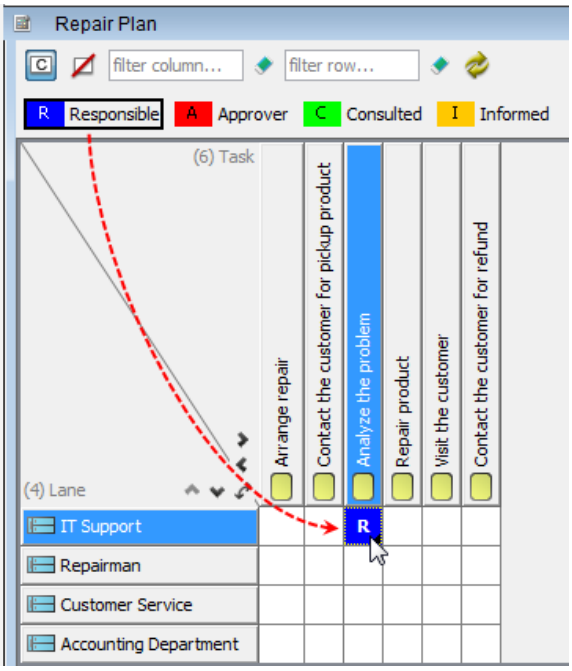
> **NOTE:** The choice of model element types will be in accordance with your existing model elements in the same project.

4. Click **OK** button.
5. You can then assign the role to participant(s) by clicking **Mark Code to Cell** button on the top of chart.



*Click **Mark Code to Cell** button*

6. When the roles reveal, click the target role and select the target cell to assignment.



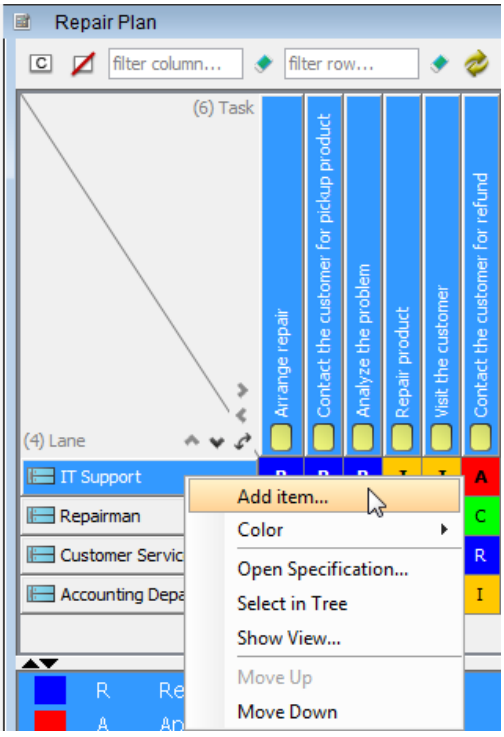
Assign role

7. Upon finish editing, click **Mark Code to Cell** button again.

Adding an item

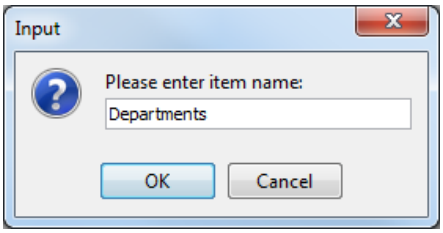
You can add an item, for example a text item, on row or column. The main characteristic of an item is that it doesn't have model view. It is because it isn't for diagramming, but catergorizing.

1. Right click on a model element and select **Add item...** from the pop-up menu. Note that an item will be inserted in front of the selected model element.



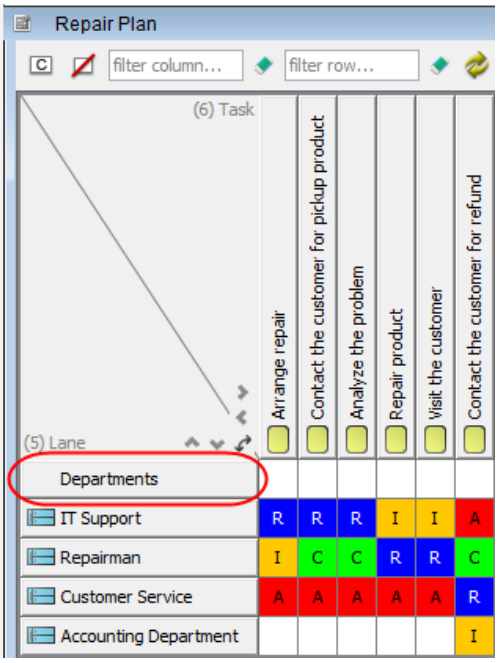
Add an item

2. Enter name for new item in the pop-up **Input** dialog box. Click **OK** button to confirm and close the dialog box.



Enter item's name

As a result, the newly created item is shown.



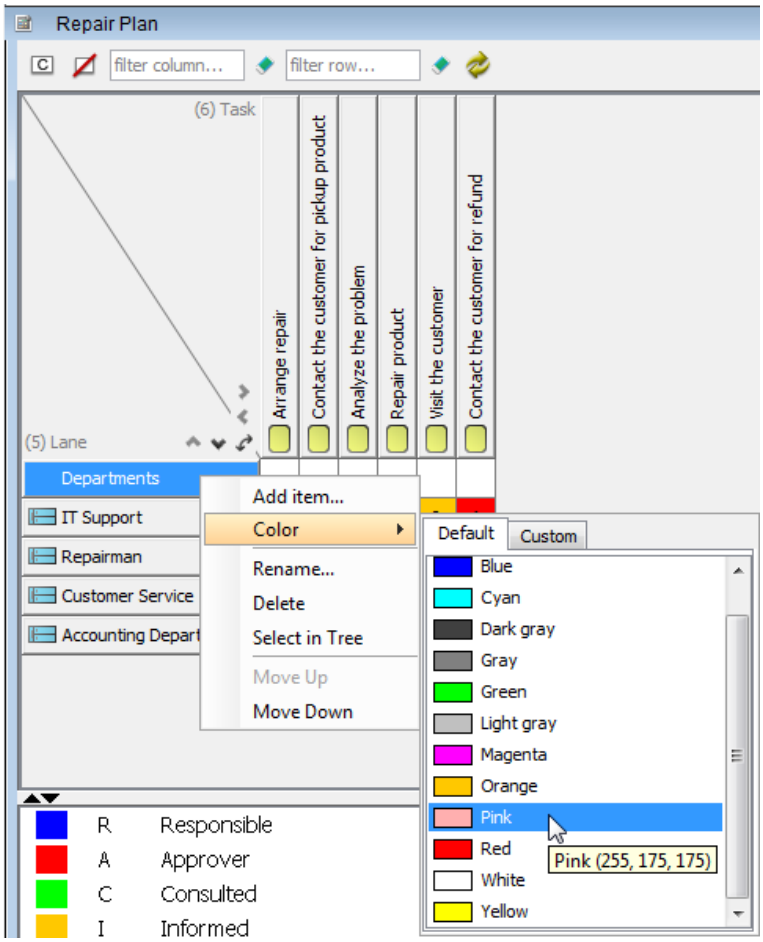
The newly created item is shown

Specifying background color for column/ row

You can specify background color for of entire column/row for emphasizing particular row/column item, or just for styling and formatting.

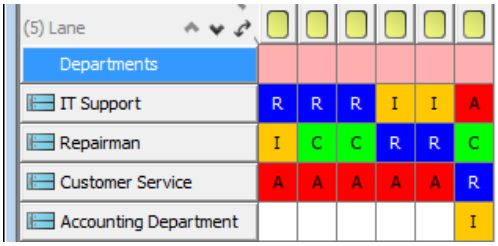
To set background color for column/ row:

Right click on the target row/ column and select **Color** > [target color] from the pop-up menu. You can select a color from either **Default** category or **Custom** category.



Select a background color

As a result, the background color for selected row/ column is set.



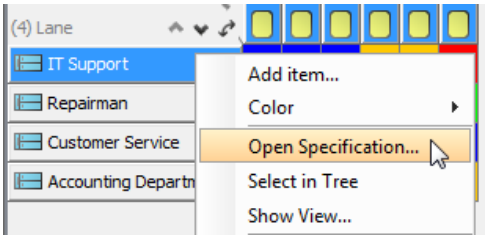
Background color is set

Opening model element's specification

You can view and specify the properties of model elements by opening their specification dialog box.

To open a model element's specification:

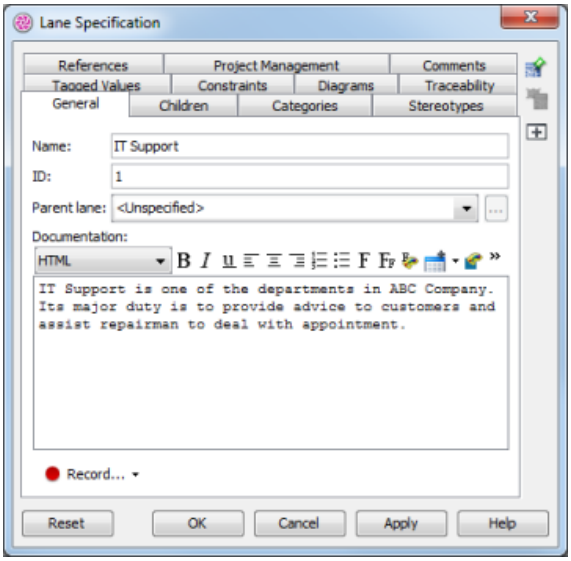
1. Right click on the target model element and select **Open Specification...** from the pop-up menu.



Open Lane's Specification dialog box

> **NOTE:** You cannot open a text item's specification dialog box.

2. In specification dialog box, specify its properties, for example, documentation, references, tagged values and comments. Click **OK** button.



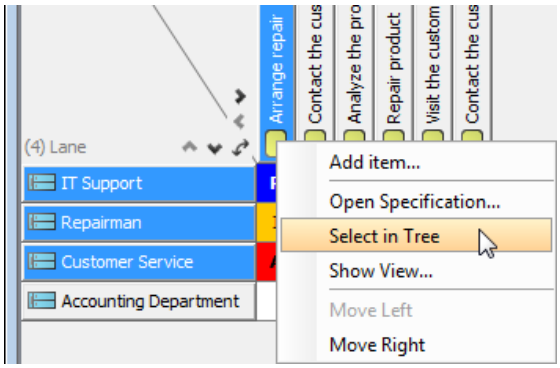
The Lane Specification

Selecting model element in Tree

You can view a model element on **Model Explorer** by selecting it on chart.

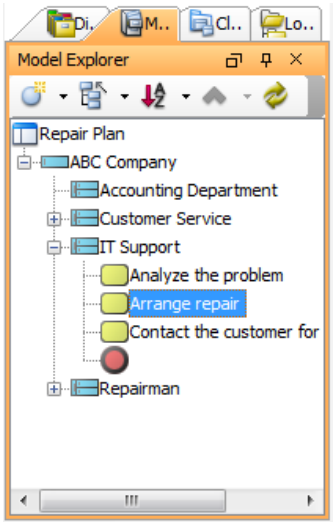
To select a model element on **Model Explorer**:

Right click on the target model element and select **Select in Tree** from the pop-up menu.



Select in Tree

As a result, the model element is highlighted on **Model Explorer**.

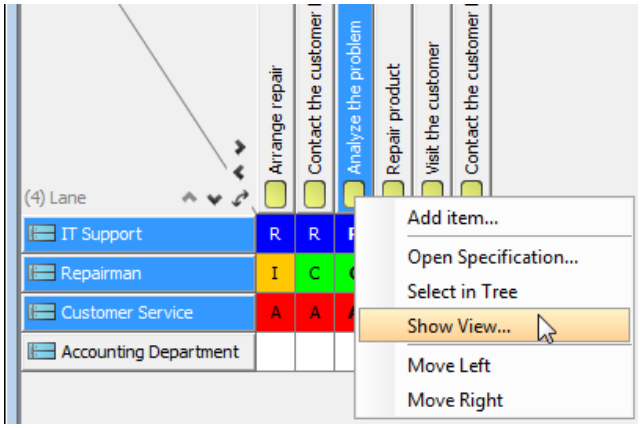


*Model element is highlighted on **Model Explorer***

Showing the view of model element

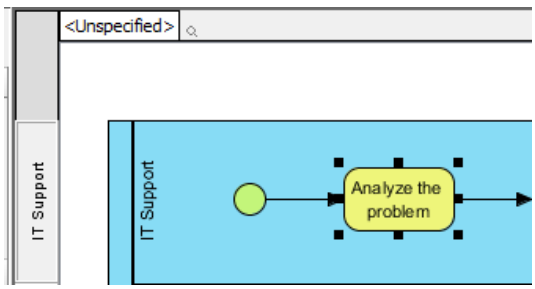
When reading the chart, you may want to view the model element on its source diagram.

You can view the model element by right clicking on the target model element and selecting **Show View...** from the pop-up menu.



Show the view of model element

As a result, the view of model element is selected on the diagram.

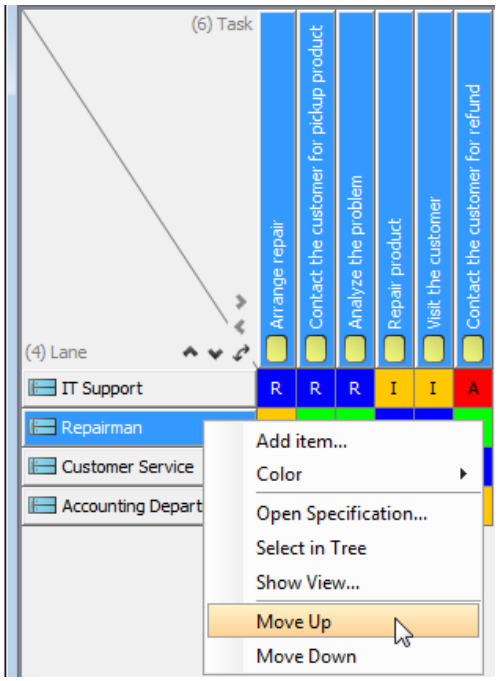


Model element is selected on its source diagram

Reordering row/ column

You can arrange the order of model elements with your preference.

Right click on target row/ column and select **Move up** or **Move Down** from the pop-up menu.

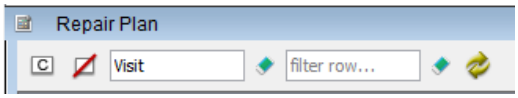


Move the model element up

Filtering model elements

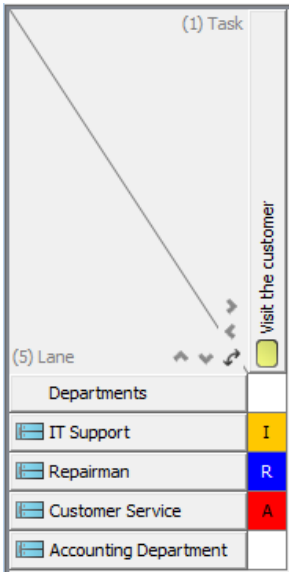
Chart provides a shortcut to search your target model elements

Type the word/ phrase in **filter column.../ filter row...**.



*Type a word in **filter column...***

Model elements matching the word/ phrase you typed will only be shown on chart.



The matched model element is shown

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian

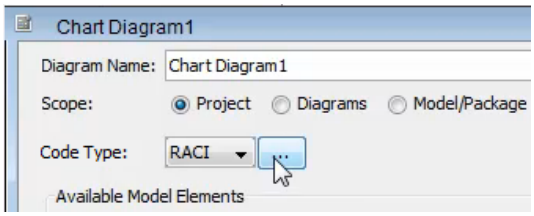
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Configuring a chart

Agilian enables you to build a chart. In addition to the built-in RACI chart available for general purposes, you can define your own code type for problem-specific purposes. In this page, you can learn how to define color for roles and define a new code type.

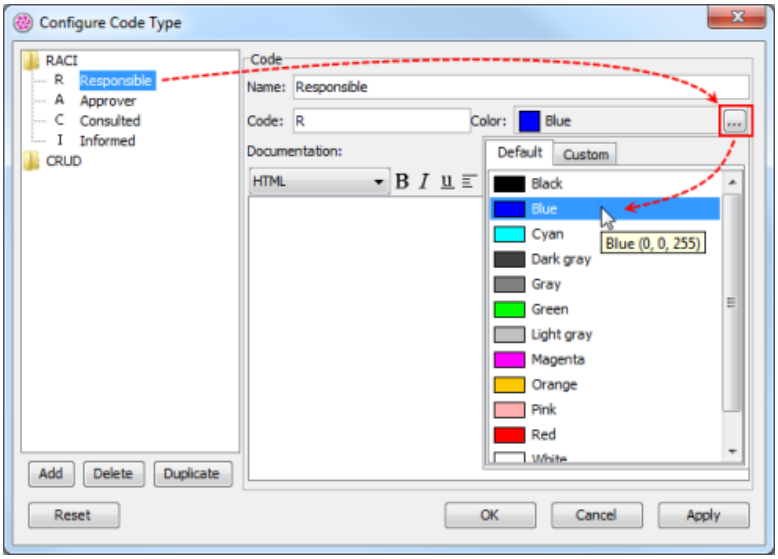
Defining color for roles

1. To define a color for roles, you can modify the existing code type by clicking the **...** button of **Code Type** in **Chart Diagram**.



Define code type

2. To define a color for a role, click the target role you want to change its color on the left, click the **...** button next to **Color** and select a color from the pop-up menu.

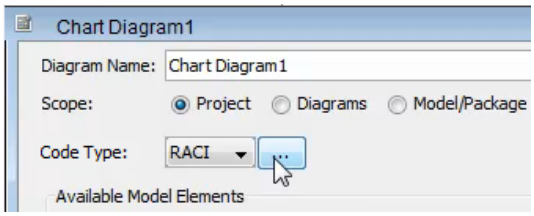


Select a color for target role

3. Click **OK** button.

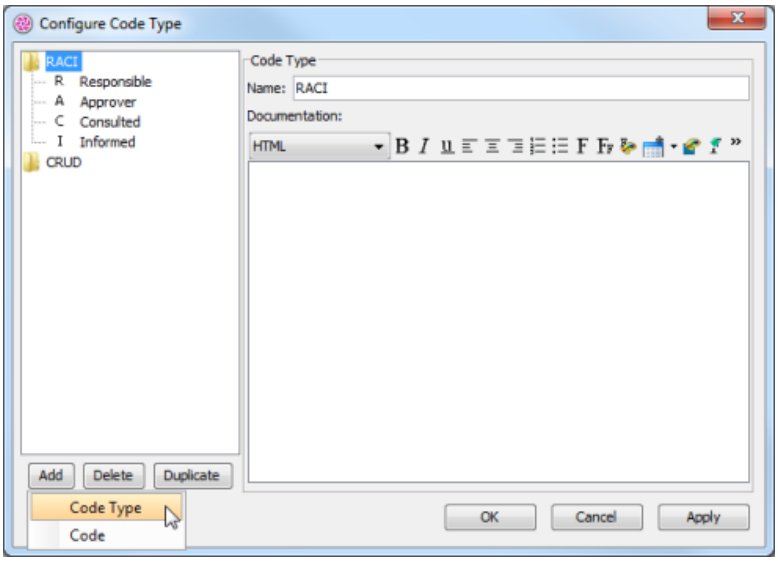
Defining a new code type

1. In **Chart Diagram**, apart from the built-in code type, you can configure a new code type by clicking the **...** button of **Code Type**.



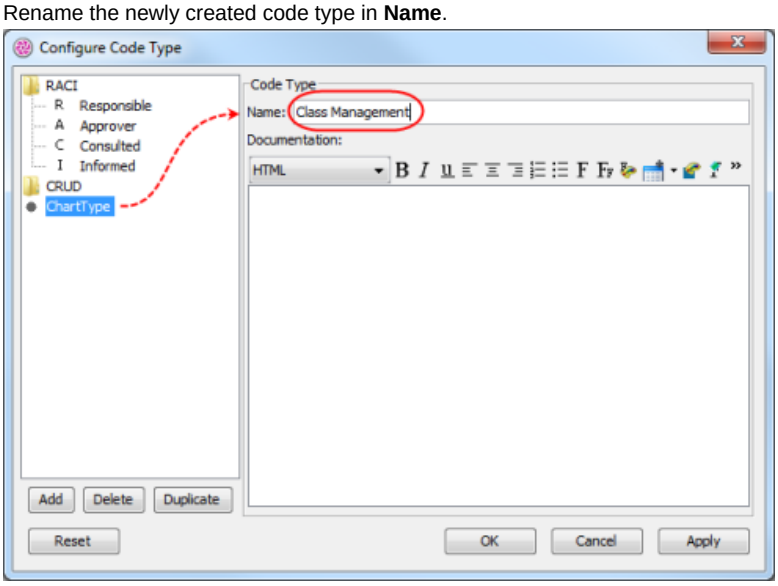
Define a new type of chart

2. In the pop-up **Configure Code Type** dialog box, select **Add > Code Type** from the pop-up menu at the bottom to add a code type.



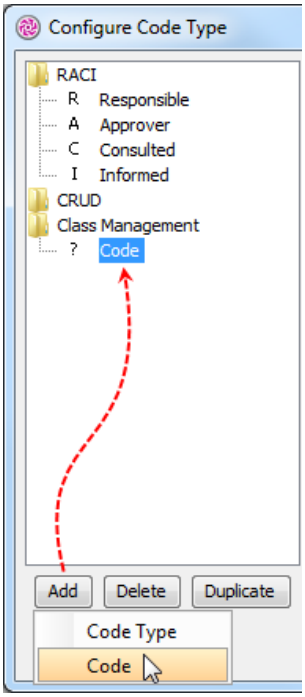
Add a code type

3. Rename the newly created code type in **Name**.



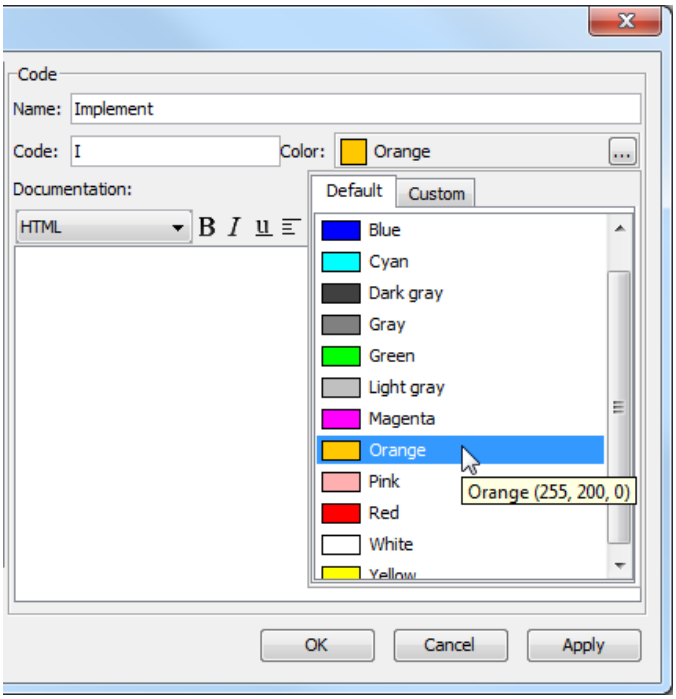
Rename the code type

4. To configure the roles under the new code type, select **Add > Code** from the pop-up menu.



Add a new code

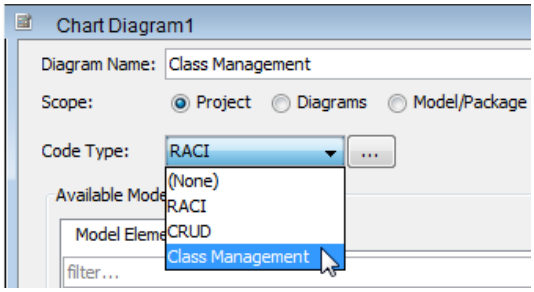
5. Enter the name of the code in **Name** and a letter which presents the code in **Code**. Click the **...** button next to **Color** and select a color for the code from the pop-up menu.



Enter the properties of code

6. Click **OK** button.

7. The chart type you created previously is currently available from the combo box of **Code Type**. Select it from the combo box of **Code Type**.



*Select a code type from the combo box of **Code Type***

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Generating HTML/PDF/Word report

Report generation is the process of producing a report for sharing your design work and model specification with teammates and clients. You can generate reports in different formats such as HTML, PDF or MS Word for reading or publishing in different environments. In this chapter, you will see how to generate a report, and how to configure the generation process and output.

Generating report

Shows you the basic steps in generating a HTML, PDF or MS Word report.

Configuring report

Gives you a description of all the report generation options.

Generating report

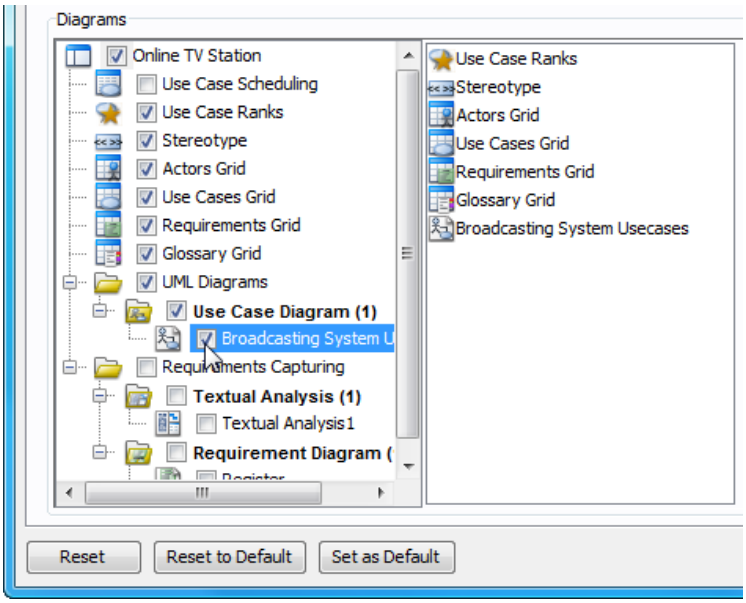
Report generation is the process of producing a report for sharing your design work and model specification with teammates and clients. You can generate reports in different formats such as HTML, PDF or MS Word for reading or publishing in different environments. They differ in file format, but have the same layout. In this chapter, we will go through the core steps in report generation.

To generate a report:

1. Select **Tools > Report** from the main menu, and then select **Generate HTML Report...**, **Generate PDF Report...** or **Generate Word Report...** depending on the type of report you want to generate.
2. In the **Generate HTML/PDF/Word** dialog box, fill in the output path where the report should be generated to.

> **NOTE:** For HTML report, specify the folder of the HTML files to be generated.
> For PDF report, specify the file path of PDF file (*.pdf) to be generated.
> For MS Word report, specify the file path of the document file (*.docx) to be generated.

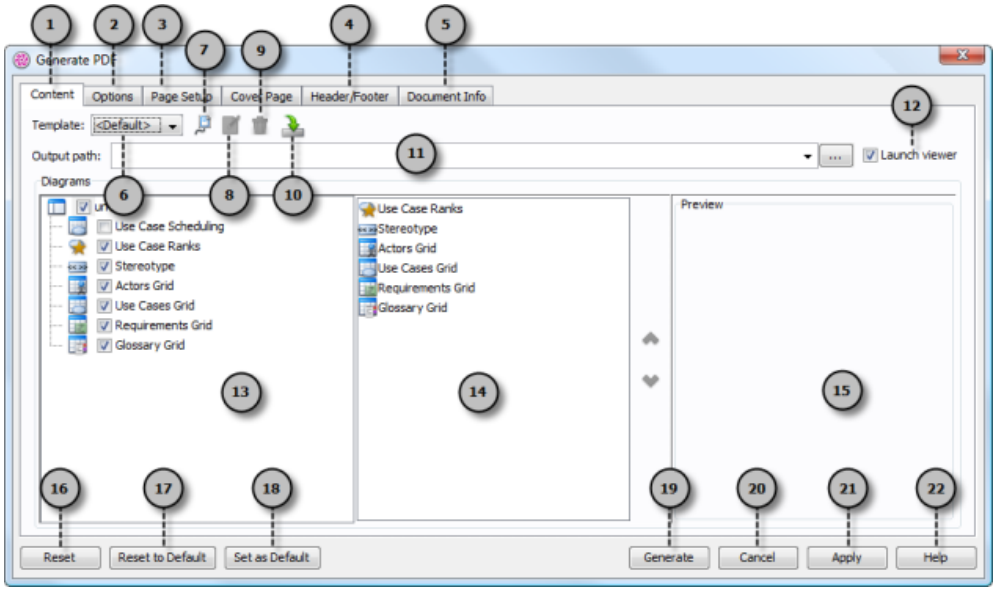
3. Select the grid(s) and/or diagram(s) to be included in report.



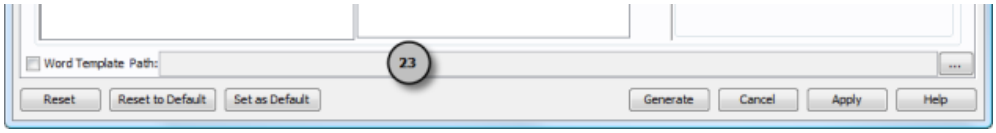
Select diagram to include in report

4. Make any necessary configuration such as the page layout, cover page, etc. For details about configuration, refer to the next chapter.
5. Click **Generate** to proceed with generation.

Overview of report generation dialog box



Overview of report generation dialog box



The bottom part of report generaition dialog box for MS Word report

No.	Name	Description
1	Content	The main page of report generation that lets you select the diagram(s) to generate report.
2	Options	Configurable options for detailed report configuration.
3	Page Setup	The setup of layout of report.
4	Header/Footer	The content of header and footer.
5	Document info	To define document info.
6	Template	You can define a template for report generation by clicking on the drop down menu next to **Template**, and selecting **<New>**. Once a template is defined, you can select it from the same drop down menu, and proceed with generation with the template. For details about report customization, read the next chapter.
7	Use external template	Click to link to an external template file.
8	Edit template	Click to edit the template selected in the drop down menu of **Template**.
9	Delete template	Click to delete the template selected in the drop down menu of **Template**.
10	Import template	Click to import a template file into the current project.
11	Output path	The output path of report to be generated.
12	Launch viewer	Check to open the report automatically after generation.
13	Available diagram list	The list of diagrams in opening project.
14	Selected diagram list	The list of diagrams selected to generate report.
15	Preview	Preview of diagram being selected in the list of selected diagram
16	Reset	Reset changes made in this dialog box
17	Reset to default	Reset changes made in this dialog box to default settings.
18	Set as default	Set the settings in this dialog box to default.
19	Generate	Click to generate report.
20	Cancel	Click to close the report dialog box.
21	Apply	Click to apply the changes made in report, causing reopening of this dialog box to restore the applied settings.
22	Help	Click to read the help contents.
23	Word template path	Available only to MS Word report generation, this option enables you to specify the path of MS Word document file that you want the generator to use as template. Report generator will append the template file content in front of generated report. In other words, you can prepare a file for cover page. Apart from this, style will also follow the definition in template file. For details, please read the section *Generating MS Word report with template (MS Word report only)*.

Description of report generation dialog box

Generating MS Word report with template (MS Word report only)

At the bottom of the MS Word report generation dialog box, there is an option **Word template path**, with a text box next to it for filling in the path of template file. A Word template file provides the start up contents and defines the style of report. During report generation, the generator will make a copy of the template file, treat the copied file as base, append the generated content to the copied file, and save the file to the destination path. By using a word template, you can define your own headers/footers, cover page, start up content, styles for your generated report.

Details

User

Name	Value
Visibility	public
Abstract	false
Leaf	false
Root	false
Author	Peter
Create Date Time	Feb 9, 2010 8:30:18 AM
Last Modified	Feb 9, 2010 8:38:14 AM
Business Model	false
Actor ID	1

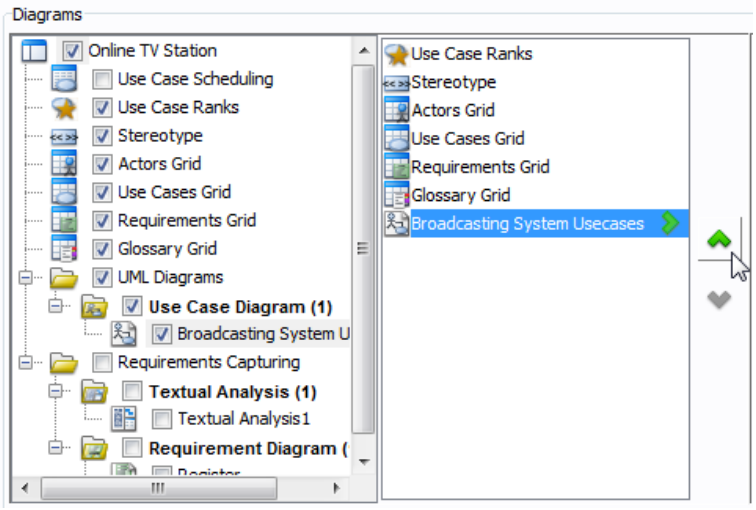
Relationships

Unamed Association		
To	Name	Value
	End Model Element	Compose Mail
	Author	Peter
	Create Date Time	Feb 9, 2010 8:30:27 AM
	Last Modified	Feb 9, 2010 8:38:14 AM

A generated report with style defined in template applied

Sorting diagrams in report

By default, diagrams show in report follows the order defined in diagram tree in the **Generate PDF** dialog box. We may, however, sort the diagrams to make them appear in desired sequence. To sort diagram(s), select the diagram(s) to be ordered on the list at the center of dialog box, and click or to sort.

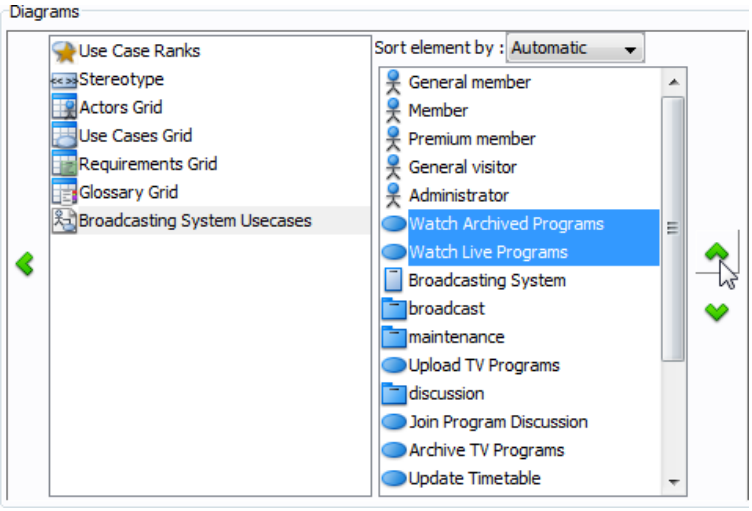


Reorder diagram

Sorting shapes in report

Custom sorting

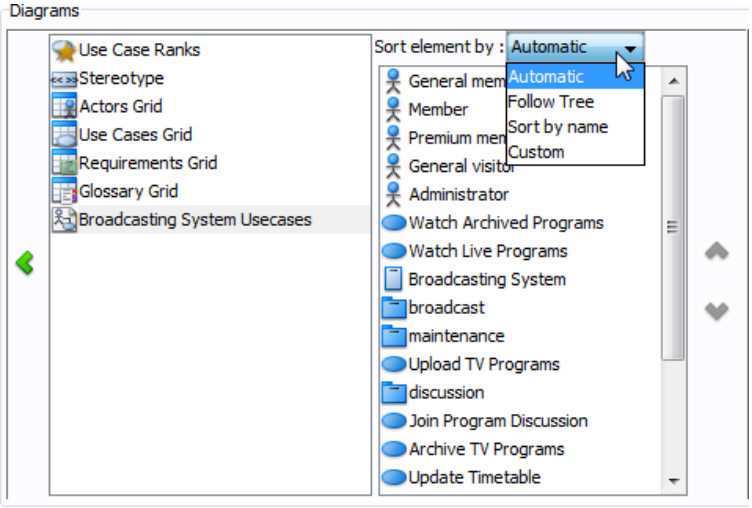
1. Select the diagram to sort and click to expand it.
2. Select the shape(s) to sort.
3. Click or to sort.



Sort shapes

Automatic sorting

1. Select the diagram to sort and click to expand it.
2. Select from the drop down menu **Sort element** a way to sort.



Select the way to sort shape

Way of sorting	Description
Automatic	The way of sorting elements is automatically managed. The order is often based on the flow and/or position of elements, which is the most logical order, following most users' understanding of that kind of diagram.
Sort by Tree	Sort diagram elements by following the order defined in Diagram Navigator.
Sort by Name	Sort diagram elements alphabetically base on their name, in ascending order.
Custom	The way of sorting elements is controlled by user, through selecting elements and pressing or .

Different ways of sorting

Related Resources

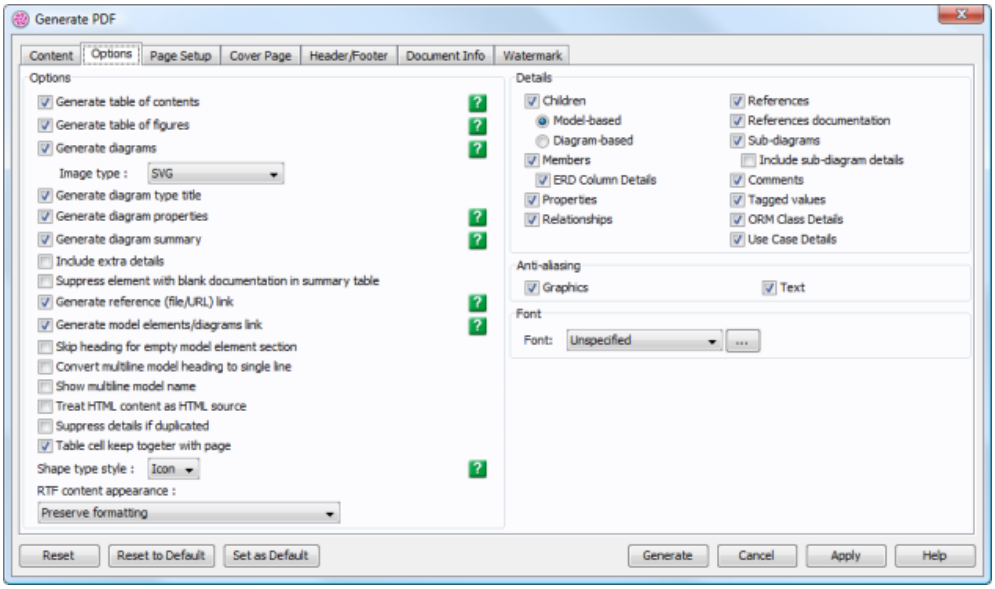
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Configuring report

Report generation can be configured to make the output more close to your expectation. Common configuration options include whether to generate table of contents/figure or not, whether to generate shape/diagram type as icon or text, and whether to generate a particular type of detail such as children. Besides configuration options, you can also adjust the page setup, design the cover page and define header/footer. In this chapter, we will go through all the options one by one.

Options



Options

Option	Description
Generate table of contents	If this option is selected, table of content for this document will be generated to the report
Generate table of figures	If this option is selected, table of figures for this document will be generated to the report
Generate diagrams	If this option is selected, the image of the selected diagrams will be generated to the report. For PDF report, you can select the type of diagram. Here are the possible selections: • PNG - Images will look exactly the same as the diagrams in your project, but not scalable against zooming. • SVG - Due to its scalable nature, image content will remain clear regardless of the level of zooming. However, image may look a bit different from the original diagram as there is a conversion re-construction from raster graphic data to SVG image. • SVG (text as shape) - Base on SVG, this option makes any text on diagram become text object, making it possible to select them in report content.
Generate diagram type title	If this option is selected, the type of diagram (e.g. archimate diagram, business motivation model diagram, etc) will be generated to the report, above the diagram name.
Generate diagram properties	If this option is selected, the properties of the selected diagrams will be generated to the report.
Generate diagram summary	If the option is selected, the summary of the selected diagrams will be generated to the report.
Include extra details	If the option is selected, information like ID and stereotypes will be generated to the summary table of report.
Suppress element with blank documentation in summary table	If the option is selected, diagram elements without documentation defined will not be generated to summary table.
Generate reference (file/URL) link	Select to generate links for referenced files/URLs defined in models.

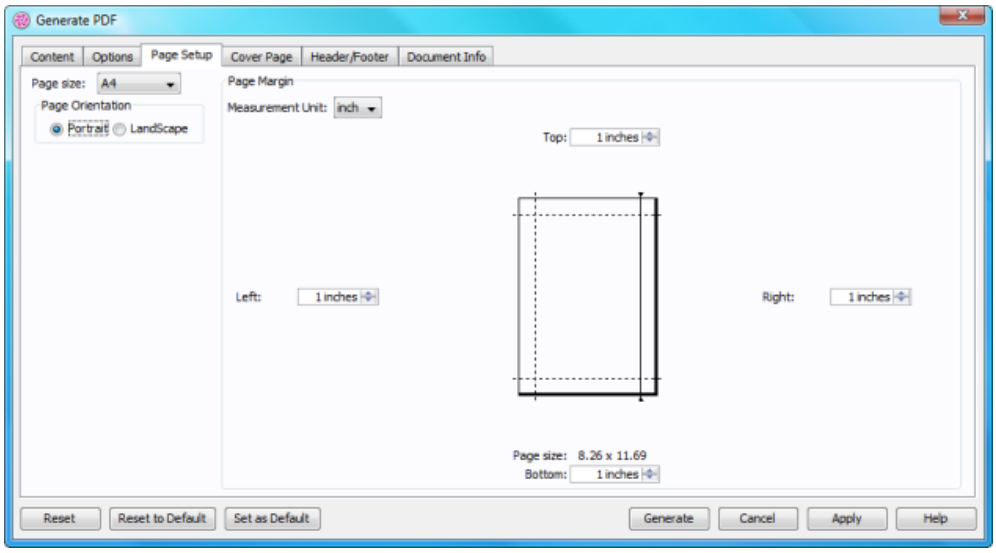
Generate model elements/diagrams link	Select to generate links for navigating to related models and diagrams.
Skip heading for empty model element section	If this option is selected, heading for empty model element section will be skipped.
Convert multiline model heading to single line	If this option is selected, multiline model heading will be converted to single line.
Show multiline model name	If this option is selected, non heading multiline model name will remain in multiline, instead of being converted to single line.
Treat HTML content as HTML source	If this option is selected, HTML content will be treat as HTML source.
Suppress details if duplicated	If this option is selected, duplicated details will be suppressed.
Table cell keep together with page	If this option is selected, table cell will try to show on a page completely instead of breaking into separate pages. This option is only available to PDF report.
Shape type style	Icon - using Icon to represent the type of shape and diagram elements Text - using text to represent the type of shape and diagram elements
RTF content appearance	Preserve formatting - using original formatting for RTF content Make font size consistent with the rest of the report - using same font size for RTF content in whole report Display in plain text - using plain text for RTF content
Copy reference files	If this option is selected, referenced files will be copy to output folder of report. With this option, you can copy the whole report folder to another machine and read there, without having broken file linkage for references.
Details	Select a kinds of content to generate it. Children - Everything a shape is containing. When selected, you can further select Model-based or Diagram-based for controlling the scope of children. Model-based consider all children the model of view contained. Diagram-based only consider the view in generating diagram. Let say if you have a package containing several classes. By selecting Model-based, all classes will be considered. By selecting Diagram-based, only the classes that are contained by the package in the generating diagram will be considered. Members - Attributes and operations are example of members. Properties - Name, documentation, abstract, leaf are example of properties. Project management properties - Author, create date, version are example of project management properties. Relationships - Association, dependency are example of relationships Quality information - The quality of model elements - Bad, Fair, Good, etc. References - File, diagram, folder, URL, shape are possible kinds of reference References documentation - Determine whether to generate the referenced shape/diagram's documentation in reference table Sub-diagrams - Sub-diagrams of a shape Comments - Comments of shape Tagged values - Tagged values of shape ORM Class Details - ORM class details specialized for ORM Persistable class Use Case Details - Use case details of use case
Anti-aliasing	Determine the quality of report content. Graphics - To enable/disable the graphic anti-aliasing of the diagram images.

	Text - To enable/disable the text anti-aliasing of the diagram images.
Font	Determines the font family of report content. This option is only available to PDF report.
Encoding	Determines the encoding of HTML file to be generated. This option is only available to HTML report.

A description of general options

Page Setup

Page setup controls the layout of report. You can adjust a report size, page orientation and margin.



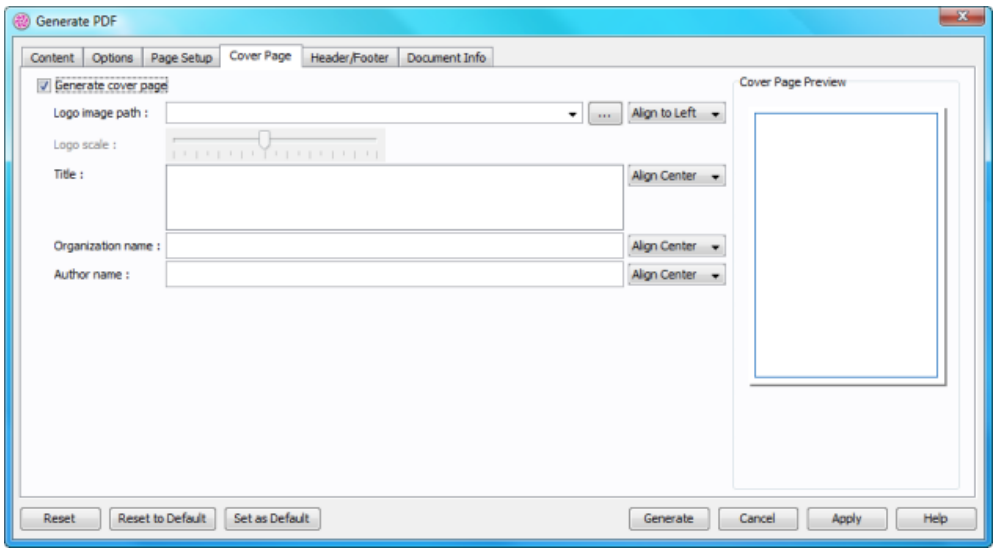
Page setup

Option	Description
Page size	To select the paper size of the generated report.
Page orientation	This option is used to select the orientation of the report (portrait/landscape). This option is only available to PDF and MS Word report.
Page margin	To specify the page margins of the report. This option is only available to PDF and MS Word report.

A description of options of page setup

Cover Page (Front Page for HTML report generation)

Cover page is the first page of report. You can add your company logo there, and enter the report title, organization namd and author name. Notice that in HTML report generation, the tab **Cover Page** is named as **Front Page**.



Cover page

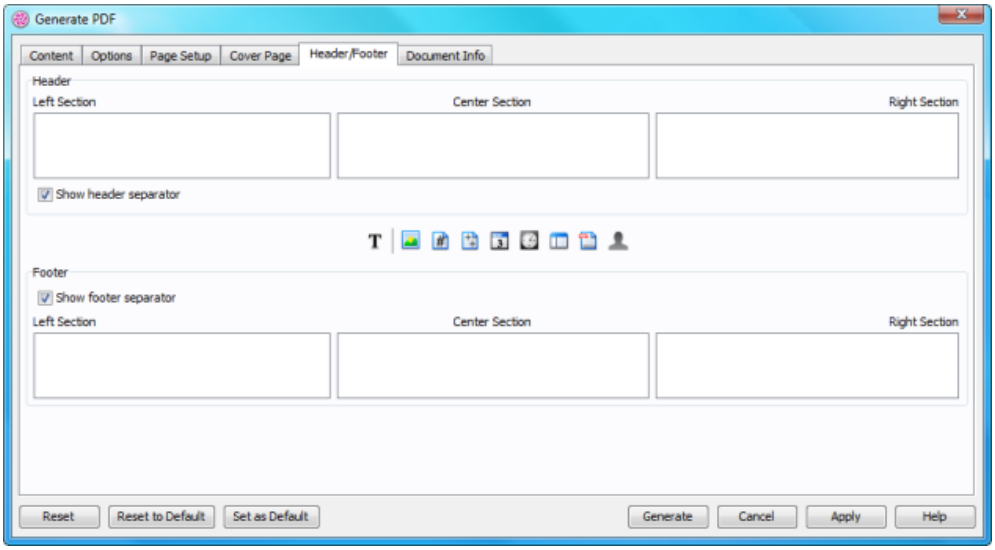
Option	Description
Generate cover page (PDF and MS Word) Generate front page (HTML)	If this option is selected, there will be a cover page generated to the report.
Logo image path	An image that appears at the report. You are expected to supply the file path of the image file. The drop down menu at the right hand side is for controlling the position of image.
Logo scale	Control the scale of logo image. This option is only available to PDF and MS Word report.
Title	The title text
Organization name	The organization name text
Author name	The author name text

A description of options of cover page

Header/Footer

Header and footer refers to the content that appear in the top and bottom of every page in report. For MS Word report, there are two text boxes for you to edit the header and footer. For PDF report, there are six boxes, three for each of header and footer. Each of the text box represent a region in header/footer, such as the top left text box refers to the left region of header, while the bottom right text box refers to the right region of footer.

Instead of typing in the content of header/footer, there are a set of variables for you to apply with. The following table provides you with description with each of the variable.



Header/Footer

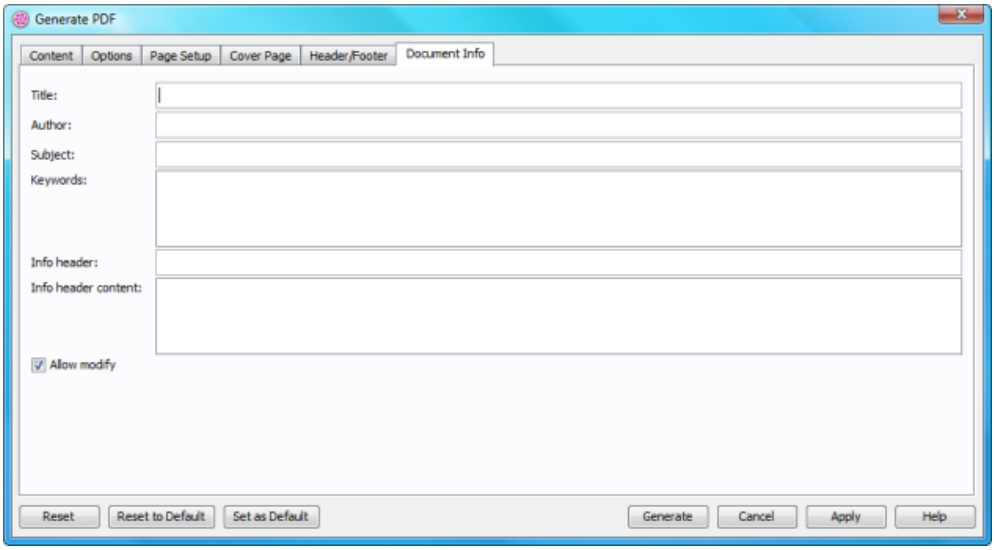
Variable	Name	Description
T	Selection font	Font settings of selected content
	Align to left	Align content to the left of header/footer. This option is available only to MS Word report.
	Align to center	Align content to the center of header/footer. This option is available only to MS Word report.
	Align to right	Align content to the right of header/footer. This option is available only to MS Word report.
	Add image	Insert an image to the position where the text cursor is placing.
	Insert page number	Insert page number to the position where the text cursor is placing.
	Insert page count	Insert page count to the position where the text cursor is placing.
	Insert date	Insert the date of when the report is generated to the position where the text cursor is placing.
	Insert time	Insert the time of when the report is generated to the position where the text cursor is placing.
	Insert project name	Insert the project name to the position where the text cursor is placing.

	Insert report file name	Insert the name of report file to the position where the text cursor is placing.
	Insert user name	Insert the name of the user logging into the system to the position where the text cursor is placing.

A description of variables that can be used in header and footer

Document Info

For HTML report, document info refers to the meta information of HTML document. For PDF and MS Word report, document info refers to the possible document properties that can be defined.



Document info

Option	Description
Title	The title of report.
Author	The author of the report. This option is only available to PDF report.
Subject	The subject of the report.
Keywords	The keywords of the report.
Info header	The info header of the report. This option is only available to PDF report.
Info header content	The info header content of the report. This option is only available to PDF report.
Allow modify	Select to allow modification on the report. This option is only available to PDF report.

A description of document info

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Customizing report

Instead of generating a report in a way that Agilian defined for you, you can develop report templates to customize the report content, to make output match your needs. In this chapter, you will see how to make use of the template editor to customize a report template.

Customizing report

Shows you how to launch the template editor to perform basic editing. It also describes the editor window in brief.

Export/import report template

Shows you how to export a template to a template file and import it into another instance.

Diagram loop

Define diagram loop and show you how to use a diagram loop in practice.

Diagram summary

Define diagram summary and show you how to use a diagram summary in practice.

Diagram paragraph

Define diagram paragraph and show you how to use a diagram paragraph in practice.

Element loop

Define element loop and show you how to use a element loop in practice.

Element summary

Define element summary and show you how to use a element summary in practice.

Element paragraph

Define element paragraph and show you how to use a element paragraph in practice.

Custom content

Define custom content and show you how to make use of it to enter formatted free text.

Diagram image

Define diagram image and show you how to make use of it to add an image of diagrams.

Property value

Define property value and show you how to make use it to extract a property from a parent model element.

Page break

Shows you how to insert a page break.

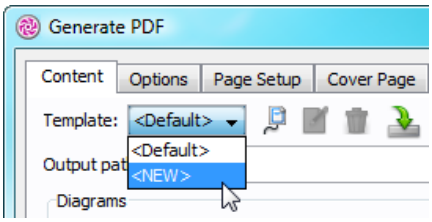
Customizing report

Instead of generating a report in a way that Agilian defined for you, you can develop report templates to customize the report content, to make output match your needs. With report customization, you can select model elements and properties to generate to report. You also can add custom text content. To customize report:

1. Select **Tools > Report** from the main menu. Then select **Generate HTML Report...**, **Generate PDF Report...** or **Generate Word Report...** depending on the type of report you want to generate.

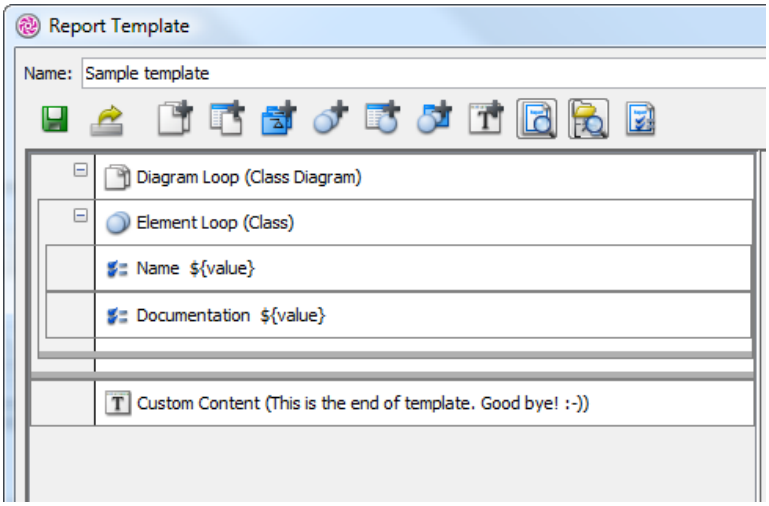
> **NOTE:** Report templates are shared among HTML, PDF and Word reports. If you just want to design template, but have no preference on the type of report to generate yet, just select either HTML, PDF or Word.

2. Press on the **Template** drop down menu and select **<New>** from the popup menu to open the report template editor and start editing template.



To create a template

3. In the **Report Template** dialog box, enter the template name at the top of dialog box.
4. Construct the report template. A report template is formed by different kinds of components that put together in a hierarchical structure. A common way of building a template is to start with a diagram loop, which can be created from the editor toolbar. Then, select the diagram loop and create children components through its toolbar, such as to create an image component or a element loop for accessing diagram elements on diagrams being looped. To learn how to use the tools in detail, refer to the coming chapters.



A sample template showing the use of diagram loop, element loop, property and custom text components

5. Click **Save** button to save your work.
6. Click **Close** button.

An overview of tools in template editor

Tool	Name	Description
	Save	Save the opening template.
	Save as	Save the opening template as a new one.
	Exoprt	To export the opening template to a report template file (.vpr). You can import the file to other machines for reusing it.
	Add diagram loop	To add a component to template editor, indicating the need of looping specific type(s) of diagram.
	Add diagram summary	To add a component to template editor, indicating the need of looping specific type(s) of diagram for constructing a tabular diagram summary.
	Add diagram paragraph	To add a component to template editor, indicating the need of looping specific type(s) of diagram for printing its properties in paragraph form.

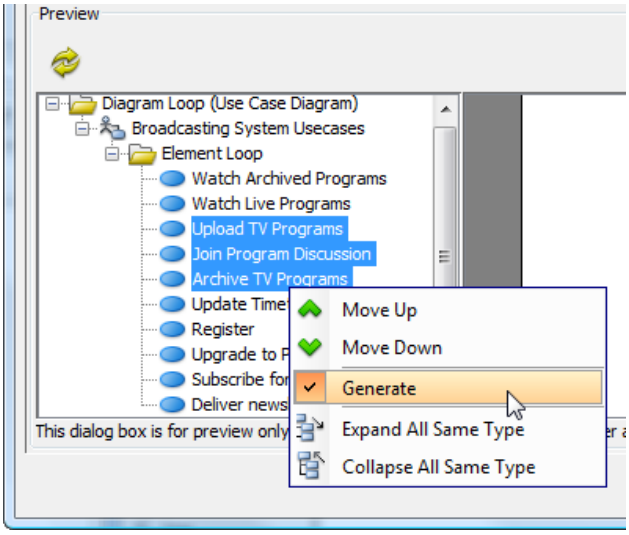
Add element loop (Model)	To add a component to template editor, indicating the need of looping Model.
Add root level element loop	To add a component to template editor, indicating the need of looping specific type(s) of model element that are at project root (i.e. not being contained).
Add element summary	To add a component to template editor, indicating the need of looping specific type(s) of model element for constructing a tabular element summary.
Add element paragraph	To add a component to template editor, indicating the need of looping specific type(s) of model element for printing its properties in paragraph form.
Add all level element loop	To add a component to template editor, indicating the need of looping model elements of specific type(s) in whole project.
Add custom content	To add a component to template editor, indicating the placement of text written by user.
Preview template content	Preview the template against the project content for report content. Rendering of report is costly especially when the project is large. If your project is large, be patient when waiting for outcome.
Preview template structure	Preview the template against the project content for report structure.
Options	Configure report options.

Description of differnet tools in template editor

Ignoring specific diagrams/shapes

A report template is independent of any project file. But if you try to apply a template on a project, you can ignore generating specific diagrams or shapes in that project. To ignore specific diagrams/shapes:

1. Open the template in template editor.
2. Preview the report by pressing or from the toolbar.
3. In the report structure tree, select the diagrams or shapes that you want to ignore in generated report.
4. Right click on your selection and de-select **Generate** from the popup menu.



To not generate model elements

More about Preview

When you try to preview a template, it tries to apply the template on the opening project to form the report structure and to render the preview. Due to the connection between template and project, there are several actions that you can perform with the preview.

Ignoring specific diagrams/shapes

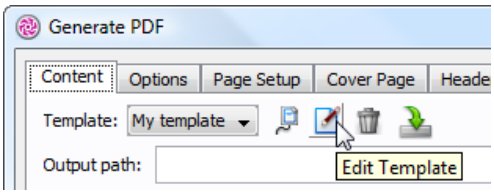
In the report structure tree, right click on the diagram(s) or shape(s) that you want to ignore when generating report and de-select **Generate** from the popup menu.

Reordering diagrams/shapes

In the report structure tree, right click on the diagram(s) or shape(s) that you want to re-order and select **Move Up** or **Move Down** to reposition them.

Editing a template

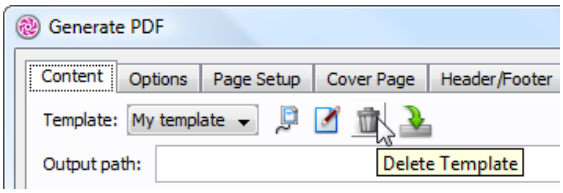
To edit an existing report template, select it in the generate dialog box such as **Generate PDF**, then click on the edit button.



Edit a template

Deleting a template

To delete an existing report template, select it in the generate dialog box such as **Generate PDF**, then click on the delete button.



Delete a template

Related Resources

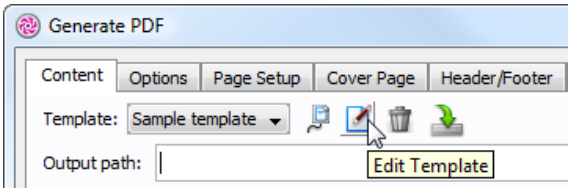
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Export/import report template

You can apply a report template on another project by exporting it as a report template file, and importing it to the target project. To export and import template:

1. Open the report template in template editor.



To open a template in template editor

2. Click on the button at the top of template editor to export template.
3. In the **Export** dialog box, enter the file name and click **Save**.
4. Open the project that you want the template to import to. Open the report dialog box by selecting **Tools > Generate HTML/PDF/Word Report** from the main menu.
5. Click on the button.
6. In the **Import** dialog box, select the report template file and click **Open**.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

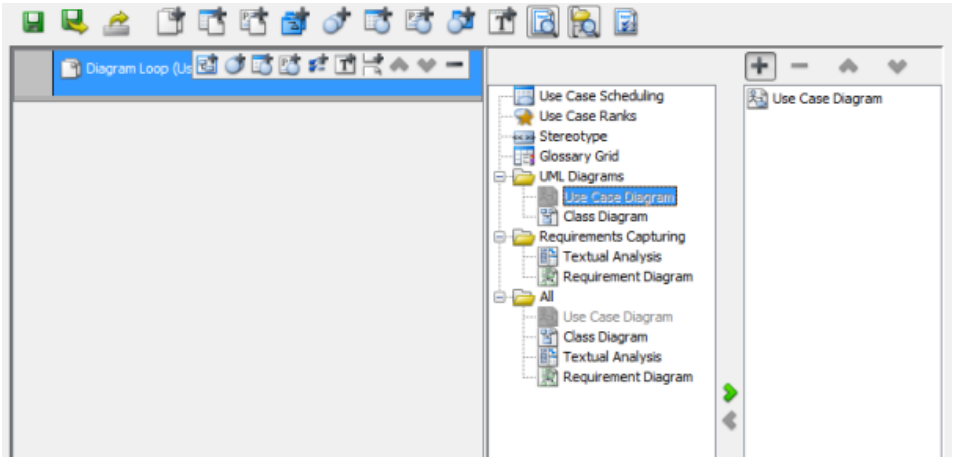
Diagram loop

A diagram loop is a component in a report template, indicating the need of looping (i.e. walking through) specific type(s) of diagram. For example, to loop all use case diagrams and class diagrams in a project. Diagram loop means nothing more than just to loop diagram. The content to print for each diagram being looped is to be determined by the children components of loop.

Looping diagrams in project

To loop specific type(s) of diagrams in project, create diagram loop at template root. To create diagram loop at template root:

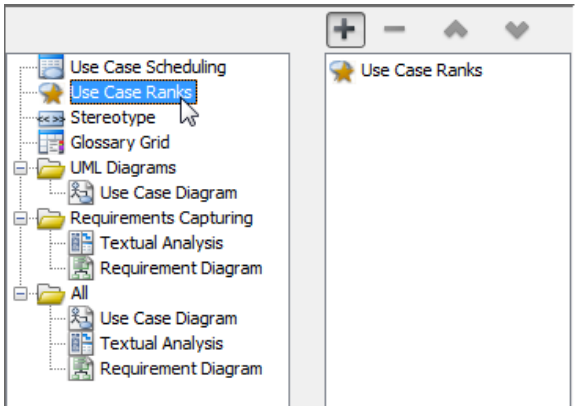
1. In the template editor, click on the on toolbar.
2. On the right hand side of the template editor, select the type of diagram(s) to loop and click to confirm the selection.



Select to loop use case diagram

Including use case scheduling,ranks, stereotypes and various grid in report

Use case scheduling, use case ranks, stereotypes and various grid fall into the category of diagram. If you want to print them to report, follow the steps as described in the previous section, and to select the appropriate content to include at the final step.



Choosing to include information of use case ranks in template

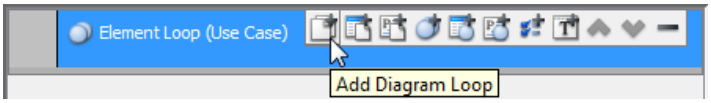
Looping sub-diagrams of specific element

Instead of looping diagrams in a project, you can also place diagram loop under an element loop to loop for sub-diagrams of specific type(s) of model element. To create diagram loop under an element loop:

1. Select the element loop that you want to loop for its sub-diagrams.

Selecting an element loop

2. Click on the **Add Diagram Loop** button from the toolbar of element loop.



Adding a diagram loop

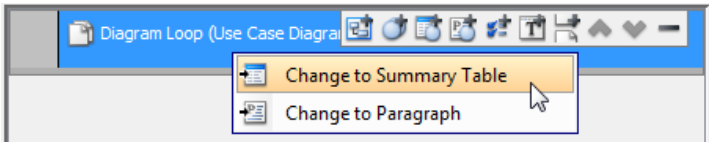
NOTE: Make sure you are clicking on the button from the toolbar of element loop. If you are clicking on the button at editor toolbar, you will create a loop at template root.

3. On the right hand side of the template editor, select the type of sub-diagram(s) to loop and click to confirm the selection.

Switching from diagram loop to diagram summary table/paragraph

A diagram summary indicates the need of looping specific type(s) of diagram for constructing a tabular diagram summary, while a diagram paragraph indicates the need of looping specific type(s) of diagram for constructing paragraphs of properties.

You can convert a diagram loop to diagram summary table or paragraph by right clicking on a diagram loop and selecting **Change to Summary Table/ Paragraph** from popup menu.



To convert a diagram loop to diagram summary table/paragraph

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

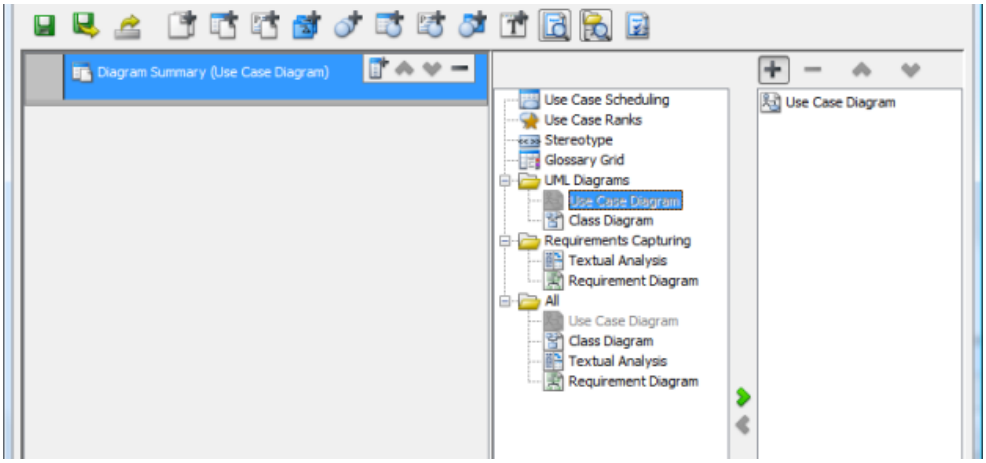
Diagram summary

A diagram summary is a component in a report template, indicating the need of looping (i.e. walking through) specific type(s) of diagram, and presenting their properties in tabular form. For example, to loop all use case diagrams in a project and form a table consisting of diagram names and documentation. By default, a table of diagram names will be printed. You are expected to add property column(s) under a diagram summary to indicate the diagram properties to print in the table.

Showing a property table for diagrams in project

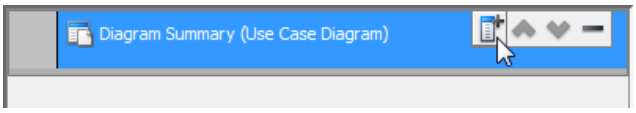
To loop specific type(s) of diagrams in project for listing their properties in a table, create diagram summary at template root. To create diagram summary at template root:

1. In the template editor, click on the on toolbar.
2. On the right hand side of the template editor, select the type of diagram(s) to loop and click to confirm the selection.



Select to loop use case diagram

3. If you generate report with a template like this, you will obtain a table of diagram names, provided that there exists diagram(s) of the chosen type(s). If you need to print specific diagram properties in table other than just name, click on the **Add Property Column** button in the toolbar of diagram summary, then select the property(ies) to add into the table as column(s). For more details about the use of property, read the chapter *Property*.

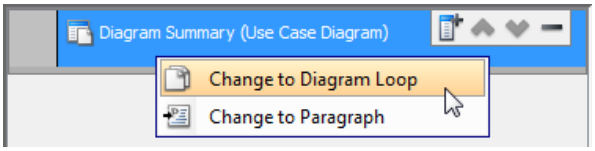


To add a property column in diagram summary

Switching from diagram summary to diagram loop/paragraph

A diagram loop indicates the need of looping specific type(s) of diagram, while a diagram paragraph indicates the need of looping specific type(s) of diagram for constructing paragraphs of properties.

You can convert a diagram summary to diagram loop or paragraph by right clicking on a diagram summary and selecting **Change to Diagram Loop/ Paragraph** from pop-up menu.



To convert a diagram summary to diagram loop/paragraph

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

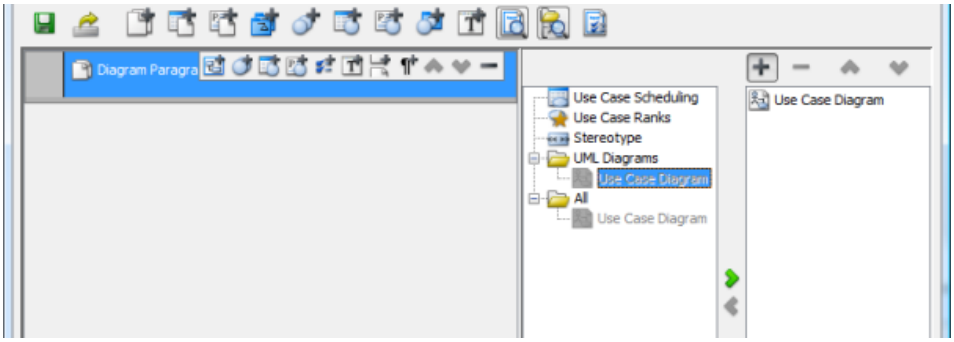
Diagram paragraph

A diagram paragraph is a component in a report template, indicating the need of looping (i.e. walking through) specific type(s) of diagram, and presenting their properties in paragraph form. For example, to loop all use case diagrams in a project and print out their documentation and last modified data in a paragraph, one paragraph per diagram. You are expected to add property value(s) under a diagram paragraph to indicate the diagram properties to print out.

Showing a property paragraph for diagrams in project

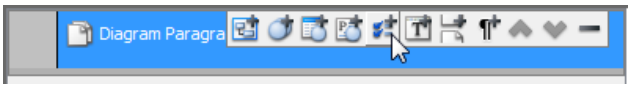
To loop specific type(s) of diagrams in project for listing their properties in paragraph form, create diagram paragraph at template root. To create diagram paragraph at template root:

1. In the template editor, click on the on toolbar.
2. On the right hand side of the template editor, select the type of diagram(s) to loop and click to confirm the selection.



Select to loop use case diagram

3. If you generate report with a template like this, you will obtain a list of diagram types and names, provided that there exists diagram(s) of the chosen type(s). If you need to print specific diagram properties other than just name, click on the **Add Property Value** button in the toolbar of diagram paragraph, then select the property(ies) to show. For more details about the use of property, read the chapter *Property*.

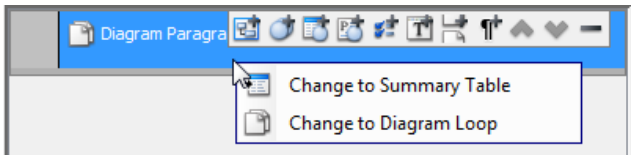


To add a property value in diagram paragraph

Switching from diagram paragraph to diagram summary table/loop

A diagram summary indicates the need of looping specific type(s) of diagram for constructing a tabular diagram summary, while a diagram loop indicates the need of looping specific type(s) of diagram.

You can convert a diagram paragraph to diagram summary table or loop by right clicking on a diagram paragraph and selecting **Change to Summary Table/Diagram Loop** from popup menu.



To convert a diagram paragraph to diagram summary table/loop

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Element loop

An element loop is a component in a report template, indicating the need of looping (i.e. walking through) specific type(s) of diagram/model elements. For example, to loop all use case and class in a project. Element loop means nothing more than just to loop diagram/model element. The content to print for each diagram/model element being looped is to be determined by the children components of loop.

Looping model elements in project

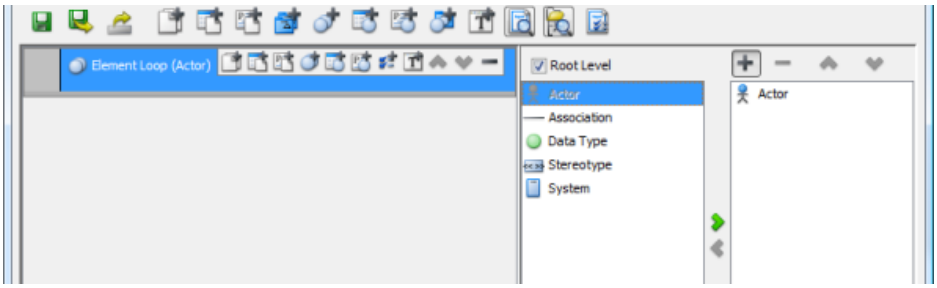
To loop specific type(s) of model elements in project, create element loop at template root. To create element loop at template root:

1. In the template editor, click on any of the buttons in the toolbar depending on your need.

Button	Name	Description
	Add Element Loop (Model)	To loop through all models in the project root. In other words, model being contained by other model element will not be accessed. This is a shortcut for creating a root level element loop whose chosen element type is model.
	Add Element Loop (Package)	To loop through all packages in the project root. In other words, package being contained by other model element will not be accessed. This is a shortcut for creating a root level element loop whose chosen element type is package.
	Add Root Level Element Loop	To loop through any kind of model element in project root. By selecting this option, you can choose the type of model element to be looped.
	Add All Level Element Loop	To loop through any kind of model element within the project, regardless of their leveling. By selecting this option, you can choose the type of model element to be looped.

Description of available type of element loop

2. If you have chosen to add a model or a package loop, you do not need to perform any actions in further.
If you have chosen to add either root level or all level element loop, select the type of element(s) to loop on the right hand side of the template editor and click to confirm the selection.



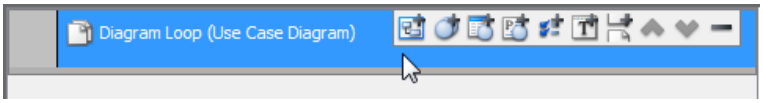
Select to loop actor

> **NOTE:** You can switch between a root level and a all-level loop by changing the option **Root Level** above the element list. Notice that changing the value of **Root Level** is not about changing available element selection, but also the end result, whether to access only root level elements or not.

Looping diagram elements in a diagram

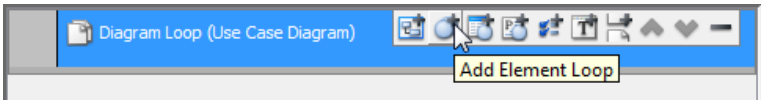
Instead of looping model elements in a project, you can also place element loop under a diagram loop to loop for diagram elements in specific type(s) of diagram. To create element loop under a diagram loop:

1. Select the element loop that you want to loop for its sub-diagrams.



Selecting a diagram loop

2. Click on the **Add Element Loop** button from the toolbar of element loop.



Adding an element loop

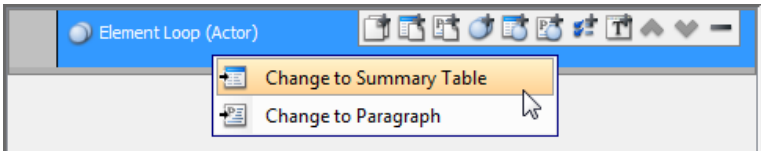
> **NOTE:** Make sure you are clicking on the button from the toolbar of diagram loop. If you are clicking on the button at editor toolbar, you will create a loop at template root.

3. On the right hand side of the template editor, select the type of diagram element to loop and click to confirm the selection.

Switching from element loop to element summary table/paragraph

An element summary indicates the need of looping (i.e. walking through) specific type(s) of diagram/model element, and presenting their properties in tabular form, while an element paragraph indicates the need of looping specific type(s) of diagram/model element for constructing paragraphs of properties.

You can convert an element loop to element summary table or paragraph by right clicking on a element loop and selecting **Change to Summary Table/Paragraph** from popup menu.



To convert an element loop to element summary table/paragraph

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Element summary

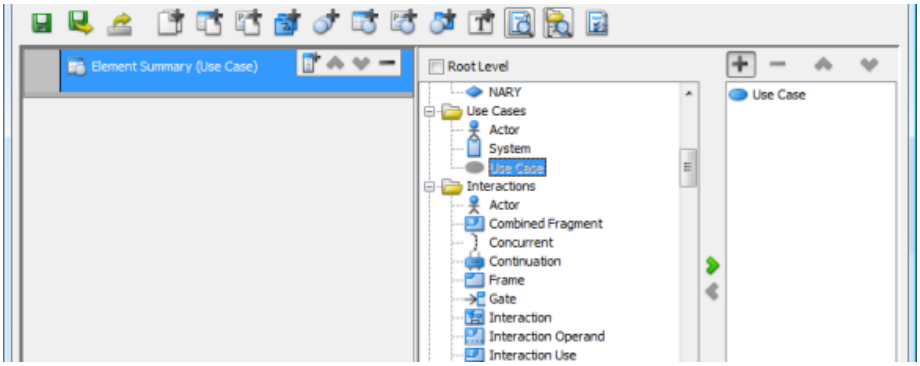
An element summary is a component in a report template, indicating the need of looping (i.e. walking through) specific type(s) of diagram/model element, and presenting their properties in tabular form. For example, to loop all use cases in a project and form a table consisting of use case IDs and documentation. By default, a table of element names will be printed. You are expected to add property column(s) under an element summary to indicate the element properties to print in the table.

Showing a property table for elements in project

To loop specific type(s) of model elements in project for listing their properties in a table, create element summary at template root. To create element summary at template root:

1. In the template editor, click on the on toolbar.

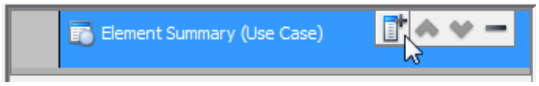
2. On the right hand side of the template editor, select the type of model element(s) to loop and click to confirm the selection.



Select to loop use case

> **NOTE:** By default, element summary added to project root enables the looping of root level elements.
> If you want to change to access elements in all levels, change the option **Root Level** above the element list.
> Notice that changing the value of **Root Level** is not about changing available element selection, but also the end result, whether to access only root level elements or not.

3. If you generate report with a template like this, you will obtain a table of element names, provided that there exists element(s) of the chosen type(s). If you need to print specific element properties in table other than just name, click on the **Add Property Column** button in the toolbar of element summary, then select the property(ies) to add into the table as column(s). For more details about the use of property, read the chapter *Property*.

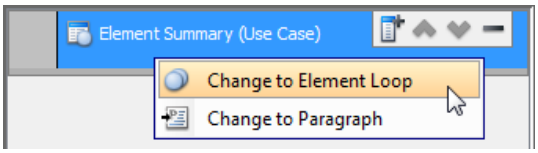


To add a property column in element summary

Switching from element summary to element loop/paragraph

An element loop indicates the need of looping specific type(s) of model/diagram element, while an element paragraph indicates the need of looping specific type(s) of diagram/model element for constructing paragraphs of properties.

You can convert an element summary to element loop/paragraph by right clicking on an element summary and selecting **Change to Element Loop/ Paragraph** from popup menu.



To convert a element summary to element loop/paragraph

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Element paragraph

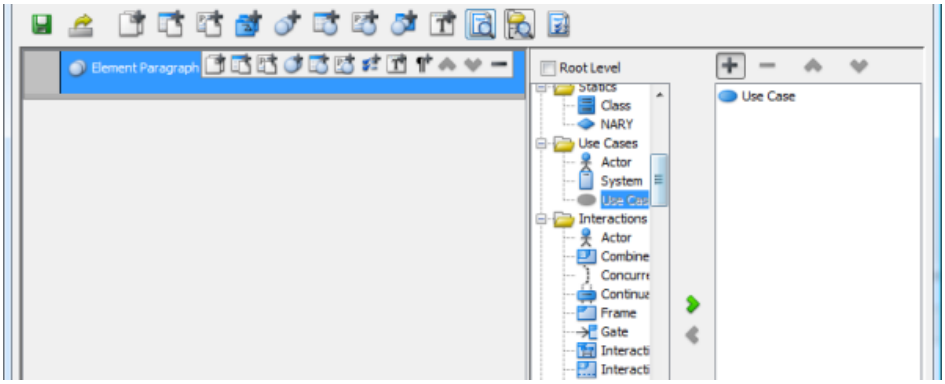
An element paragraph is a component in a report template, indicating the need of looping (i.e. walking through) specific type(s) of diagram/model element, and presenting their properties in paragraph form. For example, to loop all use cases in a project and form paragraphs consisting of use case IDs and documentation. By default, a list of element names will be printed. You are expected to add property value(s) under an element paragraph to indicate the element properties to print out.

Showing a property paragraph for elements in project

To loop specific type(s) of model elements in project for listing their properties in paragraph form, create element paragraph at template root. To create element paragraph at template root:

1. In the template editor, click on the on toolbar.

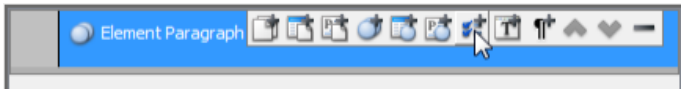
2. On the right hand side of the template editor, select the type of model element(s) to loop and click to confirm the selection.



Select to loop use case

> **NOTE:** By default, element paragraph added to project root enables the looping of root level elements.
> If you want to change to access elements in all levels, change the option **Root Level** above the element list.
> Notice that changing the value of **Root Level** is not about changing available element selection, but also the end result, whether to access only root level elements or not.

3. If you generate report with a template like this, you will obtain a list of element names, provided that there exists element(s) of the chosen type(s). If you need to print specific element properties other than just name, click on the **Add Property Column** button in the toolbar of element paragraph, then select the property(ies) to show. For more details about the use of property, read the chapter *Property*.

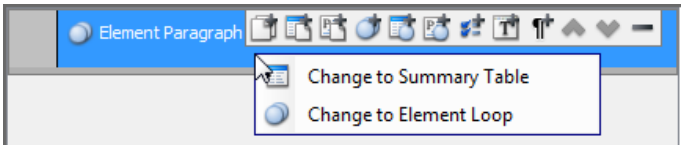


To add a property value in element paragraph

Switching from element paragraph to element summary table/loop

An element summary indicates the need of looping specific type(s) of diagram/model element for constructing a tabular element summary, while an element loop indicates the need of looping specific type(s) of diagram/model element.

You can convert an element paragraph to element summary table or loop by right clicking on a element paragraph and selecting **Change to Summary Table/Element Loop** from popup menu.



To convert a element paragraph to element summary table/loop

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

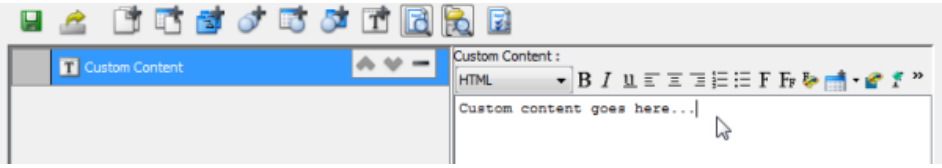
Custom content

Custom content is a component in a report template, which acts as a placeholder of user written text. For example, to add a section of acknowledgment at the beginning of report with custom text. You can write custom content in rich text, and you can add custom content to template root or under a diagram/element loop.

Adding custom content at template root

To add custom content at template root:

1. In the template editor, click on on toolbar.
2. On the right hand side of the template editor, enter the custom content. You can format your content through the buttons in the content pane. For details, read the section below about editing custom content.



Entering custom content

Adding custom content into a loop

To add custom content into a diagram/element loop:

1. Select the loop that you want to add custom content under it.

Selecting an element loop

2. Click on the **Add Custom Content** button from the toolbar of loop.

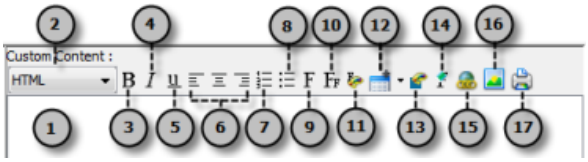
Adding custom content

> **NOTE:** Make sure you are clicking on the button from the toolbar of element loop. If you are clicking on the button at editor toolbar, you will create a loop at template root.

3. On the right hand side of the template editor, enter the custom content. You can format your content through the buttons in the content pane. For details, read the section below about editing custom content.

Editing custom content

You can write plain text in custom content as well as to add formatted text, images and tables through the help of the tools in the editor.



Custom content editor

No.	Name	Description
1	Editor pane	The editor where you can enter and edit custom content.
2	HTML	HTML - Read and edit the real content. HTML Source - Read and edit the HTML source of content.
3	Bold	Set the highlighted text to bold.
4	Italic	Set the highlighted text to italic.
5	Underline	Underline the highlighted text
6	Alignments	Set the alignment of highlighted text to right, center or left.
7	Ordered list	Add a numbered list.

8	Un-ordered list	Add a list with bullet points.
9	Font	Select the font family of highlighted text.
10	Font size	Select the size of highlighted text.
11	Font color	Select the color of highlighted text.
12	Table	Add a table.
13	Background color	Select the background color of highlighted text.
14	Clear formats	Clear formats of whole editor to convert the content to plain text.
15	Link	Add a hyperlink.
16	Image	Add an image.
17	Print	Print the custom content.

A description of custom content editor

Related Resources

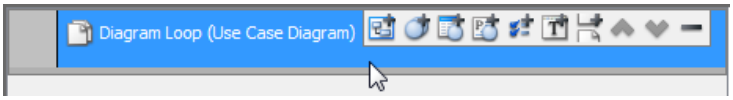
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagram image

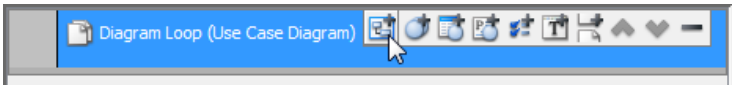
Diagram image is a component in a report template, which represents a placeholder of the image of a diagram under a diagram loop. You must place a diagram image under diagram loop. To add a diagram image:

1. Select the diagram loop that you want the images of diagrams to be printed.



Selecting a diagram loop

2. Click on the **Diagram Image** button from the toolbar of loop.



Adding a diagram image

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

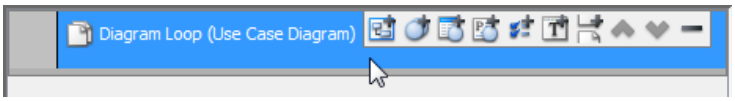
Property value

Property value is an element component in a report template, which represents the access of certain property of a diagram, model element or diagram element. For example, to print out the ID (property) of a use case. You can add property value into a diagram loop, a diagram summary, an element loop, an element summary. Property value added to a summary component will become a property column in summary table.

Adding property value into a loop or summary

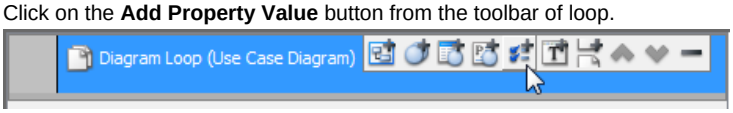
To add property value into a loop or summary:

1. Select the loop or summary that you want to add property value under it.



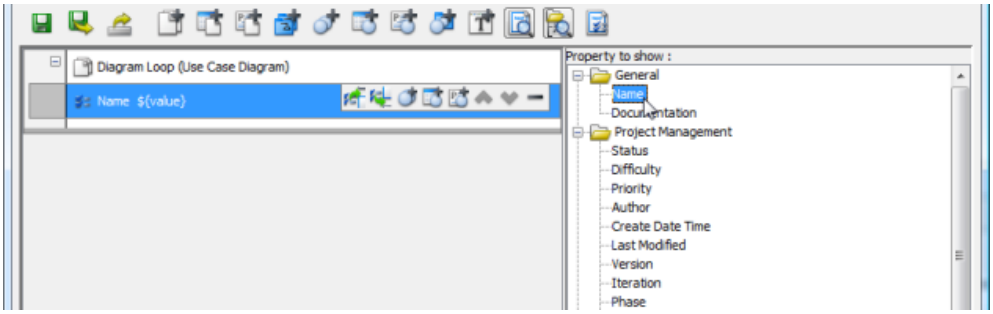
Selecting a diagram loop

2. Click on the **Add Property Value** button from the toolbar of loop.



Adding property value

3. On the right hand side of the template editor, select the property to access.

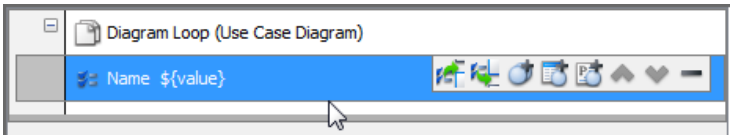


Selecting a property to access

Adding a property below another property

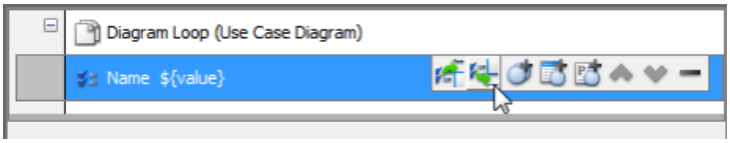
To add property below another property:

1. Select an existing property value.



Selecting a property value

2. Click on the **Add Property Value Below** or **Add Property Value Above** button.



To add a property value below an existing one

3. On the right hand side of the template editor, select the property to access.

Related Resources

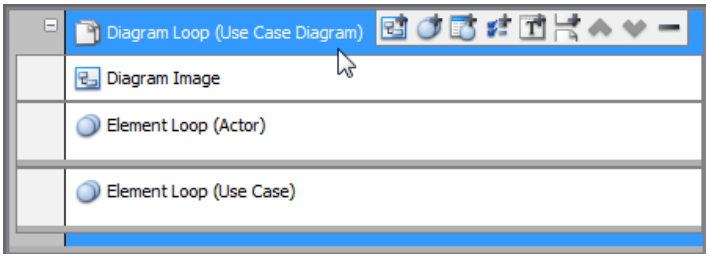
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Page break

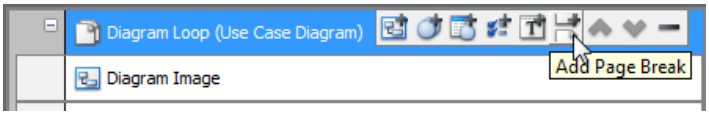
Page Break is where the report should start the contents which follows in a new page. It can be placed within a loop, either a diagram or element loop.
To create a Page Break:

1. Select the loop that you want to insert a page break.



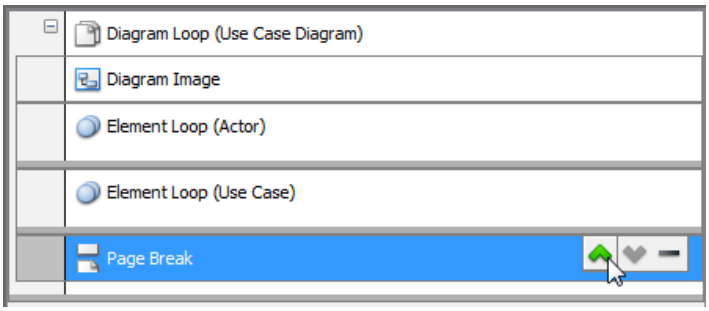
Selecting the loop for adding a page break

2. Click on the **Add Page Break** button from the toolbar of loop.



Adding a page break

3. Reposition the page break by pressing **Move Up** or **Move Down** from the toolbar of page break.



Repositioning a page break

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Publishing project to web site

The Project Publisher is a tool that exports the project, including detailed information in diagrams and models, into interactive and well-organized web pages. This chapter shows you how to publish a project.

Publish project using project publisher

Gives a brief description of publisher dialog box and guides you through the steps of publishing.

About publisher output

Describes the interactive features supported in published content.

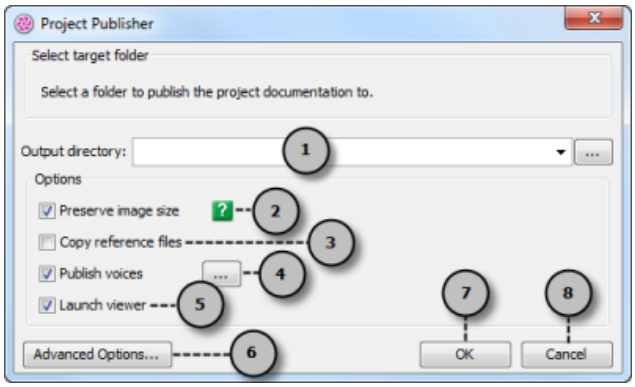
Publish project using project publisher

The Project Publisher is a tool that exports the project, including detailed information in diagrams and models, into interactive and well-organized web pages. The generated web pages can be read in any web browser with no additional plug-in required, so collaborative partners may see the published product even if they do not have Visual Paradigm products installed.

To launch Project Publisher:

1. Select **Tools > Project Publisher...** from main menu.
2. In the **Project Publisher** dialog box, specify the output directory where you want to save the published content.
3. You can optionally configure the publisher by adjust the options or options. For details, refer to the sections below.
4. Click **OK** button to start publishing.

An overview of project publisher



Overview of project publisher

No.	Name	Description
1	Output directory	The folder where you want to publish the content to.
2	Preserve image size	When checked, published content will show images in exact width and height.
3	Copy reference files	You can add file references to model elements. When this option is checked, referenced files will be copied to the output directory so that you can access any referenced file when browsing the published content in other machine easily.
4	Publish voices	When checked, audio clips recorded/linked to documentation will be publisher so that you can play them in published content.
5	Launch viewer	When checked, the system will launch the web browser and open the published Web contents.
6	Advanced options	Click to configure advanced publisher options. For details about the options, read the next section.
7	OK	Click to publish.
8	Cancel	Click to close the dialog box without publishing.

Description of project publisher

Advanced options

On the **Project Publisher** dialog box you can configure some of the common options. You can also configure the advanced options for more detailed settings by clicking the **Advanced Options...** button.

Option	Description
Generate model element list in diagram page	Check to generate a list of model element in a diagram page.
Generate only documentation in model element page	Check to generate only documentation in model element page, and exclude other contents.
Generate only when documentation is defined	Generate element page only when that element has documentation. When this option is off, the shape will have no linkage from image in diagram page due to the absent of element page.
Generate page header	Check to generate pre-defined header.
Generate page footer	Check to generate pre-defined footer.
Show documentation when hover over a shape	Check to show element's documentation when moving the mouse pointer over a shape in an image of diagram.

	Show procedure for BP task and sub-process - Popup the working procedures when you move the mouse pointer over BPMN tasks and sub-processes in an image. **Show test plan for test case** - Popup the test plan when you move the mouse pointer over a test case in an image.
Generate diagram type	Check to generate the diagram type in addition to diagram name.
Remove paragraph's top and bottom margin in RTF documentation	By default, top and bottom margins are added above and below RTF documentation text, due to the default style applied to the RTF documentation. If you want to remove the margins, you can override the setting by unchecking this option.
Generate referenced project diagrams	Check to include contents for diagrams in referenced project.
Generate quality information	Quality of model is assessed during the modeling. If you want to see the comment for each of your model elements, check this option to include in report.
Overwrite report style sheets	Project publishing is an "overwrite" action. If you publish to the same folder twice, files produced the first time would be overwritten. This option enables you to keep the style sheets file (.css) without being overwritten. This enables you to edit the styles defined and re-use it in subsequent publishing.
Generate grid configuration	Check to include configuration for grids, such as the type of elements to list and the scope (e.g. project/model/diagram, etc)
Generate menu	By default a diagram and model menus are shown on the left hand side of the published outcome. You can decide whether or not to generate the menus, or produce two index files, one with menu and another one without.
Always show indicators	Indicators refer to the small icons that show over shape(s) in image(s). They appear to reflect different status of the view or model element - Is documentation entered? Is it a master or auxiliary view? Is sub-process/reference added? Is it a referenced element?
Drill down effect for general models	Choose the action when pressing on model elements on a diagram.
Drill down effect for business sub-process	Choose the action when pressing on sub-processes on a diagram.
Drill down effect for process (Process Map)	Choose the action when pressing on processes on a process map diagram.
Drill down effect for events (BPMN)	Choose the action when pressing on (BPMN) events on a diagram.
Drill down effect for diagram overview	Choose the action when pressing on diagram overviews on a diagram.
Drill down effect for action	Choose the action when pressing on (activity diagram) action on a diagram.
Drill down effect for UI elements	Choose the action when pressing on UI elements (e.g. Frame, Panel, Button) on a diagram.
Publisher engine	Choose the engine for publishing. You are advised to use the new engine.
Default diagram	Choose the diagram that first appear when opening the published content. If unspecified, a default page with project information will be presented.
Sort elements in type groups by	Choose the way of sorting elements show in summary or drop down menu in diagram page
Filtered content	Choose the content for not to show in published content

Description of project publisher advanced options

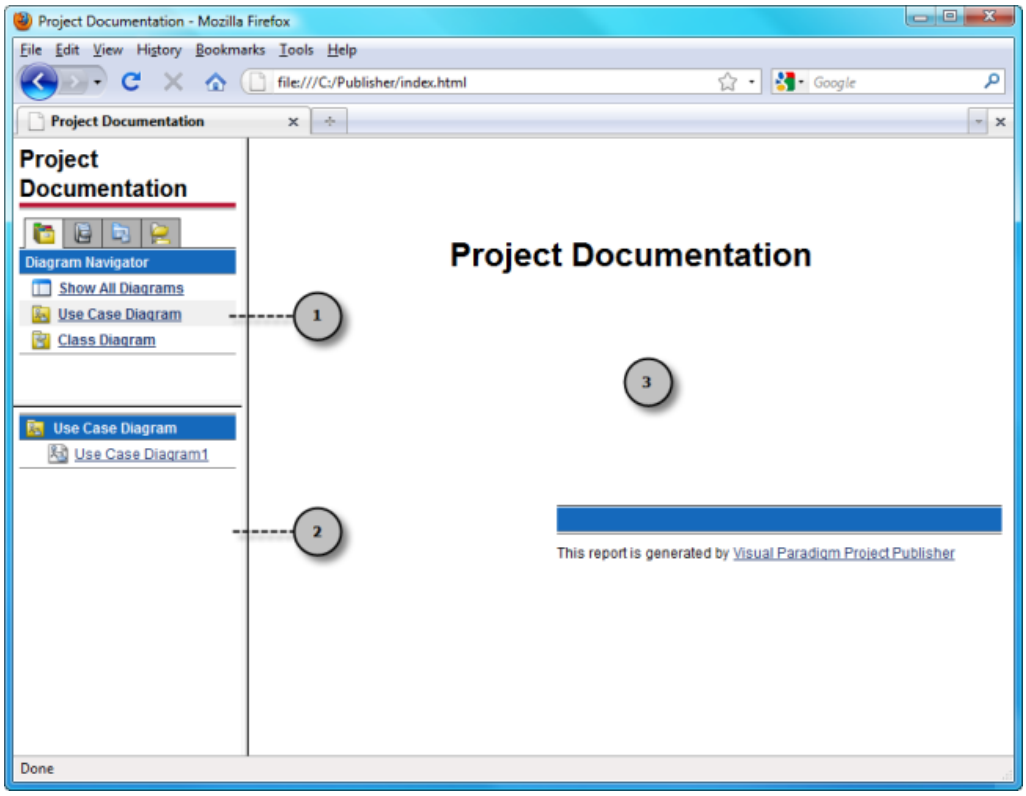
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

About publisher output

Go to the output directory of the published project and open the file 'index.html' with a web browser. The web page is organized in frames, namely the Navigator Pane, Menu Pane and Content Pane.



Published project

No.	Name	Description
1	Navigator pane	It comprises of the Diagram Navigator, Model Navigator and Class Navigator.
2	Menu pane	It shows the sub-menus of the Navigator pane. The contents shown in this pane varies with the link you clicked in the Navigator Pane. For more details about the possible contents please refer to the Navigator Pane section.
3	Content pane	It shows the details of the item (diagram, model or package/class) you clicked in the Menu Pane or Content Pane.

Description of the interface of published Web content

Navigator pane

There are four tabs within the navigator pane - **Diagram Navigator**, **Model Explorer**, **Class Navigator** and **Logical View**. They are responsible for eading the project from different angle.

Diagram Navigator

Diagram Navigator shows the categories of diagrams in the project. You can click on a category to view its diagrams in the Menu Pane, or click Show All Diagrams to view all diagrams.

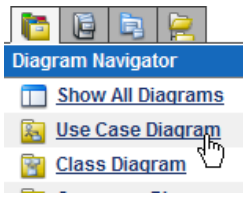
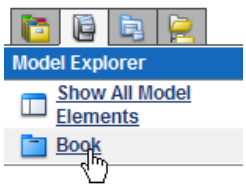


Diagram navigator

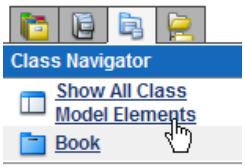
Model Explorer

Shows the package models in the project. You can click on a package to view its child models in the **Menu Pane**, or click **Show All Models** to view all model elements.



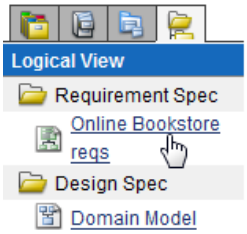
Model Navigator

Class navigator
Shows the Package models in the project. You can click on a package to view its child packages/classes in the **Menu Pane**, or click **Show All Models** to view all packages/classes.



Class navigator

Logical view
Echos the logical view defined in project. You can click on a diagram to open it.



Logical view

Menu pane
To view the details of something (i.e. diagram, shape, model element), click on its link in the **Menu Pane** and its details will be shown in the **Content Pane**.

Menu pane

Diagram Content

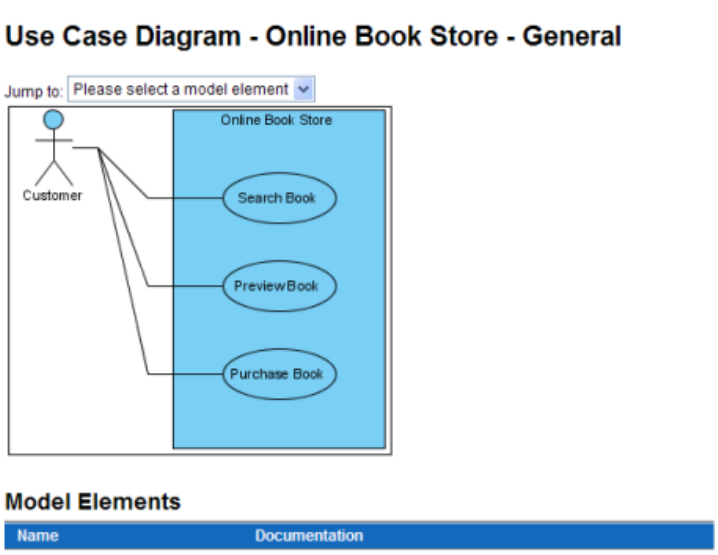
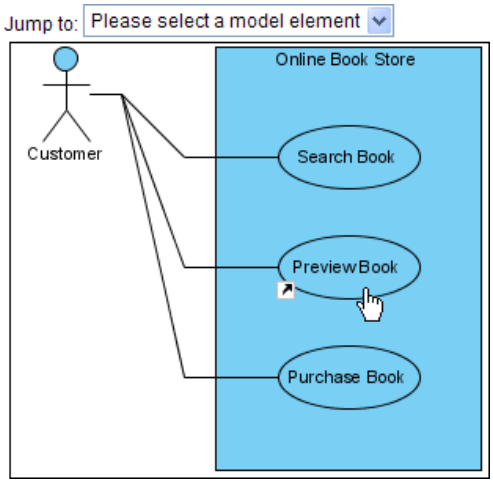


Diagram content

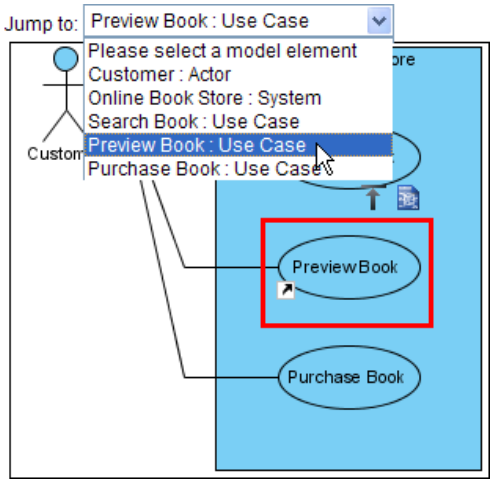
The diagram type, name, description, together with a full size image of the diagram are shown in the Content Pane. The image is mapped to different clickable regions for each shape, so you can click on a shape in the image to view its details.



Shape link to descriptions

Using jump to

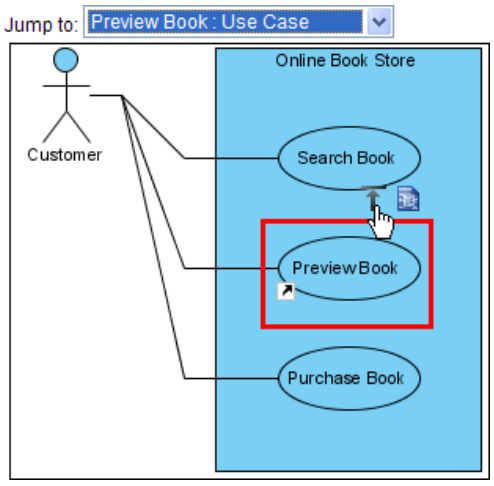
The **Jump to** drop down menu in the diagram content page lists all shapes in the diagram. You can select a shape to jump to. The content page will scroll to the selected shape and the shape will be highlighted by a red border.



Jump to

Besides, there will be two shortcut buttons above the selected shape.

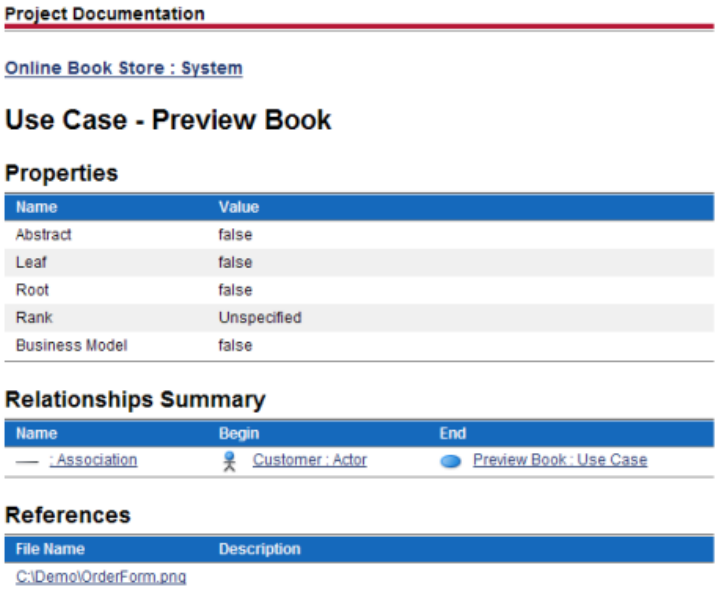
The **Back to top** button brings you to the top of the page, while the **Open specification** button, next to **Back to Top**, brings you to the details page of the shape.



Back to top

Model element content

The type, name and general model properties of a model are shown in the content page.



Project Documentation

Online Book Store : System

Use Case - Preview Book

Properties

Name	Value
Abstract	false
Leaf	false
Root	false
Rank	Unspecified
Business Model	false

Relationships Summary

Name	Begin	End
: Association	Customer : Actor	Preview Book : Use Case

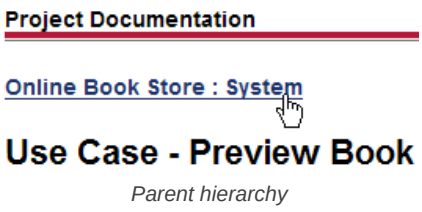
References

File Name	Description
C:\Demo\OrderForm.png	

Model element content

Parent hierarchy

The parent hierarchy is shown as a list of models on top of the page. You can click on a parent in the hierarchy to view its details.



Project Documentation

Online Book Store : System

Use Case - Preview Book

Parent hierarchy

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Report composer

Report composer provides a flexible way for you to design and construct report. After that, you can export the final to a HTML/ PDF/ Word document. This chapter will instruct you how to develop a report and then how to export the report.

Developing a report

This page shows you how to drag and drop selected model elements to create a report. Moreover, various functions are demonstarted, such as inserting an image, a table, a text box, and refreshing the report content and selecting page display option.

Exporting a report

This page shows you how to export your final report. There are three types of report for exporting: HTML, PDF and Word.

Inserting a table of contents

This page shows you how to insert a table of contents to the beginning of a report.

Inserting a revision log

This page shows you how to add and edit a revision log.

Inserting a cover page

This page shows you how to add a cover page to your report.

Working with page section in Report Composer

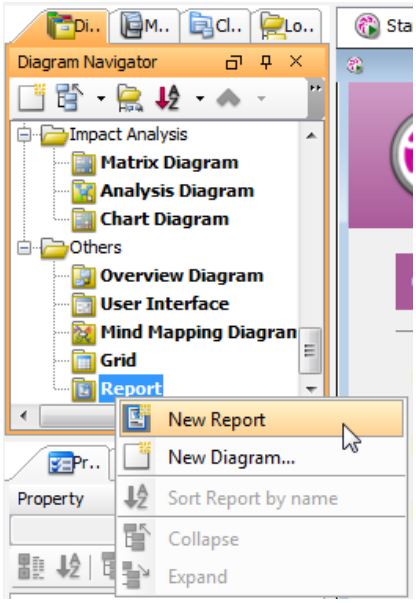
This page shows you how to make pages in different layout by using section .

Developing a report

Report provides a flexible way for you to design and construct report. All you need to do is to select your target model element/diagram, drag the target template(s) from **Property** pane and then drop it/ them on the report (diagram). You cannot type in the report directly, however, Report allows you to add text, image and table to diagram as necessary. After that, you can export the result to a HTML/ PDF/ Word document. Since Report maintains the linkage between project data and report content, you can refresh the report upon project changes. As a result, it saves your time on repeating the steps for creating report. All predefined templates can be edited in order to fit into your specific needs. An efficient way to customize your own templates is also available. Details about how to edit template will be covered in next chapter.

Creating a Report

Right click on **Report** on **Diagram Navigator** and select **New Report** from the pop-up menu.

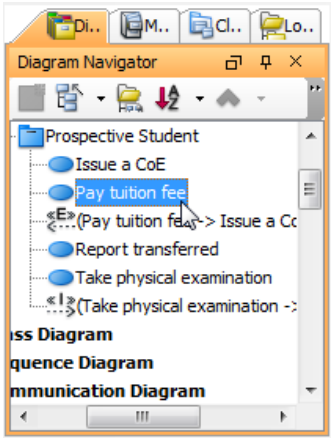


Create a Report

Developing a report in Report

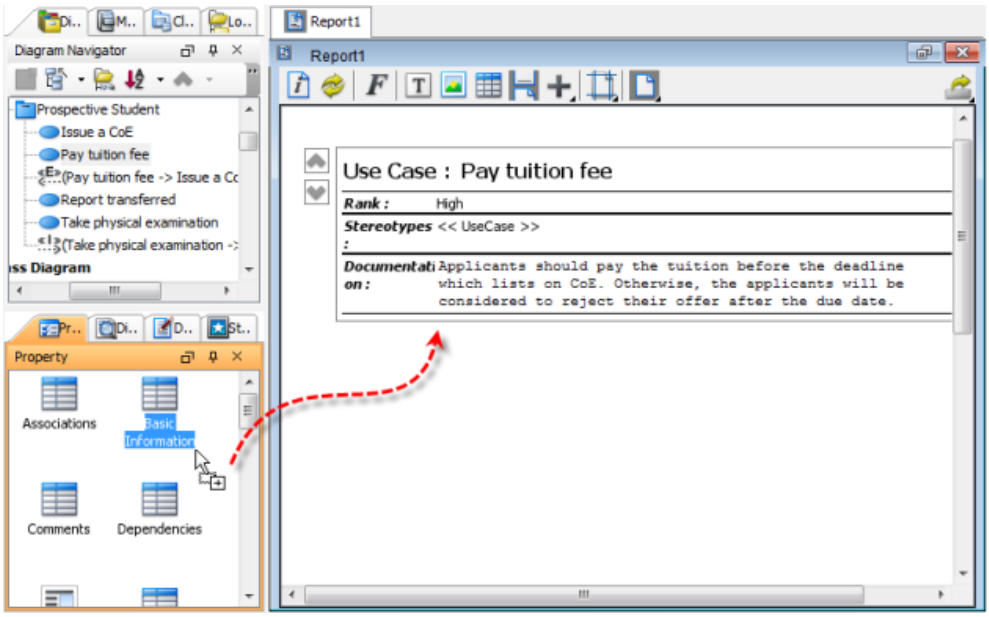
After you have created a Report, you can select your model elements, and drag and drop them on it.

1. Select model element(s) you created previously on **Diagram Navigator**/ **Model Explorer**/ **Class Repository**.



Select a model element

2. Drag your target template(s) from **Property** pane, and drop it/ them on the diagram.



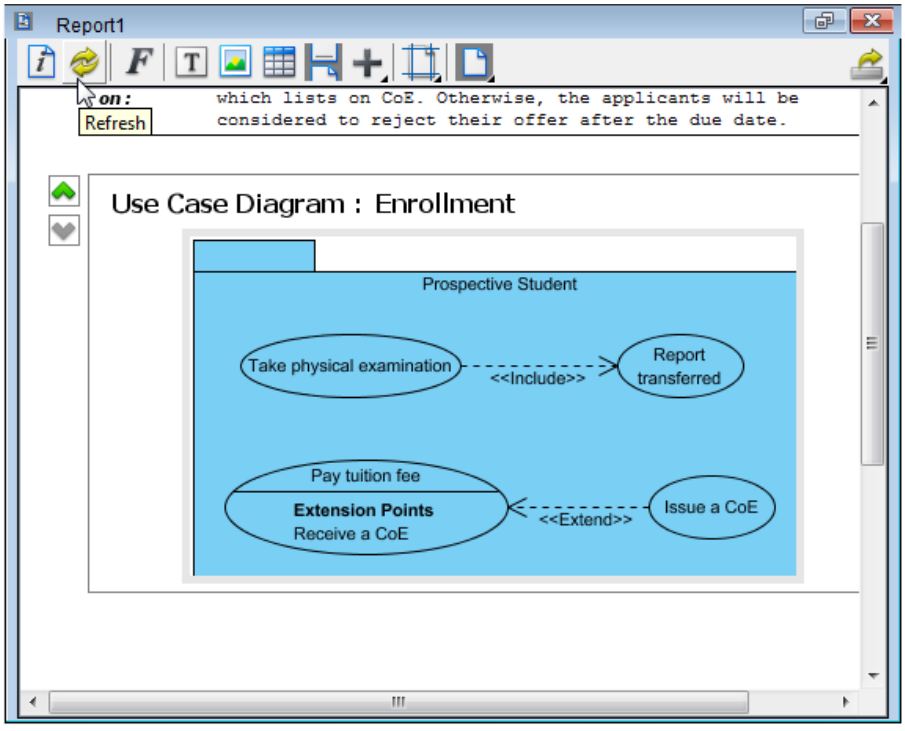
Drag and drop target template on the diagram

NOTE: If the **Property** pane is hidden, you can open it by selecting **View > Panes > Property** from the main menu.

Refreshing report content

Refreshing report content would be time-consuming if you have to repeat the same steps of creating report to update your report content. Agilian accommodates the refresh button on Report’s toolbar to help you refresh your report content shortly. You can refresh the content of reports, including model elements and diagrams, to retrieve the latest updates and changes.

Return to your Report and click **Refresh** button on its toolbar.



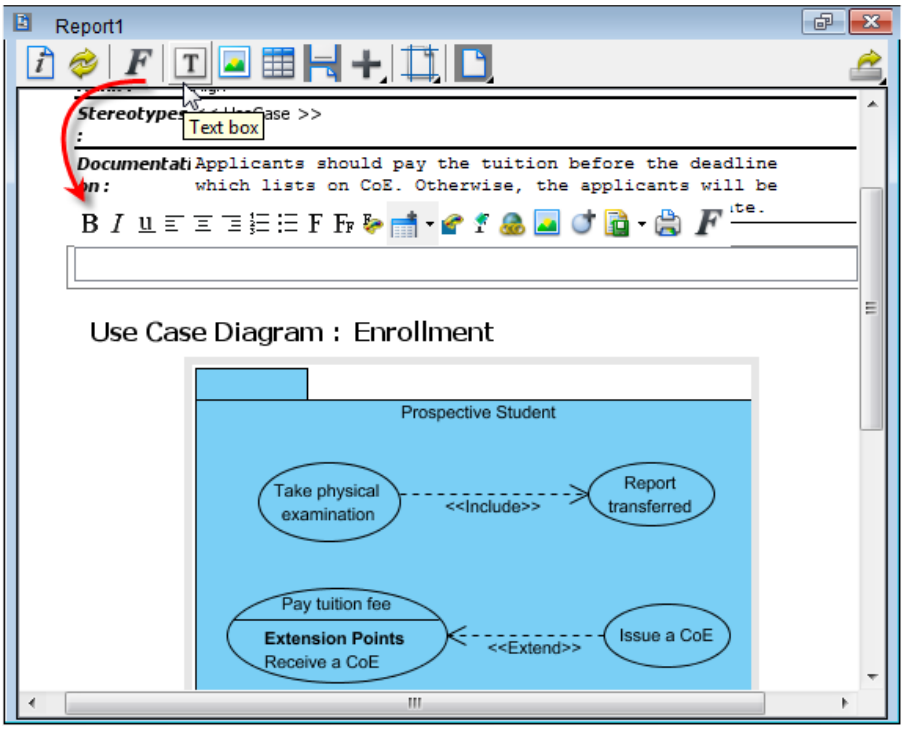
Refresh report content

As a result, the modified element(s) will be updated.

Adding custom text

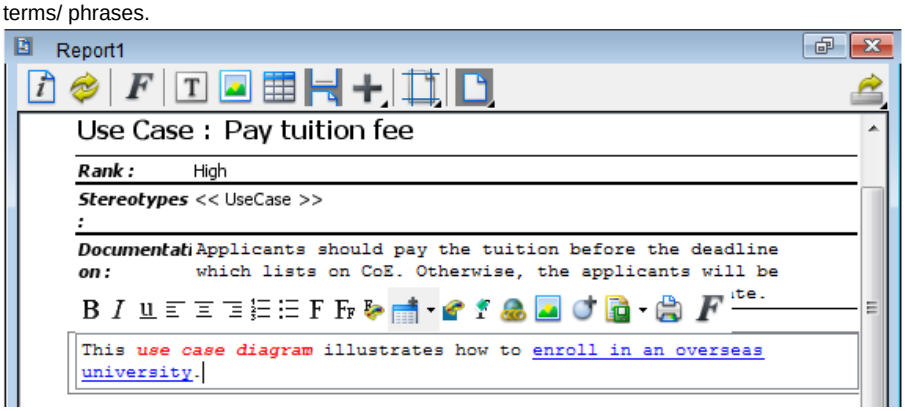
The text box is used for editing data on report. The significant characteristic is, you can display many different types of data with applying RTF within the text boxes.

1. Select the space where you want to insert text beforehand.
2. Click **Text box** button on the Report's toolbar.



Insert a text box

3. Enter text in the text box. You can use the pop-up formatting toolbar to convert your plain text into RTF when you want to emphasize some terms/ phrases.

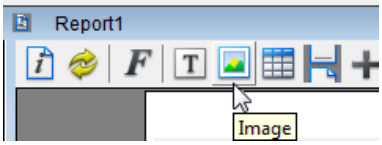


Format text

Adding image

Report supports inserting images. An image can be a logo or picture that is placed on the report. You can not only place pictures on the empty space of report, but also fit them inside table cells. In this sense, you can insert your company logo into any preferred place within the report when you are doing a company report. The advantage is, you can spare no effort in arranging a series of images in report and then resize them.

1. Select the space where you want to insert an image beforehand.
2. Click **Image** button on the Report's toolbar.



Insert an image

3. Select the directory of your target image and then click **Open** button in **Choose image(s)** dialog box.
   As a result, the selected image is inserted.

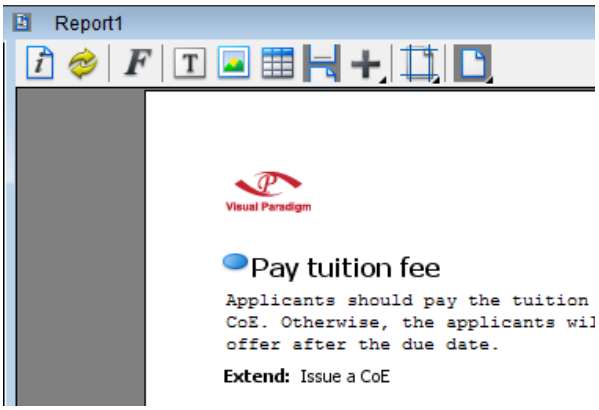
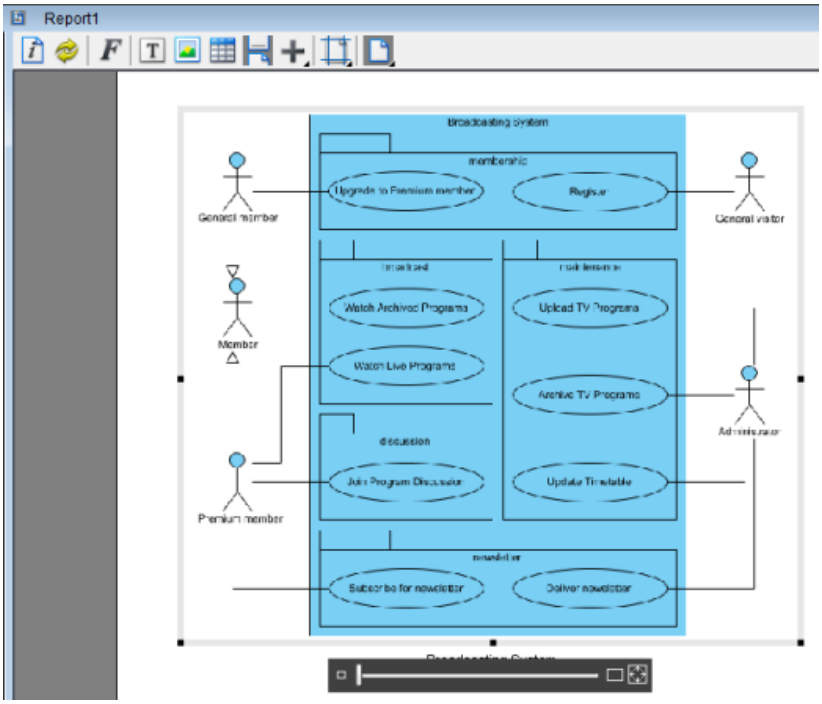


Image is inserted

Editing image

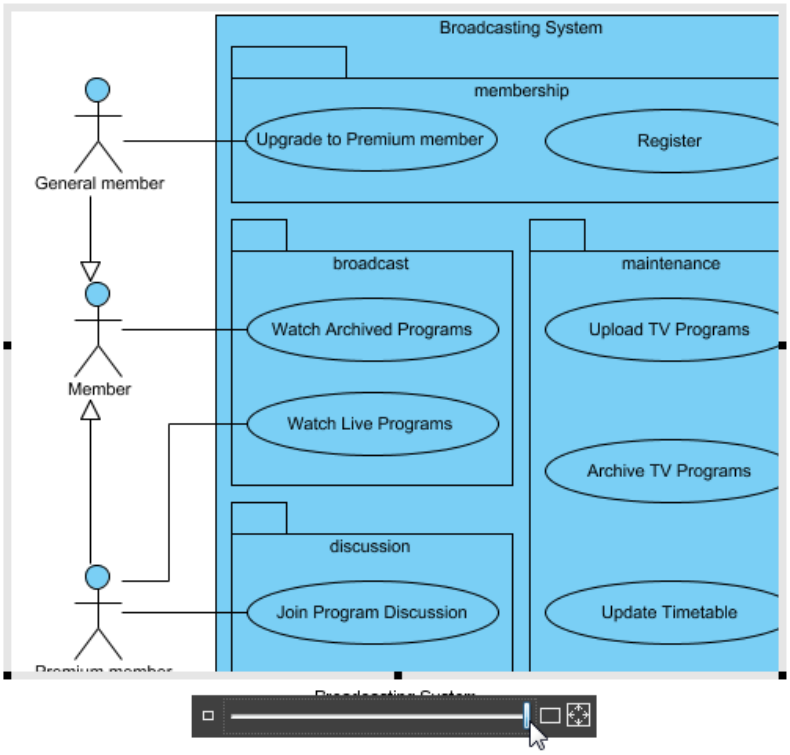
You may find that the image on report template is not clear enough since the image is oversized. This also affects the quality of image in both exported report and the printout. It is recommended that you would edit an image and split a diagram on report template to make it legible. After creating multiple diagram templates (the same diagram template) on report, edit each diagram template to show different parts of diagram. It is regarded as splitting diagram. Finally, you can combine those fragmentary diagrams (separate diagram templates) to form a complete diagram after you print out the report.

1. Drag multiple diagram templates on the diagram in advance.
2. Click each image so that a bar will appear at the bottom of the image. You can edit the image through the bar. Initially, the whole diagram is displayed to fit the placeholder. As you can see, the slider is placed on **Display whole diagram**. Now, the extra large diagram is quite blurred.



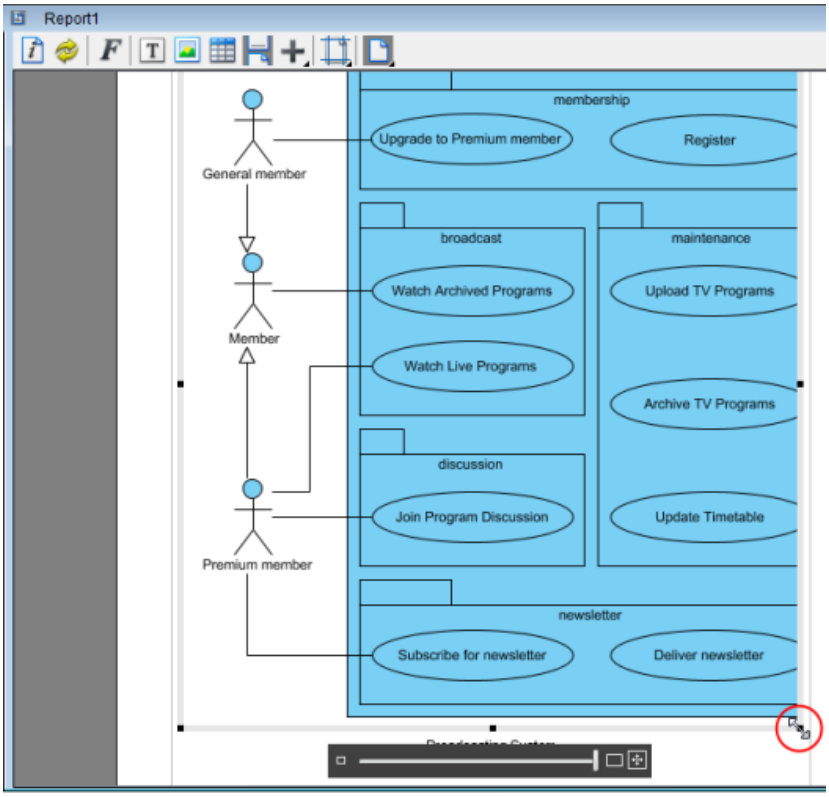
The whole diagram is displayed inside the placeholder

To zoom in the particular part of diagram, drag the slider to **Zoom 100% (Actual Size)**.



Zoom in the image

3. To resize the image, drag the border of placeholder.

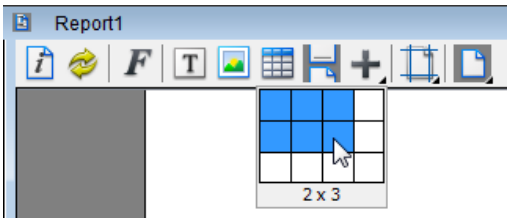


Resize the image

Adding table

Table is one of the important elements while displaying data. It provides a presentable format for data representation. Report enables you to present data with RTF in tabular form by simply pressing **Table** button on the Report’s toolbar.

1. Select the space where you want to insert a table beforehand.
2. Click **Table** button on the Report's toolbar and select the table size, i.e. the number of columns and rows in the table.



Select the table size

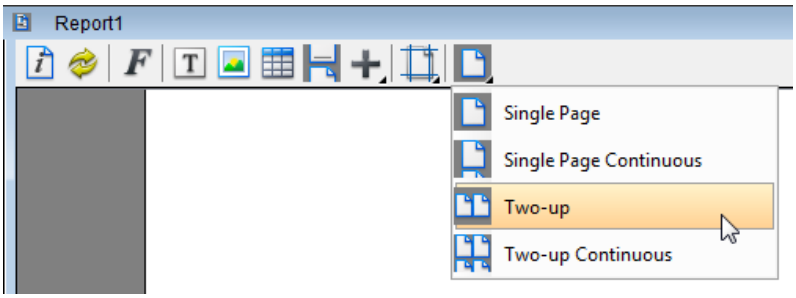
3. Complete the table.

Various page display options

Page display is especially useful when you view the overview of report layout. Agilian supports 4 display options: single page, single page continuous, two-up and two-up continuous.

- **Single Page** displays only one page at a time.
- **Single Page Continuous** displays pages in a consecutive and vertical column.
- **Two-Up** displays two pages side by side simultaneously.
- **Two-Up Continuous** displays pages side by side in two consecutive vertical columns.

Click the **Page Display Option** button to select a page display option from the drop-down menu.



Select an option

Related Resources

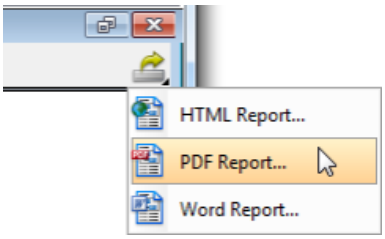
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Exporting a report

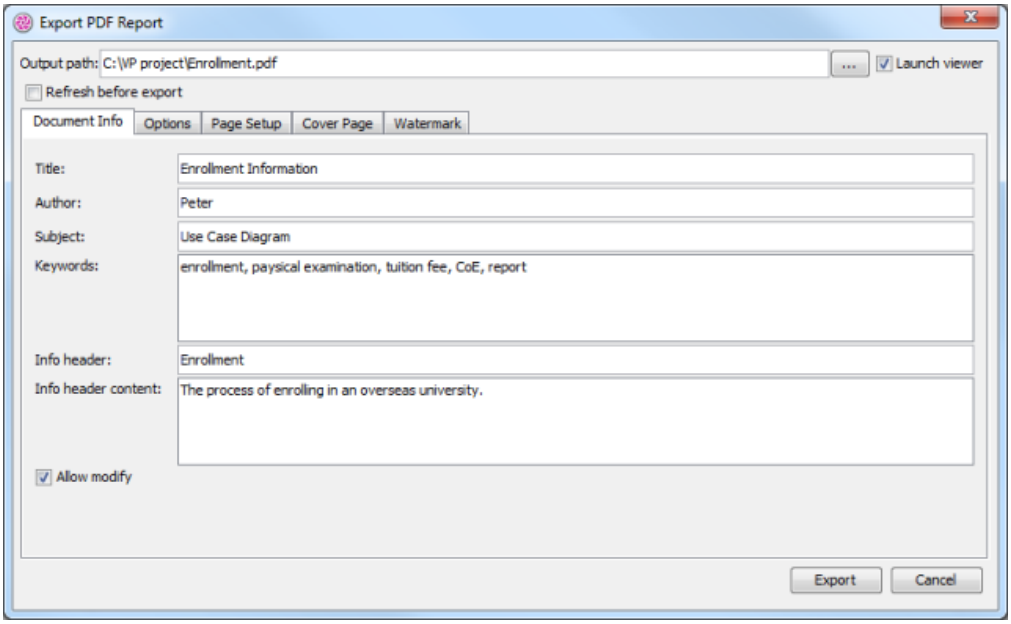
After you have customized your report template on report, you can export it into report. There are three types of report available for exporting: HTML, PDF and Word.

In report, click the **Export** button at the top right corner and select a type of report for exporting.



*Click **Export** button*

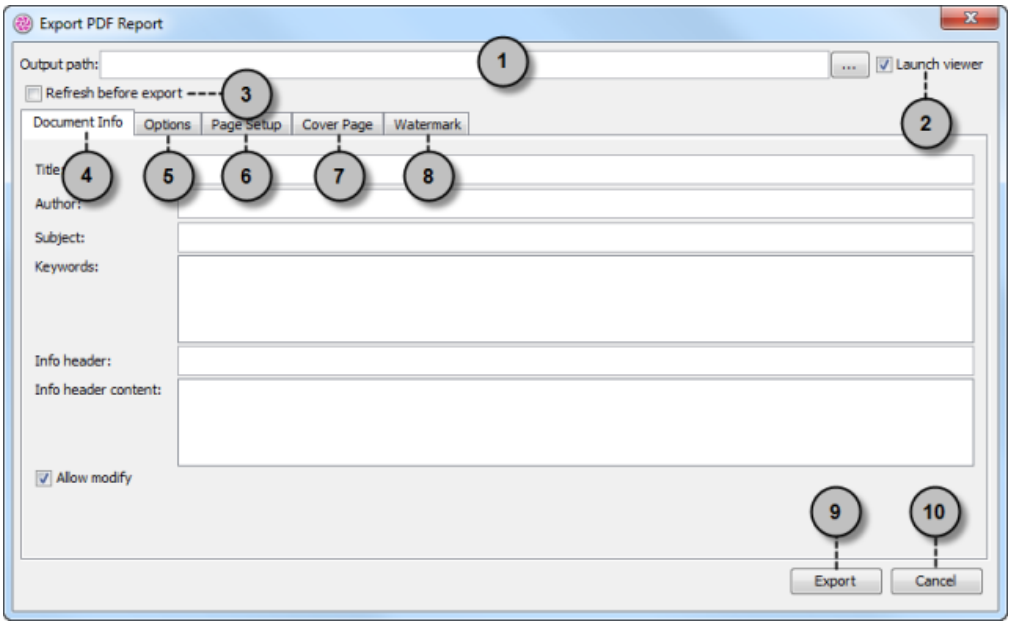
In the pop-up **Export [report type] Report** dialog box, specify output path and document info, and customize page setup, cover page and watermark.



Specify output path

At last, click **Export** button.

The overview of export report dialog box



The overview of Export PDF Report dialog box

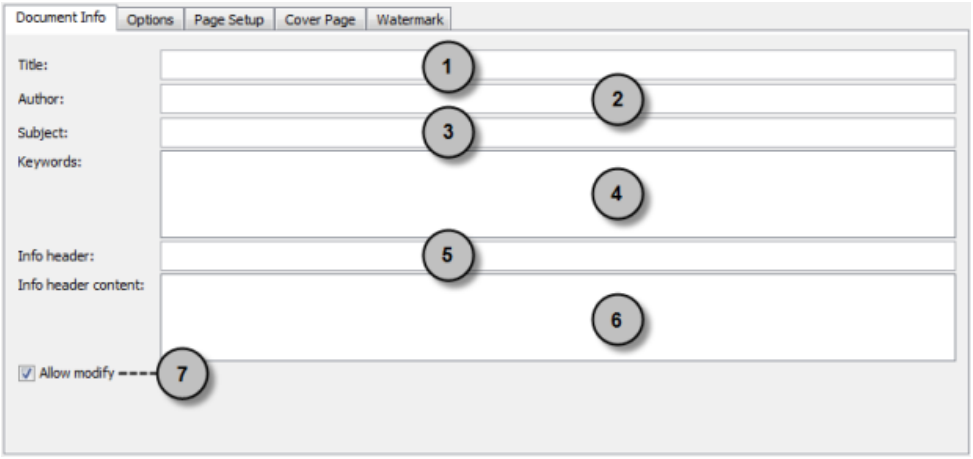
No.	Name	Description

1	Output path	The output path of report to be generated.
2	Launch viewer	Check to open the report automatically after generation.
3	Refresh before export	Before proceed exporting, refresh the report content.
4	Document info	To define document information.
5	Options	To determine how data is to be printed in report by setting some of the configurable options.
6	Page Setup	To customize the layout of report.
7	Cover Page	To customize the first page of report.
8	Watermark	To customize the watermark on report.
9	Export	Confirm and export the report.
10	Cancel	Close the export report dialog box without exporting.

Description of export report dialog box

NOTE: An additional **Content** tab is attached to **Export Word Report** dialog box.

The overview of Document Info

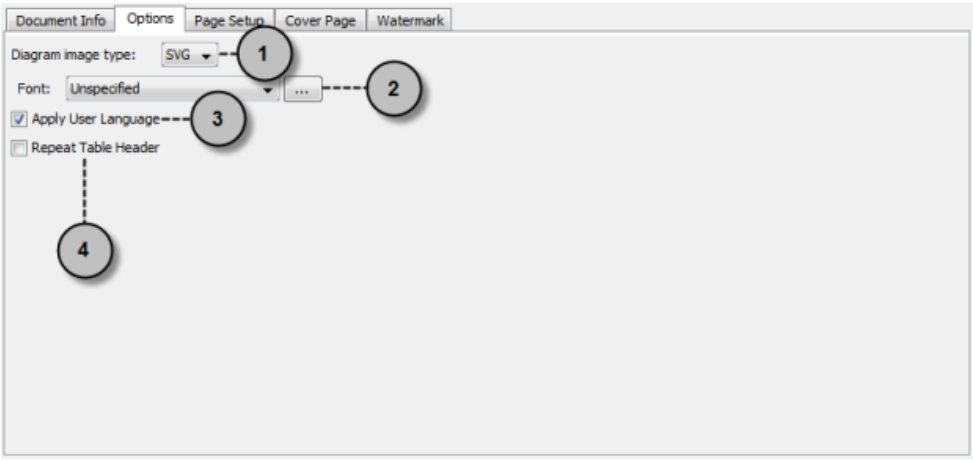


The overview of Document Info tab

No.	Option	Description
1	Title	The title of report. This option is only available for exporting PDF report.
2	Author	The author of the report.
3	Subject	The subject of the report. This option is only available for exporting PDF and Word report.
4	Keywords	The keywords of the report.
5	Info header	The info header of the report. This option is only available for exporting PDF report.
6	Info header content	The info header content of the report. This option is only available for exporting PDF report.
7	Allow modify	Select to allow modification on the report. This option is only available for exporting PDF report.

Description of Document Info tab

The overview of Options Setup

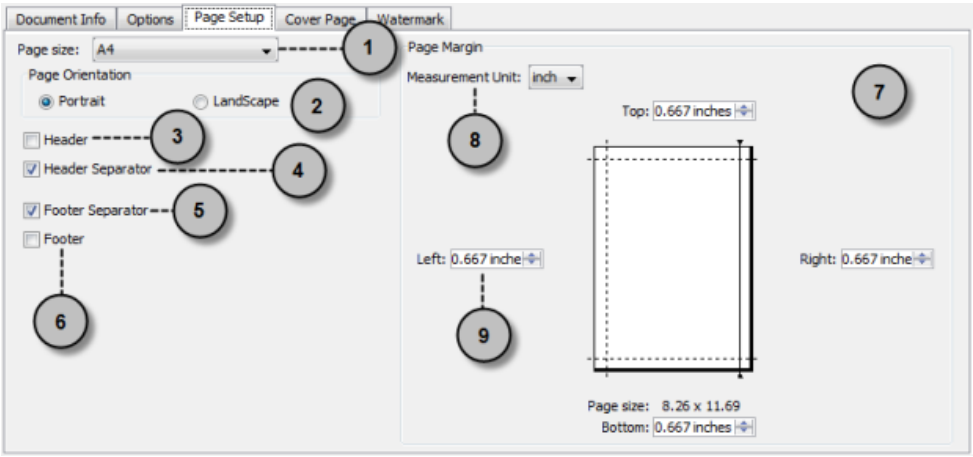


The overview of Page Setup tab

No.	Option	Description
1	Diagram image type	Select the type of image format for image that appear in the exported report.
2	Font	Control the font of report text.
3	Apply User Language	By default, report content will be printed in English. By checking this option, it will follow the language setting chosen in global options.
4	Repeat Table Header	By checking this option, table header would be repeatedly printed when the table span multiple pages.

Description of Page Setup tab

The overview of Page Setup



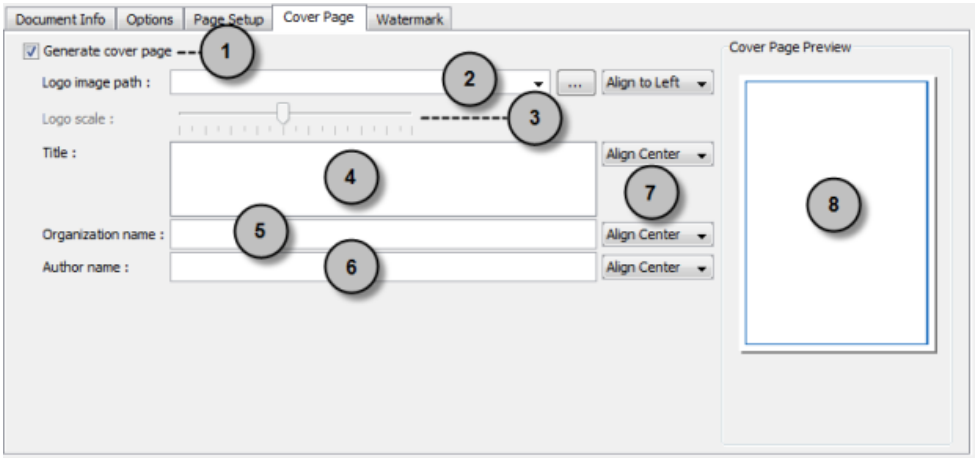
The overview of Page Setup tab

No.	Option	Description
1	Page size	To select the paper size of the exported report.
2	Page Orientation	This option is used to select the orientation of the report (portrait/ landscape). This option is only available to PDF and Word report.
3	Header	Check this option to insert header to the exported report. This option is only available to PDF and Word report.
4	Header Separator	Check this option to insert header separator to the exported report. This option is only available to PDF and Word report.
5	Footer Separator	Check this option to insert footer separator to the exported report. This option is only available to PDF and Word report.
6	Footer	Check this option to insert footer to the exported report. This option is only available to PDF and Word report.
7	Page Margin	To specify the page margins of the report: top, left, right and bottom. This option is only available to PDF and Word report.

8	Measurement Unit	To choose the measurement unit of page margin of the report: inch and cm. This option is only available to PDF and Word report.
9	Margin (Left/ Top/ Right/ Bottom)	Specify the width of spaces between the content and the page border.

Description of Page Setup tab

The overview of Cover Page

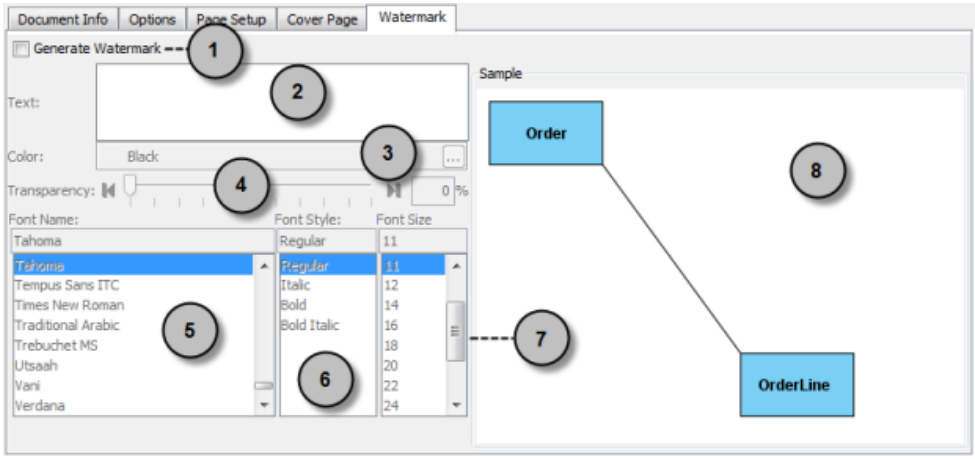


The overview of Cover Page tab

No.	Option	Description
1	Generate cover page	Check this option to generate a cover page to the report.
2	Logo image path	Insert an image to the cover page. You can specify the image's directory or select the directory by clicking the ... button.
3	Logo scale	Resize the inserted image by adjusting the slider.
4	Title	Specify the title of your report on cover page.
5	Organization name	Specify the organization name of your report on cover page.
6	Author name	Specify the author name on cover page.
7	Alignment	Control the position of content, whether to appear at the left, center of right hand side of the page.
8	Cover Page Preview	You can preview your cover page here.

Description of Cover Page tab

The overview of Watermark



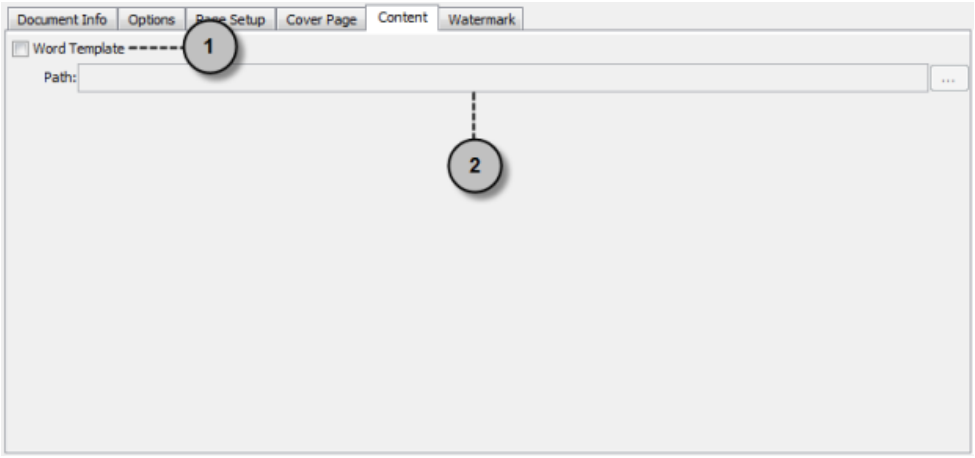
The overview of Watermark tab

No.	Option	Description
1	Generate Watermark	Check this option to generate watermark on all diagrams of report.
2	Text	Specify the text will be used for watermark.

3	Color	Specify the color of text will be used for watermark by clicking the **...** button.
4	Transparency	To change the background transparency for watermark, move the **Transparency** slider or specify the percentage of transparency directly.
5	Font Name	Select the font name for watermark.
6	Font Style	Select the font style for watermark.
7	Font Size	Select the font size for watermark.
8	Sample	Preview watermark here.

Description of Watermark tab

The overview of Content (Only for Word report)



The overview of Content tab

No.	Option	Description
1	Word Template	Check this option to select a Word template for Word report.
2	Path	Specify the directory of word template by clicking the **...** button.

Description of Content tab

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Inserting a table of contents

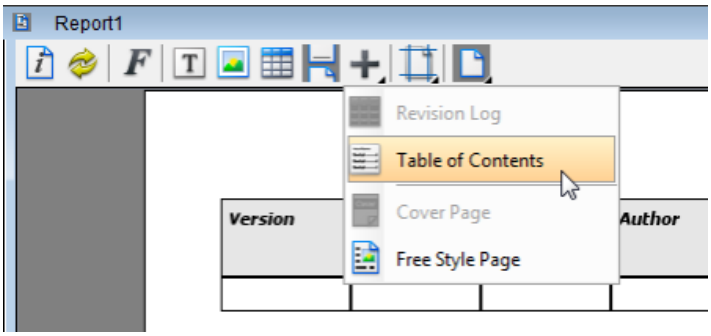
A table of contents, often be abbreviated as TOC, is a list of key parts of a document. It is often constructed by headers or key titles in a document, to present readers with and outline of the whole document.

Report composer lets you insert a table of contents into a report. A table of contents can be formed not only from traditional headings styles like Heading 1 (VP) and Heading 2 (VP), but from any kind of style, even from user-defined styles. In generated report, table of contents provide readers with access to different part of the report content.

Inserting a table of contents

Through toolbar

By inserting a table of contents through the toolbar, the table of contents will be put right under the currently selected content, if any, or at the end of report if there is no active selection made in the report.

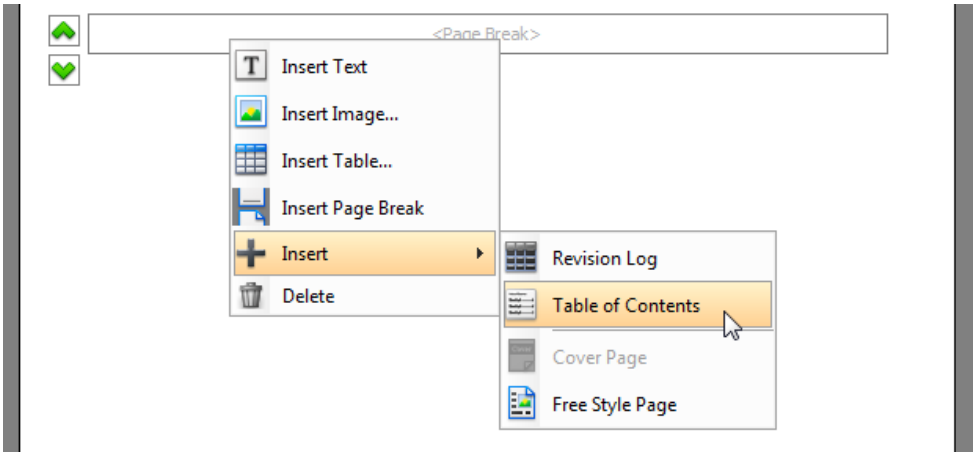


Creating a table of contents below the Revision Log

To insert a table of contents through toolbar, click on the plus button and select Table of Contents from the toolbar. Configure the table of contents and click **OK** to create it. About the configuration of table of contents, please read the next section *Configuring table of contents*.

Through popup menu

You can insert a table of contents to a specific location in report by right clicking on a part of contents (e.g. the "Children" table of a diagram) and selecting **Insert > Table of Contents** from the popup menu. This will add the table of contents below the clicked content.

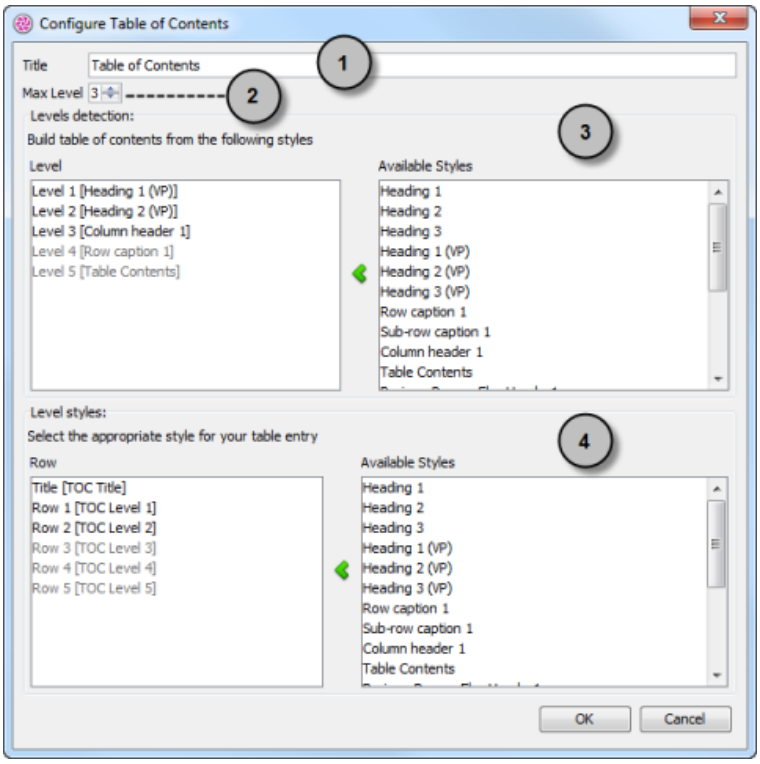


Creating a table of contents below a page break

If you right click on the background (i.e. whitespace) within a report and select **Insert > Table of Contents**, this will add the table of contents to the end of report.

Configuring table of contents

To edit a table of contents for chaging its title, maximum number of level, level detection or styles, you need to configure it. Right click on the table of contents and select **Configure Table of Contents...** from the popup menu.



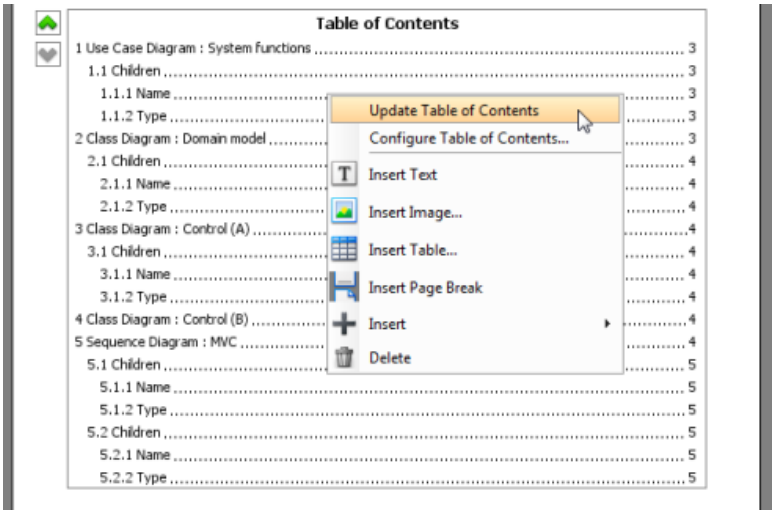
The overview of Configure Table of Contents window

Option	Description
1	The title of the table of contents. This is the text that appear above the table of contents in report.
2	Determines the depth of the table of contents.
3	Specify the style to check for each level. If you want level 1 shows all content with Heading 1 as style, select Level 1 on the left hand side, Heading 1 on right hand side, and click < to match them up.
4	Specify the appearance of text in table of contents. You can apply different styles for different rows (levels).

Description of Configure Table of Contents window

Updating table of contents

To update a table of contents to make it reflect the structure of the latest report content, right click on the table of contents and select Update Table of Contents from the popup menu.



Updating a table of contents

NOTE: Refreshing a report would not update a table of contents. The only way to update a table of contents is to update via its popup menu **Update Table of Contents**.

Deleting table of contents

To remove a table of contents from a document, either select it in the report and press the **Delete** key, or right click on it and select **Delete** from the popup menu.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

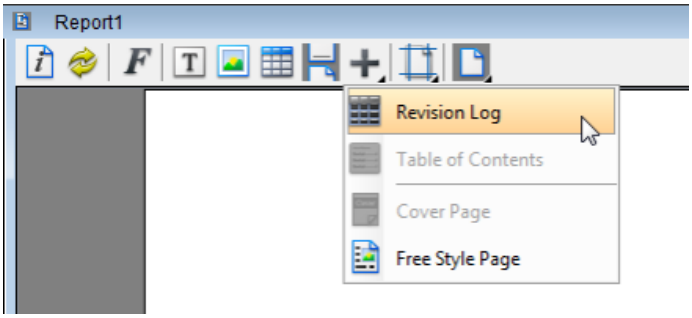
Inserting a revision log

When your team attempt to or has made signaficant changes on a report, you may want to record the version of report, the date/time when the change took place, the person who made the change and other necessarily remarks regarding the changes. Revision log is a piece of content you can add to a report to record all these information. With a revision log, you fill in the revision detail, as well as to add/remove columns to suit the requirement of your team.

Inserting a revision log

Through toolbar

By inserting a revision log through the toolbar, the revision log will be put right under the currently selected content, if any, or at the end of report if there is no active selection made in the report.

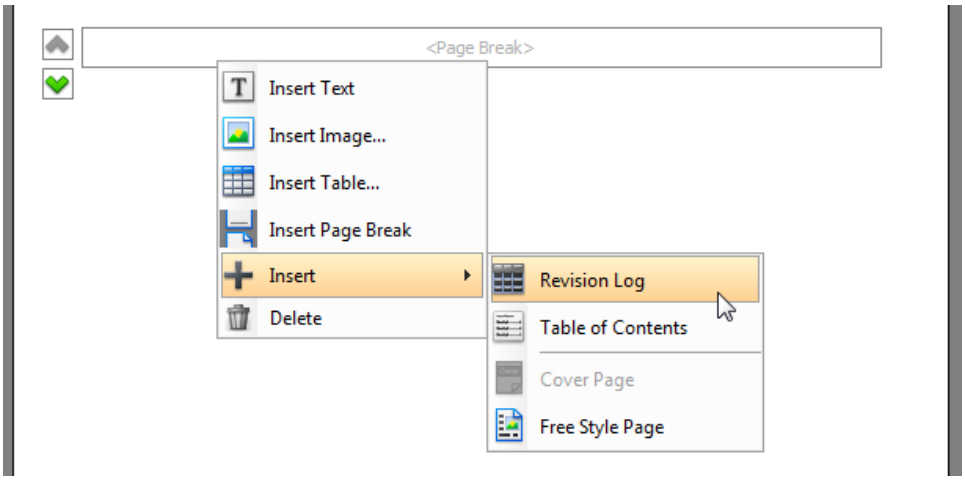


Creating a revision log through the toolbar

To insert a revision log through toolbar, click on the plus button and select **Revision Log** from the toolbar.

Through popup menu

You can insert a revision log to a specific location in report by right clicking on a part of contents (e.g. the "Children" table of a diagram) and selecting **Insert > Revision Log** from the popup menu. This will add the revision log below the clicked content.

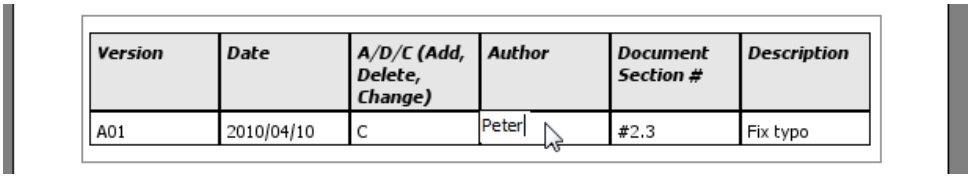


Creating a revision log through a popup menu

If you right click on the background (i.e. whitespace) within a report and select Insert > **Revision Log**, this will add the revision log to the end of report.

Editing a revision log

To enter a revision, simply double click on the cells and enter the values one by one.

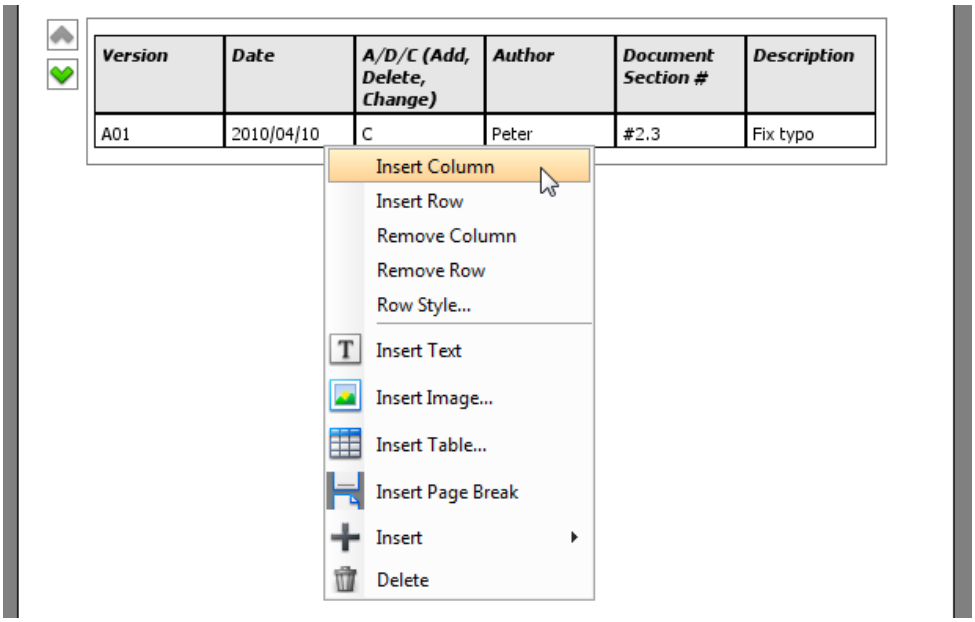


Version	Date	A/D/C (Add, Delete, Change)	Author	Document Section #	Description
A01	2010/04/10	C	Peter	#2.3	Fix typo

Editing a log

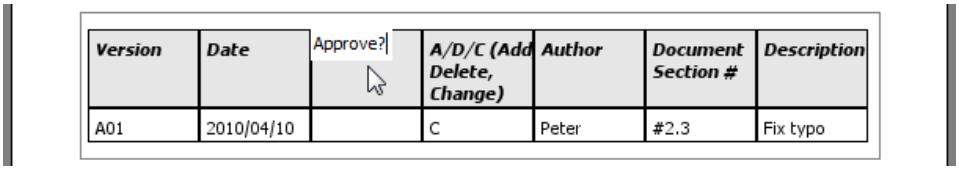
Inserting a column

If you want to record an extra kind of content for revisions, you can add a column to the revision log table. Right click at the column where you want to insert a column after it, select **Insert Column** from the popup menu.



Inserting a column

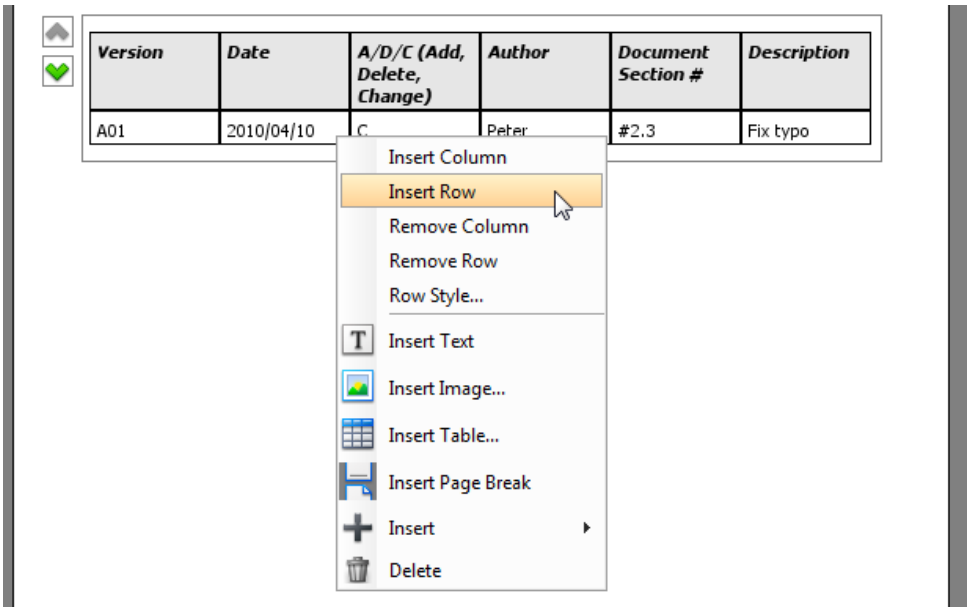
After that, double click on the header and enter the column header.



Entering the column header

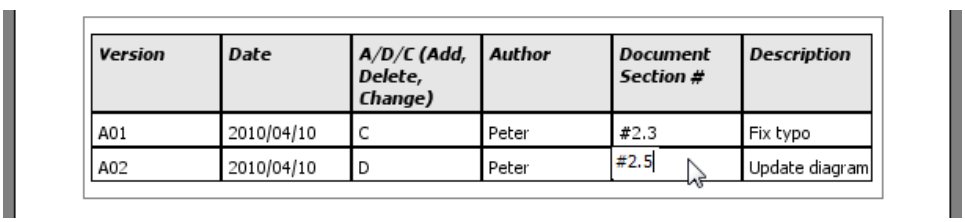
Inserting a row

If you want to record a new revision, you need to insert a row. Right click at the row where you want to insert a row after it, select Insert Row from the popup menu.



Inserting a new row

After that, fill in the cells to record the new revision.



Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

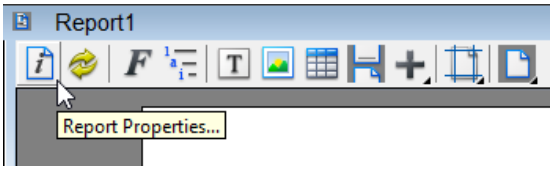
Inserting a cover page

A cover page is a page that can be added at the beginning of a report. Whether or not to add such page is up to the writer. There are two kinds of cover page you can add into a report. The first one is to print the cover page in a program defined way. This approach require you fill in some of the background information like the report title, organization name and author name, etc. The second kind of cover page is fully designed by you, the writer. It is called a free style cover page.

Build-in cover page

When you export a report from report composer to a report file, a default cover page would be generated for you. You should configure the cover page to make your company logo, report title, organization name and author name print on it. Otherwise, only a blank page would be printed. To configure the default cover page:

1. Click on **Report** **Properties...** in the toolbar.



Open the Report Properties

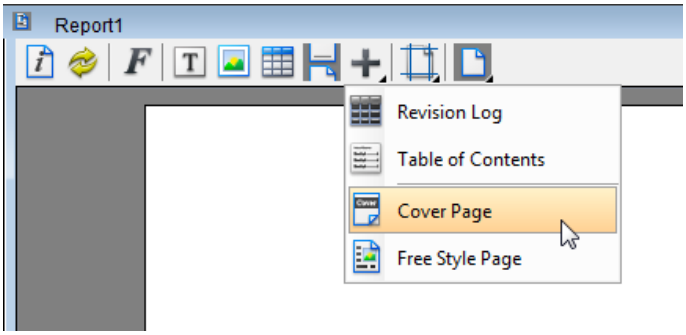
2. Open the **Cover Page** tab.
3. Configure the cover page by specifying file path of logo image, title, orgainzation name, author name. You can preview the page at the right hand side of the **Report Properties** window.
4. Click **OK**.

> **NOTE:** You will only see the cover page in the generated report.

Free style cover page

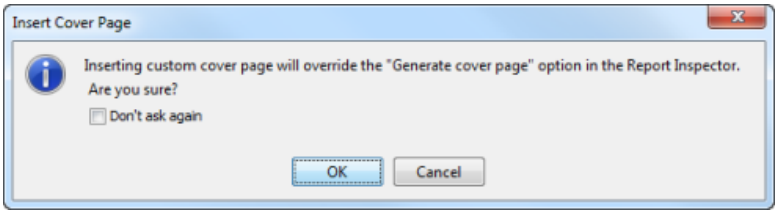
Free style cover page provides you with a page that appear at the beginning of a report for you to design the page. You can add any text and image freely on the cover page, and position them in any position you like, within the cover page. To insert a free style cover page:

1. Click on the plus button and select **Cover Page** from the toolbar.



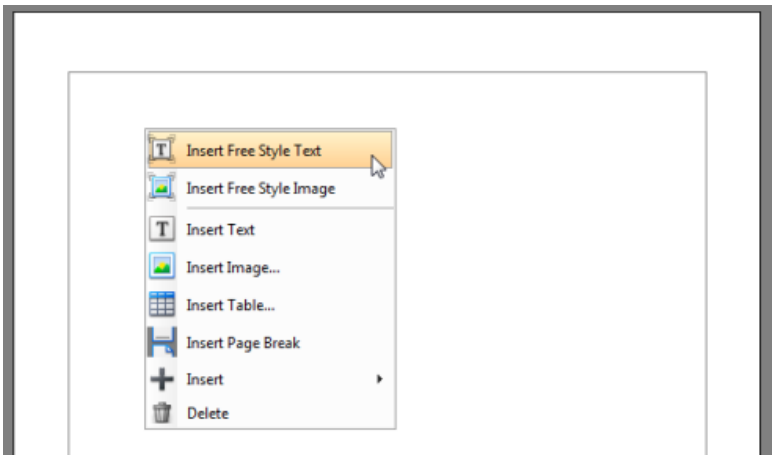
Insert a free style cover page

2. When you insert a free style page the first time, you are prompted to override the generate cover page option. By default, the build in cover page (as described in the above section) would be chosen as cover page. When you try to insert a free style cover page, the build in cover page would be ignored. This option is to ask for your confirmation for ignoring the build in cover page. Click **OK** to confirm.



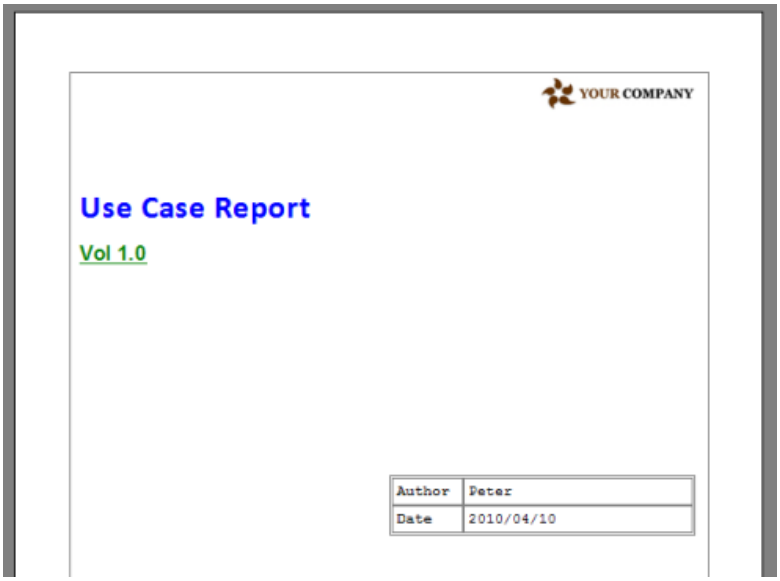
To overwrite the default cover page option

3. You will see an empty cover page is added to the beginning of the report. Note that the page MUST be added to the beginning of report. You cannot control its location. If you want to add a free style page in the middle of the report, insert a **Free Style** page instead.
You can start editing the page by inserting text and image. To insert text or image into the page, right click on the background of cover page and select **Insert Free Style Text** or **Insert Free Style Image** from the popup menu.



Insert a free style text in cover page

4. Fill in the text or select the image file to insert to the page. Repeat step 3 and 4 to complete the page.



Cover page designed

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Working with page section in Report Composer

A section is a number of continued pages that apply the same set of page properties. These properties include page size, page orientation, page margin, visibility of header/footer, content of header/footer, etc.

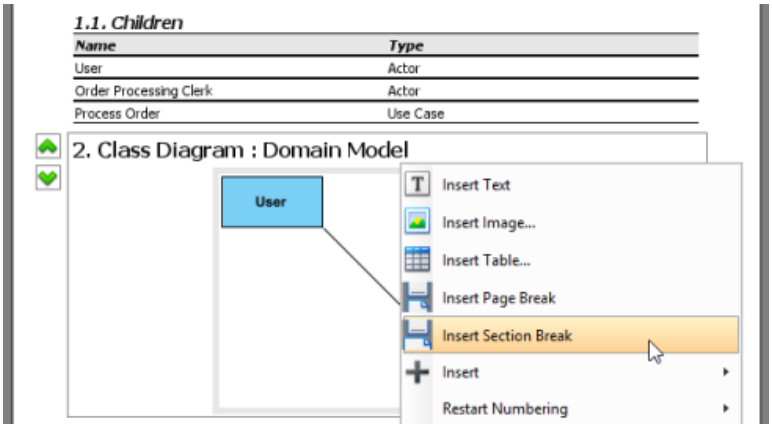
Because section allows you to define different layouts for different pages, you can make pages that consist of wide tables show in landscape, with the other pages remain in portrait. You can also add content-specific header and footer.

Inserting a section break

By inserting a section break, pages that appear after the break will apply the same set of page properties as defined in the break. To insert a section break:

In the middle of a report

1. Right click at the content that you want to add a section break **BEFORE** it.
2. Select **Insert Section Break** from the popup menu.

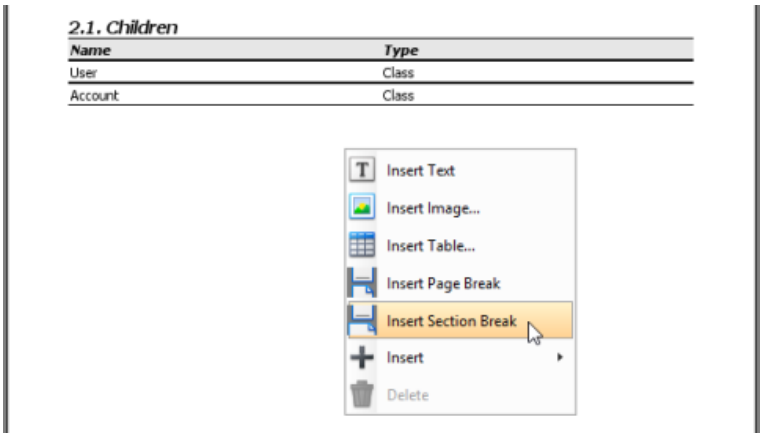


Inserting section break

A section page is inserted, which moves the chosen content to a new page. Read the next section to see how to configure the properties of a section break.

At the end of the report

1. Right click at the empty region after the last content block in the report.
2. Select **Insert Section Break** from the popup menu.



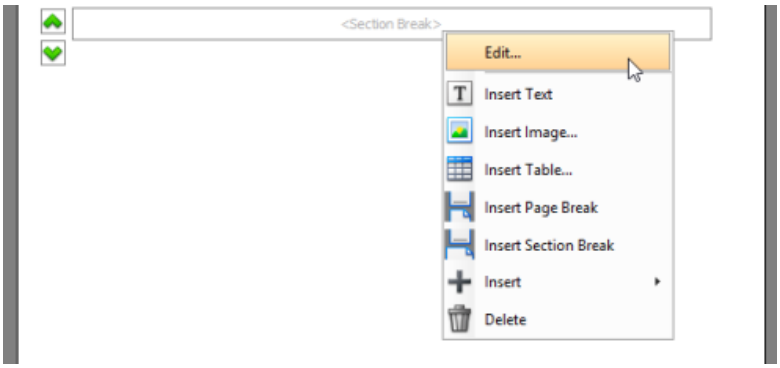
Inserting section break to the end of the report

A section page is inserted to the end of the report. Read the next section to see how to configure the properties of a section break.

Configuring section break

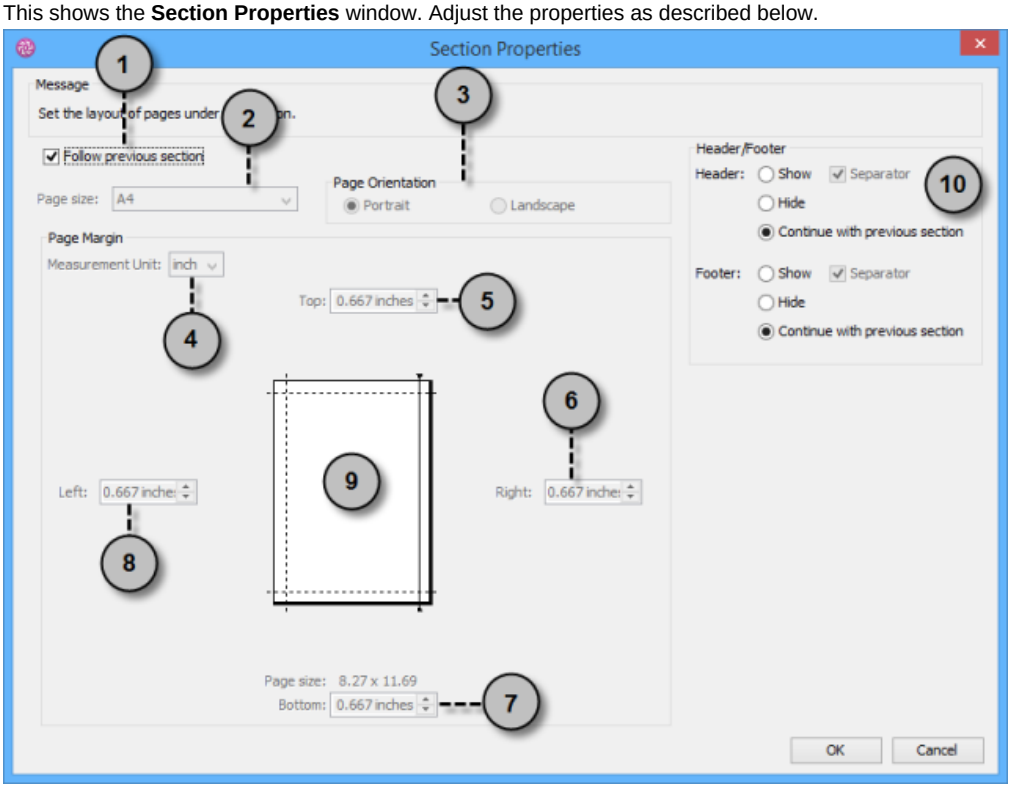
The layout of pages within a section are controlled by the setting configured in the section break. To configure a section break:

1. Right click on the desired section break.
2. Select **Edit...** from the popup menu.



Edit a section break

3. This shows the **Section Properties** window. Adjust the properties as described below.



Section Properties window

No.	Name	Description
1	Follow previous section	When checked, section properties will follow that defined in previous section. For the first section, it follows the properties set to the whole report.
2	Page size	Determine the dimension of page.
3	Page Orientation	Orientation of page, either in portrait or landscape.
4	Measurement unit	Unit of margin.
5	Top margin	Determine the empty space at the top of a page. You can edit the margins size by entering the sizes into the text fields. Alternatively, click the spinner buttons to increase/ decrease the margin sizes.
6	Right margin	Determine the empty space on the right hand side of a page. You can edit the margins size by entering the sizes into the text fields. Alternatively, click the spinner buttons to increase/ decrease the margin sizes.
7	Bottom margin	Determine the empty space at the bottom of a page. You can edit the margins size by entering the sizes into the text fields. Alternatively, click the spinner buttons to increase/ decrease the margin sizes.
8	Left margin	Determine the empty space on the left hand side of a page. You can edit the margins size by entering the sizes into the text fields. Alternatively, click the spinner buttons to increase/ decrease the margin sizes.
9	Preview of page	Preview the effect of adjusting margin.

10	Header/Footer	**Show** - Show the header/footer in report **Hide** - Hide the header/footer in report **Separator** - When checked, a line will be shown between header/footer and the main content. **Continue with previous section** - Following the previous section means to have the visibility of header, footer and separator follow that defined in the previous section. If you have added page number to header/footer, the numbering will continue with the previous section, too. If not to follow the previous section, the numbering will reset to 1.

Description of Section Properties window

4. Click **OK** to confirm.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Report composer template

Other than using the build-in templates in developing a report, you can also define your own templates, to construct report in your own way.

Introduction

Gives you an introduction to template development.

Creating a template

Shows you the steps of creating a template from an existing one through duplication.

Beginning of template

The template elements to use at the beginning of template.

Reading model - Looping

The template elements to use for looping through a type of model element through its parent.

Reading model - Conditional checking

The template elements to use for performing checking, i.e. to do something only when a condition is met.

Reading model - Model/diagram element

The template elements to use to query diagram/model element from a model element.

Text and break

The template elements to use for printing text or break to report. Text can be a custom text or to extract from a property of model element.

Table

The template elements to use at the beginning of template.

Image

The template elements to use for producing diagram image or icon of shape types.

Miscellaneous

The template elements to use to build a table.

Introduction

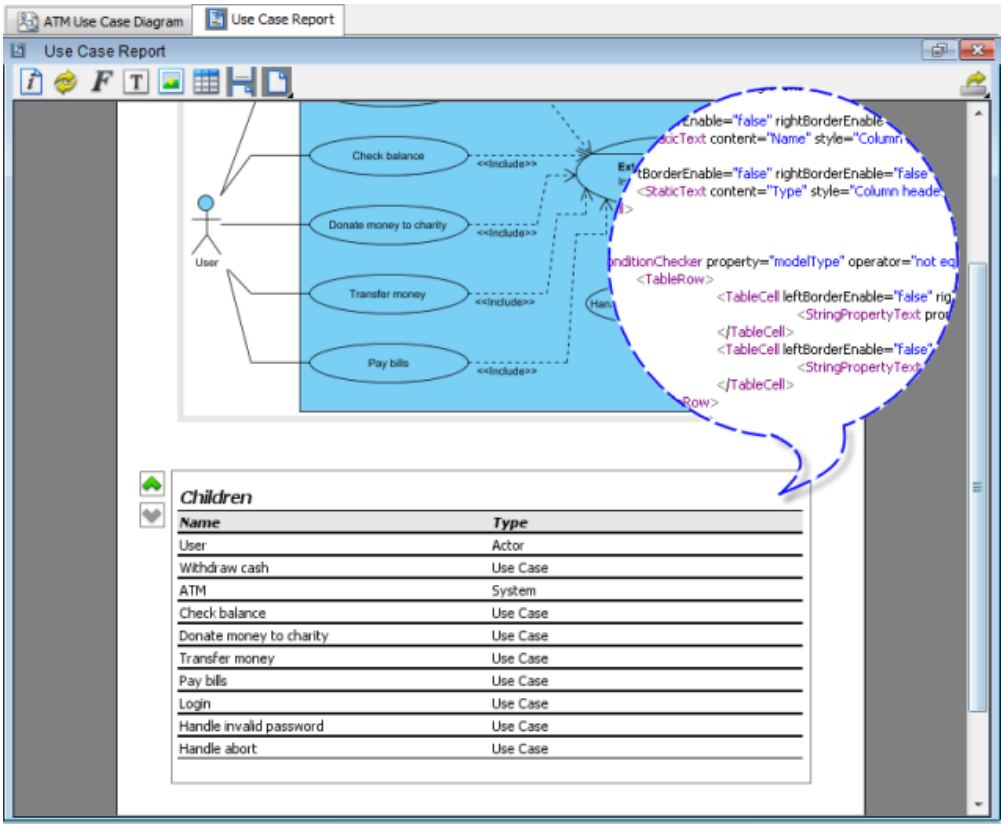
Report diagram provides you with a dynamic and efficient report design experience by letting you design report through simple drag-and-drop. You just need to select a piece of model data, like a use case or a sequence diagram, then drag out a build-in template and drop it onto the report diagram to create content.

All the build-in templates are opened for editing. To make report design more adoptable to your company’s needs, you can edit a template or to design your own templates and re-use it report-by-report. For example, design a class specification by printing out the documentation of classes that contains "Controller" in their names.

This section is divided into two main parts. The first part is going to talk about the steps for creating a template in report diagram. The second part comes with a detailed specification of template constructs you can use in constructing a template.

An element template

An element template is a XML-based document which defines the way to construct a part of a report. You can form a report by dragging and dropping appropriate element templates onto report diagram. You can program a template by right clicking on it in property pane and selecting Edit in the popup menu.



An element template is an XML based document

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Creating a template

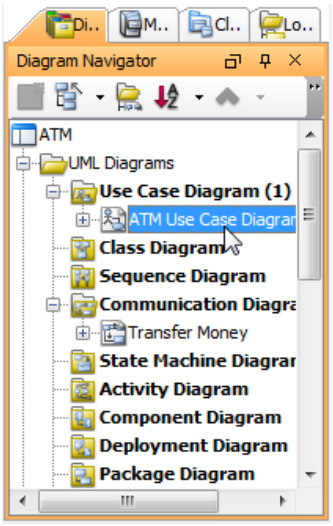
Although report diagram comes with a complete set of build-in element templates, you may still want to customize, or even to create a template for maximizing the efficiency of report design. You can accomplish this by creating and programming your own element template.

Before you create a template, please pay attention to the following points:

- Element templates are product installation based. All the projects opened under the same VP Suite installation share the same set of element templates.
- If you need to open a project file that used an edited or new template in another environment, make sure that environment has the element templates ready. Otherwise, the content may lost upon refreshing, or even make it unable to export report.

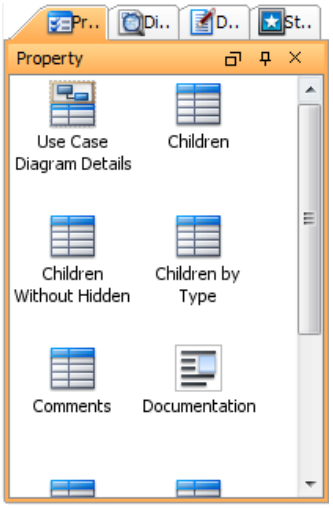
To create a template:

1. Editing or creating template must be done when report diagram is active. Open any report diagram.
2. Select the type of element to create template. For example, select ANY use case diagram in diagram navigator if you want to create a template to list specific shapes in use case diagram. You can select project/diagram/model element/diagram element in various trees like **Diagram Navigator**, **Model Explorer** or **Class Repository**.



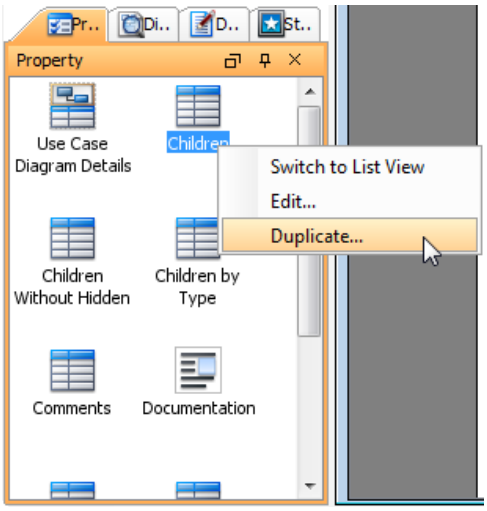
Select a use case diagram

3. The **Property** pane lists the templates available for the selected element type. If your **Property** pane is hidden, select **View > Panes > Property** (or press **Ctrl+Shift+P**) to show it.



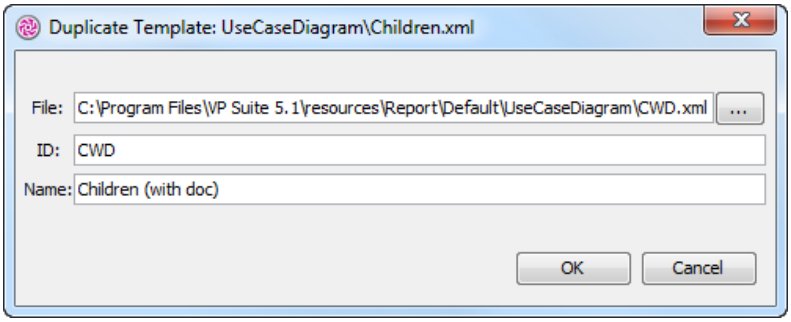
Property pane

4. To simplify the programming of template, you are suggested to duplicate an existing template and start editing it, rather than do do everything from scratch. Target on a template that gives the cloest outcome to what you want to show in report. If you want to start from an empty document, select any templates. Right click on your selection and select **Duplicate...** from the popup menu.



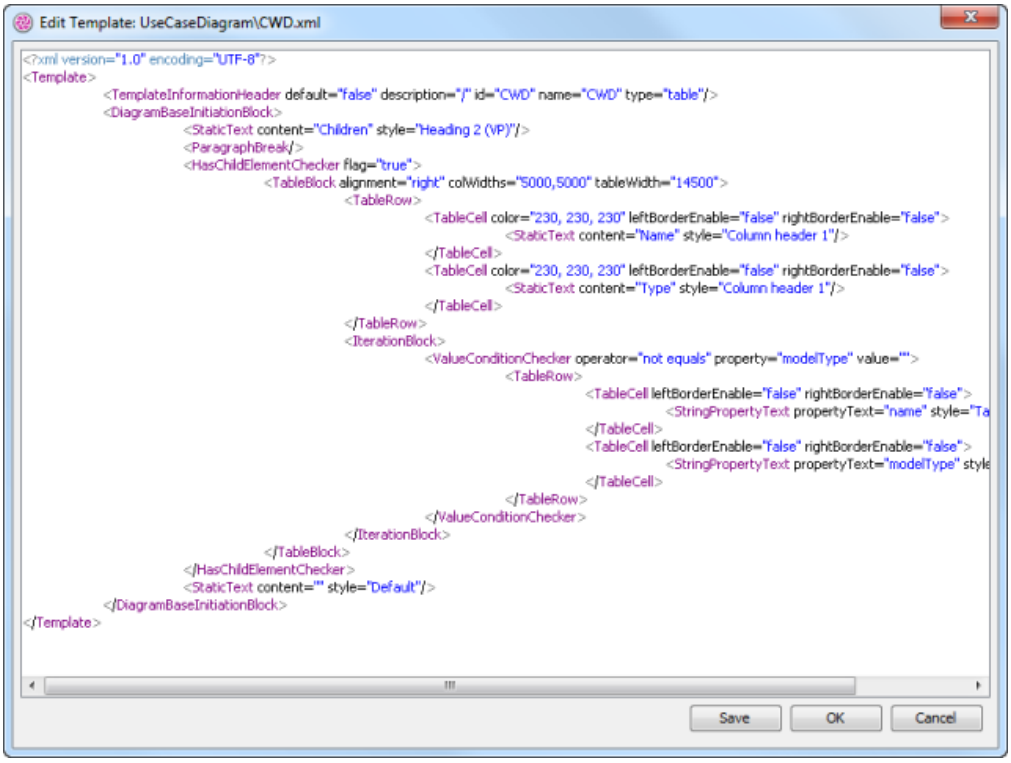
Duplicate a template

5. In the **Duplicate Template** dialog box, specify the following information and click **OK** to continue.
File - The physical file path of template file. You MUST store the template to the folder where the build-in templates are stored, which is the suggested path.
ID - A string that can uniquely identify the template.
Name - The name of the template. This is also the name to appear in **Property** pane.



Duplicate template dialog box

6. In the **Edit Template** dialog box, customize your template and click **OK** to apply the changes.



Editing a template

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Beginning of template

<Template>

Description
<Template> is the root element of a template document.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Basic Information" description = "/" type = "table" default = "true"/>
   ...
</Template>
```

<TemplateInformationHeader>

Description
<TemplateInformationHeader> is an element that must exist as a child of <Template>. It is used primarily for specifying the name, description and type of template. For details, please refer to the attributes section.

Attributes

Name	Description/Usage	Required?
id : string	A value for uniquely identifying a template.	Optional
name : string	The name of template. It serves two purposes: Act as the template display name to show in Template pane in report diagram Act as a key to relate the content displayed in report diagram and the template file Account for the second reason, do not change the name once the template has been used by a project, or else the linkage between report content and template file will lost, making it failed to update the content upon refreshing.	Required
description : string	A text to describe the template.	Optional
default : boolean = false	Specifies whether the template is the default template of the type of model element/diagram the template serves. When editing a report diagram, you may select a template, drag and drop it onto the diagram to form content. Alternatively, drag and drop a model element/diagram onto the diagram. At that moment, the default template will be applied. Note that each model element/diagram can have zero or up to one default template. If there are more than one default templates, just one of them will be chosen during report editing.	Optional
type = text {text | image | table}	A template can be classified as text, image or table template, depending on what the template will print on report diagram. A text template prints paragraphs of content, such as the documentation of model elements. An image template prints a diagram image, such as the image of a use case diagram. A table template prints a table of model element, diagram or their properties, such as a table of elements inside a package. The selection of type affects just the icon to use in Template pane. If you do not specify a type, the template will be classified as a text template.	Optional

Supported attributes for <TemplateInformationHeader>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Basic Information" description = "/" type = "table" default = "true"/>
   ...
</Template>
```

<ElementBaseInitiationBlock>

Description
<ElementBaseInitiationBlock> is an element that must exist as a child of <Template>. As an initiation block, <ElementBaseInitiationBlock> provides a hints to the template engine, to tell the engine that the template is going to apply on a model element, not for any other kind of project data. In other words, if you are creating a template for a model element (e.g. use case, package...), add a <ElementBaseInitiationBlock> under <Template>.

<ElementBaseInitiationBlock> represents the beginning of template. The remaining parts of template, its logic, shall be defined as children of <ElementBaseInitiationBlock>.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Children of Package" description = "/" type = "table" default = "true"/>
   <ElementBaseInitiationBlock>
   ...
   </ElementBaseInitiationBlock>
</Template>
```

<DiagramBaseInitiationBlock>

Description

<DiagramBaseInitiationBlock> is an element that must exist as a child of <Template>. As an initiation block, <DiagramBaseInitiationBlock> provides a hints to the template engine, to tell the engine that the template is going to apply on a diagram, not for any other kind of project data. In other words, if you are creating a template for a diagram (e.g. class diagram), add a <DiagramBaseInitiationBlock> under <Template>.

<DiagramBaseInitiationBlock> represents the beginning of template. The remaining parts of template, its logic, shall be defined as children of <DiagramBaseInitiationBlock>.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "My Diagram" description = "/" type = "image" default = "true"/>
   <DiagramBaseInitiationBlock>
   ...
   </DiagramBaseInitiationBlock>
</Template>
```

<DiagramElementBaseInitiationBlock>

Description

<DiagramElementBaseInitiationBlock> is an element that must exist as a child of <Template>. As an initiation block, <DiagramElementBaseInitiationBlock> provides a hints to the template engine, to tell the engine that the template is going to apply on a diagram element, not for any other kind of project data. A diagram element is a view (shape) of model element. You can visualize a model element (on multiple diagrams) to create as many diagram element(s) as you need. If you are creating a template for a diagram element (e.g. a class on class diagram), add a <DiagramElementBaseInitiationBlock> under <Template>.

<DiagramElementBaseInitiationBlock> represents the beginning of template. The remaining parts of template, its logic, shall be defined as children of <DiagramElementBaseInitiationBlock>.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Children of Package (diagram base)" description = "/" type = "text" default = "true"/>
   <DiagramElementBaseInitiationBlock>
   ...
   </DiagramElementBaseInitiationBlock>
</Template>
```

<ProjectBaseInitiationBlock>

Description

<ProjectBaseInitiationBlock> is an element that must exist as a child of <Template>. As an initiation block, <ProjectBaseInitiationBlock> provides a hints to the template engine, to tell the engine that the template is going to apply on a project – the origin of model, not for any other kind of project data. In other words, if you are creating a template for project, add a <ProjectBaseInitiationBlock> under <Template>.

<ProjectBaseInitiationBlock> represents the beginning of template. The remaining parts of template, its logic, shall be defined as children of <ProjectBaseInitiationBlock>.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "My diagrams in project" description = "/" type = "text" default = "true"/>
   <ProjectBaseInitiationBlock>
   ...
   </ProjectBaseInitiationBlock>
</Template>
```

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Reading model – Looping

<ForEachSimpleRelationship>

Description
<ForEachSimpleRelationship> is a construct for retrieving SimpleRelationship elements from model element, or connector from diagram element.

Attributes

Name	Description/Usage	Required?
Type : string [Deprecated]	Deprecated. Replaced by @modelType.	Optional
modelType : string	Filter relationship by specified model element type (e.g. Generalization). Note that not all kinds of relationship belong to simple relationship. Here are the possible types of simple relationship: Abstraction, ActivityObjectFlow, AnalysisComposition, AnalysisDiagramTransitor, AnalysisParentChild, AnalysisReference, AnalysisRelationship, AnalysisSubDiagram, AnalysisTransitor, AnalysisUsed, AnalysisView, Anchor, ArchiMateAccess, ArchiMateAggregation, ArchiMateAssignment, ArchiMateAssociation, ArchiMateCommunicationPath, ArchiMateFlow, ArchiMateNetwork, ArchiMateProvide, ArchiMateRealization, ArchiMateRequire, ArchiMateSpecialization, ArchiMateTriggering, ArchiMateUsedBy, AssociationClass, BPAssociation, BPDataAssociation, BPMessageFlow, BPSequenceFlow, BindingDependency, BusinessRuleAssociation, Constraint, ControlFlow, ConversationLink, DBForeignKey, DFDataFlow, Dependency, Deployment, EPCControlFlow, EPCInformationFlow, EPCOrganizationUnitAssignment, ExceptionHandler, Extend, Generalization, GenericConnector, GlossaryFactTypeAssociation, Include, InteractionDiagramDurationConstraint, Link, MindConnector, MindLink, OCLine, ObjectFlow, PMProcessLink, Permission, RQRefine, RQTrace, Realization, RequirementDerive, Satisfy, Transition, Transition2, Usage, Verify	Optional
modelTypes : string	Filter relationships by modelTypes. If @modelType is defined, @modelTypes will be ignored.	Optional
direction {all | from | to}	Filter relationship by direction.	Optional
ignoreParagraphBreakForLastModel : boolean = false	Ignore the break for the last element of current for-each loop. Break can be <ParagraphBreak> or <StaticText @content= "\n" or ",">.	Optional
breakString : string	Insert a string between model elements of current for-each.	Optional
sortBy [Deprecated]	Deprecated. Replaced by @sortBys.	Optional
sortBys {name | modelType | property | followTree}	Used in sorting model elements in the for-each loop. This is to sort model elements by name and/or model type and/or property. E.g. sortBys="modelType, name" Sorts model elements with their model type first. When same model type is met, sort by their names.	Optional
sortProperty : string	Used in sorting. This is to sort model elements by specific property. This attribute works together with sortBys="property". E.g. sortBys="property" sortProperty="visibility" Sort model elements by their "visibility" property.	Optional
sortValues : strings	Used in Sorting. This is to sort model elements base on the value of specific property. E.g. sortValues="private, protected, public" sortBys="property" sortProperty="visibility" The model elements will be sorted by "visibility" property in the order private -> protected -> public.	Optional
defaultPropertyValue : string	Used in Sorting. This is to specify a default value for the model elemet(s) that have no value for its property. This works together with sortBys="property".	Optional
descendingSort : boolean = false [Deprecated]	Deprecated. Replaced by @descendingSort.	Optional
descendingSorts : booleans	Determines whether the sorting is in descending order or not. True for descending, false for ascending.	Optional

	The numbers of descendingSorts should match the number of sortBys. For example, when @sortBy="name, modelType", @descendingSorts="true, false". This means to sort by name in descending order, and sort by modelType in ascending order.	

Supported attributes for <ForEachSimpleRelationship>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Generalizations of class" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <ForEachSimpleRelationship type="Generalization" direction="all">
         <StringPropertyText propertyText = "name"/>
      </ForEachSimpleRelationship>
   </ElementBaseInitiationBlock>
</Template>
```

<ForEachSubDiagram>

Description

<ForEachSubDiagram> is a construct for retrieving sub-diagram(s) from a model element. For example, retrieve sub-sequence-diagrams of a controller class. Note that you can only use <ForEachSubDiagram> to retrieve sub-diagram(s) of model element. If you want to retrieve diagrams from project, use <ForEachDiagram> instead.

Attributes

Name	Description/Usage	Required?
diagramType : string	The type of diagram to retrieve.	Optional
ignoreParagraphBreakForLastModel : boolean = false	Ignore the break for the last element of current for-each loop. Break can be <ParagraphBreak> or <StaticText @content= "\n" or ",">.	Optional
sortBy [Deprecated]	Deprecated. Replaced by @sortBys.	Optional
sortBys {name | modelType | property | followTree}	Used in sorting model elements in the for-each loop. This is to sort model elements by name and/or model type and/or property. E.g. sortBys="modelType, name" Sorts model elements with their model type first. When same model type is met, sort by their names.	Optional
sortProperty : string	Used in sorting. This is to sort model elements by specific property. This attribute works together with sortBys="property". E.g. sortBys="property" sortProperty="visibility" Sort model elements by their "visibility" property.	Optional
sortValues : strings	Used in Sorting. This is to sort model elements base on the value of specific property. E.g. sortValues="private, protected, public" sortBys="property" sortProperty="visibility" The model elements will be sorted by "visibility" property in the order private -> protected -> public.	Optional
defaultPropertyValue : string	Used in Sorting. This is to specify a default value for the model elemet(s) that have no value for its property. This works together with sortBys="property".	Optional
descendingSort : boolean = false [Deprecated]	Deprecated. Replaced by @descendingSort.	Optional
descendingSorts : booleans	Determines whether the sorting is in descending order or not. True for descending, false for ascending. The numbers of descendingSorts should match the number of sortBys. For example, when @sortBy="name, modelType", @descendingSorts="true, false". This means to sort by name in descending order, and sort by modelType in ascending order.	Optional

Supported attributes for <ForEachSubDiagram>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Name of sub-diagrams" description = "/" type = "text" default = "false"/>
```

```
  <ElementBaseInitiationBlock>
    <ForEachSubDiagram>
      <StringPropertyText propertyText = "name"/>
    </ForEachSubDiagram>
  </ElementBaseInitiationBlock>
</Template>
```

<ForEachOwnerDiagram>

Description

<ForEachOwnerDiagram> is a construct for retrieving the diagram(s) that owns a specific model element. For example, class diagram "Domain Diagram" and "Security" both contain class "Login" (same model element), by applying <ForEachOwnerDiagram> on the "Login" class, diagram "Domain Diagram" and "Security" will be returned.

Attributes

Name	Description/Usage	Required?
diagramType : string	The type of diagram to retrieve.	Optional
ignoreParagraphBreakForLastModel : boolean = false	Ignore the break for the last element of current for-each loop. Break can be <ParagraphBreak> or <StaticText @content= "\n" or ",">.	Optional
sortBy [Deprecated]	Deprecated. Replaced by @sortBys.	Optional
sortBys {name | modelType | property | followTree}	Used in sorting model elements in the for-each loop. This is to sort model elements by name and/or model type and/or property. E.g. sortBys="modelType, name" Sorts model elements with their model type first. When same model type is met, sort by their names.	Optional
sortProperty : string	Used in sorting. This is to sort model elements by specific property. This attribute works together with sortBys="property". E.g. sortBys="property" sortProperty="visibility" Sort model elements by their "visibility" property.	Optional
sortValues : strings	Used in Sorting. This is to sort model elements base on the value of specific property. E.g. sortValues="private, protected, public" sortBys="property" sortProperty="visibility" The model elements will be sorted by "visibility" property in the order private -> protected -> public.	Optional
defaultPropertyValue : string	Used in Sorting. This is to specify a default value for the model elemet(s) that have no value for its property. This works together with sortBys="property".	Optional
descendingSort : boolean = false [Deprecated]	Deprecated. Replaced by @descendingSort.	Optional
descendingSorts : booleans	Determines whether the sorting is in descending order or not. True for descending, false for ascending. The numbers of descendingSorts should match the number of sortBys. For example, when @sortBy="name, modelType", @descendingSorts="true, false". This means to sort by name in descending order, and sort by modelType in ascending order.	Optional

Supported attributes for <ForEachOwnerDiagram>

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Name of parent diagrams" description = "/" type = "text" default = "false"/>
  <ElementBaseInitiationBlock>
    <ForEachOwnerDiagram>
      <StringPropertyText propertyText = "name"/>
    </ForEachOwnerDiagram>
  </ElementBaseInitiationBlock>
</Template>
```

<ForEachDiagram>

Description
<ForEachDiagram> is a construct for retrieving diagram(s) from project. Like other for-each constructs, you can specify the type of diagram to retrieve. For example, retrieve all class diagrams from project. Note that you can only use <ForEachDiagram> to retrieve diagram from project. If you want to retrieve sub-diagrams from model element, use <ForEachSubDiagram> instead.

Attributes

Name	Description/Usage	Required?
diagramType : string	The type of diagram to retrieve.	Optional
ignoreParagraphBreakForLastModel : boolean = false	Ignore the break for the last element of current for-each loop. Break can be <ParagraphBreak> or <StaticText @content= "\n" or ",">.	Optional
sortBy [Deprecated]	Deprecated. Replaced by @sortBys.	Optional
sortBys {name | modelType | property | followTree}	Used in sorting model elements in the for-each loop. This is to sort model elements by name and/or model type and/or property. E.g. sortBys="modelType, name" Sorts model elements with their model type first. When same model type is met, sort by their names.	Optional
sortProperty : string	Used in sorting. This is to sort model elements by specific property. This attribute works together with sortBys="property". E.g. sortBys="property" sortProperty="visibility" Sort model elements by their "visibility" property.	Optional
sortValues : strings	Used in Sorting. This is to sort model elements base on the value of specific property. E.g. sortValues="private, protected, public" sortBys="property" sortProperty="visibility" The model elements will be sorted by "visibility" property in the order private -> protected -> public.	Optional
defaultPropertyValue : string	Used in Sorting. This is to specify a default value for the model elemet(s) that have no value for its property. This works together with sortBys="property".	Optional
descendingSort : boolean = false [Deprecated]	Deprecated. Replaced by @descendingSort.	Optional
descendingSorts : booleans	Determines whether the sorting is in descending order or not. True for descending, false for ascending. The numbers of descendingSorts should match the number of sortBys. For example, when @sortBy="name, modelType", @descendingSorts="true, false". This means to sort by name in descending order, and sort by modelType in ascending order.	Optional

Supported attributes for <ForEachDiagram>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Name of diagrams" description = "/" type = "text" default = "false"/>
   <ProjectBaseInitiationBlock>
      <ForEachDiagram>
         <StringPropertyText propertyText = "name"/>
      </ForEachDiagram>
   </ProjectBaseInitiationBlock>
</Template>
```

<IterationBlock>

Description
<IterationBlock> is a construct for retrieving children elements from project/model element/diagram/diagram element. Iterating through project and model element will retrieve a list of model element, while iterating through diagram and diagram element will retrieve a list of diagram element.

If you want to retrieve sub-diagrams of a model element, check <ForEachDiagram>. If you want to retrieve all diagrams of project, check < ForEachSubDiagram>.

Attributes

Name	Description/Usage	Required?

elementType : string [Deprecated]	Deprecated. Replaced by @modelType.	Optional
modelType : string	Filter the children by specified model element type (e.g. package).	Optional
elementTypes : string [Deprecated]	Deprecated. Replaced by @modelTypes.	Optional
modelTypes : string	Filter the children by a number of model element types. (e.g. actor, usecase)	Optional
stereotypes : string	Filter the children by a number of stereotypes.	Optional
name : string	Filter the children by their name.	Optional
filterhidden : boolean = false	Filter hidden children diagram element. This is for retrieving from diagram/ diagram element only.	Optional
includeConnector : boolean = false [Deprecated]	Deprecated. Replaced by @includeConnectors.	Optional
includeConnectors : boolean = false	Determines whether to retrieve shape or shape+connectors from diagram. This is for retrieving from diagram only.	Optional
byBounds : boolean = false	Determines whether to retrieve shape by their bounds. This is for retriever from diagram or diagram element only.	Optional
byBoundsInAllLevel : boolean = false	Determines whether to retrieve shape by their bounds (include nested level). This is for retriever from diagram or diagram element only.	Optional
allLevel : boolean = false	Determines whether to retrieve all model elements from project. When false, only the root level elements will be retrieved. This attribute is only useful when retrieving elements from project.	Optional
ignoreParagraphBreakForLastModel : boolean = false	Ignore the break for the last element of current for-each loop. Break can be <ParagraphBreak> or <StaticText @content= "\n" or ",">.	Optional
breakString : string	Insert a string between model elements of current for-each.	Optional
sortBy [Deprecated]	Deprecated. Replaced by @sortBys.	Optional
sortBys {name | modelType | property | followTree}	Used in sorting model elements in the for-each loop. This is to sort model elements by name and/or model type and/or property. E.g. sortBys="modelType, name" Sorts model elements with their model type first. When same model type is met, sort by their names.	Optional
sortProperty : string	Used in sorting. This is to sort model elements by specific property. This attribute works together with sortBys="property". E.g. sortBys="property" sortProperty="visibility" Sort model elements by their "visibility" property.	Optional
sortValues : strings	Used in Sorting. This is to sort model elements base on the value of specific property. E.g. sortValues="private, protected, public" sortBys="property" sortProperty="visibility" The model elements will be sorted by "visibility" property in the order private -> protected -> public.	Optional
defaultPropertyValue : string	Used in Sorting. This is to specify a default value for the model elemet(s) that have no value for its property. This works together with sortBys="property".	Optional
descendingSort : boolean = false [Deprecated]	Deprecated. Replaced by @descendingSort.	Optional
descendingSorts : booleans	Determines whether the sorting is in descending order or not. True for descending, false for ascending. The numbers of descendingSorts should match the number of sortBys. For example, when @sortBy="name, modelType", @descendingSorts="true, false". This means to sort by name in descending order, and sort by modelType in ascending order.	Optional

Supported attributes for <IterationBlock>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Children" description = "/" type = "table" default = "false"/>
   <ElementBaseInitiationBlock>
      <IterationBlock modelType="class">
         <StringPropertyText propertyText = "name"/>
      </IterationBlock>
   </ElementBaseInitiationBlock>
</Template>
```

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Reading model - Conditional checking

<DefaultConditionChecker>

Description

<DefaultConditionChecker> is a construct which enables you to check whether the value of a property of a model element is the same as the default value, to control whether or not to continue processing the children template nodes base on the result of checking. Let’s take use case, a kind of model element, as example. For each use case, you can optionally set its rank to Low/Medium/High. When you are preparing a report to list the use cases with respect to their ranks, you probably interested in seeing the use cases that have rank set, but not those without rank set. In that situation you can make use of <DefaultConditionChecker> to make the report ignore those use cases with rank be Unspecified, which is the default value of rank.

Attributes

Name	Description/Usage	Required?
property : string	The property to check.	Required
equalFlag : boolean = false [Deprecated]	Deprecated. Replaced by @flag.	Optional
flag : boolean = false	Determines whether you expect the value to be the same as the default or not.	Optional

Supported attributes for <DefaultConditionChecker>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Visibility of classes" description = "/" type = "text" default = "false"/>
   <DiagramBaseInitiationBlock>
      <IterationBlock>
         <DefaultConditionChecker property="visibility" flag="false">
            <StringPropertyText propertyText = "visibility"/>
         </DefaultConditionChecker>
      </IterationBlock>
   </DiagramBaseInitiationBlock>
</Template>
```

<ValueConditionChecker>

Description

<ValueConditionChecker> is a construct which enables you to check whether the value of a property of a model element is the same as what you expect, to control whether or not to continue processing the children template nodes base on the result of checking. For example, print out the name of classes that contain "Controller" in their names.

Attributes

Name	Description/Usage	Required?
property : string	The property to check.	Required
propertyType : string {string | int | boolean | model}	The type of property.	Optional
operator : string {equals, not equals, less than, equals or less than, greater than, equals or greater than, like, not like, equals, not equal}	Specify the way to compare the property value of model against your expectation. equals - The value of property must be the same as the expected value not equals - The value of property must be different from the expected value less than - The value of property must be smaller than the expected value. equals or less than - The value of property must be the same or smaller than the expected value. greater than - The value of property must be larger than the expected value. equals or greater than - The value of property must be the same or larger than the expected value. like – The value of property must contain the expected value not like - The value of property must not contain the expected value equal – Deprecated. Replaced by "equals ". not equal - Deprecated. Replaced by "not equals ".	Required
value : string	The value expected for the property	Optional
caseSensitive : boolean = true	Determines whether the checking of string property need to take care of the use of upper and lower case.	Optional

Supported attributes for <ValueConditionChecker>

Sample template fragment

```
<Template>
```

```
  <TemplateInformationHeader name = "Controller classes" description = "/" type = "text" default = "false"/>
  <DiagramBaseInitiationBlock>
    <IterationBlock>
      <ValueConditionChecker property="name" operator = "like" value = "Controller" >
        <StringPropertyText propertyText = "name"/>
      </ValueConditionChecker>
    </IterationBlock>
  </DiagramBaseInitiationBlock>
</Template>
```

<HasChildElementChecker>

Description

<HasChildElementChecker> is a construct which enables you to check whether a project or a model element contains a child element, to control whether or not to continue processing the children template nodes base on the result of checking.

Attributes

Name	Description/Usage	Required?
flag : boolean = true	Check whether you want any child exists or not.	Optional
type : string [Deprecated]	Deprecated. Replaced by @modelType.	Optional
modelType : string	The type of model element you want the parent to contain or not contain.	Optional
elementTypes : string [Deprecated]	Deprecated. Replaced by @modelTypes.	Optional
stereotypes : strings	Filter the children by a number of stereotypes.	Optional
includeConnector : boolean [Deprecated]	Deprecated. Replaced by @includeConnectors	Optional
includeConnectors : boolean = false	Determines whether to retrieve shape or shape+connectors from diagram. This is for retrieving from diagram only.	Optional
allLevel : boolean = false	Determines whether to retrieve all model elements from project. When false, only the root level elements will be retrieved. This attribute is only useful when retrieving elements from project.	Optional

Supported attributes for <HasChildElementChecker>

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Package has children?" description = "/" type = "text" default = "false"/>
  <ElementBaseInitiationBlock>
    <HasChildElementChecker flag = "true">
      <StaticText content = "This package has children!"/>
    </HasChildElementChecker>
  </ElementBaseInitiationBlock>
</Template>
```

<HasDiagramChecker>

Description

<HasDiagramChecker> is a construct which enables you to check whether a project contains any diagram, or a specific type of diagram, to control whether or not to continue processing the children template nodes base on the result of checking.

Attributes

Name	Description/Usage	Required?
flag : boolean = true	Check whether you want any diagram exists or not.	Optional
type : string [Deprecated]	Deprecated. Replaced by @diagramType.	Optional
diagramType : string	The type of diagram you want the project to contain or not contain.	Optional

Supported attributes for <HasDiagramChecker>

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Has Class Diagram?" description = "/" type = "text" default = "false"/>
```

```
    <ProjectBaseInitiationBlock>
        <HasDiagramChecker type = "ClassDiagram">
            <StaticText content = "This project contains class diagram!"/>
        </HasDiagramChecker>
    </ProjectBaseInitiationBlock>
</Template>
```

<HasMetaModelPropertyChecker>

Description

<HasMetaModelPropertyChecker> is a construct which enables you to check whether a model element contains any model element, or a specific type of model element, to control whether or not to continue processing the children template nodes base on the result of checking.

Attributes

Name	Description/Usage	Required?
flag : boolean = true	Check whether you want the model element exists or not.	Optional
property : string	The name of model element property you want to contain or not contain.	Required

Supported attributes for <HasMetaModelPropertyChecker>

Sample template fragment

```
<Template>
    <TemplateInformationHeader name = "Has Stereotype?" description = "/" type = "text" default = "false"/>
    <ElementBaseInitiationBlock>
        <HasMetaModelPropertyChecker property = "stereotypes" flag = "true">
            <StaticText content = "This element contains stereotype!"/>
        </HasMetaModelPropertyChecker>
    </ElementBaseInitiationBlock>
</Template>
```

<HasParentModelChecker>

Description

<HasParentModelChecker> is a construct which enables you to check whether a model element is contained by another model element, to control whether or not to continue processing the children template nodes base on the result of checking. For example, check whether a class is under (i.e. contained by) a package.

Attributes

Name	Description/Usage	Required?
flag : boolean = true	Check whether you want the parent exists or not.	Optional
modelType : string	The type of parent model element you want the model element to be/not to be contained by.	Optional

Supported attributes for <HasParentModelChecker>

Sample template fragment

```
<Template>
    <TemplateInformationHeader name = "Contained by parent?" description = "/" type = "text" default = "false"/>
    <ElementBaseInitiationBlock>
        <HasParentModelChecker flag = "true">
            <StaticText content = "This element is contained by a parent!"/>
        </HasParentModelChecker>
    </ElementBaseInitiationBlock>
</Template>
```

<HasSubDiagramChecker>

Description

<HasSubDiagramChecker> is a construct which enables you to check whether a model element contains any sub-diagram, to control whether or not to continue processing the children template nodes base on the result of checking.

Attributes

Name	Description/Usage	Required?
flag : boolean = true	Check whether you want any sub-diagram exists or not.	Optional
type : string [Deprecated]	Deprecated. Replaced by @diagramType	Optional
diagramType : string	The type of sub-diagram you want the model element to contain.	Optional

Supported attributes for <HasSubDiagramChecker>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Contained sub-diagram?" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <HasSubDiagramChecker flag = "true">
         <StaticText content = "This element contains sub-diagram!"/>
      </HasSubDiagramChecker>
   </ElementBaseInitiationBlock>
</Template>
```

<ReportOptionChecker>

Description

<ReportOptionChecker> is a construct which enables you to check specific report option, to control whether or not to continue processing the children template nodes base on the result of checking.

Attributes

Name	Description/Usage	Required?
equalFlag : Boolean [Deprecated]	Deprecated. Replaced by @flag	Optional
flag : boolean = false	Check whether you want a report option to be true or false.	Optional
option : string [Deprecated]	The option to check. For example, basicInformationShowsDocumentation.	Optional

Supported attributes for <ReportOptionChecker>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Documentation" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <ReportOptionChecker option = "basicInformationShowsDocumentation" equalFlag = "true">
         <StringPropertyText propertyText = "documentation"/>
      </ReportOptionChecker>
   </ElementBaseInitiationBlock>
</Template>
```

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Reading model - Model/diagram element

<MetaModelElement>

Description
<MetaModelElement> is a construct which enables you to retrieve a model element from another model element.

Attributes

Name	Description/Usage	Required?
modelProperty : string	The name of property to query from model element, which should result in returning a model element from the given model element.	Optional

Supported attributes for <MetaModelElement>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Tagged values" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <MetaModelElement modelProperty = "taggedValues">
         <IterationBlock>
            <StringPropertyText propertyText = "name"/>
         </ IterationBlock >
      </MetaModelElement>
   </ElementBaseInitiationBlock>
</Template>
```

<FromEnd>

Description
<FromEnd> is a construct for retrieving the from-side of a relationship. Here are some of the key points you need to know when using <FromEnd> in your template:

- The from-end of relationship is determined by the side a connector (shape) begins with. It has nothing related to the semantic or definition of notation. Take generalization as example, the from-side is the side with arrow head (unless you have configured the beginning of generalization in Options window).
- When you try to retrieve the from-end of an association, you are retrieving the connected element (e.g. a class, a use case) instead of a role. If you need to retrieve a role, use <ForEachRelationshipEnd>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Associations and Generalizations" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <!-- Print from ends of associations -->
      <ForEachRelationshipEnd type = "AssociationEnd" endPointer = "self">
         <RelationshipEndEndRelationship>
            <FromEnd>
               <StringPropertyText propertyText = "name"/>
            </FromEnd>
         </RelationshipEndEndRelationship>
      </ForEachRelationshipEnd>
      <!-- Print from ends of generalizations -->
      <ForEachSimpleRelationship type = "Generalization" direction = "all">
         <FromEnd>
            <StringPropertyText propertyText = "name"/>
         </FromEnd>
      </ForEachRelationshipEnd>
   </ElementBaseInitiationBlock>
</Template>
```

<ToEnd>

Description
<ToEnd> is a construct for retrieving the to-side of a relationship. Here are some of the key points you need to know when using <ToEnd> in your template:

- The to-end of relationship is determined by the side a connector (shape) begins with. It has nothing related to the semantic or definition of notation. Take generalization as example, the to-side is the side without arrow head (unless you have configured the beginning of generalization in Options window).
- When you try to retrieve the to-end of an association, you are retrieving the connected element (e.g. a class, a use case) instead of a role. If you need to retrieve a role, use <ForEachRelationshipEnd> to retrieve the association ends of an elements. The association ends returned is then the roles.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Associations and Generalizations" description = "/" type = "text" default = "false"/>
```

```
  <ElementBaseInitiationBlock>
    <!-- Print to ends of associations -->
    <ForEachRelationshipEnd type = "AssociationEnd" endPointer = "self">
      <RelationshipEndEndRelationship>
        <ToEnd>
          <StringPropertyText propertyText = "name"/>
        </ToEnd>
      </RelationshipEndEndRelationship>
    </ForEachRelationshipEnd>
    <!-- Print to ends of generalizations -->
    <ForEachSimpleRelationship type = "Generalization" direction = "all">
      <ToEnd>
        <StringPropertyText propertyText = "name"/>
      </ToEnd>
    </ForEachRelationshipEnd>
  </ElementBaseInitiationBlock>
</Template>
```

<RelationshipEndEndRelationship>

Description

<RelationshipEndEndRelationship> is a construct for retrieving relationship from a relationship end (e.g. association end). Note that <RelationshipEndEndRelationship> has no effect with <FromEnd> and <ToEnd>.

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Associations" description = "/" type = "text" default = "false"/>
  <ElementBaseInitiationBlock>
    <!-- Print to ends of associations -->
    <ForEachRelationshipEnd type = "AssociationEnd" endPointer = "self">
      <RelationshipEndEndRelationship>
        <ToEnd>
          <StringPropertyText propertyText = "name"/>
        </ToEnd>
      </RelationshipEndEndRelationship>
    </ForEachRelationshipEnd>
  </ElementBaseInitiationBlock>
</Template>
```

<RelationshipEndOppositeEnd>

Description

<RelationshipEndOppositeEnd> is a construct for retrieving the opposite end of a relationship end (e.g. association end). Note that <RelationshipEndOppositeEnd> has no effect with <FromEnd> and <ToEnd>.

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Associations" description = "/" type = "text" default = "false"/>
  <ElementBaseInitiationBlock>
    <ForEachRelationshipEnd type = "Print myself" endPointer = "self">
      <RelationshipEndOppositeEnd>
        <MetaModelElement property="EndModelElement">
          <StringPropertyText propertyText = "name"/>
        </ MetaModelElement>
      </RelationshipEndOppositeEnd>
    </ForEachRelationshipEnd>
  </ElementBaseInitiationBlock>
</Template>
```

<ParentModel>

Description

<ParentModel> is a construct for retrieving the parent model element that contains a model element. For example, retrieve the package of a class.

Attributes

Name	Description/Usage	Required?
modelProperty : string	The type of parent model element you want the model element to be/not to be contained by.	Optional

Supported attributes for <ParentModel>

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Parent" description = "/" type = "text" default = "false"/>
  <ElementBaseInitiationBlock>
```

```
    <ParentModel>
      <StringPropertyText propertyText = "name"/>
    </ParentModel>
  </ElementBaseInitiationBlock>
</Template>
```

<DiagramProperty>

Description

<DiagramProperty> is a construct for retrieving a diagram from property. For example, retrieve embedded process diagram of a BPMN sub-process. Note that <DiagramProperty> is not for retrieving sub-diagrams. If you want to retrieve sub-diagrams, use <ForEachSubDiagram> instead.

Attributes

Name	Description/Usage	Required?
property : string	The name of diagram property to retrieve from model element.	Required

Supported attributes for <DiagramProperty>

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Diagram name of analysis diagram" description = "/" type = "text" default = "false"/>
  <DiagramBaseInitiationBlock>
    <IterationBlock elementType="AnalysisDiagramNode">
      <DiagramProperty property="analysisDiagramId">
        <StringPropertyText property="name"/>
      </DiagramProperty>
    </IterationBlock>
  </DiagramBaseInitiationBlock>
</Template>
```

<OwnerDiagram>

Description

<OwnerDiagram> is a construct that enables you to retrieve the diagram where a diagram element exists.

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Diagram name of analysis diagram node" description = "/" type = "text" default = "false"/>
  <DiagramBaseInitiationBlock>
    <IterationBlock elementType="AnalysisViewNode">
      <DiagramElementProperty property="analysisViewId">
        <OwnerDiagram>
          <StringPropertyText property="name"/>
        </OwnerDiagram>
      </ DiagramElementProperty >
    </IterationBlock>
  </DiagramBaseInitiationBlock>
</Template>
```

<ElementImage>

Description

<ElementImage> is a construct for printing an image of diagram or diagram element, or icon for model type.

Attributes

Name	Description/Usage	Required?
imageType : string = icon {diagram | icon}	Specify the type of image. diagram – image for diagram/diagram element icon – icon for model type	Optional
imageFormat = png {png | jpg | wmf}	The image format (png/jpg/wmf) of the image to present.	Optional
alignment = left {left | center | right}	The position of image – left/center/right.	Optional
width : integer	Restrict the image to specific width.	Optional
height : integer	Restrict the image to specific height.	Optional
maxWidth : integer	Set the maximum width of image.	Optional
maxHeight : integer	Set the maximum height of image.	Optional

landscape : boolean = false	Adjust the orientation of image. True for landscape. False for portrait.	Optional

Supported attributes for <ElementImage>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Class diagrams in project" description = "/" type = "text" default = "false"/>
   <ProjectBaseInitiationBlock>
      <ForEachDiagram diagramType="ClassDiagram">
         <ElementImage imageType="diagram"/>
      </ForEachDiagram>
   </ProjectBaseInitiationBlock>
</Template>
```

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Text and break

<StaticText>

Description
<StaticText> is a construct which enables you to insert your own text into a report, at the position where <StatcText> is processed. You can add your text with custom formats.

Attributes

Name	Description/Usage	Required?
content : string	The text to add.	Optional
isBold : boolean	Set the text to bold.	Optional
isItalic : boolean	Set the text italic.	Optional
isUnderline : boolean	Set the text to underline.	Optional
isSuperscript : boolean	Set the text to superscript.	Optional
fontFamily : string = Times New Roman	Specify the name of font to apply to the text.	Optional
alignment = left {left | center | right}	Set the alignment of text.	Optional
fontSize : integer = 12	Set the font size.	Optional
style : string	Set the name of style. You can add and edit style in report diagram.	Optional
numberingLevel : short	Detemine the Numbering Level if the text is showing as a number or bullet list.	Optional
foreColor = color	The color of text.	Optional
indentation : double	Determine the left margin of text.	Optional
hyperlink : boolean = false	Specify whether the text is a hyperlink or not. If true, the text will be linkable.	Optional
keepWithNextInWord : boolean = false	Make sure the text will be shown in same page with next item. (Used for WORD Report only)	Optional

Supported attributes for <StaticText>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Say Hello" description = "/" type = "text" default = "false"/>
   <ProjectBaseInitiationBlock>
      <StaticText content="Hello!" />
   </ProjectBaseInitiationBlock>
</Template>
```

<StringPropertyText>

Description
<StringPropertyText> is the most commonly used construct in a report template document. It is used to extract a string property from model element (e.g. name of a class) and print it to report.

Attributes

Name	Description/Usage	Required?
propertyText : string	Deprecated. Replaced by @property.	Optional
isIgnoreHTMLFontSize : boolean = false	Ignore the font size of the HTML text.	Optional
isIgnoreHTMLFontFamily : boolean = false	Ignore the font selection of the HTML text	Optional
forcePlainText : boolean	Force HTML text to show as plain text by removing formatting, if any.	Optional
defaultValue : string	The text to show when the value of property has not ever been specified.	Optional
property : string	The name of string property to retrieve from model element.	Required
isBold : boolean	Set the text to bold.	Optional
isItalic : boolean	Set the text italic.	Optional
fontFamily : string = Times New Roman	Specify the name of font to apply to the text.	Optional

alignment = left {left | center | right}	Set the alignment of text.	Optional
fontSize : integer = 12	Set the font size.	Optional
style : string	Set the name of style. You can add and edit style in report diagram.	Optional
numberingLevel : short	Detemine the Numbering Level if the text is showing as a number or bullet list.	Optional
foreColor = color	The color of text.	Optional
indentation : double	Determine the left margin of text.	Optional
hyperlink : boolean = false	Specify whether the text is a hyperlink or not. If true, the text will be linkable.	Optional
keepWithNextInWord : boolean = false	Make sure the text will be shown in same page with next item. (Used for WORD Report only)	Optional

Supported attributes for <StringPropertyText>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Name of use cases" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <StringPropertyText propertyText="name/>
   </ElementBaseInitiationBlock>
</Template>
```

<IntPropertyText>

Description

<IntPropertyText> is a construct which enables you to extract an integer property from model element (e.g. length of an entity column) and print it to report.

Attributes

Name	Description/Usage	Required?
propertyText : string	Deprecated. Replaced by @property.	Optional
isIgnoreHTMLFontSize : boolean = false	Ignore the font size on the HTML text.	Optional
property : string	The name of integer property to retrieve from model element.	Required
isBold : boolean	Set the text to bold.	Optional
isItalic : boolean	Set the text italic.	Optional
fontFamily : string = Times New Roman	Specify the name of font to apply to the text.	Optional
alignment = left {left | center | right}	Set the alignment of text.	Optional
fontSize : integer = 12	Set the font size.	Optional
style : string	Set the name of style. You can add and edit style in report diagram.	Optional
numberingLevel : short	Detemine the Numbering Level if the text is showing as a number or bullet list.	Optional
foreColor = color	The color of text.	Optional
keepWithNextInWord : boolean = false	Make sure the text will be shown in same page with next item. (Used for WORD Report only)	Optional

Supported attributes for <IntPropertyText>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Entity columns" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <IterationBlock elementType = "DBColumn">
         <IntPropertyText propertyText = "length"/>
      </IterationBlock>
   </ElementBaseInitiationBlock>
</Template>
```

<BooleanPropertyText>

Description
<BooleanPropertyText> is a construct which enables you to extract a boolean property (i.e. true/false) from model element (e.g. abstract of a class) and print it to report.

Attributes

Name	Description/Usage	Required?
propertyText : string	Deprecated. Replaced by @property.	Optional
property : string	The name of boolean property to retrieve from model element.	Required
isBold : boolean	Set the text to bold.	Optional
isItalic : boolean	Set the text italic.	Optional
fontFamily : string = Times New Roman	Specify the name of font to apply to the text.	Optional
alignment = left {left | center | right}	Set the alignment of text.	Optional
fontSize : integer = 12	Set the font size.	Optional
style : string	Set the name of style. You can add and edit style in report diagram.	Optional
numberingLevel : short	Detemine the Numbering Level if the text is showing as a number or bullet list.	Optional
foreColor = color	The color of text.	Optional
keepWithNextInWord : boolean = false	Make sure the text will be shown in same page with next item. (Used for WORD Report only)	Optional

Supported attributes for <BooleanPropertyText>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Is class abstract?" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <StaticText content = "Abstract?" style = "Row caption 1"/>
      <BooleanPropertyText propertyText = "abstract"/>
   </ElementBaseInitiationBlock>
</Template>
```

<ParagraphBreak>

Description
<ParagraphBreak> is a construct which enables you to add a break in report to separate text into paragraphs. <ParagraphBreak> does not carry any text.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Two paragraphs" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <StaticText content = "Paragraph 1" style = "Row caption 1"/>
      <ParagraphBreak>
      <StaticText content = "Paragraph 2" style = "Row caption 1"/>
   </ElementBaseInitiationBlock>
</Template>
```

<PageBreak>

Description
<PageBreak> is a construct which enables you to insert a new page at where <PageBreak> is processed.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Two pages" description = "/" type = "text" default = "false"/>
   <ElementBaseInitiationBlock>
      <StaticText content = "Page 1" style = "Row caption 1"/>
      <ParagraphBreak>
      <StaticText content = " Page 2" style = "Row caption 1"/>
   </ElementBaseInitiationBlock>
</Template>
```

Related Resources
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Table

<TableBlock>

Description
<TableBlock> is a construct that enables you to insert a table to report diagram. It is typically used to present table of elements or element properties. You must combine the use of <TableRow> and <TableCell> in order to form a complete table. Most of the build-in templates are formed by tables and contains <TableBlock>. You can look for references easily.

Attributes

Name	Description/Usage	Required?
tableWidth : integer	The width of table, in percentage. For example: 50% means to occupy 50% of page width.	Optional
colWidths : integers	Specify the widths of table columns in ratio, separate by comma. Note that the number of columns specify in @colWidths must match the number of <TableCell> to add under <TableRow>, under this table. For example, specify "1, 1, 2" for a table with 20000 as width will result in creating a table with three columns, and have widths 5000, 5000, 10000.	Optional
alignment = center {left | center | right}	Alignment of table.	Optional
rowBackgroundColors : colors	Background color of rows in table.	Optional
generateHeaderOnNewPage : boolean = false	When true, generate the header (the first row of this table) again if this table is generated across multiple pages.	Optional
margin	Specify the space around the table by providing the value of top, right, bottom, and left margins. For example, specify "10, 20, 30, 40" for a table that has 10 as top margin, 20 as right margin, 30 as bottom margin and 40 as left margin.	Optional
margin-top: integer	Specify the top margin of table.	Optional
margin-right: integer	Specify the right margin of table.	Optional
margin-bottom: integer	Specify the bottom margin of table.	Optional
margin-left: integer	Specify the left margin of table.	Optional

Supported attributes for <TableBlock>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Classes in diagram" description = "/" type = "table" default = "false"/>
   <DiagramBaseInitiationBlock>
      <TableBlock tableWidth="10000" colWidths="1,3" alignment="right">
         <IterationBlock modelType="Class">
            <TableRow>
               <TableCell>
                  <StringPropertyText propertyText="name"/>
               </TableCell>
               <TableCell>
                  <StringPropertyText propertyText="documentation"/>
               </TableCell>
            </TableRow>
         </IterationBlock>
      </TableBlock>
   </DiagramBaseInitiationBlock>
</Template>
```

<TableRow>

Description
<TableRow> is a construct that enables you to add rows to a <TableBlock>. Without <TableRow>, <TableBlock> is useless. You should add <TableCell> to <TableRow> in order to complete a table.

Attributes

Name	Description/Usage	Required?
height : integer	How tall it is for the table row.	Optional
backgroundColor : color	Background color of row.	Optional

Supported attributes for <TableRow>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Classes in diagram" description = "/" type = "table" default = "false"/>
   <DiagramBaseInitiationBlock>
      <TableBlock tableWidth="10000" colWidths="1,3" alignment="right">
         <IterationBlock modelType="Class">
            <TableRow>
               <TableCell>
                  <StringPropertyText propertyText="name"/>
               </TableCell>
               <TableCell>
                  <StringPropertyText propertyText="documentation"/>
               </TableCell>
            </TableRow>
         </IterationBlock>
      </TableBlock>
   </DiagramBaseInitiationBlock>
</Template>
```

<TableCell>

Description
<TableCell> is a construct that enables you to add cells to a <TableRow>.

Attributes

Name	Description/Usage	Required?
topBorderEnable : boolean = true	True to draw the top border of cell.	Optional
bottomBorderEnable : boolean = true	True to draw the bottom border of cell.	Optional
leftBorderEnable : boolean = true	True to draw the left border of cell.	Optional
rightBorderEnable : boolean = true	True to draw the right border of cell.	Optional
color : color	The background color of cell.	Optional
splitted : boolean = false	Determine this cell is splitted. If true, <TableRow> + <TableCell> cannot be added into this cell to make the cell splitted by more cells.	Optional
mergeColumns : integer	Deprecated. Replaced by @colspan.	Optional
colspan : integer	Specify the number of cell this cell consumes horizontally. For example, a colspan of 2 means to consume this and the cell on the right. This is equivalent to HTML’s colspan.	Optional

Supported attributes for <TableCell>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Classes in diagram" description = "/" type = "table" default = "false"/>
   <DiagramBaseInitiationBlock>
      <TableBlock tableWidth="10000" colWidths="1,3" alignment="right">
         <IterationBlock modelType="Class">
            <TableRow>
               <TableCell>
                  <StringPropertyText propertyText="name"/>
               </TableCell>
               <TableCell>
                  <StringPropertyText propertyText="documentation"/>
               </TableCell>
            </TableRow>
         </IterationBlock>
      </TableBlock>
   </DiagramBaseInitiationBlock>
</Template>
```

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Image

<ElementImage>

Description

<ElementImage> is a construct which enables you to insert diagram image or an icon of specific shape type (e.g. icon of use case shape), at the position where <ElementImage> is processed.

Attributes

Name	Description/Usage	Required?
imageType : string {diagram | icon}	The text to add.	Optional
alignment = left {left | center | right}	Set the alignment of text.	Optional
imageFormat : string {png | jpg | wmf}	Set the file format of image.	Optional
keepWithPreviousInPDF : boolean	Make sure the previous item will be shown in same page with this item. (Used for PDF Report only)	Optional
keepWithNextInWord : boolean	Make sure the this item will be shown in same page with next item. (Used for WORD Report only)	Optional
width : string	Set the width of image. It can be an absolute value (e.g. "15500") or a scale to the original image width (e.g. "80%")	Optional
height : string	Set the height of image. It can be an absolute value (e.g. "15500") or a scale to the original image height (e.g. "80%")	Optional
maxWidth : integer	Set the maximum width of image.	Optional
maxHeight : integer	Set the maximum height of image.	Optional
rotate : string {none | right | left}	Rotate the image to right (90 degree) or left (270 degree)	Optional

Supported attributes for <ElementImage>

Sample template fragment

```
<Template>
  <TemplateInformationHeader name = "Print All Diagram Images" description = "/" type = "text" default = "false"/>
  <ProjectBaseInitiationBlock>
    <ForEachDiagram>
      <!-- Print the image of diagrams in project -->
      <ElementImage imageType="diagram" alignment="center"/>
      <IterationBlock>
        <!-- Print the icon of shapes in each diagram -->
        <ElementImage type="icon"/>
      </IterationBlock>
    </ForEachDiagram>
  </ProjectBaseInitiationBlock>
</Template>
```

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Miscellaneous

<BPProcedures>

Description
<BPProcedures> is a construct that prints in report the procedures of task/sub-process, if specified.

Attributes

Name	Description/Usage	Required?
tableWidth : integer	The width of table.	Optional
showRowBorder : boolean = false	True to draw border between rows.	Optional
showColumnBorder : boolean = true	True to draw border between columns.	Optional
showName : boolean	True to show the name of procedure.	Optional
name : string	Filter procedures by name.	Optional

Supported attributes for <BPProcedures>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Procedures" description = "/" type = "table" default = "false"/>
   <ElementBaseInitiationBlock>
      <BPProcedures/>
   </ElementBaseInitiationBlock>
</Template>
```

<TestPlans>

Description
<TestPlans> is a construct that prints in report the test plans of a requirement test case, if specified.

Attributes

Name	Description/Usage	Required?
tableWidth : integer	The width of table.	Optional
showRowBorder : boolean = false	True to draw border between rows.	Optional
showColumnBorder : boolean = true	True to draw border between columns.	Optional

Supported attributes for <TestPlans>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Test Plans" description = "/" type = "table" default = "false"/>
   <ElementBaseInitiationBlock>
      <TestPlans/>
   </ElementBaseInitiationBlock>
</Template>
```

<TestPlanStepIterationBlock>

Description
<TestPlanStepIterationBlock> is a construct that enables you to retrieve the steps within a test plan.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Test Plans" description = "/" type = "table" default = "false"/>
   <TestPlanStepIterationBlock>
      <StringPropertyText propertyText="index" />
      <StringPropertyText propertyText="name"/>
      <StringPropertyText propertyText="expectedResults"/>
   </TestPlanStepIterationBlock>
</Template>
```

<MatrixDiagram>

Description
<MatrixDiagram> is a construct that prints in report the content of matrix diagram.

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Matrix diagram" description = "/" type = "table" default = "false"/>
   <DiagramBaseInitiationBlock>
      <MatrixDiagram/>
   </DiagramBaseInitiationBlock>
</Template>
```

<GridDiagram>

Description

<GridDiagram> is a construct that prints in report a grid’s content.

Attributes

Name	Description/Usage	Required?
tableWidth : integer	The width of table.	Optional

Supported attributes for <GridDiagram>

Sample template fragment

```
<Template>
   <TemplateInformationHeader name = "Grid" description = "/" type = "table" default = "false"/>
   <DiagramBaseInitiationBlock>
      <GridDiagram/>
   </DiagramBaseInitiationBlock>
</Template>
```

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Export and import XML

Interoperability is the ability to exchange information between two systems and to use the information that has been exchanged. This chapter focuses on the interoperability with XML file.

Exporting XML

Shows you how to export project data to XML file.

Importing XML

Shows you how to import XML to an opening project.

Exporting XML

Interoperability is the ability to exchange information between two systems and to use the information that has been exchanged. Agilian supports interoperability with XML file . You can export project data to an XML, manipulate it externally, and feed the changes back to Agilian. In this chapter, you will see how to export XML file of whole project or specific diagram in project.

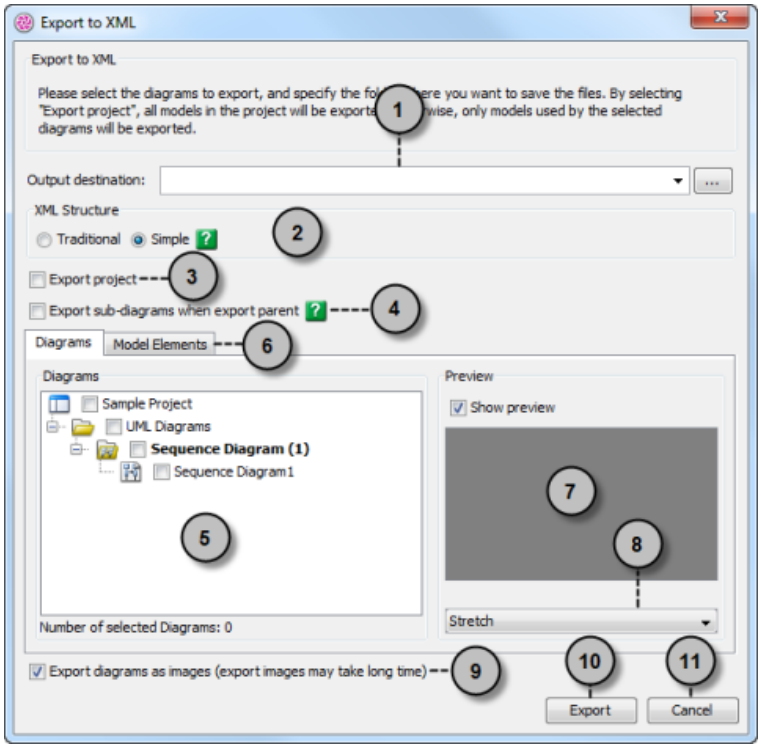
Exporting whole project to XML

1. Select **File > Export > XML...** from the main menu.
2. Specify the output destination.
3. Check **Export project** to export everything in the project to XML, or keep it unchecked, and check the diagram(s) to export only their content.
4. Click **Export**.

Exporting active diagram to XML

1. Right click on diagram and select **Export > Export XML...** from the popup menu.
2. In the **Export to XML** window, specify the output destination, which is a folder for containing the exported XML files and optionally the image files of diagrams.
3. Click **Export**.

Overview of exporting XML



Overview of Export XML dialog box

No.	Name	Description
1	Output destination	The location where you want to save the file.
2	XML structure	Controls how model data is to be presented in the exported XML file. **Traditional**: XML elements are named in more general manner. For example, Model, StringProperty, etc. **Simple**: XML elements are named using the name of the model. For example, UseCase, Frame, etc.
3	Export project	By checking **Export project**, all models in the project will be exported.
4	Export sub-diagrams when export parent	By selecting **Export sub-diagrams when export parent**, the sub-diagram(s) will also be exported when the parent diagram(s) is exported. For example, package *MyWorks* contains diagrams A and B. If you select to export diagram A, its parent "MyWork" will get exported, too. With the option **Export sub-diagrams when export parent** on, B will get exported too since B is a sub-diagram of package *MyWork*s. By turning off the option, B will be ignored when export.
5	Diagram list	A list of diagram of your project. Select the diagrams to export to XML.
6	Model elements	A list of model elements of your project. Select the model elements to export to XML.

7	Preview window	By checking the selected diagram and **Show preview**, it will be shown in preview window.
8	Preview mode	You can choose either **Stretch** or **Real size** to preview your diagram. **Stretch**: The ratio of your diagram will be fit in the size of preview window. **Real size**: The ratio of your diagram will be shown on the preview window as its real size.
9	Export diagrams as images	When checked, image file of the selected diagrams will also be exported along with the exported XML.
10	Export	Click **Export** to proceed with exporting to XML.
11	Cancel	Click **Cancel** to discard exporting to XML.

Description of Export XML dialog box

Exporting diagrams to XML with command line interface

1. Start the command console.
2. Change directory to **%AG_Installation_Dir%\scripts** .
3. Execute **ExportXML** with the parameters required (e.g. ExportXML -project C:\MyProjects\sample.vpp -out C:\MyXML). Here are the available parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	Folder for storing the exported XML and images	C:\Demo\Output
-diagram	One or more diagrams to be exported	"Diagram A" "Diagram B"
-noimage	Do not export image files for diagrams	N/A

Parameters for ExportXML

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing XML

You can import changes made externally in XML back to Agilian , to update the project data. In this chapter, you will see how to import an XML exported before. Notice that XML import is made in response to XML export in Agilian. Only XML exported from Agilian can be imported.

Importing XML to current project

1. Select **File > Import > XML...** from the main menu.
2. Specify the file path of the XML to import.
3. Click **OK**.

> **NOTE:** All changes made in project will be overwritten by data in XML. For example, if class Foo is renamed to Bar. By importing an XML exported before renaming class, Bar will be renamed to Foo.

Importing XML to project with command line interface

1. Start the command console.
2. Change directory to **%AG_Installation_Dir%\scripts**.
3. Execute **ImportXML** with the parameters required (e.g. ImportXML -project C:\MyProjects\sample.vpp -file C:\MyXML\project.xml). Here are the available parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-file	The filepath of the XML file to import	C:\Demo\input\sample.xml

Parameters for ImportXML

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Export and import VP project

Interoperability is the ability to exchange information between two systems and to use the information that has been exchanged. This chapter focuses on the interoperability with native .vpp project file.

Exporting VP project

Shows you how to export project data to project file.

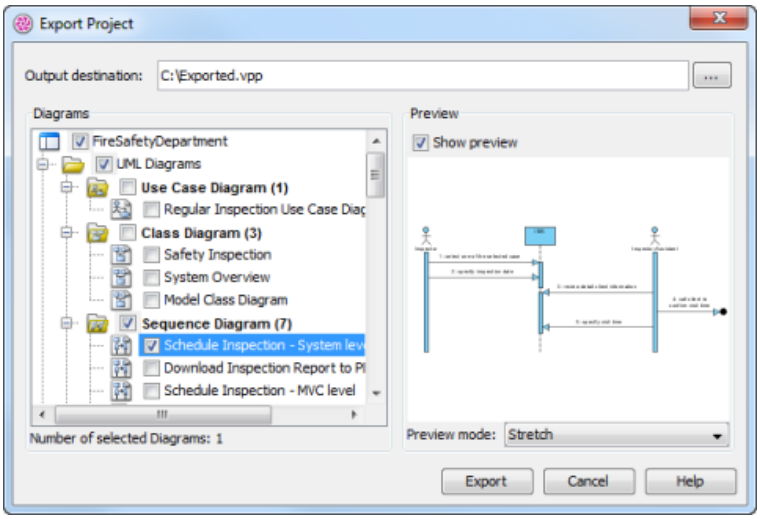
Importing VP project

Shows you how to import project file to an opening project.

Exporting VP project

Interoperability is the ability to exchange information between two systems and to use the information that has been exchanged. Agilian supports interoperability with native .vpp project file. You can export some of the diagrams to a project file, send to your team member for editing, and feed the changes back to Agilian. In this chapter, you will see how to export project file for diagrams in project. To export VP project:

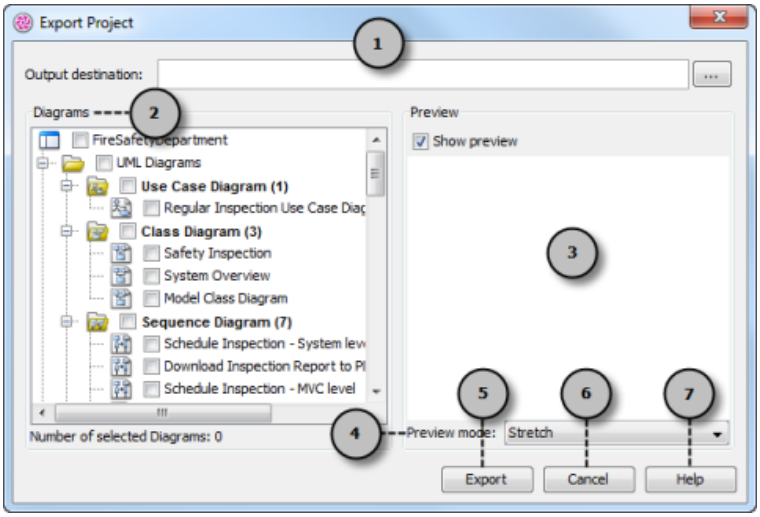
1. Select **File > Export > Agilian Project...** from the main menu.
2. Specify the output destination.



Inputting output destination

3. Check in the diagram tree the diagrams to export. If you want to export the whole project, check the top most root node.
4. Click **Export**. Upon finishing, you can visit the output destination specified to obtain the .vpp project file.

Overview of exporting VP project



*Overview of **Export Project** dialog box*

No.	Name	Description
1	Output destination	The location where you want to save the file.
2	Diagrams	All diagrams in your project will be shown in here.
3	Preview window	By checking the selected diagram and Show preview, it will be shown in preview window.
4	Preview mode	You can choose either **Stretch** or **Real size** to preview your diagram. **Stretch**: The ratio of your diagram will be fit in the size of preview window. **Real size**: The ratio of your diagram will be shown on the preview window as its real size.
5	Export	Click **Export** to proceed with exporting to VP project.
6	Cancel	Click **Cancel** to discard exporting to VP project.
7	Help	More information about how to export VP Project can be obtained by clicking this button.

*Description of **Export Project** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing VP project

You can import a VP project to Agilian for importing changes made in an exported project, or to import model in another project. In this chapter, you will see how to import a VP project. To import a project:

1. Select **File > Import > Agilian Project...** from the main menu.
2. Select the file path of the .vpp file to import in file chooser.
3. Click **Open**.

> **NOTE:** All changes made in project will be overwritten by imported project.
> For example, if class Foo is renamed to Bar. By importing a project exported before renaming class, Bar will be renamed to Foo.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Export and import Microsoft Excel

Interoperability is the ability to exchange information between two systems and to use the information that has been exchanged. This chapter focuses on the interoperability with Microsoft Excel file.

Exporting to Microsoft Excel

Shows you how to export project data to Excel file.

Importing Microsoft Excel file

Shows you how to import Excel file to an opening project. (The Excel must be exported from Agilian through the export feature)

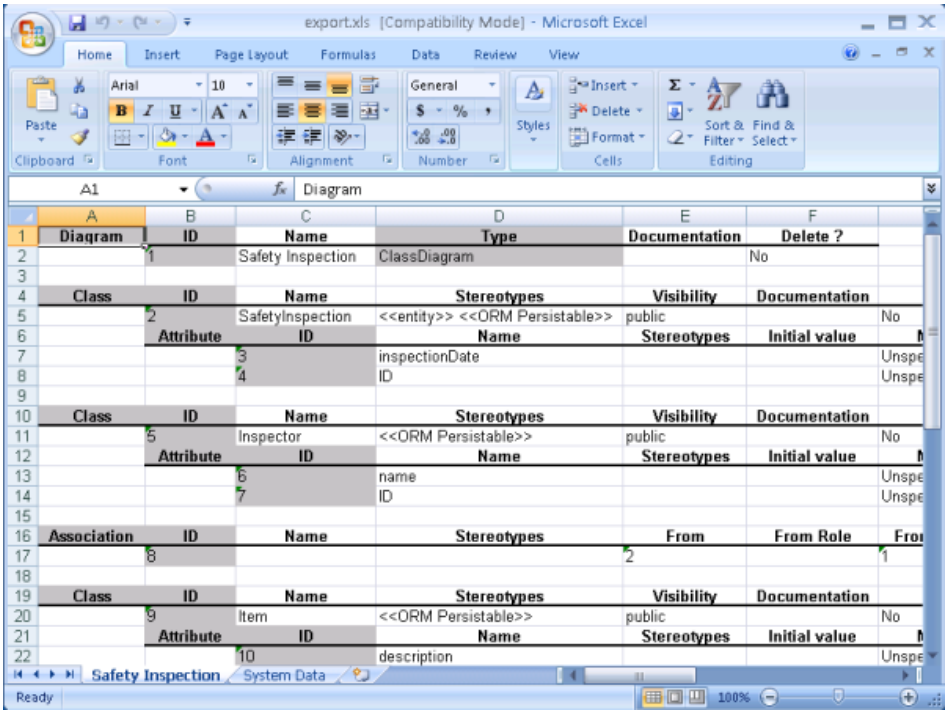
Excel modification guidelines

Non-desired changes made in Excel may damage its structure and cause import failed to work. This page outlines some of the points you need to pay attention to when modifying an exported Excel file.

Exporting diagrams to Microsoft Excel format

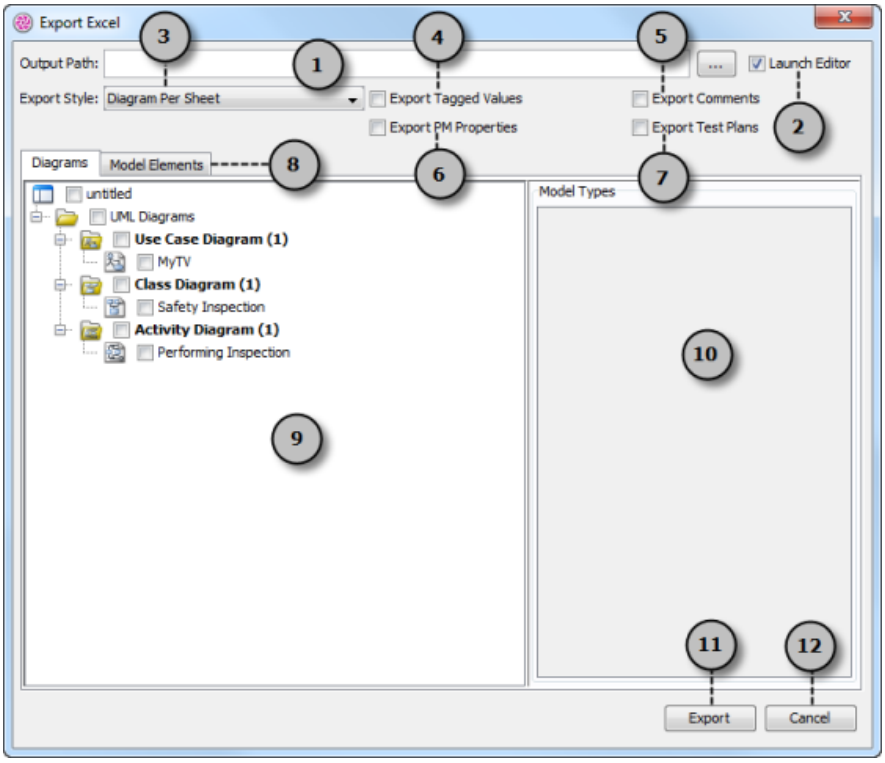
Interoperability is the ability to exchange information between two systems and to use the information that has been exchanged. Agilian supports interoperability with Microsoft Excel file. You can export project data to an excel file, make changes in exported Excel's worksheets, or further processing like to query data with formula, and eventually feed the changes back to Agilian. In this chapter, you will see how to export an Excel file from a project. To export Excel:

1. Select **File > Export > Excel...** from the main menu.
   This opens the **Export Excel** dialog box.
2. Specify the output path of Excel file.
3. Select the diagrams to export to Excel.
4. On the **Model Types** list at the right hand side, you can optionally select the type(s) of model elements to be exported.
5. Click **Export**. Upon finishing, you can visit the output destination specified to obtain the Excel file.



The exported Excel file

An overview of Export Excel dialog box



*Overview of the **Export Excel** dialog box*

No.	Field	Description
1	Output Path	The file path of the Excel file.
2	Launch Editor	Check to open the exported Excel file after exported.
3	Export Style	Determine how data will be presented in Excel. **Diagram Per Sheet** - Selected diagrams will be exported to separate sheets. **All in One Sheet** - Selected diagrams will be exported to a single sheet. **Model Type Per Sheet** - Selected model elements will be exported to separate sheets, grouped by type. **Raw** - Only one sheet will be generated with all elements listed in it. All the properties of selected elements will be listed as columns, which each row represents a model element.
4	Export Tagged Values	Tagged values can be added to model elements to specify domain specific properties. You can add tagged values in the specification dialog box of model element. Check **Export Tagged Values** if you want to export tagged values of model elements.
5	Export Comments	You can add your own comments to model elements in their specification dialog box. Check **Export Comments** if you want to export the comments to the Excel file.
6	Export PM Properties	PM properties is the short form of project management properties. You can set project management properties, such as version, phrase, author, in the specification dialog box of a model element. Check **Export PM Properties** if you want to export PM properties of model elements.
7	Export Test Plans	Test plan refers to the test plans that can be specified for Test Case elements in requirement diagram. Check **Export Test Plans** to export the information of test plan to a separate sheet in Excel file.
8	Model Elements	Select this tab and check the model elements you want to export to Excel.
9	Diagrams	All diagrams in the opening project are listed here. Select the diagrams to export to Excel.
10	Model Types	When you have updated the diagram selection in **Diagrams** list, the **Model Types** list will be updated to list the types of diagram elements that appear in the chosen diagrams. By default, all diagram element types are checked, meaning that diagram elements in those types will be exported to Excel. You can uncheck type(s) to ignore certain type(s) of diagram elements when exporting.
11	Export	Click **Export** to proceed with exporting Excel.
12	Cancel	Click **Cancel** to diacard exporting to Excel.

*Overview of the **Export Excel** dialog box*

Exporting diagrams to Microsoft Excel with command line interface

1. Start the command console.
2. In the console, change directory to the scripts folder and execute **ExportExcel** with parameters required.

Parameters for ExportExcel

This displays the usage of the command. Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	The filepath of the Excel file to import	C:\Demo\Output\Sample.xls

Parameters for ExportExcel

Upon finishing, you can visit the output destination specified to obtain the Excel file.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

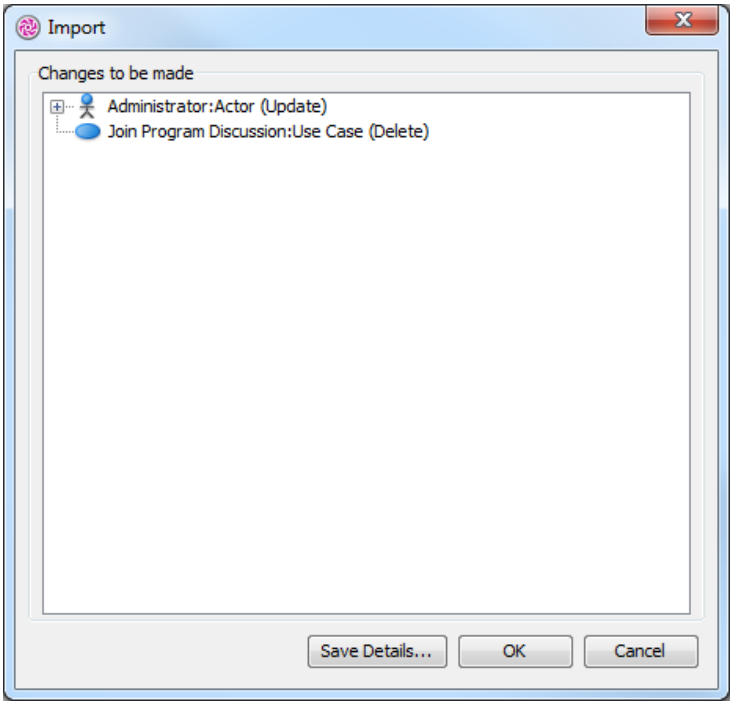
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing Microsoft Excel file

You can import changes made externally in Excel file back to Agilian , to update the project data. In this chapter, you will see how to import an Excel exported before. Notice that Excel import is made in response to Excel export in Agilian. Only Excel exported from Agilian can be imported.

Importing an Excel file

1. Select **File > Import > Excel...** from the main menu.
2. In the file chooser, select the Excel file to import and click **Open** to confirm.
3. In the **Import** dialog box, you can preview the changes you have made in the previous Excel file. If you want to keep a record of changes, click on **Save Details...** at the bottom of dialog box to export the changes to an Excel file.

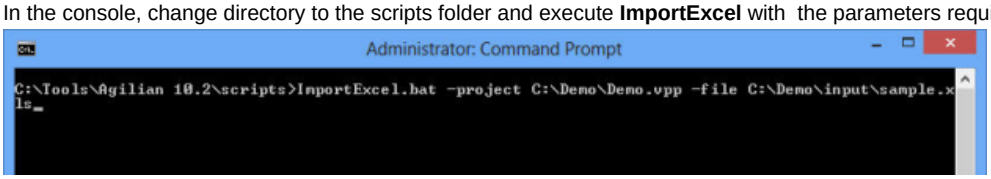


***Import** dialog box*

4. Click **OK** to proceed.

Importing Excel to project with command line interface

1. Start the command console.
2. In the console, change directory to the scripts folder and execute **ImportExcel** with the parameters required.



Parameters for ImportExcel

This displays the usage of the export command. Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-file	The filepath of the Excel file to import	C:\Demo\input\sample.xls

Parameters for ImportExcel

Upon finishing, the project file will be updated with the data presented in the XML file.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Excel modification guidelines

When you are going to modify Excel file exported from Agilian , make sure the changes you make are all valid. Making invalid changes will cause the file cannot be imported back to Agilian. In this chapter, we will go through some of the points that you need to pay attention to when editing.

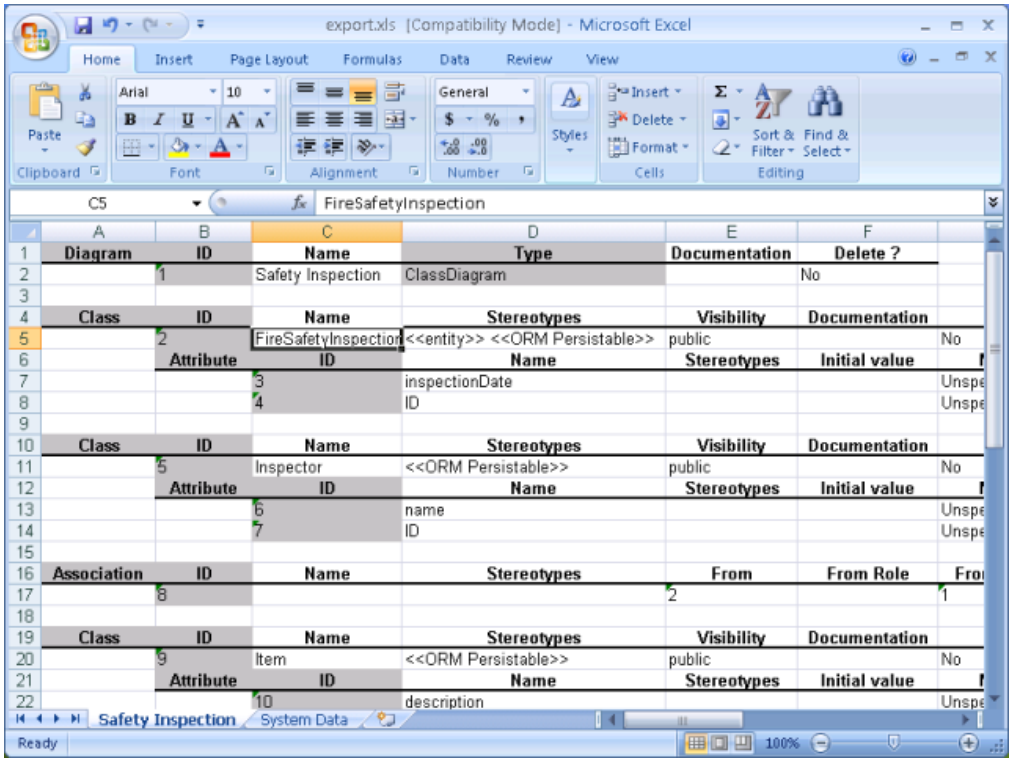
Caution

When you start to modify the Excel, pay attention to the following points to avoid having problems when importing the file back to project:

- Do NOT modify the gray cells
- Do NOT just delete a row for deleting a model element. Instead, change the value *No* to *Yes* under the **Delete?** column.
- Do NOT modify the System Data sheet

Renaming a model element

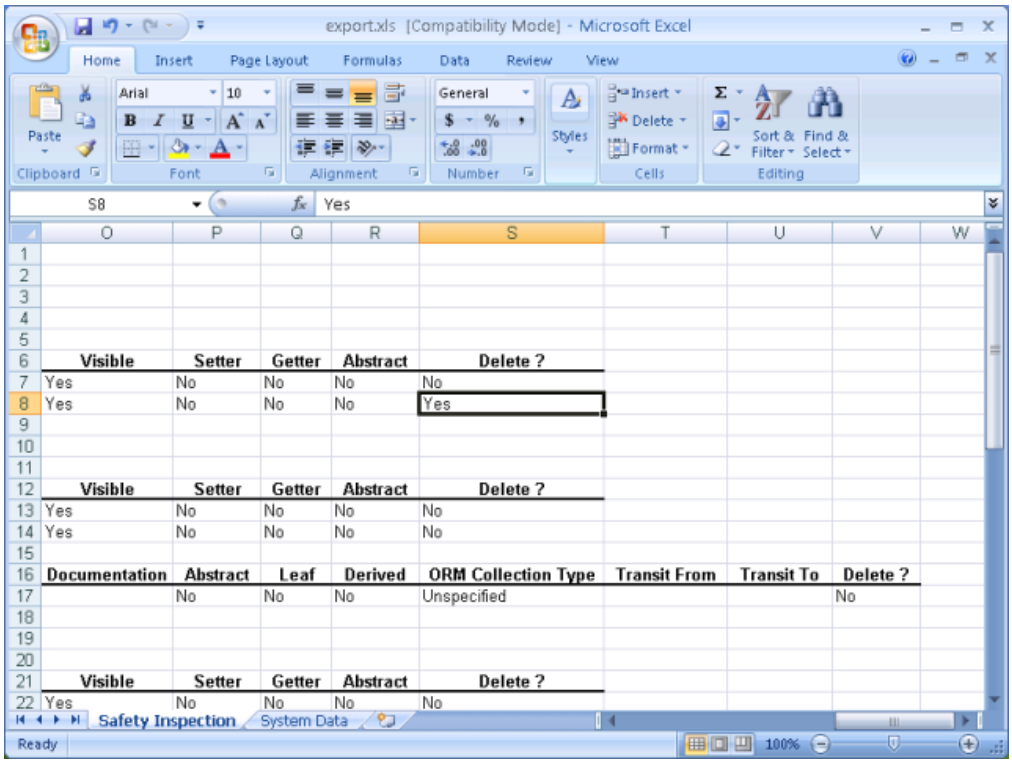
Double click on a cell and enter a new name.



Renaming class in Excel

Deleting a model element

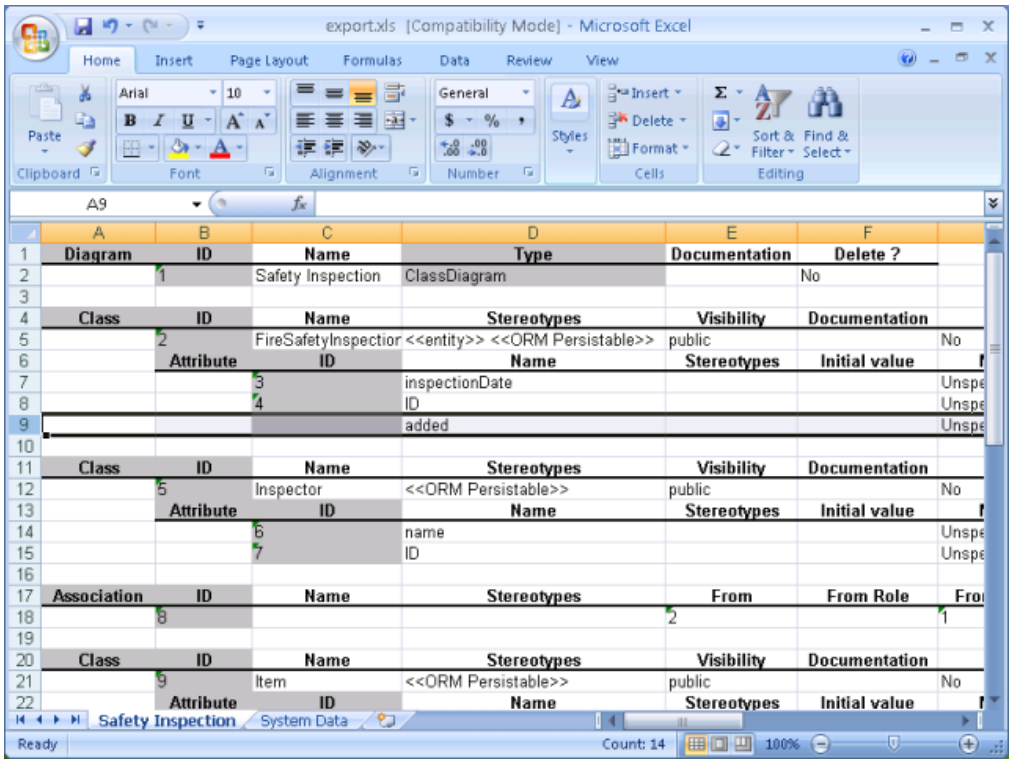
To delete a model element, change *No* to *Yes* under the **Delete?** column. Do NOT delete the row.



Deleting attribute in Excel

Adding a model element

Suppose you want to add an attribute, select the last attribute row and insert a row in Excel, right under the last one. Then, start editing it. The gray cells can leave blank.



Adding attribute in Excel

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Export and import XMI

Interoperability is the ability to exchange information between two systems and to use the information that has been exchanged. This chapter focuses on the interoperability with XMI file.

Exporting XML

Shows you how to export project data to XMI file.

Importing XML

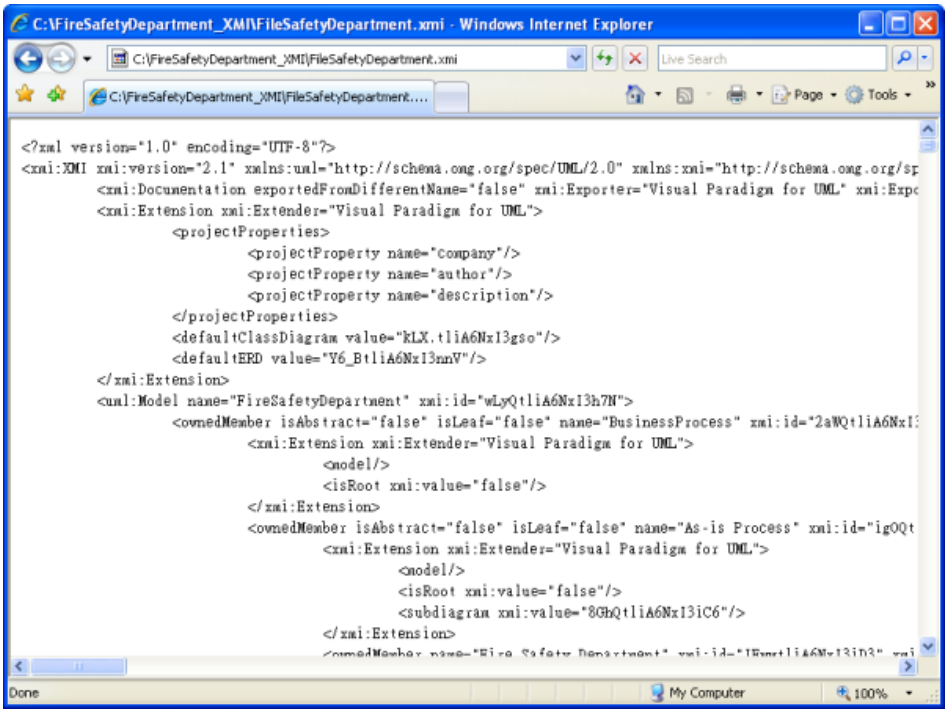
Shows you how to import XMI to an opening project.

Exporting XMI

Interoperability is the ability to exchange information between two systems and to use the information that has been exchanged. Agilian supports interoperability with XMI file , a standard made for data exchange. You can export project data to an XMI, edit it externally with other softwares that accepts XMI. In this chapter, you will see how to export XMI file.

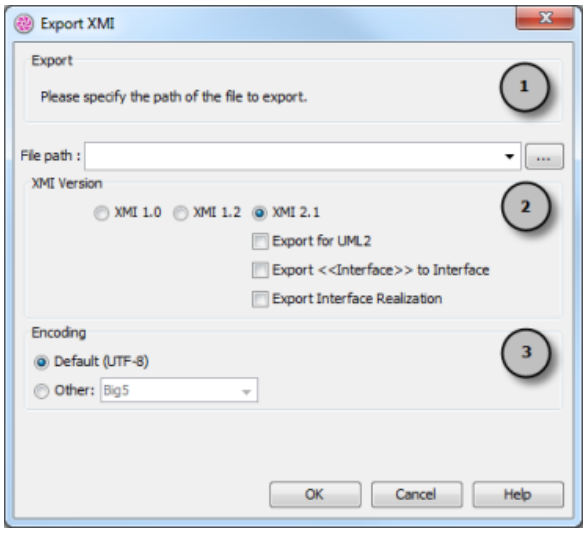
Exporting project to XMI

1. Select **File > Export > XMI...** from the main menu.
2. Specify the file path of the XMI file.
3. Configure the exporting such as setting the XMI version and changing the encoding of XMI.
4. Click **OK** to start exporting. Upon finishing, you can visit the output destination specified to obtain the XMI.



Review exported XMI

An overview of Export XMI window



*An overview of the **Export XMI** window*

No.	Name	Description
1	File path	The file path for the XMI file to export.
2	XMI version	There are three versions of XMI to suit different interoperability needs, including 1.0, 1.2 and 2.1. If the latest version 2.1 is selected, the following options are then available for selection: • **Export for UML 2** - By checking this option, the exported XMI will conform to the tool UML2's standard. The following options will appear in further.

- **Export Data Type to** - Determine whether to export data type to UML or Ecore primitive type
- **Export Java Annotation to EAnnotation** - Determine whether to export Java annotation to EAnnotation

- **Export <<Interface>> to Interface** - Determine whether to export stereotype "interface" as stereotype, or a segment of interface element.
- **Export Interface Realization** - Determine whether to keep exporting realization between interface and concrete class as realization, or export it as interface realization.

3	Encoding	Select the encoding of XMI file.

*Description of **Export XMI** dialog box*

Exporting diagrams to XMI with command line interface

1. Start the command console.
2. Change directory to **%AG_Installation_Dir%\scripts** .
3. Execute **ExportXMI** with the parameters required (e.g. ExportXMI -project C:\MyProjects\sample.vpp -out C:\myproject.xmi). Here are the available parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	The filepath of XMI file	C:\Demo\Output\sample.xmi
-type [optional]	Version of XMI. Unless specified, the lastly generated version will be selected. Here are the possible options: • 1.0 • 1.2 • 2.1 • 2.1UML2	2.1
-encoding [optional]	Encoding of XMI file	UTF-8

Parameters for ExportXMI

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing XMI

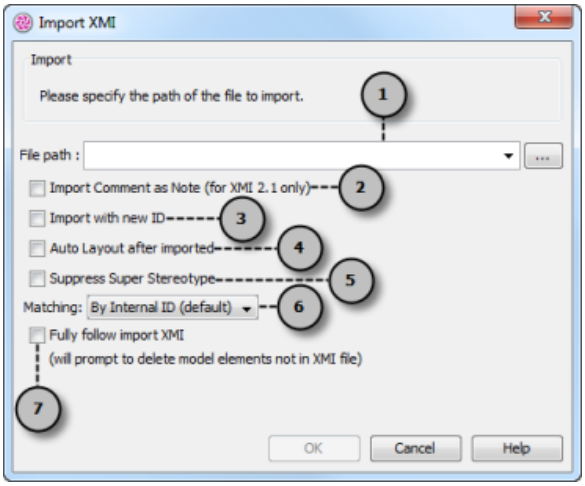
You can migrate your works done in another software to Agilian through XMI, provided that the software supports XMI. In this chapter, you will see how to import an XMI file.

Importing XMI to current project

1. Select **File > Import > XMI...** from the main menu.
2. Specify the file path of the XMI to import, and configure the import if necessary.
3. Click **OK** button.

> **NOTE:** All changes made in project will be overwritten by data in XMI. For example, if class Foo is renamed to Bar. By importing an XMI exported before renaming class, Bar will be renamed to Foo.

An overview of Import XMI window



An overview of Import XMI window

No.	Name	Description
1	File path	The file path of the XMI file to import.
2	Import Comment as Note (for XMI 2.1 only)	Determines whether to import comment as documentation, or as note of model.
3	Auto Layout after imported	Determines whether to run a layout on diagram after import. Note that running layout may takes time for a massive amount of diagram data.
4	Suppress Super Stereotype	Determines whether to ignore super stereotype when importing.
5	Matching	The importing of XMI is a procedure to merge the data in XMI into the opening project. The matching option is to determine how to match between data in XMI and the opening project during export, and to perform changes accordingly.

*Description of **Import XMI** dialog box*

Importing XMI to project with command line interface

1. Start the command console.
2. Change directory to **%AG_Installation_Dir%\scripts**.
3. Execute **ImportXMI** with the parameters required (e.g. ImportXMI -project C:\MyProjects\sample.vpp -file C:\myproject.xmi). Here are the available parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-file	The filepath of the XMI file to import	C:\Demo\input\sample.xmi

Parameters for ImportXMI

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Importing Visio drawing

You can import your previous work drawn in Visio to Agilian through the import feature, and continue modeling in Agilian. This chapter shows you how to do.

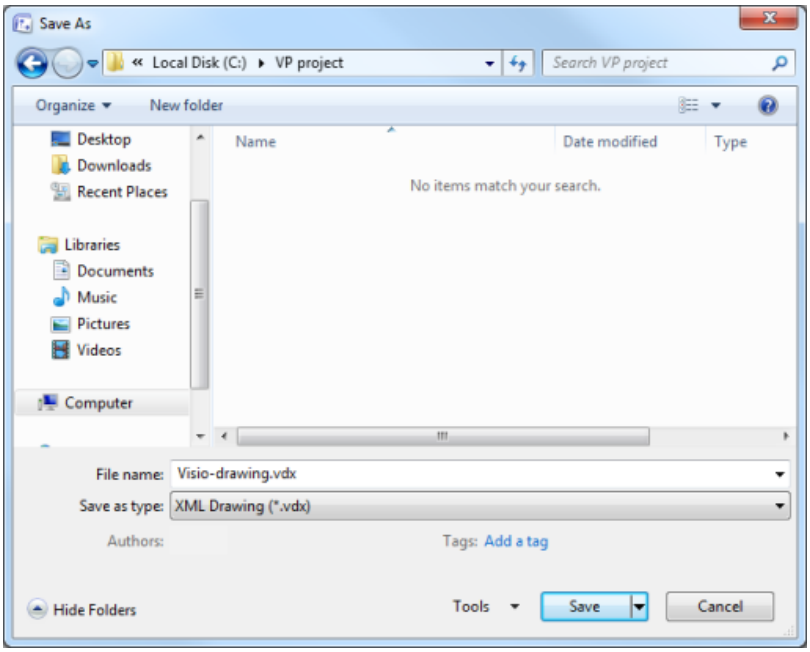
Importing Visio drawing

Outlines the steps involved in importing an Visio drawing.

Importing Visio drawing

After you have drawn a diagram in Visio, you can now import your work through the import feature.

1. Save your Visio drawing as a .vdx file.

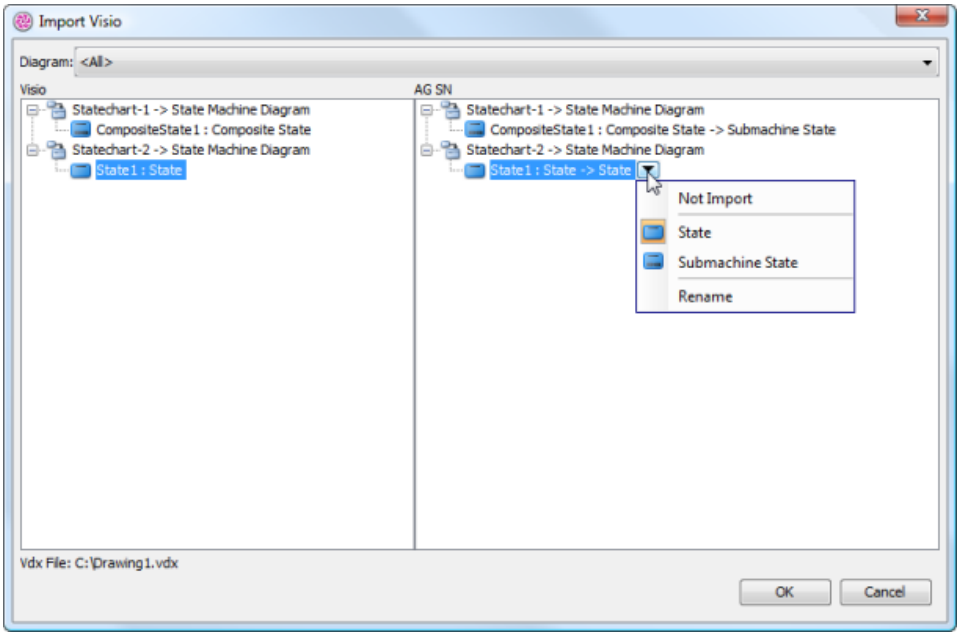


Save Visio drawing

2. To import a Visio drawing into Agilian , select **File > Import > Visio...** from the main menu.
3. Specify the file path of the Visio drawing.

> **NOTE:** Only valid XML Drawing (*.vdx) can be imported.
> If your Visio drawing does not have a .vdx as extension, open it in Visio and save it as a .vdx file.

4. Click **OK** to start importing. This popup another **Import Visio** dialog box. As the model structure is different among Visio and Visual Paradigm, this dialog enables users to resolve inconsistency between Visio and Agilian.



The Import Visio dialog box

The left hand side of the dialog box represents the structure of Visio drawing, while the right hand side represents the expected outcome in Agilian through importing. Users can perform the following actions in this dialog box.

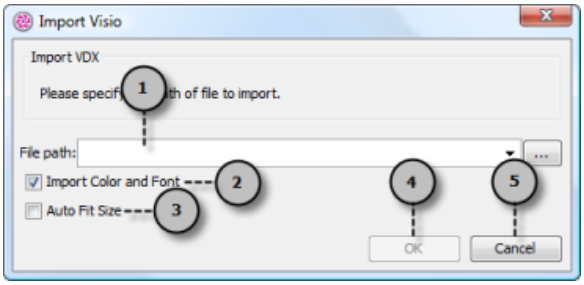
Action	Description and steps
Not to import a shape	Click on the button beside the shape node and select **Not Import** in the popup menu.
Rename a shape when importing	Click on the button beside the shape node and select **Rename** in the popup menu. Then, enter the new name of shape and press the **Enter** key to confirm renaming/.

Reset the shape to another type	Click on the button beside the shape node and select an appropriate shape type to reset to.

Description of available import options on individual shape

5. Click **OK** when the import is configured. The drawings can then be accessible in the **Diagram Navigator**. You can then double click on the diagram node to open the diagram.

An overview of Import Visio dialog box



*An overview of **Import Visio** dialog box*

No.	Name	Description
1	File path	The path of Visio .vdx file to be imported.
2	Import Color and Font	By selecting this option, colors and fonts of the shapes to be imported will remain unchanged. Otherwise, Visual Paradigm's default settings will be applied.
3	Auto Fit Size	By selecting this option, shapes' size will be optimized to their minimum possible size. Otherwise, the original size of the imported shapes will remain unchanged.
4	OK	Click to import.
5	Cancel	Click to cancel importing.

*Description of **Import Visio** dialog box*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing Rational Rose model

You can import your previous work drawn in Rose to Agilian through the import feature, and continue modeling in Agilian. This chapter shows you how to do.

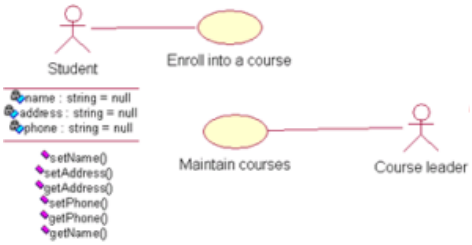
Importing Rational Rose Model into Agilian

Outlines the steps involved in importing a Rational Rose model file.

Importing Rational Rose model

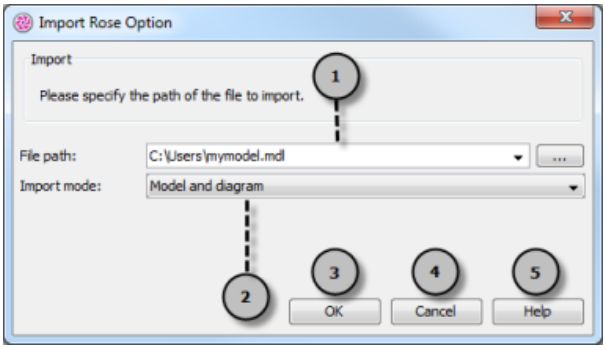
Rational Rose is one of the most widely used UML CASE Tool in the 90's. Visual Paradigm supports the importing of Rational Rose model . With this, users can import legacy design made in Rose into Agilian, with all the model data as well as formatting retained.

1. Save your work in Rose.



A Rose drawing

2. To import a Rose model into Agilian, select **File > Import > Rose Project...** from the main menu.
3. Specify the file path of the Rose model.



Specifying Rose model path

No.	Name	Description
1	File path	The path of Rose .mdl file to be imported.
2	Import mode	You can choose the mode to be imported. **Model only**: By selecting this option, only the model elements (e.g. Actor, Use Case, Class, etc.) will be imported. NO diagrams will be imported. **Model and diagram**: By selecting this option, both model elements and diagrams will be imported.
3	OK	Click to import.
4	Cancel	Click to cancel importing.
5	Help	Click to obtain more information from the help system.

Description of import rose properties

4. Click **OK** to start importing.
5. When import is completed, click **Close** to close the progress dialog box. If **Model and diagram** was set for **Import mode**, the drawings can then be accessible in the **Diagram Navigator**.

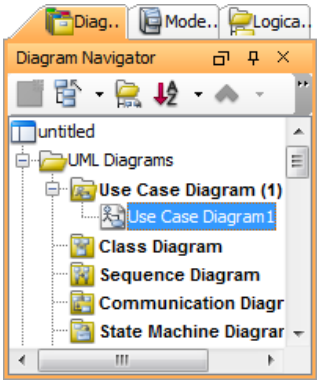
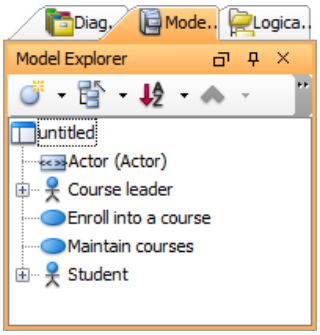


Diagram Navigator listing the imported diagram(s)

Model element can be accessed in the **Model Explorer**. User can form diagrams with them by dragging and dropping them onto diagrams.



Model Explorer listing the imported model element(s)

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing Rational Software Architect File

You can import your previous work drawn in Rational Software Architect to Agilian through the import feature, and continue modeling in Agilian. This chapter shows you how to do.

Importing Rational Software Architect EMX

Outlines the steps involved in importing a .emx file.

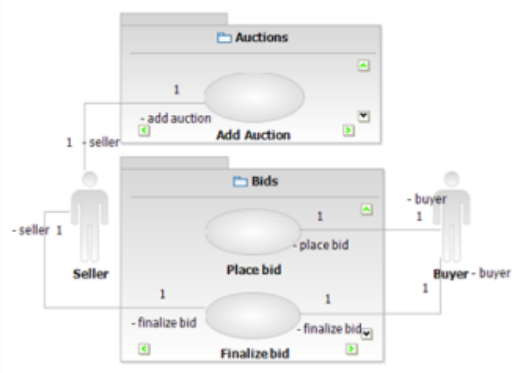
Importing Rational Software Architect DNX

Outlines the steps involved in importing a .dnx file.

Importing Rational Software Architect EMX

Rational Software Architect (RSA) is a modeling and development environment, which leverages UML for architectural design for C++ and Java 2 Enterprise Edition (Java2EE) applications and web services. Import of the RSA file, i.e. the .emx file, is supported in Agilian so that users can simply migrate the work from RSA to VP, also perform further modeling on the imported models in the VP tool.

1. Save your work in Rational.



A Rational drawing

2. To import a Rational model into Agilian, select **File > Import > Rational Model...** in the main menu of Agilian.
3. In the **Import Rational Software Architect UML Model** dialog box, specify the file path of the .emx file and click **OK**.
4. The drawings can then be accessible in the **Diagram Navigator**. You can then double click on the diagram node to open the diagram.

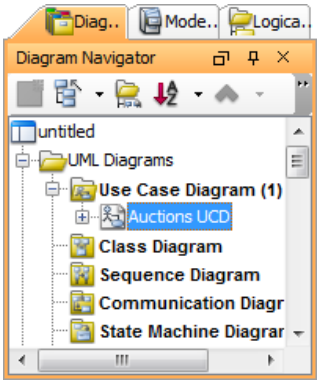


Diagram Navigator listing the imported diagram(s)

Related Resources

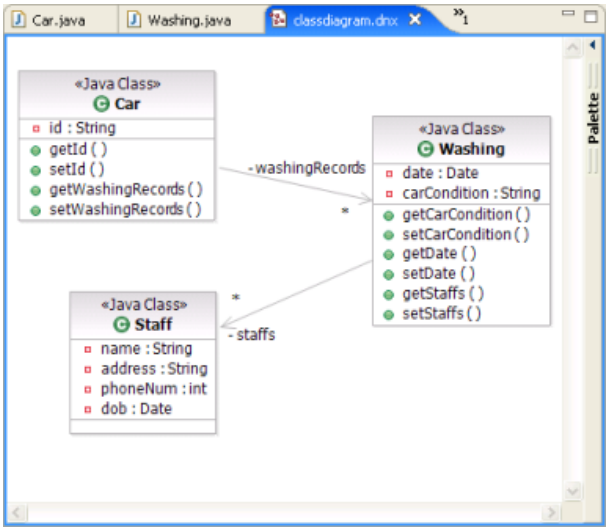
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing Rational Software Architect DNX

Agilian supports importing drawing drawn in Rational Software Architect with a .dnx extension. By importing a drawing, all diagrams, shapes and model information will be imported.

1. Save the drawing in Rational Software Architect.



A Rational drawing

2. To import the drawing into Agilian, select **File > Import > Rational DNX...** from the main menu.
3. In the **Import Rational Diagram DNX** dialog box, specify the file path of the .dnx file and click **OK** button.
4. After importing is completed, you can then double click on the diagram node to open the diagram.

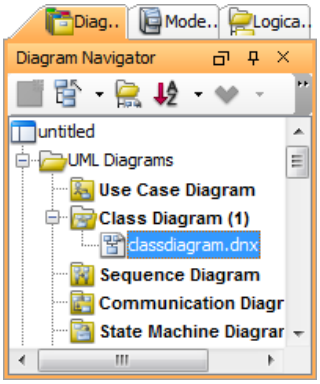


***Diagram Navigator** listing the imported diagram(s)*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing ERwin data modeler project

You can import your previous work drawn in ERwin Data Modeler to Agilian through the import feature, and continue modeling in Agilian. This chapter shows you how to do.

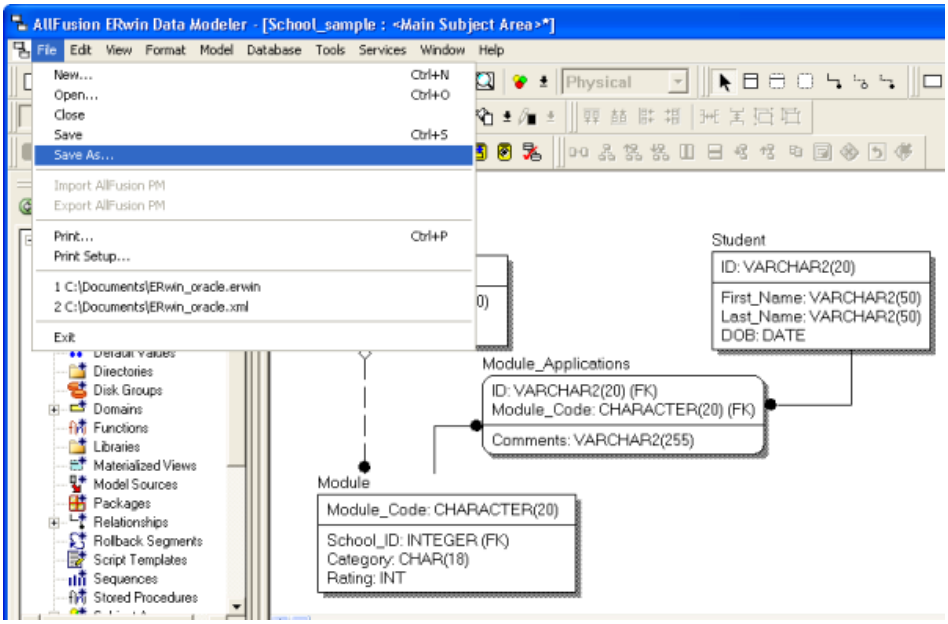
Importing ERwin data modeler project

Outlines the steps involved in importing an ERwin project.

Importing ERwin data modeler project

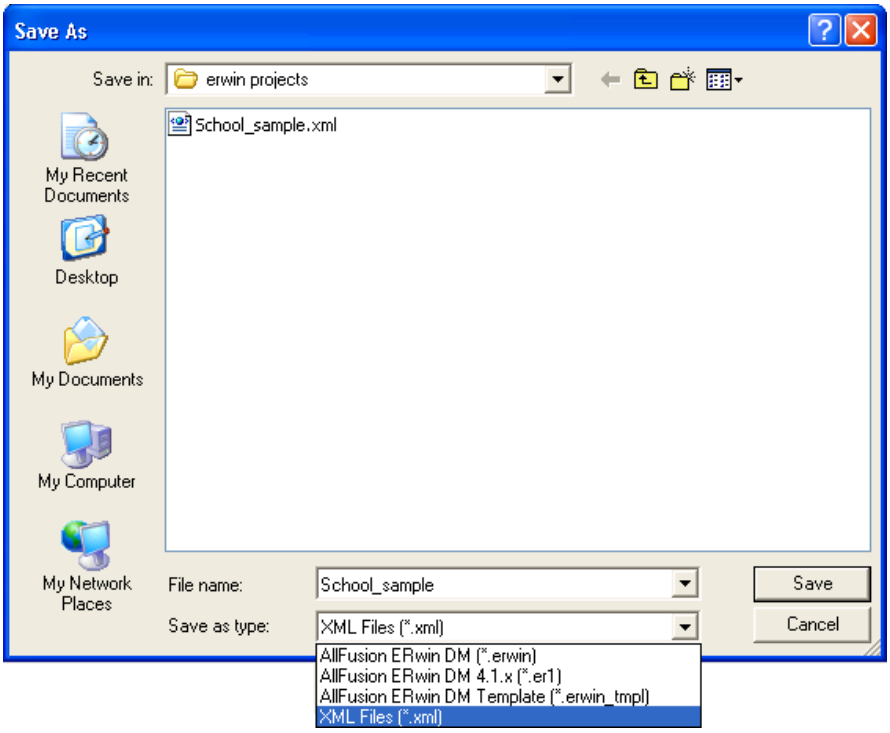
AllFusion ERwin Data Modeler is a popular tool for data modeling. You can import ERwin diagrams and entity models into Agilian with all properties preserved.

1. Here is a ERwin Data Modeler Project. In order to let Agilian import it, you need to save it as an XML file. Select **File > Save As...** from the menu.



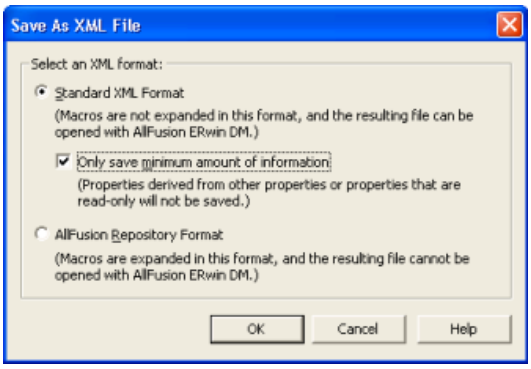
To save an ERwin Data Modeler project as XML

2. Select **XML Files (*.xml)** in **Save as type** and enter the file name in the **Save As** dialog box.



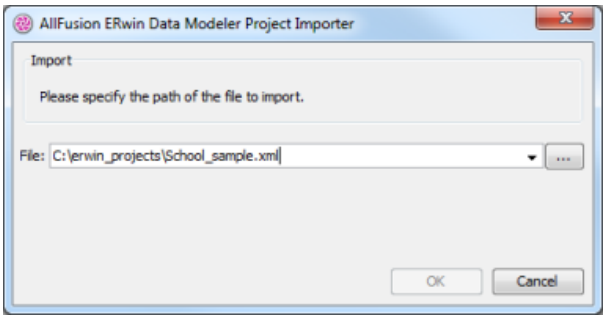
Save the ERwin Data Modeler project as XML

3. Click **Save**. This popup the **Save as XML File** dialog box.
4. Keep using the default settings **Standard XML Format** and **Only save minimum amount of information**.



Exporting the XML in standard XML format

5. Click **OK** to confirm. This saves an XML file that can be used for importing into Agilian.
6. To import an ERwin Data Modeler project into Agilian, select **File > Import > ERwin Project(XML)...** from the main menu.
7. Specify the file path of the XML file.



Specifying ERwin Data Modeler project file path

8. Click **OK** to start importing. When import is completed, the **Open Imported Entity Relationship Diagram(s)** dialog box will appear.
9. Select the diagram(s) to open and click **Open** to open them. The drawings will then be opened.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing Telelogic Rhapsody and System Architect project file

You can import your previous work drawn in Telelogic Rhapsody or Telelogic System Architect to Agilian through the import feature, and continue modeling in Agilian. This chapter shows you how to do.

Importing Telelogic Rhapsody project

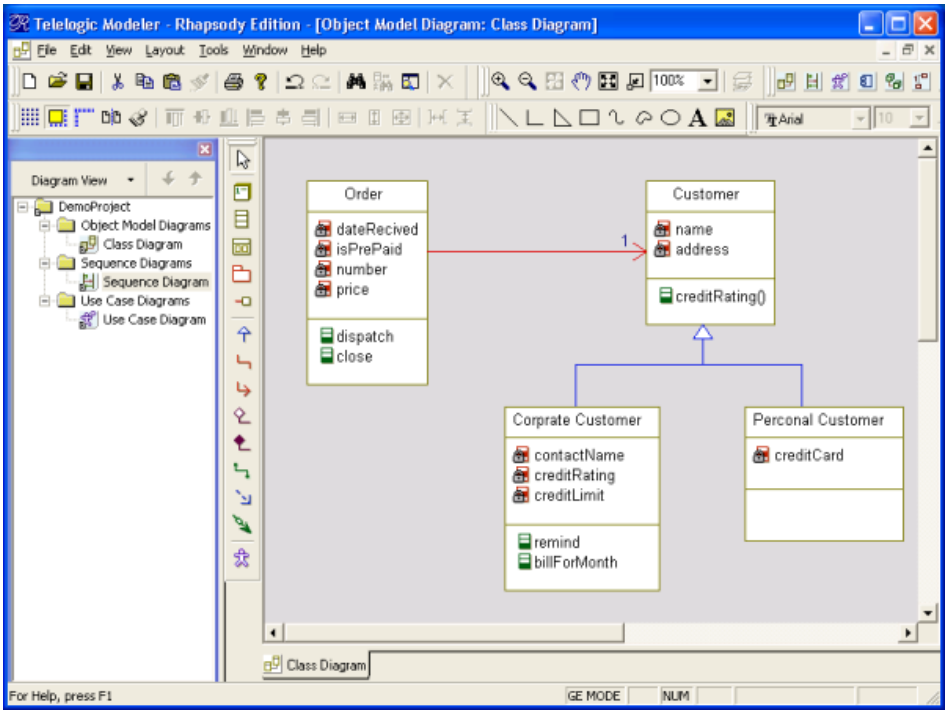
Outlines the steps involved in importing a Telelogic Rhapsody file.

Importing Telelogic System Architect

Outlines the steps involved in importing a Telelogic System Architect file.

Importing Telelogic Rhapsody project

1. Save your Telelogic Rhapsody Project.



A drawing in Telelogic Modeler

2. To import a Telelogic Phapsody project into Agilian, select **File > Import > Telelogic Rhapsody Project ...** from the main menu.
3. Specify the file path of the Telelogic Rhapsody Project in the pop-up **Import Rhapsody** dialog box.
4. Click **OK** when the import is configured. The drawings can then be accessible in the **Diagram Navigator**. You can then double click on the diagram node to open the diagram.

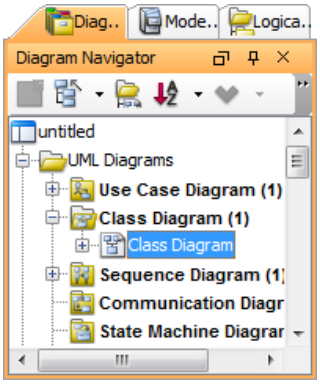


Diagram Navigator listing the imported diagram(s)

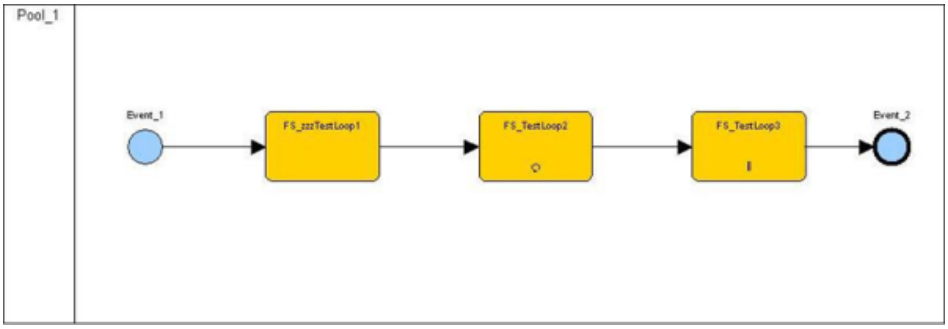
Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

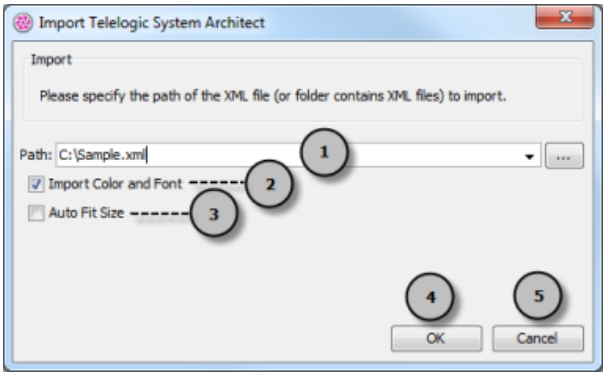
Importing Telelogic System Architect

1. Save your Telelogic System Architect drawing:



A Telelogic System Architect drawing

2. To import a Telelogic System Architect project into Agilian, select **File > Import > Telelogic System Architect ...** from the main menu.
3. Specify the file path of the Telelogic System Architect project.



Specifying Telelogic System Architect path

No.	Name	Description
1	File path	The path of Telelogic System Architect .xml file to be imported.
2	Import Color and Font	By selecting this option, colors and fonts of the shapes to be imported will remain unchanged. Otherwise, Visual Paradigm's default settings will be applied.
3	Auto Fit Size	By selecting this option, shapes' size will be optimized to their possible minimum size. Otherwise, the original size of the imported shapes will remain unchanged.
4	OK	Click **OK** to proceed with importing Telelogic System Architect.
5	Cancel	Click **Cancel** to discard importing XML.

*Description of **Import Telelogic System Architect** dialog box*

4. Click **OK** to start importing. When import is completed, the message pane will popup, with a notification appear in it. The drawings can then be accessible in the **Diagram Navigator**. You can then double click on the diagram node to open the diagram.

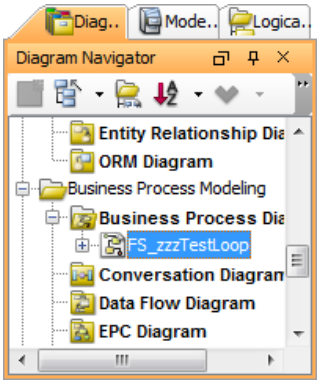


***Diagram Navigator** listing the imported diagram(s)*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian

- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing NetBeans 6.x UML diagrams

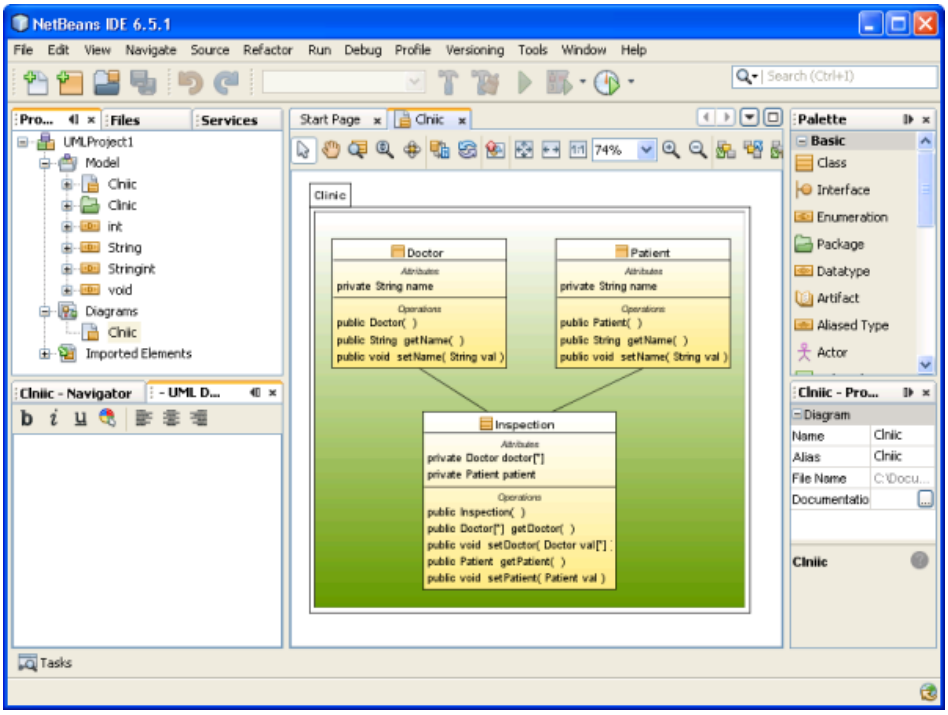
You can import the UML diagrams you previously drawn in NetBeans 6.x to Agilian through the import feature, and continue modeling in Agilian. This chapter shows you how to do.

Importing NetBeans 6.x UML Diagrams into Agilian

Outlines the steps involved in importing NetBeans 6.x UML diagrams.

Importing NetBeans 6.x UML diagrams

1. Here is a NetBeans UML Diagram:



A Class Diagram drew in NetBeans

2. To import a NetBeans UML project into Agilian, select **File > Import > NetBeans UML Project...** from the main menu.
3. In the **Import NetBeans UML Project** dialog box, specify the file path of the NB project and click **OK** button.
4. The drawings can then be accessible in the **Diagram Navigator**. You can then double click on the diagram node to open the diagram.

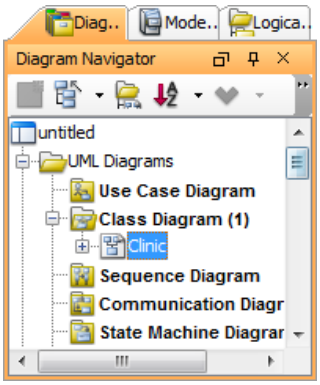
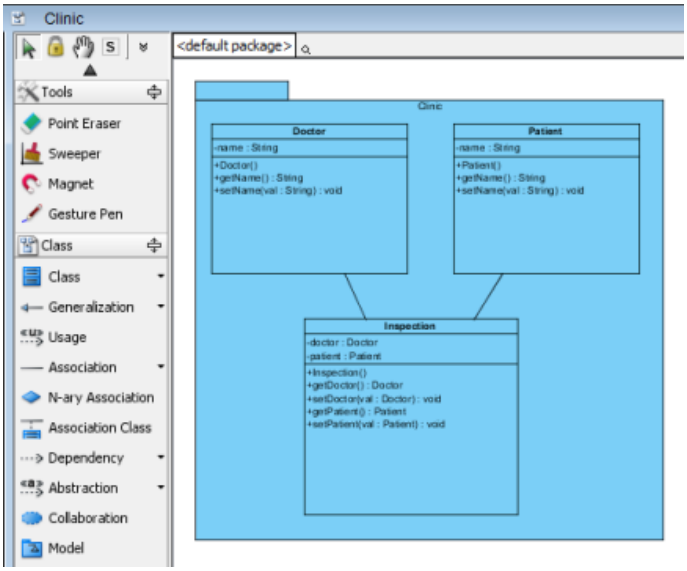


Diagram Navigator listing the imported diagram(s)

You can then double click on the diagram node to open the diagram.



A Class Diagram imported from NetBeans UML Project

NOTE: Due to different ways in presenting diagrams in Agilian and NetBeans, the imported shapes may be bigger than normal.
To fit a shape's size, move the mouse cover over it and press on the resource icon at the bottom right of shape.
To fit size for all shapes, right click on the diagram background and select **Diagram Content > Auto Fit Shapes Size** in the popup menu.

Related Resources
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Importing BizAgi

You can import your previous work drawn in BizAgi to Agilian through the import feature, and continue modeling in Agilian. This chapter shows you how to do.

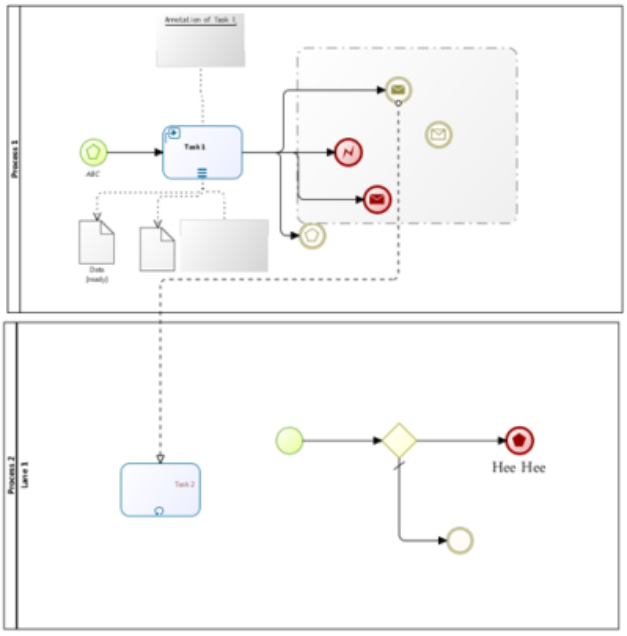
Importing BizAgi

Outlines the steps involved in importing a BizAgi project.

Importing BizAgi

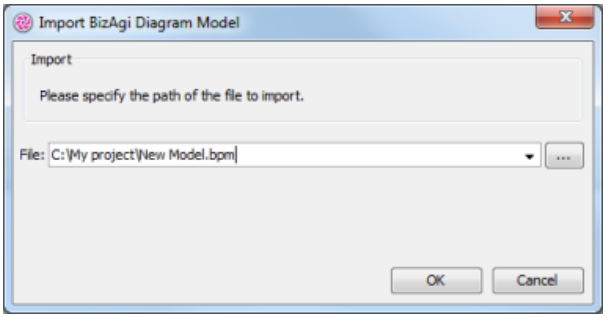
BizAgi is one of the BPM softwares in the market. With Agilian, you can import BizAgi's .bpm files into Agilian. Importing BizAgi is as simple as migrating the work from BizAgi to Agilian. You can perform further modeling on the imported models in Agilian when necessary.

1. Save your BizAgi drawing.



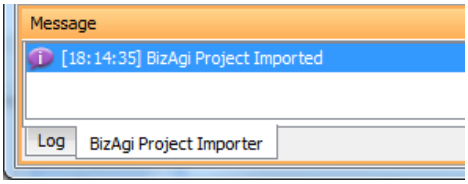
A BizAgi drawing

2. To import a BizAgi project into Agilian, select **File > Import > BIZ AGI ...** from the main menu.
3. Specify the file path of the BizAgi project.



Specify BizAgi model path

4. Click **OK** to start importing.
5. It will notify you that your BizAgi file is imported successfully in **Message** pane.



Successful import message

6. The drawing can then be accessible in the **Diagram Navigator**. You can then double click on the diagram node to open the diagram.

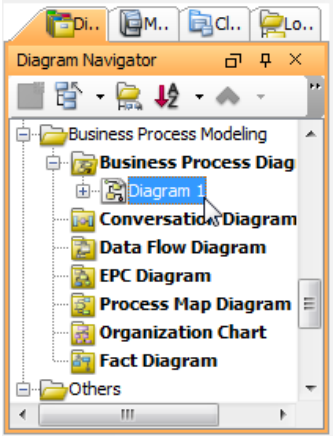
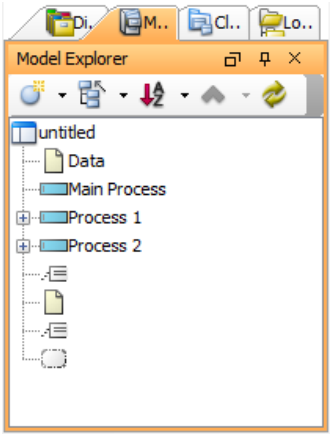


Diagram Navigator lists the imported diagram(s)

Model element can be accessed in the **Model Explorer**. You can create diagram with them by dragging and dropping them onto diagrams.



Model Explorer lists the imported model element(s)

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Export diagram to various graphic formats

You can export the opening diagram to image file. There are three ways of exporting. This chapter will shows you the instruction as well as some of the configuration of export, like how to slice diagram into parts.

Exporting active diagram as image

Shows you how to export the opening diagram to image.

Exporting multiple diagrams as images

Shows you how to export selected diagrams to image, as well as the steps of slicing image into parts.

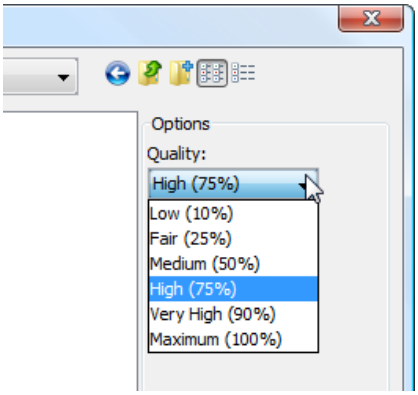
Exporting portion of diagram as image

You can export part of a drawing (i.e. some shapes) as an image. This page shows you how to do.

Exporting active diagram as image

You can export the opening diagram to image file. To export the active diagram as an image file:

1. Select **File > Export > Active Diagram as Image...** from main menu.
2. In the **Save** dialog box, set the image quality. The higher the quality, the clearer the image, the larger the image size.



Set image quality

3. Select the image format at the bottom of dialog box.

> **NOTE:** There are two options for exporting as PNG files - with and without background:
> - With background: export diagram's background color.
> - Without background: ignore the background color by exporting transparent background.

> **NOTE:** You can export diagrams to native PDF format. Since the exported PDF is of a small size, it can save a lot of space. Also, because the diagram in PDF is a vector, it is scalable. There are two export options:
> - PDF(diagram per page): selected diagrams will be exported to the same PDF file. Each diagram will occupy one page.
> - PDF(diagram per file): each diagram will be exported in one new PDF file.

4. Specify the filename.
5. Click **Save** to export.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

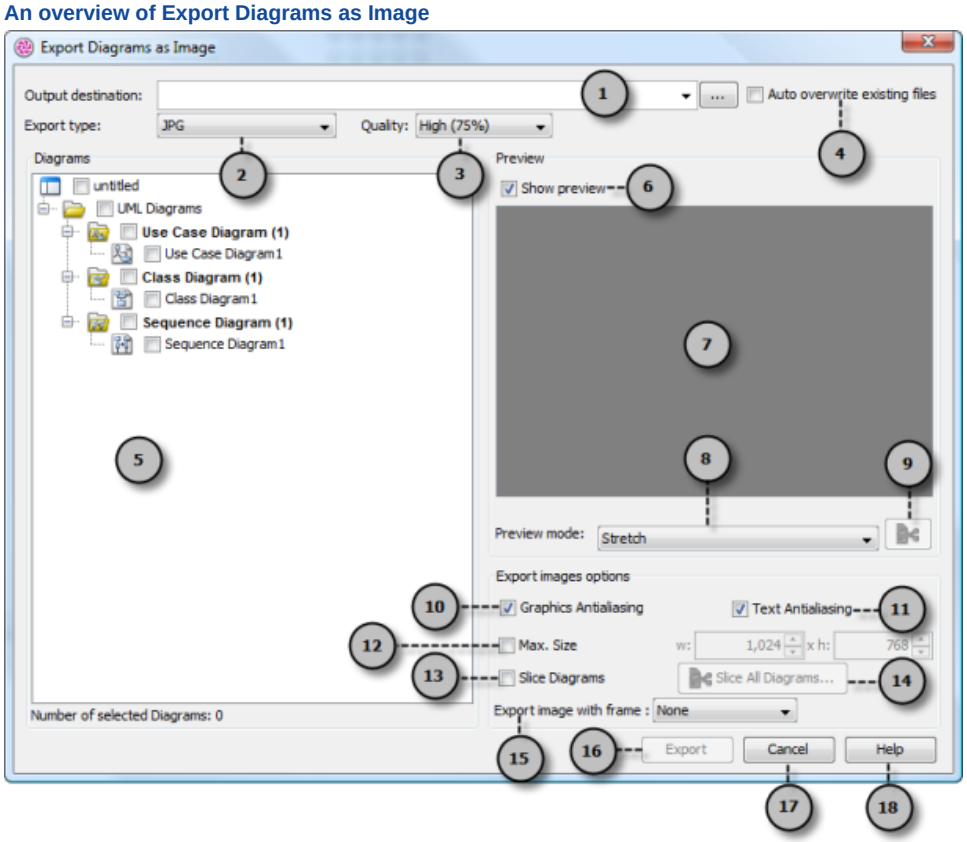
Exporting multiple diagrams as images

You can export diagrams in your project to image files. You can not only specify the quality and format (e.g. JPG, PNG, SVG, EMF, PDF) of images, but also slice diagram into pieces, for easier insertion into documents.

Export diagrams

1. Select **File > Export > Diagrams as Image...** from main menu.
2. In the **Export Diagrams as Image** dialog box, select the diagram(s) to export.
3. Specify the output destination for storing the image files.
4. Click **Export** button to export the diagrams.

An overview of Export Diagrams as Image



*An overview of **Export Diagrams as Image** dialog box*

No.	Name	Description
1	Output destination	The **Output destination** is the directory where all the exported images are saved to. You can enter the path in the text field directly, or you can click on the **...** button to browse for the directory.
2	Export type	To select the image format of the exported image click on the pull-down box beside the **Export type** field and select the format you want to use. There are two options for exporting as PNG files - with and without background. With background will export diagram's background color. Without diagram will ignore the background color by exporting transparent background. You can export Agilian diagram to native PDF format. Since the exported PDF is of a small size, it can save a lot of space. Also, because the diagram in PDF is a vector, it is scalable. There are two different options when you export. For PDF(diagram per page), all the diagrams selected will be exported in the same PDF file. Each diagram will occupy one page. For PDF (diagram per file), each diagram selected will be exported in one new PDF file.
3	Quality	The quality of image. By applying a higher quality, the images will be more clear, but larger in file size. By applying a lower quality, the images will look more blur, but smaller in file size.
4	Auto overwrite existing files	You can check the **Auto overwrite existing files** checkbox to allow overwriting of files in the export process.
5	Diagrams	The **Diagrams** pane shows the diagrams in the current project. Check the checkbox beside the diagram you want to export. The number of selected diagrams is displayed at the bottom of the **Diagram** pane. The Preview pane also allows you to preview the exported image of the selected diagram.
6	Show preview	Check or uncheck to enable or disable the preview.

7	Preview	The **Preview** pane shows the preview of the exported image of the selected diagram in the Diagrams pane.
8	Preview mode	Select the size of the preview image by selecting from the pull-down box beside the **Preview mode** field. Selecting **Stretch** will show the image in scaled size that fits to the preview area, while selecting **Real size** will show the image in its actual size.
9	Diagram slicer	Click to configure how diagram is sliced into pieces. This is enabled only when the check box for **Slice Diagrams** (for slicing all diagrams) is unchecked. For details about slicing diagrams, please refer to the following section.
10	Graphics Anti-aliasing	Anti-aliasing is a method which handles the staircase pixels of slanted lines and curves to make them look smoother. You can apply anti-aliasing to the exported images. To apply anti-aliasing to graphics, check the **Graphics Anti-aliasing** checkbox.
11	Text Anti-aliasing	Anti-aliasing is a method which handles the staircase pixels of slanted lines and curves to make them look smoother. You can apply anti-aliasing to the exported images. To apply anti-aliasing to text, check the **Text Anti-aliasing** checkbox.
12	Max. Size	Maximum size of exported images. If the diagram size is larger than the maximum size, it will be resized.
13	Slice Diagrams	Enable it to slice all diagrams into pieces to obtain multiple image files for a single diagram. For details, please refer to the following section.
14	Slice all diagrams	Click to configure slice settings on all diagrams.
15	Export image with frame	A frame is a border that prints around a diagram. By selecting **None**, frame won't be printed. By selecting **Export with frame,** a frame will be added to exported images, making the diagram name show at the top left of diagram. By selecting **Export with border**, a black and thin border will be added to exported images.
16	Export	Click to proceed with exporting.
17	Cancel	Click to close the exporter without exporting diagrams.
18	Help	Click to show the help contents.

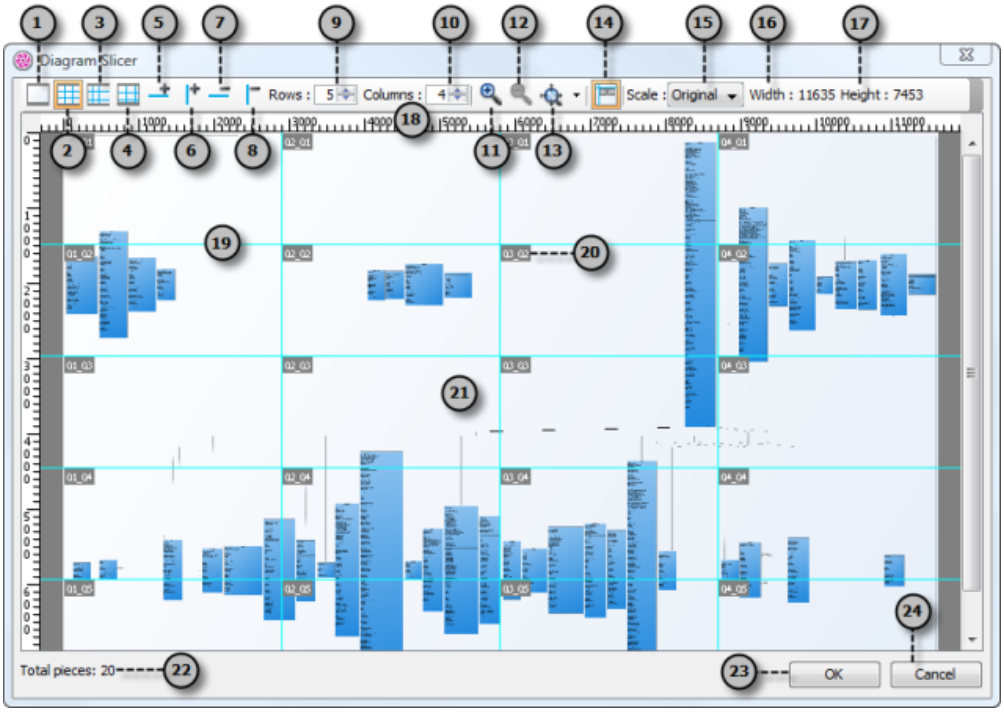
*Description of **Diagram Exporter** dialog box*

Slice a diagram into pieces with diagram slicer

You can slice diagrams into pieces (number of files), as well as restrict the size of the exported diagrams.

The slice diagram dialog box

The way how diagram is sliced can be set per diagram, or to all diagrams. To slice a diagram, click on the slice button right under the diagram preview in the **Diagram Exporter** dialog box. To slice all diagrams, enable **Slice Diagrams** and click on **Slice All Diagrams** button. Both ways open the **Diagram Slicer** for configuring how diagram(s) is to be sliced.



An overview of Diagram Slicer dialog box

Below is the description of different parts of the dialog box.

No.	Name	Description
1	No slicing	Do not slice diagram.
2	Fixed size	A simple strategy which slice exported diagram into pieces that have the same size.
3	Free slicing	User can customize how to slice the exported diagram by specifying the position of vertical slices and horizontal slices.
4	Fixed ratio	User can customize how to slice the exported diagram by specifying the position of vertical slices and horizontal slices.
5	Add row	Slice the diagram into specific number of rows.
6	Add column	Slice the diagram into specific number of columns.
7	Remove row	Reduce the number of rows to slice.
8	Remove column	Reduce the number of columns to slice.
9	Rows	The number of rows for slicing a diagram.
10	Columns	The number of columns for slicing a diagram.
11	Zoom in	Magnify the diagram content.
12	Zoom out	Magnify the diagram content.
13	Zoom fit to screen	Adjust the diagram to make it fit well on the slicer window.
14	Show label	Click to show/hide index label.
15	Scale	The way of scaling. When configuring slicing for all diagrams, this part will not be displayed.
16	Width	The total width of diagram.
17	Height	The total height of diagram.
18	Ruler	Shows the size of the diagram. When the slicing strategy **Free Slicing** is selected, a new row and column can be created by dragging a new one from the ruler.
19	Slice line	Lines that divide the diagram into pieces. The show the vertical and horizontal position that the diagram will be sliced at. When **Free Slicing** or **Fixed Ratio** is selected, the lines can be dragged and moved.
20	Index label	Shows the index of the pieces. This index will be printed on the exported file as well.
21	Diagram	The diagram being sliced.

22	Total pieces	The number of pieces to produced.
23	OK	Click to confirm the slice settings.
24	Cancel	Click to close the diagram slicer with changes discarded.

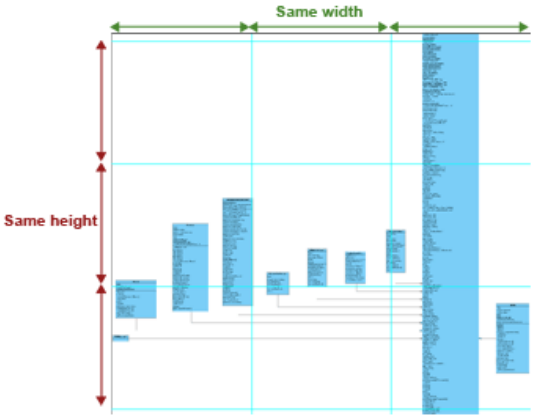
*Description of **Diagram Slicer** dialog box*

Slicing strategies

There are three slicing strategies - **Fixed Size**, **Free Slicing** and **Fixed Ratio**. Each gives a distinct way of slicing images.

Fixed size

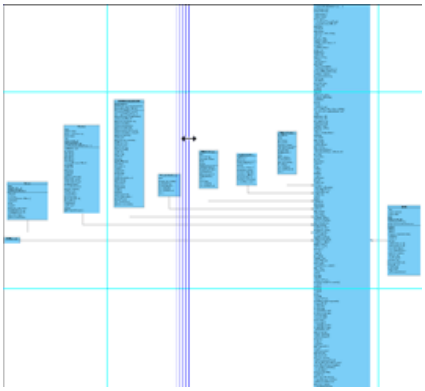
Fixed Size is a simple strategy which slice exported diagram into pieces that have the same size. User specifies the number of columns and rows to slice and then the exported diagram will be sliced into specific pieces.



Fixed Size

Free slicing

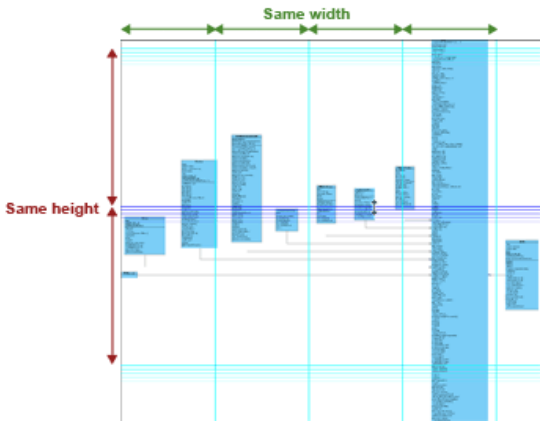
User can customize how to slice the exported diagram by specifying the position of vertical slices and horizontal slices. It is particularly useful to prevent a shape from being sliced into pieces.



Free Slicing

Fixed ratio

Fixed Ratio strategy gains the benefit of **Fixed Slice** and **Free Slicing**. The width and height of pieces are the same but last row and column. User can also customize the width and height of sliced pieces. Like **Free Slicing**, **Fixed Ratio** is size oriented. User modifies the size of pieces and **Diagram Slicer** calculates the number of row and column to slice.



Fixed Ratio

Controlling size of exported image

Diagram Slicer not only slice diagram into pieces but also controls the total size of the exported diagrams. There are scale controls on the right of the toolbar from the **Diagram Slicer** dialog. By default, the type of scale is **Original**. And it shows the size of diagram. The following are some of the possible ways of controlling diagram size.

Control by size

To control the total size of the exported diagram by specific width and height, select **Size** from the **Scale** combo box, and then enter the width and height of the diagram. The ruler shows the size of the diagram.

Control by ratio

To control the total size of the exported diagram by specific ratio, select **Ratio** from the **Scale** combo box and enter the ratio in the field next to the combo box. The total size of the exported diagram shows next to that field.

Related Resources

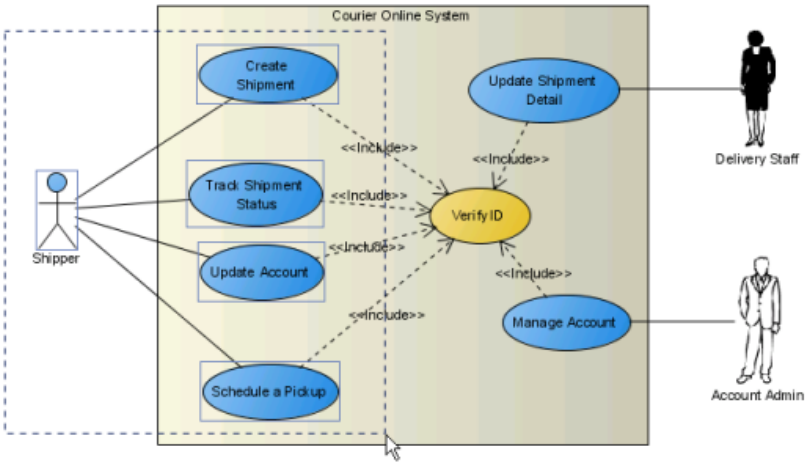
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Exporting portion of diagram as image

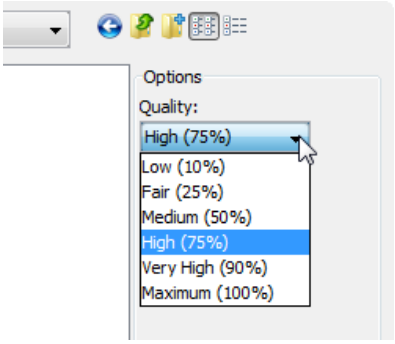
You can export some shapes in a diagram as an image file by selecting the shapes you want to export then perform export. To export selected shapes to image:

1. On a diagram, select the shapes to be exported.



Selecting shapes to export as image

2. Select **File > Export > Selection as Image...** from main menu.
3. In the **Save** dialog box, set the image quality. The higher the quality, the clearer the image, the larger the image size.



Set image quality

4. Select the image format at the bottom of dialog box.

> **NOTE:** There are two options for exporting as PNG files - with and without background:
> - With background: export diagram's background color.
> - Without background: ignore the background color by exporting transparent background.

> **NOTE:** You can export diagrams to native PDF format. Since the exported PDF is of a small size, it can save a lot of space. Also, because the diagram in PDF is a vector, it is scalable. There are two export options:
> - PDF(diagram per page): selected diagrams will be exported to the same PDF file. Each diagram will occupy one page.
> - PDF(diagram per file): each diagram will be exported in one new PDF file.

5. Specify the filename of the image file.
6. Click **Save** to export.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

The Open API

Plugin Support provides an interface for developers to integrate with Agilian. Developers can develop their plugins to achieve domain related operations. In this chapter, the overview of API architecture as well as the development of plugin will be covered.

Introduction to plugin support

Provides basic information about plugin support. Some of the key concepts like diagram, model element and action will be mentioned.

Implementing plugin

Detailed information about how to implement (code) a plugin will be covered.

Deploying plugin

Shows you how to deploy a plugin to Agilian.

Introduction to plugin support

Plugin Support provides an interface for developers to integrate with Agilian. Developers can develop their plugins for what they want. In this chapter, we will introduce the structure of a plugin.

Plugin.xml

A plugin is defined in a XML file (plugin.xml). It includes the information (such as plugin id, provider, required libraries, etc...), custom actions (menu, toolbar and popup menu) and custom shapes/connector of the plugin.

For working with Agilian in plugin, there are 4 main components must be known by developers: Model, Diagram, Diagram Element and Action/Action Controller.

Model element

Model Elements are basic construct of a model. Plugin allows developer to create, retrieve, update and delete model elements through the popup menu context or through the project (by iterating model elements within a project).

Diagram

Diagram is contains diagram elements on different domain (such as Business Process Diagram, DFD, etc...).

Plugin allows developer to create, retrieve, update and delete diagrams through the popup menu context or through the project (by iterating diagrams within a project)

Diagram element

A model element does not contain information of appearance (such as x, y, width, height, etc...). It is the diagram element, which appear on the user interface, that owns the appearance data. Diagram Element represents a view of a model element. A model element can be shown on different diagrams (such as a class can be shown on 2 different class diagrams).

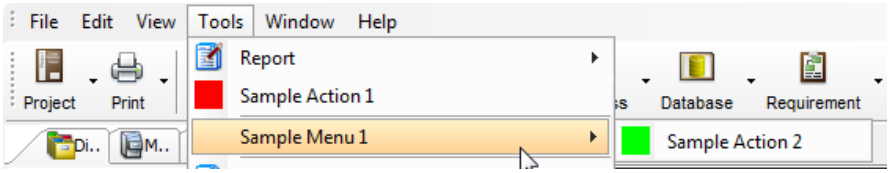
There are 2 kinds of diagram element: Shape and Connector. Shape represents the non-relationships diagram element (such as sub-process). Connector represents the relationships (such as sequence flow). Plugin allows developer to create, retrieve, update and delete diagram elements through the popup menu context or through the project (to iterate all the diagrams and then the diagram elements appear on a diagram).

Action/Action controller

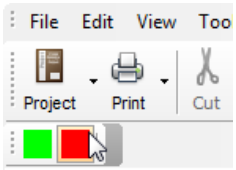
Action represents buttons and menus (menu, toolbar and popup menu), which contains the information on outlook (such as label, icon, mnemonic, etc...) and responses to trigger the function call.

Action is used to represent the button on 3 regions: menu/toolbar, popup menu and diagram toolbar

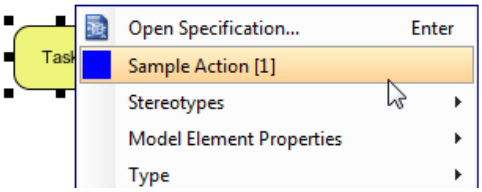
Action Controller is the control (function call) of actions. Developer needs to implement different Action Controller on different region's actions.



Menu with user-defined menus



Toolbar with user-defined buttons



Popup menu with user-defined menus

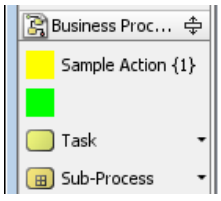


Diagram toolbar with user-defined buttons

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Implementing plugin

Configuring development environment

Plugin Support API is placed on **%Agilian-Install-Dir%/lib/openapi.jar**. In order to work with Agilian, developer must import the jar into the development classpaths.

Beginning of plugin.xml

plugin.xml is the base of a plugin, to develop a plugin, should be start from writing the plugin.xml. The basic directory structure is "Agilian-Install-Dir/plugins/YOUR_PLUGIN_ID/plugin.xml"

For improving the variability of the plugin.xml, a properties file (plugin.properties) can be used for storing the value of the xml. Developer can assignment the value of the attributes in xml starts with '%', then the value will be read from the properties file. For example

In plugin.xml: <plugin id="sample.plugin name="%plugin.name" .../>

In plugin.properties: plugin.name=sample.plugin

Sample on XML:

```
<  plugin
        id= "sample.plugin"
        name= "Sample Plugin"
        description= "Sample Plugin"
        provider= "Visual Paradigm"
        class= "sample.plugin.SamplePlugin">
        <  runtime >
                < library path= "lib/sampleplugin.jar" relativePath= "true"/>
        </  runtime >
        <!-- to be continued -->
</  plugin >
```

Table shows the description of elements in the plugin.xml.

Element	Attribute	Description
plugin		The root element of plugin.xml, specify the basic information of the plugin (id, name, provider, etc...)
plugin	**class**	The class of the plugin, required to implements **com.vp.plugin.VPPlugin**.
runtime		The element specified the runtime environment of the plugin.
library (1..*)		Specifies the .jar or directory as the classpaths required on the plugin. Such as the classes of the plugin and some libraries the plugin required.
library (1..*)	**Path**	The path of the .jar or directory.
library (1..*)	**relativePath** (optional, default: **true**)	Specifies whether the path is relative path.

plugin.xml element description

Description on Code:

VPPlugin (com.vp.plugin.VPPlugin)

This class must be implemented and ref on <plugin class="xxx"... Otherwise, the plugin will not be loaded completely. In fact, the class can do nothing on it.

The following is the sample code:

```
package sample.plugin;
public class SamplePlugin implements com.vp.plugin.VPPlugin {
        // make sure there is a constructor without any parameters
        public void loaded(com.vp.plugin.VPPluginInfo info) {
                // called when the plugin is loaded
                // developer can get the current plugin's id and the
                // current plugin directory (default: %VP-Suite-Install-DIR%/plugins)of VP-Suite from the VPPluginInfo.
        }
        public void unloaded() {
                // called when the plugin is unloaded (when Agilian will be exited)
        }
}
```

Implementing custom action

There are 2 main components for an Action: Action and Action Controller. Action represents the outlook, Action Controller responses to work as function call. In order to create custom action, developer needs to define the Action on xml, and implement the Action Controller on code.

Sample on XML:

```
<plugin>
        < actionSets>
                <!-- to be continued -->
        </ actionSets>
        <!-- to be continued -->
</plugin>
```

Table shows the description of elements in the above XML.

Element	Attribute	Description
actionSets		It is a collection of ActionSet. There 2 kinds of ActionSet: **actionSet** and **contextSensitiveActionSet**. **actionSet** is a set of actions which will be shown on menu/toolbar or diagram toolbar. **contextSensitiveActionSet** is set of actions which will be shown on popup menu.

XML sample for custom action

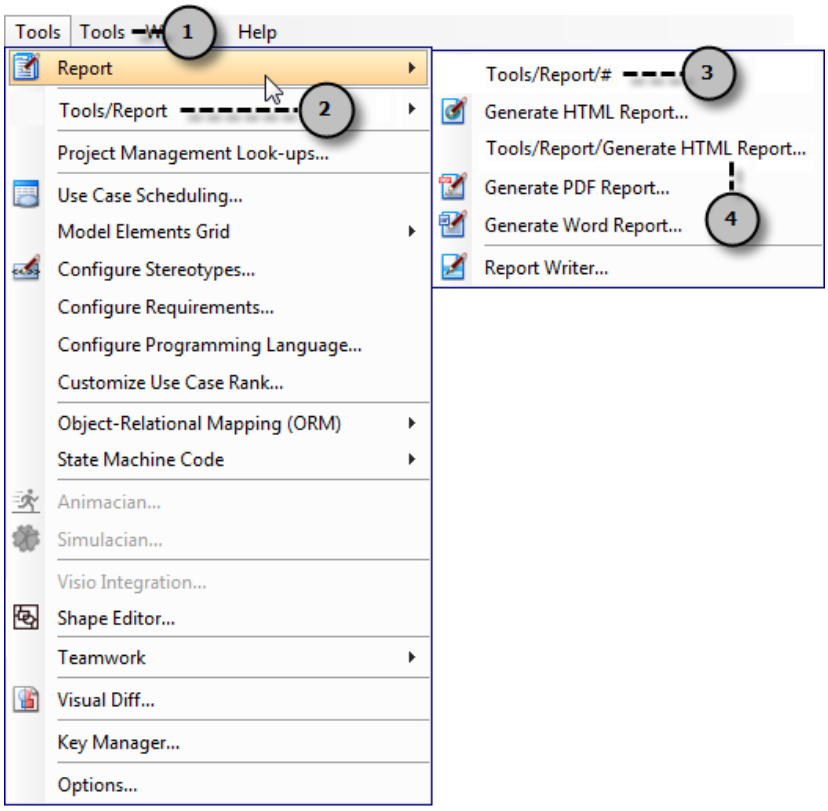
There are differences on xml definition and code implementation of the 3 kinds of Actions (menu/toolbar, popup menu, diagram toolbar).

Custom action on menu/Toolbar

Developer can define the menu, menu item, toolbar, toolbar button and etc... on the plugin.xml. In order to trigger the menu item and toolbar button's function call, Action Controller is required to be implemented and added into the Action. The Action Controller class on menu/toolbar actions is com.vp.plugin.action.VPActionController.

There are 2 important attributes used on menu, action and separator: **menuPath** and **toolbarPath** .

menuPath is the path specified where is the item placed on menu, toolbarPath is the path specified where is the item placed on toolbar. The path is formed by a set of 'name'. The 'name' is similar with the caption of the menu items (caption in English, ignores the "..." and remind the 'space'). '/' is used as delimiter of the path. '#' is used to represent the front of the menu. Here is 4 examples will be given:



Custom Action on MenuBar

Below is the menupaths required for implementing the menus shown in the above images.

Menu	"label" in XML	"menupath" in XML	Remarks
1	Tools	Tools	After the Tools menu
2	Tools/Report	Tools/Report	Under the Tools menu, after the Report menu
3	Tools/Report/#	Tools/Report/#	Under the Tools menu, and under the Report menu, place on the front

4	Tools/Report/Generate HTML Report	Tools/Report/Generate HTML Report	Under the Tools menu, and under the Report menu, after the Generate HTML Report menu item

Different menupaths settings

Sample on XML:

```
< actionSet id="sample.plugin.actions.ActionSet1">
        < toolbar
                id= "sample.plugin.actions.Toolbar1"
                orientation= "north"
                index= "last"/>
        < menu
                id= "sample.plugin.actions.Menu1"
                label= "Sample Menu 1"
                mnemonic= "M"
                menuPath= "Tools/Report"/>
        < action
                id= "sample.plugin.actions.Action1"
                actionType= "generalAction"
                label= "Sample Action 1"
                tooltip= "Sample Action 1"
                icon= "icons/red.png"
                style= "normal"
                menuPath= "Tools/Report"
                toolbarPath= "sample.plugin.actions.Toolbar1/#">
                < actionController class= "sample.plugin.actions.ActionController"/>
        </ action>
        < separator
                id= "sample.plugin.actions.Separator1"
                menuPath= "Tools/sample.plugin.actions.Action1"
                toolbarPath= "sample.plugin.actions.Toolbar1/sample.plugin.action.Action1"/>
</ actionSet>
```

Table shows the description of elements in the above XML.

Element	Attribute	Description
actionSets		It is a collection of ActionSet. There 2 kinds of ActionSet: **actionSet** and **contextSensitiveActionSet**. **actionSet** is a set of actions which will be shown on menu/toolbar or diagram toolbar. **contextSensitiveActionSet** is set of actions which will be shown on popup menu.
toolbar (0..*)		Specifies a toolbar, contains the location information of the toolbar.
toolbar (0..*)	**orientation** [north | east |south | west]	Specifies which side will be the toolbar placed on.
toolbar (0..*)	**index** [(number) | last |new]	Based on the orientation, where will be the toolbar placed. e.g. the orientation is "north" and there is 2 rows toolbars already. If the index is "0", then the toolbar will be placed on the first row's last position. If the index is "last", the toolbar will be placed on the last row, last position. If the index is "new", the toolbar will be placed on the third row (new row).
menu (0..*)		Specifies a menu or pull down button on menu bar or toolbar. It contains the outlook information of the menu.
action (0..*)		Specifies a menu item or button on menu bar or toolbar. It contains the outlook information of the menu item.
action (0..*)	**actionType** [generalAction | shapeAction | connectorAction] (optional, default: generalAction)	There are 3 types: generalAction, shapeAction and connectorAction. As the action on menu/toolbar, generalAction should be assigned.
actionController		Specifies the Action Controller for the action (the parent node in the xml).
actionController		The class name of the Action Controller. For the action on menu/toolbar, it is required to implement **com.vp.plugin.action.VPActionController**.
separator (0..*)		Specified a separator on menu bar or toolbar.

XML sample for menus and toolbars

Description on Code:

VPActionController (com.vp.plugin.action.VPActionController)

This class is used to perform the function call when the action is clicked. One Action Controller class refers to multi Actions is allowed.

Sample:

```
package sample.plugin.actions;
public class ActionController implements com.vp.plugin.action.VPActionController {
	// make sure there is an constructor without any parameters
	public void performAction(com.vp.plugin.action.VPAction action) {
		// called when the button is clicked, the parameter action represents the Action which be clicked.
		// developer also can set the properties of the action
	}
	public void update(com.vp.plugin.action.VPAction action) {
		// *for the actions located on menu bar only
		// when the parent menu is selected, this will be called,
		// developer can set the properties of the action before it is shown (e.g. enable/disable the menu item)
	}
}
```

Custom action on popup menu (context sensitive)

Developer can define the menu, menu item and separator on the popup menu shown on the diagram. The popup menu on diagram is context sensitive which based on what diagram element or diagram is selected. In order to make the menu item trigger the function call, Action Controller is required to be implemented. For popup menu, **com.vp.plugin.action.VPContextActionController** is the interface required developer to implement.

Same as Action on Menu/Toolbar, **menuPath** is used to specify the location of the action (menu/menu item on popup menu).

Sample on XML:

```
< contextSensitiveActionSet id= "sample.plugin.actions.ActionSet2">
	< contextTypes all= "false">
		< include type="Class"/>
		<!-- ignored when contextTypes.all = true -->
		< exclude type="Package"/>
		<!-- ignored when contextTypes.all = false -->
	</ contextTypes>
	<action
		id= "sample.plugin.actions.ContextAction1"
		label= "Sample Action [1]"
		icon= "icons/blue.png"
		style= "toggle"
		menuPath= "OpenSpecification">
		< actionController class= "sample.plugin.actions.ContextActionController"/>
	</action>
</contextSensitiveActionSet>
```

Table shows the description of elements in the above XML.

Element	Attribute	Description
contextSensitvieActionSet (0..*)		It is a collection of menu, action, separator on the popup menu of the plugin. The child elements should be ordered if they have the relationship on the position (e.g. developer prefers Action1 is placed into Menu1, then please define the Menu1 on the xml first
contextTypes		It is a collection of the model of diagram element of diagram types which the **contextSensitiveActionSet** is considering.
contextTypes	**all** [true | false] (optional, default: **false**)	Specify whether all the types of the models, diagram elements and diagrams will be considered by this actionSet.
Include		Specify the model, diagram element or diagram type will be considered by this ActionSet. (This will be ignored if the **contextType's** attribute **'all'** is assigned 'true'.
Include	**type**	It is type of the element. Such as "Class", "Actor", "ClassDiagram", "Attribute", etc...
exclude		Specify the model, diagram element or diagram type will not be considered by this ActionSet. (This will be ignored if the **contextType's** attribute **'all'** is assigned 'false'.
type		It is type of the element. Such as "Class", "Actor", "ClassDiagram", "Attribute", etc...

actionController		Specifies the Action Controller for the action (the parent node in the xml)
actionController	**class**	The class name of the Action Controller. For the action on popup menu, it is required to implement **com.vp.plugin.action.VPContextActionController**.

XML sample for popup menu

Description on Code:

VPContextActionController (com.vp.plugin.action.VPContextActionController)

This class is used to perform the function call when the action is clicked. One Action Controller class refers to multi Actions is allowed.

Sample:

```
package sample.plugin.actions;
import java.awt.event.ActionEvent;
public class ContextActionController implements com.vp.plugin.action.VPContextActionController {
	// make sure there is an constructor without any parameters
	public void performAction(
		com.vp.plugin.action.VPAction action,
		com.vp.plugin.action.VPContext context,
		ActionEvent e
	) {
		// called when the button is clicked
	}
	public void update(
		com.vp.plugin.action.VPAction action,
		com.vp.plugin.action.VPContext context
	) {
		// when the popup menu is selected, this will be called,
		// developer can set the properties of the action before it is shown (e.g. enable/disable the menu item)
	}
}
```

VPContext (**com.vp.plugin.action.VPContext**)

Context will be passed into the Action Controller when the popup menu is shown or action is trigger. It is what the user selected on the diagram, can be model, diagram element or/and diagram.

A diagram may contain many diagram elements, when user right-click on the diagram element or the diagram, a popup menu will be shown. So, the context may be diagram element or diagram. However, the diagram element must be contained by diagram, then if popup menu shown on a diagram element, the context must contain both diagram element and diagram. And the diagram element always represents for a model, so that is possible the context contains model, diagram element and diagram as same time. However, sometime, the popup menu is shown for a model only (e.g. select on an attribute of a class, because there is no diagram element for the attribute, the class's diagram element will be contained in the context).

Custom diagram element (shape and connector)

Developer can define the shape of connect on the specified diagram. But it is not allowed to develop a custom model. ActionSet and Action are used on definition of custom diagram element.

Sample on XML:

```
<actionSet id= "sample.plugin.actions.ShapeActionSet">
	<action
		id=  "sample.plugin.actions.ShapeAction1"
		actionType=  "shapeAction"
		label=  "Sample Action {1}"
		tooltip=  "Sample Action {1}"
		icon=  "icons/yellow.png"
		editorToolbarPath= "com.vp.diagram.ClassDiagram/Class">
		< shapeCreatorInfo
			shapeType= "sample.plugin.shape.Shape1"
			defaultWidth= "30"
			defaultHeight= "30"
			controllerClass= "sample.plugin.actions.ShapeController1"
			multilineCaption= "false"
			captionStyle= "north"
			resizable= "true"/>
	</action>
	<action
		id=  "sample.plugin.actions.ConnectorAction1"
		actionType= "connectorAction"
		label=  "Sample Action {2}"
		tooltip=  "Sample Action {2}"
		icon=  "icons/green.png"
		editorToolbarPath= "com.vp.diagram.ClassDiagram/sample.plugin.actions.ShapeAction1">
		<connectorCreatorInfo
			shapeType= "sample.plugin.connector.Connector1"
			fromArrowHeadStyle= "Arrow1"
			toArrowHeadStyle= "Arrow2"
```

```
                                    fromArrowHeadSize= "verySmall"
                                    toArrowHeadSize= "large"
                                    dashes= "7,10"
                                    lineWeight= "3"
                                    connectorStyle= "rectilinear">
                                    < connectionRules>
                                            < connectionRule
                                                                                fromShapeType= "sample.plugin.shape.Shape1"
                                                                                toShapeType= "sample.plugin.shape.Shape1"
                                                                                bidirection= "true"/>
                                            < connectionRule
                                                                                fromShapeType= "Class"
                                                                                toShapeType= "sample.plugin.shape.Shape1"
                                                                                bidirection= "true"/>
                                            <connectionRule
                                                                                fromShapeType= "Package"
                                                                                toShapeType= "sample.plugin.shape.Shape1"
                                                                                bidirection= "true"/>
                                    </connectionRules>
                            </connectorCreatorInfo>
                    </action>
            </actionSet>
```

Table shows the description of elements in the above XML.

Element	Attribute	Description
Action		It is a collection of menu, action, separator on the popup menu of the plugin. The child elements should be ordered if they have the relationship on the position (e.g. developer prefers Action1 is placed into Menu1, then please define the Menu1 on the xml first
Action	**actionType** [generalAction | shapeAction | connectorAction] (optional, default: generalAction)	There are 3 types: generalAction, shapeAction and connectorAction. As the action for custom shape, "shapeAction" should be assigned. For custom connector, "connectorAction" should be assigned.
Action	**editorToolbarPath**	Specify which diagram toolbar contains this action. e.g. to add a shapeAction on class diagram after the button for creating a new class, "com.vp.diagram.ClassDiagram/Class" should be assigned. "com.vp.diagram.ClassDiagram" is the id of the class diagram. "/" is the delimiter. "Class" is the button id.
shapeCreatorInfo		If the actionType is "shapeAction", **shapeCreatorInfo** is required. It is used to specify the details of the custom shape.
shapeCreatorInfo	shapeType	The shape type assigned by developer, unique value is required.
shapeCreatorInfo	**captionStyle** [center | north | none] (optional)	Specify where the caption of the shape is displayed.
shapeCreatorInfo	**controllerClass**	The class name which the class is responsible to draw the shape on the diagram, **com.vp.plugin.diagram.VPShapeController** is required to be implemented. **com.vp.plugin.diagram.AbstractShapeController** is an abstract class of the **VPShapeController**.
connectorCreatorInfo		If the actionType is "connectorShape", **connectorCreatorInfo** is required. It is used to specify the defails of the custom connector.
connectorCreatorInfo	**shapeType**	The shape type assigned by developer, unique value is required.
connectorCreatorInfo	**connectorStyle** [oblique | rectilinear] (optional, default: oblique)	Specify the style of the connector.
connectorCreatorInfo	**fromArrowHeadStyle** (optional)	Specify the arrow head style of the "from" side of the connector.
connectorCreatorInfo	**toArrowHeadStyle** (optional)	Specify the arrow head style of the "to" side of the connector.

connectorCreatorInfo	**fromArrowHeadSize** [verySmall | small | medium | large | extraLarge | jumbo | colossal] (optional)	Specify the arrow head size of the "from" side of the connector.
connectorCreatorInfo	**toArrowHeadSize** [verySmall | small | medium | large | extraLarge | jumbo | colossal] (optional)	Specify the arrow head size of the "to" side of the connector
connectorCreatorInfo	**dashes** (optional)	Specify the dashes pattern of the connector. A list of float, written in the pattern "%f, %f, %f, ...".
connectorCreatorInfo	**lineWeight** (optional)	Specify the weight of the connector.
connectorRules		It is a collection of **connectorRule**.
connectorRule (1..*)		Specify this connector can connect with what diagram element.

XML sample for diagram element

Description on Code:

VPShapeController (**com.vp.plugin.diagram.VPShapeController**)

It response to handle the outlook of the shape on the diagram.

Sample:

```
package sample.plugin.actions;
// import the necessaries
public class ShapeController implement com.vp.plugin.diagram.VPShapeController {
	public void drawShape(
		Graphics2D g, Paint lineColor, Paint fillColor, Stoke stroke,
		Com.vp.plugin.diagram.VPShapeInfo shapeInfo
	){
		// draw the shape by the graphics
		// shapeInfo contains the information of the shape, e.g. the bounds of the shape.
	}
	public boolean contains( int x, int y, com.vp.plugin.diagram.VPShapeInfo shapeInfo) {
		// check whether the x, y is inside the shape,
		// it is used to checking what is selected by the user
	}
}
```

Working with models

Plugin Support provides interface for the developer to create, retrieve update and delete the models in Agilian. The base class of the model is **com.vp.plugin.model.IModelElement**. All models are contained in the project (**com.vp.plugin.model.IProject**). Each model has a model type, to access all the model type, please refers to the class **com.vp.plugin.model.IModelElementFactory**, it is the class to create the models.

Creating model

Developer can use the model element factory (**com.vp.plugin.model.IModelElementFactory**) to create the model. Or based on a parent model (**com.vp.plugin.model.IModelElementParent**) to create a child model.

IModelElementFactory can be access by **IModelElementFactory.instance()**. It provides the functions to create all the models.

IModelElementParent is subclass of **IModelElement**. It provides the function to create the child into it. If the parent class is more specified, it may support a more details function to create the child. For example, **IClass** is subclass of **IModelElementParent**, it provides **createOperation()** to create an operation into it.

Sample on Code:

```
/*
* create model by IModelElementFactory
* result of the 2 methods: "class model is created and added into the project"
*
// assume in a code segment
IClass classModel1 = IModelElementFactory.instance().createClass();
IClass classModel2 = (IClass) IModelElementFactory.instance().create(IModelElementFactory. MODEL_TYPE_CLASS);
/*
* create model by IModelElementParent
* result of the first 2 methods, "operation model is created and added into the class model"
* result of the last method, "actor model is created and added into project", because actor cannot be the child of class model
*/
// assume in a code segment
IOperation operationModel1 = classModel1.createOperation();
IOperation operationModel2 = (IOperation) classModel1.create(IModelElementFactory. MODEL_TYPE_OPREATION);
IActor actorModel1 = (IActor) classModel1.create(IModelElementFactory. MODEL_TYPE_ACTOR);
```

Retrieving model

Developer can use the project (**com.vp.plugin.model.IProject**) or the context (**com.vp.plugin.action.VPContext**) from ActionController to retrieve the models.

IProject is the project of Agilian. The project contains all models, diagram and diagram elements. It provides function (**modelElementIterator()**) for the developer to iterate the models.

VPContext is the context of a popup menu. Developer can access the context by popup menu's action controller (**com.vp.plugin.action.VPContextActionController**). Context may contain a model element if the popup menu is shown on a diagram element or model.

Sample on Code:

```
/*
* retrieve model by IProject
*/
// assume in a code segment
IProject project = ApplicationManager.instance().getProjectManager().getProject();
Iterator iter = project.modelElementIterator();
while (iter.hasNext()) {
	IModelEmenet modelElement = (IModelElement) iter.next();
	// model element retrieved
}
/*
*retrieve model by VPContext
*/
// assume on a sub-class of com.vp.plugin.action.VPContextActionController
public void update(VPAction action, VPContext context) {
	IModelElement modelElement = context.getModelElement();
	// model element retrieved, but please take care,
	// context.getModelElement() may return null if the popup menu is shown for the diagram
	// or the selected diagram element doesn't refer to a model element.
}
/*
* retrieve relationship model from a class model
* there are 2 kinds of relationships: IRelationship and IEndRelationship
*/
// assume in a code segment
IClass classModel = ...; // retrieved the class model from somewhere
// retrieve a generalization (IRelationship)
Iterator genIter = classModel.fromRelationshipIterator();
while (genIter.hasNext()) {
	IRelationship relationship = (IRelationship) genIter.next();
	// found out the another side's model of the relationship
	IModelElement otherModel = relationship.getTo();
}
// retrieve am association (IEndRelationship)
Iterator assoIter = classModel.fromRelationshipEndIterator();
while (assoIter.hasNext()) {
	IRelationshipEnd relationshipEnd = (IRelationshipEnd) assoIter.next();
	IModelElement otherModel = relationshipEnd.getEndRelationship().getToEnd().getModelElement();
}
```

Updating model

Developer can call a set of get/set methods on a model. Different model type has different properties. For setting and getting the model's property, cast the **IModelElement** into it sub-class is necessary. For example, developer get the **IModelElement** from the popup menu's context. Developer check whether the model is a **IClass**, then developer cast the **IModelElement** into **IClass**, and call the function **IClass.setVisibility(xxx)**.

Sample on Code:

```
/*
* update a class model
*/
// assume in a code segment
IModelElement model = ...; // model is retrieved from somewhere
If (IModelElementFactory. MODEL_TYPE_CLASS.equals(model.getModelType()) ) {
	IClass classModel = (IClassModel) model;
	// set the class to be 'private'
	classModel.setVisibility(IClass. VISIBILITY_PRIVATE);
	// set super class
	IClass superClassModel = ...; // another class model is retrieved, it will be set to be the previous model's super class
	IGeneralization generalizationModel = IModelElementFactory.instance().createGeneralization();
	generalizationModel.setFrom(superClassModel);
	generalizationModel.setTo(classModel);
	// get all "setName" operation from the class and set to be "protected final"
	Iterator operationIter = classModel.operationIterator();
	while (operationIter.hasNext()) {
		IOperation operation = (IOperation) operationIter.next();
		if ( "setName".equals(operation.getName()) ) {
			operation.getJavaDetail( true).setJavaFinal( true);
```

```
                operation.setVisibility(IOperation. VISIBILITY_PROTECTED);
            }
        }
}
```

Deleting model

Developer can delete the model by simple way, just call the **IModelElement.delete()**.

Working with diagrams/Diagram elements

Plugin Support provides interface for the developer to create, retrieve update and delete the diagrams or diagram elements in Agilian. The base class of the diagram is co **m.vp.plugin.diagram.IDiagramUIModel**. The base class of the diagram element is **com.vp.plugin.diagram.DiagramElement**. All diagrams are contained in the project (**com.vp.plugin.model.IProject**). And the diagram elements can be found in the diagrams. The diagram elements can contains by the diagrams.

Creating diagrams/Diagram elements

Developer can create the diagram or diagram element by com.vp.plugin.DiagramManager. DiagramManager can be access by ApplicationManager.instance().getDiagramManager().

Sample on Code:

```
// assume in a code segment
DiagramManager diagramManager = ApplicationManager.instance().getDiagramManager();
/*
* create diagram
*/
IDiagramUIModel diagram = diagramManager.createDiagram(DiagramManager. DIAGRAM_TYPE_CLASS_DIAGRAM);
/*
* create diagram element with exists models
*/
IModelElement classModel1 = ...; // retrieved a class model from somewhere
IModelElement packageModel1 = classModel1.getParent(); // assume the class model is contained by a package
IDiagramElement packageDiagramElement1 = diagramManager.createDiagramElement(diagram, packageModel1);
IDiagramElement classDiagramElement1 = diagramManager.createDiagramElement(diagram, classModel1);
// class's diagram element should be a shape, not a connector
packageDiagramElement1.addChild((IShapeUIModel) classDiagramElement1);
/*
* create diagram element without models (the model will be created automatically)
*/
IDiagramElement newClassDiagramElement =
diagramManager.createDiagramElement(diagram, IClassDiagramUIModel. SHAPETYPE_CLASS);
IModelElement newClassModel = newClassDiagramElement.getModelElement();
/*
* open the created diagram
*/
diagramManager.openDiagram(diagram);
```

Retrieving diagrams/Diagram elements

Developer can use the project (**com.vp.plugin.model.IProject**) to retrieve the diagrams. Use a diagram (**com.vp.plugin.diagram.IDiagramUIModel**) to retrieve the contained diagram elements. Or use the context (**com.vp.plugin.action.VPContext**) from ActionController to retrieve the diagram and/or diagram element.

IProject is the project of Agilian. The project contains all models, diagram and diagram elements. It provides function (**diagramIterator()**) for the developer to iterate the diagrams.

IDiagramUIModel is a diagram, which may contain many diagram elements.

VPContext is the context of a popup menu. Developer can access the context by popup menu's action controller (**com.vp.plugin.action.VPContextActionController**). Context may contain a diagram and/or diagram elements.

Sample on Code:

```
/*
* retrieve diagram from IProject
*/
// assume in a code segment
IProject project = ApplicationManager.instance().getProjectManager().getProject();
Iterator diagramIter = project.diagramIterator();
while (diagramIter.hasNext()) {
        IDiagramUIModel diagram = (IDiagramUIModel) diagramIter.next();
        /*
        * retrieve diagram element from IDiagramUIModel
        */
        Iterator diagramElementIter = diagram.diagramElementIterator();
        while (diagramElementIter.hasNext()) {
                IDiagramElement diagramElement = (IDiagramElement) diagramElementIter.next();
        }
}
/*
* retrieve diagram and diagram element from VPContext
*/
```

```
// assume on a sub-class of com.vp.plugin.action.VPContextActionController
public void update(VPAction action, VPContext context) {
	IDiagramUIModel diagram = context.getDiagram();
	IDiagramElement diagramElement = context.getDiagramElement();
	// diagramElement may be null, if the popup menu shown for the diagram
}
/*
* retrieve connected connector from a shape
* because a connector can connected with both Shape and Connector, please check the
* both getToShape() and getToConnector() or getFromShape() and getFromConnector()
*/
// assume in a code segment
IShapeUIModel shape = ...; // retrieved the shape from somewhere
IConnectUIModel[] connectors = shape.toFromConnectorArray();
int count = connectors == null ? 0 : connectors. length;
for ( int i = 0; i < count; i++) {
	IDiagramElement toDiagramElement = connectors[i].getToShape();
	if (toDiagramElement == null) {
		toDiagramElement = connectors[i].getToConnector();
	}
}
```

Updateing diagrams/Diagram elements

IDiagramUIModel provides the functions to set the diagram outlook (size, background, etc...).

IDiagramElement is the super class of **IShapeUIModel** and **IConnectorUIModel**. Because there is difference between shape and connector, the **IShapeUIModel** and **IConnectorUIModel** provide different set of functions to update them.

Sample Code:

```
/*
* update a shape's size and set a connector's connector style
*/
// assume in a code segment
IShapeUIModel shape = ...; // retrieved the shape from somewhere
shape.setBounds(20, 20, 400, 400);
IConnector connector = ...; // retrieved the connector from somewhere
connector.setConnectorStyle(IConnector. CS_CURVE);
```

Deleting diagrams/Diagram elements

Developer can delete the diagram and diagram element by simple way, just call the **IDiagramUIModel.delete()** and **IDiagramElement.delete()**.

Showing dialog on Agilian

There is special method to show any dialog with Swing technology.

com.vp.plugin.ViewManager is an interface provides function for developer to show the dialog as same as show dialog by JOptionPane. Besides that, **Viewmanager** supports developer to show message on Agilian 's message pane and show custom dialog by implementing an interface (**com.vp.plugin.view.IDialogHandler**).

Same as JOptionPane, to show a dialog, it is better to have a component as the invoker/parent component. To get the component in Agilian , just call **ViewManager.getRootFrame()**.

Showing message on message pane

ViewManager provides function **showMessage(msg:String, msgTabId:String)** to show the message on Message Pane. The parameter **msg** is the content of the message, **msgTabId** is the id to identify the tab on Message Pane, which can be defined by developer.

Sample on Code:

```
// assume in a code segment
ViewManager viewManager = ApplicationManager.instance().getViewManager();
viewManager.showMessage( "Thank you for reading Agilian  Plugin Support User's Guide. >=)", "sample.plugin");
```

Showing simple message dialog

In Swing, we may use the **javax.swing.JOptionPane** to show a message dialog (e.g. **JOptionPane.showMessageDialog(...)**). **ViewrManager** provides the functions which simulate the JOptionPane. **ViewManger** provides a set of **showXXXXDialog(...)** functions for showing the dialog. The signature of the functions are same with the JOptionPane. Developer need not feel strange on calling the **showXXXXDialog(...)** functions.

Showing custom dialog

In Swing, we may implement the **javax.swing.JDialog** and add our component on the dialog's content pane. But in plugin, developer is required to implement an interface **com.vp.plugin.view.IDialogHandler** to work for the dialog.

IDialogHandler specify the behaviors of a dialog. There are 4 functions need to be implemented.

```
getComponent() : java.awt.Component
```

It is called once before the dialog is shown. Developer should return the content of the dialog (similar to the content pane).

```
prepare(dialog : com.vp.plugin.view.IDialog) : void
```

It is called after the **getComponent()**. A dialog is created on Agilian internally (it still not shown out). Developer can set the outlook of the dialog on **prepare()**, such as title, bounds and modal, etc... For your convenience, the dialog will be shown on the screen center as default. If developer don't want change the location, there is no necessary to call the **setLocation()** function.

shown()

It is called when the dialog is shown. Developer may need to do something when the dialog is shown, such as checking something before user to input data on the dialog.

canClosed()

It is called when the dialog is closed by the user clicking on the close button of the frame. Developer may not allow the user to close the dialog (e.g. failed on validation check), then please return 'false' on **canClosed()**.

Sample on Code:

```
package sample.plugin.dialog;
// assume imported necessary classes
public class CustomDialogHandler implements IDialogHandler {
	private IDialog _dialog;
	private Component _component;
	private JTextField _inputField1, _inputField2, _answerField;
	public Component getComponent() {
		this._inputField1 = new JTextField(10);
		this._inputField2 = new JTextField(10);
		this._answerField = new JTextField(10);
		JLabel addLabel = new JLabel( " + "); JLabel equalLabel = new JLabel( " = ");
		JButton okButton = new JButton( "Apply");
		okButton.addActionListener( new ActionListener() {
		public void actionPerformed(ActionEvent e) { ok();}
		} );
		JPanel pane = new JPanel();
		pane.add( this._inputField1); pane.add(addLabel); pane.add( this._inputField2);
		pane.add(equalLabel); pane.add( this._answerField); pane.add(okButton);
		this._component = pane;
		return pane;
	}
	public void prepare(IDialog dialog) {
		this._dialog = dialog;
		dialog.setModal(true);
		dialog.setTitle( "Maths Test");
		dialog.setResizable( false ); dialog.pack();
		this._inputField1.setText(String.valueOf(( int)(Math.random()*10000)));
		this._inputField2.setText(String.valueOf(( int)(Math.random()*10000)));
	}
	public void shown() {
		ApplicationManager.instance().getViewManager().showMessageDialog(
		this._component, "Maths Test is started, you have an half hour to finish this test.",
		"Maths Test", JOptionPane. INFORMATION_MESSAGE
		);
	}
	public boolean canClosed() {
		if ( this.checkAnswer()) { return true; }
		else {
			ApplicationManager.instance().getViewManager().showMessageDialog(
				this._component, "Incorrect",
				"Maths Test", JOptionPane. ERROR_MESSAGE
			);
			return false;
		}
	}
	private void ok() {
		if ( this.checkAnswer() ) { this._dialog.close(); }
		else {
			ApplicationManager.instance().getViewManager().showMessageDialog(
				this._component, "Incorrect",
				"Maths Test", JOptionPane. ERROR_MESSAGE
			);
		}
	}
	private boolean checkAnswer() {
		try {
			int a = Integer.parse Int( this._inputField1.getText());
			int b = Integer.parse Int( this._inputField2.getText());
			int c = Integer.parse Int( this._answerField.getText());
			return (a+b == c);
		}
		catch (Exception ex) { return false; }
	}
```

```
}
```

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Deploying plugin

After prepared all the required files for a plugin (plugin.xml, plugin.properties, classes/libraries and other resources), developer can plug the plugin into Agilian.

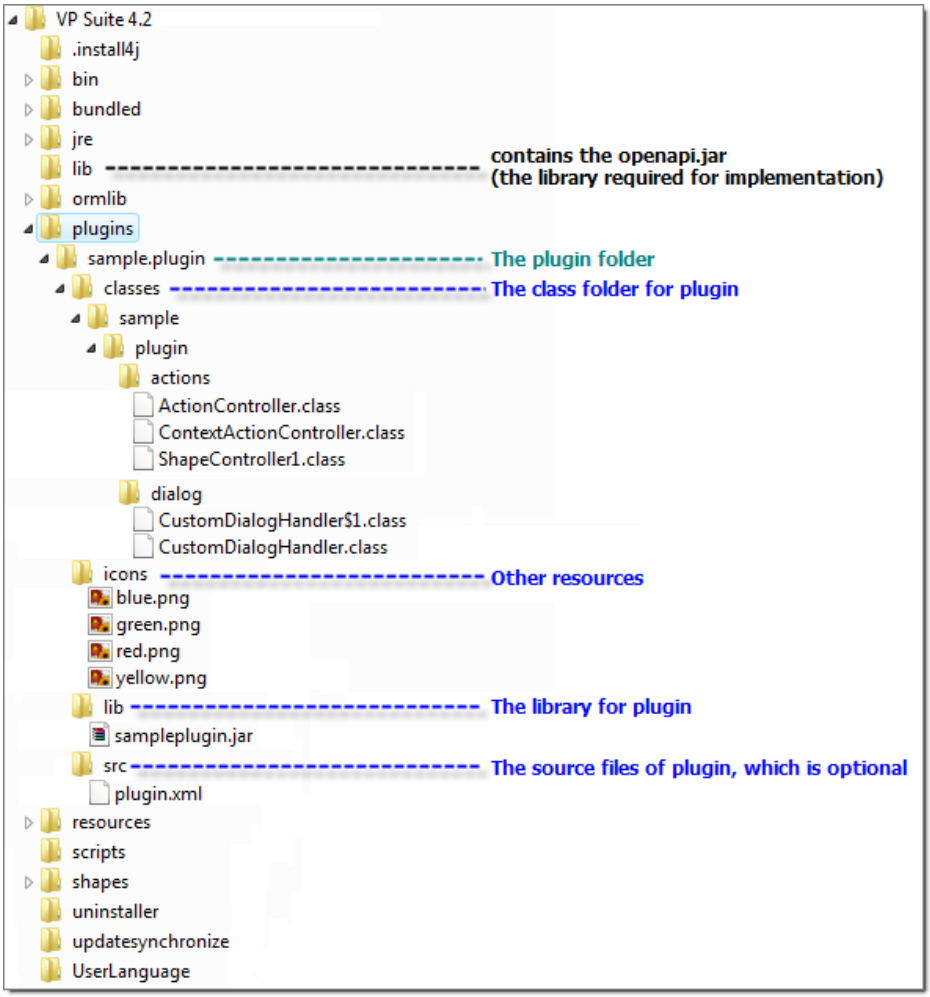
First, create a folder named **plugins** (notice the ' **s**') in the Agilian installation directory. Put the plugin files into "%VP-Suite-Install-Dir%\plugins\ %PLUGIN_ID%\". %PLUGIN_ID% is a directory named as the plugin id (use the id as the directory name to avoid duplicated directories defined in **plugins**)

The following structure should be obtained:

%VP-Suite-Install-Dir%

bin
lib
...
plugins
sample.plugin *(%PLUGIN_ID %)*
plugin.xml
plugin.properties
classes
sample (package)
... (other packages or classes or resources)
lib
sampleplugin.jar
... (others .jar)
icons (others resources)
red.png
...(other resources)

Below is an example of VP-Suite installation folder with plugin created in the **plugins** folder.



Plugin folder structure

After all, restart Agilian will see the plugin available. If not, make sure the code was written correctly and can be compiled, and you have setup the above folder structure correctly.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube

- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Command line operations

Instead of executing command through graphic user interface, you can also execute certain command in background, through the command line interface. In this chapter, all the supported command will be described in detail.

Exporting diagram image

The command needed to export diagram as image.

Exporting and importing XML

The command needed to export project data as XML.

Generating report

The command needed to generate HTML/PDF/Word report

Project publisher

The command needed to publish a project

Updating teamwork project from server

The command needed to update a local teamwork project by getting changes from server.

Executing operations with Apache Ant

How to execute commands with Ant script.

Exporting diagram image

To export images from a project through command line:

1. Browse the **scripts** folder under the Agilian installation directory.
2. Copy the script file **ExportDiagramImage.bat** and paste to the **bin** folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of Agilian installation directory.
5. Execute the script with parameters. For example:
   ExportDiagramImage -project C:\Demo\Demo.vpp -out C:\Demo\Output -diagram "" -type jpg*
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	Folder for storing the exported images	C:\Demo\Output
-diagram	A list of diagram required to export images. User can enter "*" for representing all diagrams, to supply the names of diagrams, or to supply a text file which includes the names of all diagrams. Notice that when you use an asterisk to indicate all diagrams, quote it.	diagram_1 diagram_2
-type [optional]	Type of diagrams. Here are the possible types: • png • png_with_background • jpg • svg • pdf	png

Parameters for ExportDiagramImage

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Exporting and import XMI

You can export project data to an XMI, edit it externally with other softwares that accepts XMI. In this chapter, you will see how to export and import XMI file through the command line interface.

Exporting XMI

To export XMI and images from a project through command line:

1. Browse the **scripts** folder under the Agilian installation directory.
2. Copy the script file **ExportXMI** and paste to the **bin** folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of Agilian installation directory.
5. Execute the script by supplying the required parameters. For example:
   ExportXMI -project C:\Demo\Demo.vpp -out C:\Demo\Output\Sample.xmi -type 2.1
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	The filepath of XMI file	C:\Demo\Output\sample.xmi
-type [optional]	Version of XMI. Unless specified, the lastly generated version will be selected. Here are the possible options: • 1.0 • 1.2 • 2.1 • 2.1UML2	2.1
-encoding [optional]	Encoding of XMI file	

Parameters for ExportXMI

Importing XMI

To import XMI to a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.
2. Copy the script file **ImportXMI** and paste to the **bin** folder of VP Suite installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.
5. Execute the script by supplying the required parameters. For example:
   ImportXMI -project C:\Demo\Demo.vpp -file C:\Demo\input\sample.xmi
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-file	The filepath of the XMI file to import	C:\Demo\input\sample.xmi

Parameters for ImportXMI

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Exporting and import XML

You can export project data to an XML, manipulate it externally, and feed the changes back to Agilian. In this chapter, you will see how to export and import XML file through the command line interface.

Exporting XML

To export XML and images from a project through command line:

1. Browse the **scripts** folder under the Agilian installation directory.
2. Copy the script file **ExportXML** and paste to the **bin** folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of Agilian installation directory.
5. Execute the script by supplying the required parameters. For example:
   ExportXML -project C:\Demo\Demo.vpp -out C:\Demo\Output
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	Folder for storing the exported XML and images	C:\Demo\Output
-diagram	One or more diagrams to be exported	"Diagram A" "Diagram B"
-noimage	Do not export image files for diagrams	N/A
-refmodel	Whether to embed referenced projects' content inline	true

Parameters for ExportXML

Importing XML

To import XML to a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.
2. Copy the script file **ImportXML** and paste to the **bin** folder of VP Suite installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.
5. Execute the script by supplying the required parameters. For example:
   ImportXML -project C:\Demo\Demo.vpp -file C:\Demo\Input\project.xml
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-file	The filepath of the XML file to import	C:\Demo\input\sample.xml

Parameters for ImportXML

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Exporting and import Excel

Agilian supports interoperability with Excel file. You can export project data to an Excel and edit it externally. In this chapter, you will see how to export and import Excel through the command line interface.

Exporting Excel

To export Excel from a project through command line:

1. Browse the scripts folder under the Agilian installation directory.
2. Copy the script file **ExportExcel** and paste to the bin folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of Agilian installation directory.
5. Execute the script by supplying the required parameters. For example:
   ExportExcel -project C:\Demo\Demo.vpp -out C:\Demo\Output\Sample.xls
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	The filepath of Excel file	C:\Demo\Output\Sample.xls

Parameters for ExportExcel

Importing Excel

To import Excel to a project through command line:

1. Browse the scripts folder under the Agilian installation directory.
2. Copy the script file **ImportExcel** and paste to the bin folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of Agilian installation directory.
5. Execute the script by supplying the required parameters. For example:
   ImportExcel -project C:\Demo\Demo.vpp -file C:\Demo\input\sample.xls
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-file	The filepath of the Excel file to import	C:\Demo\input\sample.xls

Parameters for ImportExcel

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of UML tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Generating report through command line

To generate HTML/PDF/Word report from a project through command line:

1. Browse the **scripts** folder under the Agilian installation directory.
2. Copy the script file **GenerateReport** and paste to the **bin** folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of Agilian installation directory.
5. Execute the script by supplying the required parameters. For example:
   GenerateReport -project C:\Demo\Demo.vpp -out C:\Demo\Output\MyReport.pdf -type pdf -all
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	The file or folder path of generated report file(s)	C:\Demo\Output\MyReport.pdf
-type	Type of report to generate. Here are the possible options: • html • pdf • word	pdf
-all [optional]	By default, only the selected diagrams (saved in vpp) will be covered when generating report. By including -all, all diagrams will be covered.	-all

Parameters for GenerateReport

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Project publisher

To publish project through command line:

1. Browse the **scripts** folder under the Agilian installation directory.
2. Copy the script file **ProjectPublisher** and paste to the **bin** folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the **bin** folder of Agilian installation directory.
5. Execute the script by supplying the required parameters. For example:
   ProjectPublisher -project C:\Demo\Demo.vpp -out C:\Demo\Output
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-out	The folder path of the files to be published	C:\Demo\Output
-nickname [optional]	By default, Project Publisher produces content for the active nickname. You can make it publish a specific nickname by specifying the nickname parameter.	Spanish

Parameters for ProjectPublisher

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Updating teamwork project from server

To update teamwork project from server through command line:

1. Browse the **scripts** folder under the Agilian installation directory.
2. Copy the script file **UpdateTeamworkProject** and paste to the **bin** folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of Agilian installation directory.
5. Execute the script by supplying the required parameters. For example:
   UpdateTeamworkProject -project "C:\vpworkspace\teamwork_client\projects\Demo\Demo.vpp" -workspace "C:\vpworkspace"
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-workspace	The path of workspace of the supplied project	C:\vpworkspace

Parameters for UpdateTeamworkProject

Related Resources

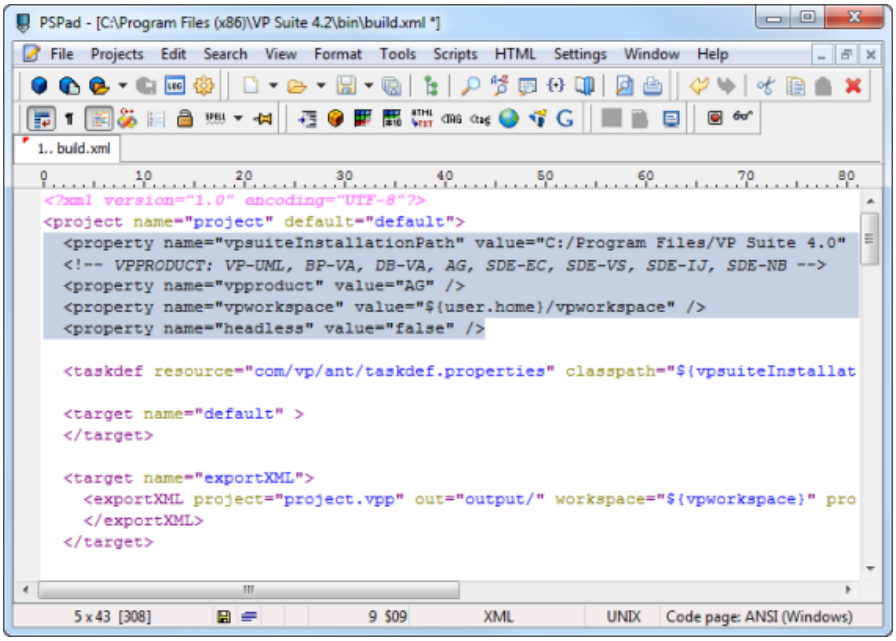
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Executing operations with Apache Ant

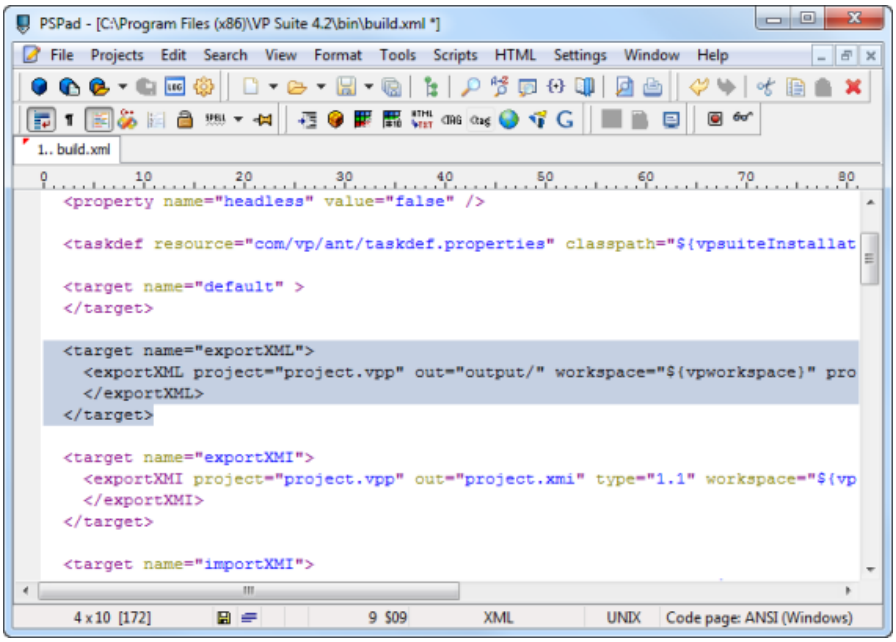
Apache Ant is a software tool for automating software build processes. It is written in the Java language and is primarily intended for use with Java. If you are not familiar with Ant, you can find more information about it at Ant's webpage. To execute commands with Ant:

1. Browse the **scripts** folder under the Agilian installation directory.
2. Copy the script file **build.xml** and paste to the **bin** folder of Agilian installation directory.
3. Open the build file in any text editor. Modify the properties **vpsuiteInstallationPath**, **vpproduct**, **vpworkspace** and **headless** to suit your environment.



To modify basic properties in build.xml

4. Modify task(s) specific parts by changing the values of parameters. For details about the parameters, refer to previous sections.



Modify task(s) specific properties in build.xml

5. Save the changes and exit.
6. Start the command prompt and navigate to the bin folder of VP Suite installation directory.
7. Enter **ant build.xml**, and then the task name to execute specific task.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Exporting report through command line (Report Composer)

To export reports created by report composer through command line:

1. Browse the **scripts** folder under the Agilian installation directory.
2. Copy the script file **ExportReportDiagram** and paste to the **bin** folder of Agilian installation directory.
3. Start the command prompt.
4. Navigate to the bin folder of Agilian installation directory.
5. Execute the script by supplying the required parameters. For example:
   ExportReportDiagram.bat -project "C:\Demo\Demo.vpp" -diagram "Report1" -out "C:\Demo\Output" -type word
   Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-diagram	The name of the report(s) to be exported. You can provide a list of diagram like "Report1" "Report2", or enter "*" to export all reports, or enter @diagramlist.txt to let the batch read the diagram list from file diagramlist.txt placed in bin folder.	"Report 1" "Report 2"
-out	The folder to store the exported file	C:\Demo\Output\
-type	Type of report to generate. Here are the possible options: • html • pdf • word	word

Parameters for ExportReportDiagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Printing diagrams

This chapters covers the steps needed to print diagram(s) to printing devices.

Printing diagrams

Introduces the standard way of printing diagram with description to several printing configuration.

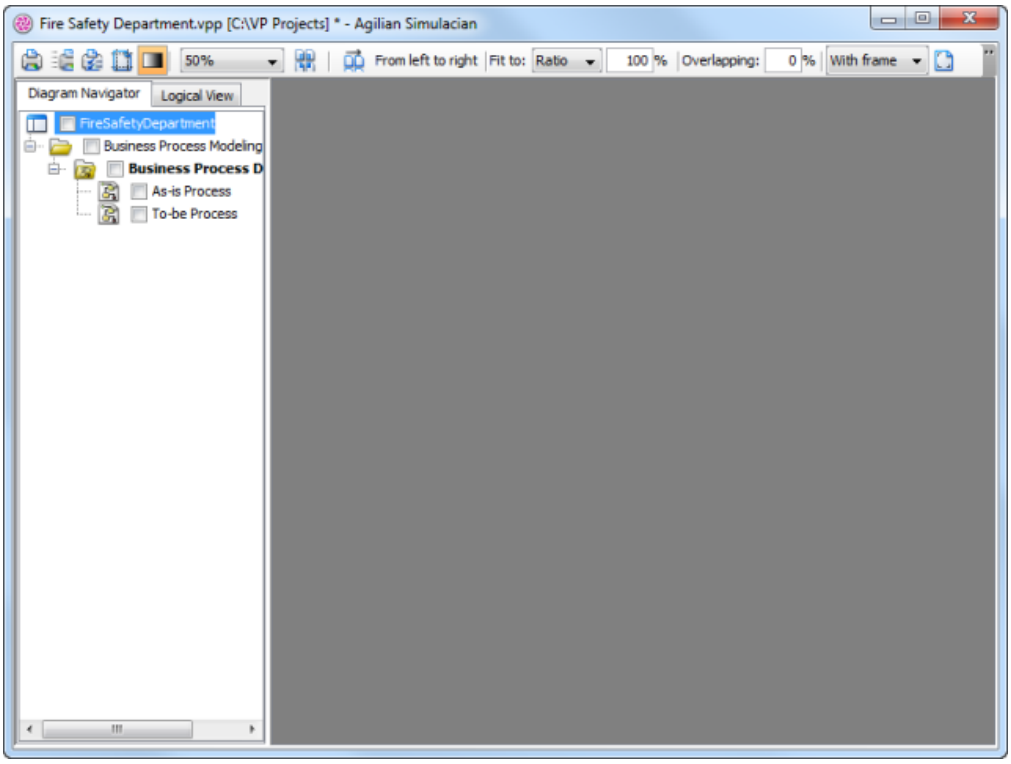
Quick print

Quick print is a lite way of printing diagram by ignoring some of the configuration options and preview of outcome.

Printing diagrams

The **Print** window dialog box allows you to preview the printout and provides a set of options for changing the printout style. To unfold the **Print** window dialog box, select **File > Print...** from main menu.

An overview of Print window



Print preview dialog box

The toolbar of the print preview pane allows you to configure the print settings. The buttons and their description are shown in the table below:

Button	Name	Description
	Print	Print the diagram(s). The **Print** dialog box will be opened.
	Quick Print	Print the diagram(s) without previewing them. The **Quick Print** dialog box will be opened.
	Page Setup	Set up the page properties, such as paper size and orientation.
	Adjust Margin	Adjust the margins of the pages.
	Use Gradient Color	Select the use gradient color in printout. Since printing gradient color will use up lots of memory, it is recommended to turn this option off for better performance.
	Print Diagram Background	Click to print diagram's background when printing. When un-clicked, background color is ignored in printing.
	Global page Number	Page number can be optionally displayed in printout. By default, the numbering of pages is diagram based, meaning that each diagram has its own set of numbers, and the numbers reset for a new diagram. The **Global Page Number** option is to enable the continuous numbering of all diagrams.
50%	Zoom	Select the percentage to reduce/enlarge the print preview of diagrams.
/	Paper Base Layout/ Diagram Base Layout	If the **Fit to Pages** option is chosen, and there are multiple pages in the printout, choosing **Paper Base Layout** will cause the distribution of pages to be paper-oriented (the diagram size is ignored in arranging the preview); while choosing **Diagram Base Layout** will cause the distribution of pages to be diagram-oriented. Note that this option affects the preview only; the order of the printout remains unchanged.
/	Paper Place Style	Change the order of the printout. A large diagram is divided into many pages, choosing **From left to right** will arrange the printout order from the pages on the left to the pages on the right, while

		choosing **From top to bottom** will arrange the print order from the pages on the top to the pages on the bottom.
Fit to: Ratio 100 %	Fit to Ratio	Set the diagram size to fit to the specified ratio.
Fit to: Pages 1 x 1	Fit to Pages	Set the diagram to be printed on the specified number of pages.
Overlapping: 0 %	Overlapping	Set the percentage of the margins to overlap among adjacent pages.
No border	Border	Determine whether to add frame or boder around diagram in printout.
	Show/ Hide Clip Marks on Page	Select/ deselect to show/hide the clip marks on the printout.
	Edit Header/ Footer	Edit the header and the footer of the printout.
	Multiple Page Mode	Switch to the Multiple Page Mode to set the multiple page options.
	Help	Call the Agilian help file.
	Close Print Preview	Close the print preview pane and return to the design area.

Details of toolbar

Printing a diagram with preview

You can use the Print command to select the printer. Set the range of pages and number of copies to be printed.

1. Select the desired diagram(s) for printing. The selected diagram(s) will be shown in the preview area.

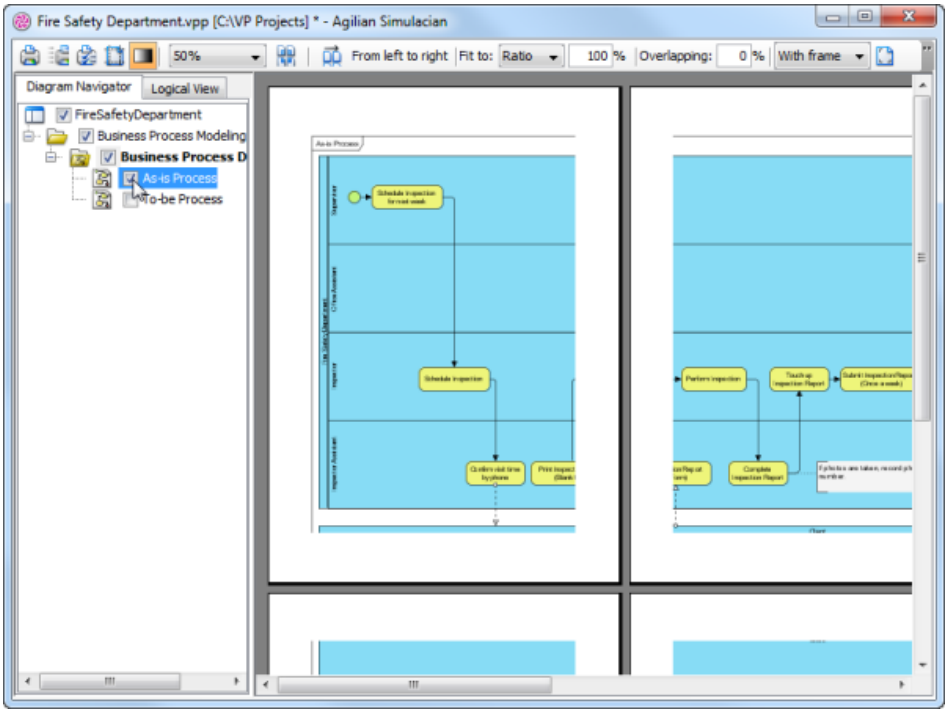
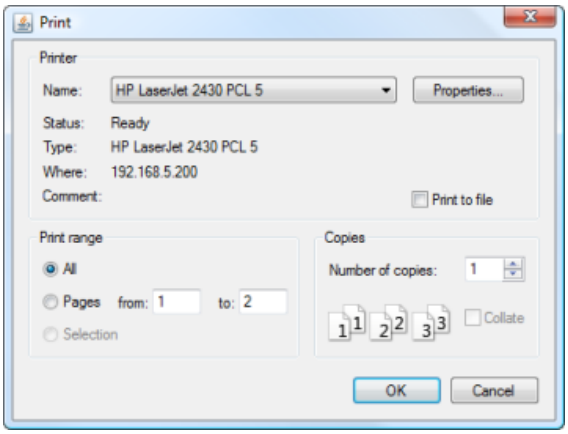


Diagram preview

2. Click the **Print** button on the Print preview toolbar. The **Print** dialog box will be shown.

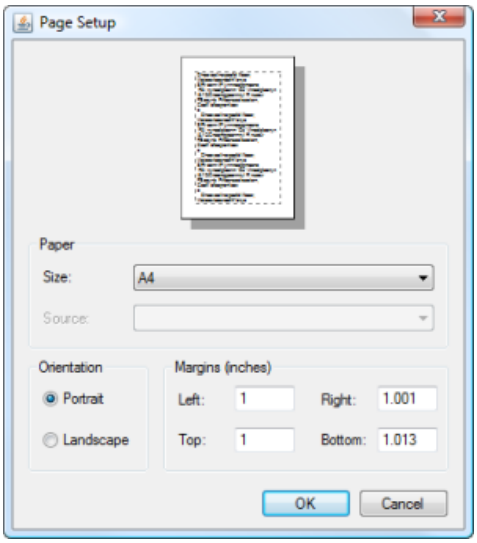


*The **Print** dialog box*

3. Select a specific printer, the page range and the number of copies to be printed. You may click the **Properties...** button to configure the printer-specific properties as well.
4. Click **OK** to start printing.

Page setup

Page Setup allows you to specify the page size, orientation, as well as the margins of the pages.

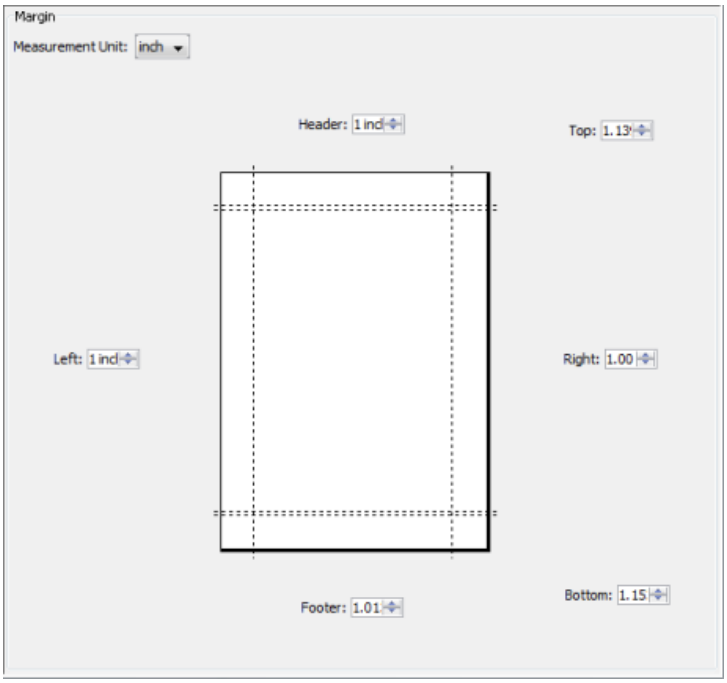


*The **Page setup** dialog box*

1. Click the **Page Setup** button on the toolbar. The **Page Setup** dialog box will appear.
2. You can click the **Size** drop-down menu to select the paper size for printing.
3. You can check either **Portrait** or **Landscape** udner **Orientation**.
4. You can enter the value into the **Left**, **Right**, **Top** and **Bottom** text fields to adjust the size of the corresponding margin.
5. Click **OK** to confirm the settings.

Adjusting margins

The Margins pane allows user to specify the margins of the pages, header and footer.



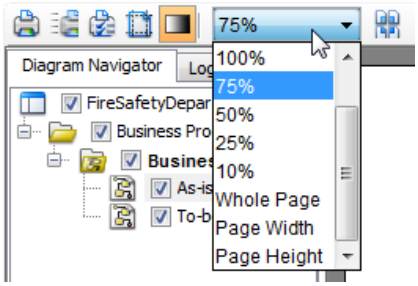
Adjusting margins

1. Click the **Adjust Margin** button on the toolbar. The margin setting page will be shown in the preview area.
2. You can edit the margins size by entering the sizes into the text fields. Alternatively, click the spinner buttons to increase/ decrease the margin sizes.
3. Click the **Finish Adjust Margin** button when you finish configuring the margin settings. The margin sizes will then be updated.

Zooming pages

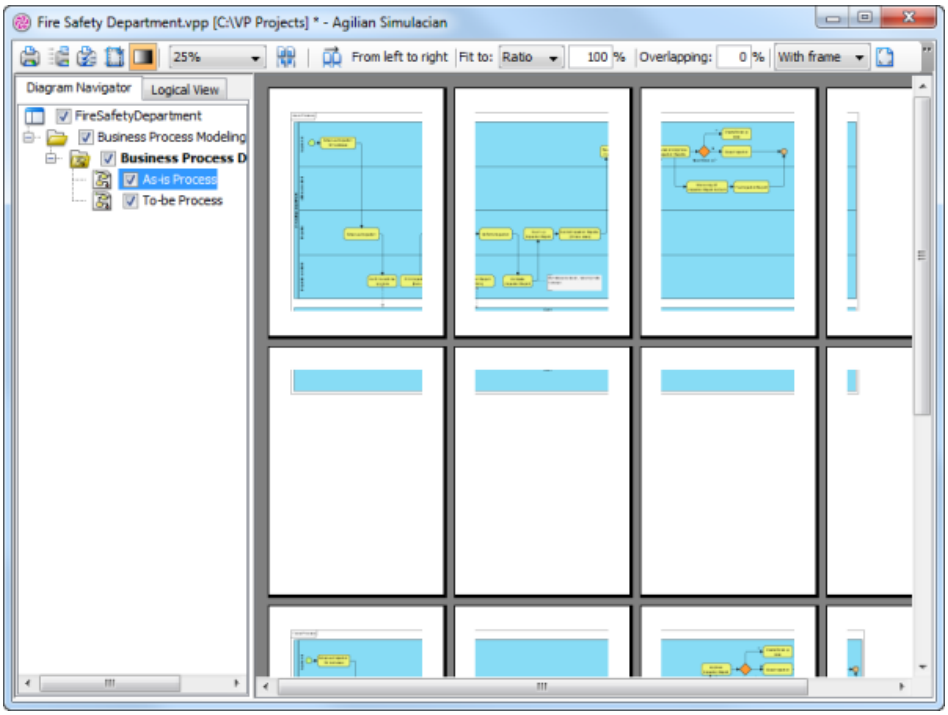
Diagrams can be zoomed in or zoomed out according to the user's preference.

1. Click the **Zoom** drop-down menu to select the desired zoom ratio.



Set zoom ratio

2. The preview area will show the diagrams in the zoom ratio that you have selected.



*Preview in **Preview** dialog box*

Selecting the preview layout

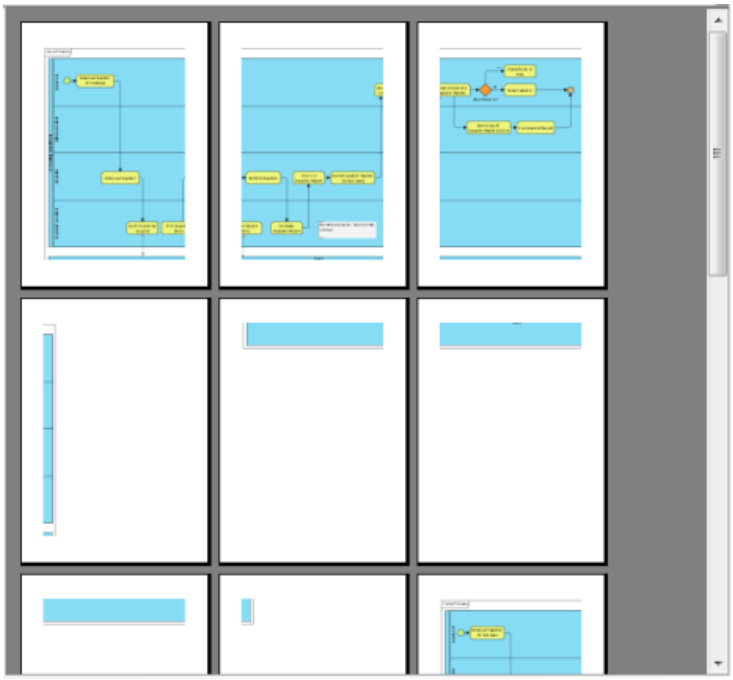
There are two layouts that you can choose for the print preview, the **Paper Base Layout** and the **Diagram Base Layout**.

If the **Fit to Pages** option is chosen and there are multiple pages in the printout, choosing **Paper Base Layout** will cause the distribution of pages to be paper-oriented (the diagram size is ignored in arranging the preview); while choosing **Diagram Base Layout** will cause the distribution of pages to be diagram-oriented.

Note that this option affects the preview only; the order of the printout remains unchanged.

To select a layout of the preview, click the **Paper Base Layout** button or **Diagram Base Layout** button on the toolbar. A pop-up menu where you can choose the layout will appear.

The preview after applying the Paper Base Layout:



Preview in paper base layout

The preview after applying the Diagram Base Layout:

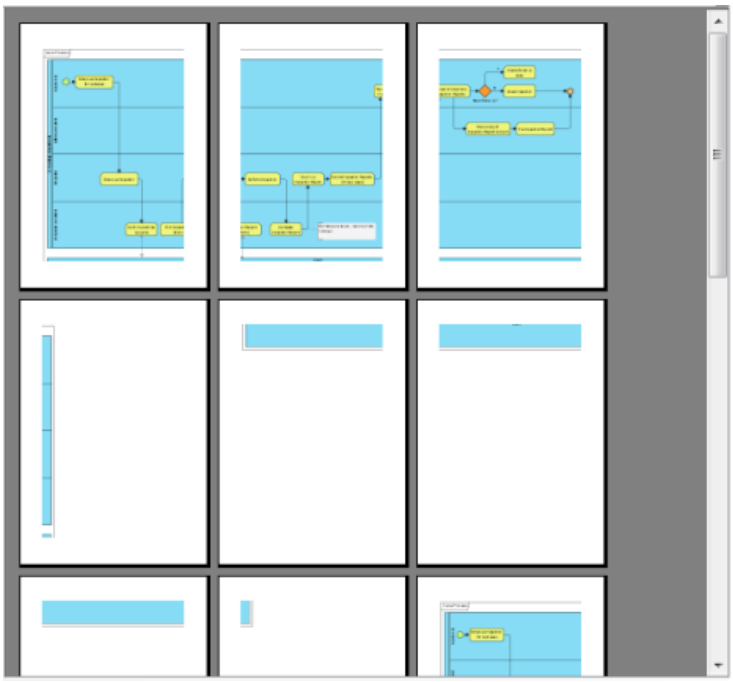
Preview in diagram base layout

Setting paper place style

You can select the paper place style to change the order of the printout. To select the paper place style, click the **Paper Place Style** button on the toolbar. A pop-up menu where you can choose a paper place style will appear.

Considering a large diagram is divided into many pages, choose **From left to right** will arrange the printout order from the pages on the left to the pages on the right, while choosing **From top to bottom** will arrange the print order from the pages on the top to the pages on the bottom.

The order of the printout after choosing **From left to right**:



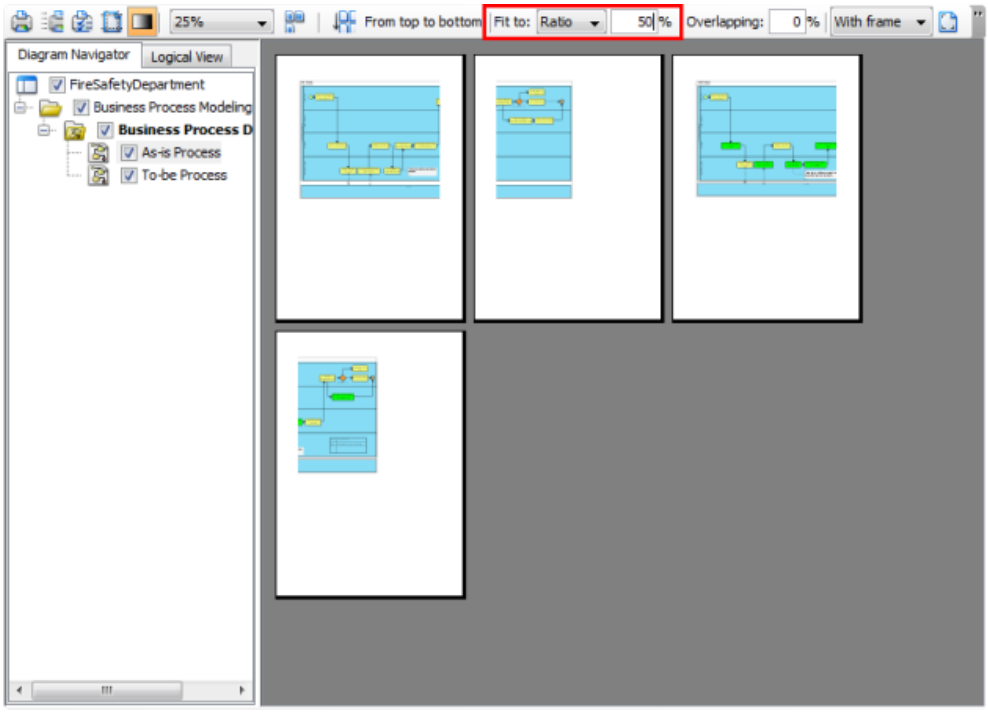
Printout order is left to right

Fit to ratio

Fit to Ratio is used to resize the diagrams in the printout to a specific ratio.

Click the **Fit** to drop-down menu and select Ratio.

You can enter the ratio into the text field. After editing the ratio, the diagrams in the printout will be resized at once.

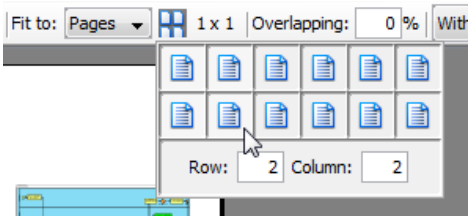


Fit to ratio

Fit to pages

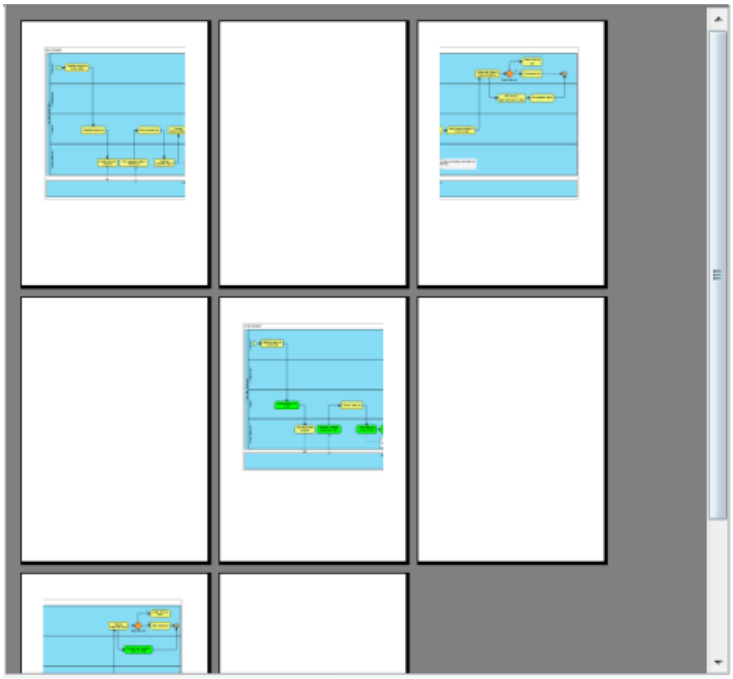
Fit to Pages is used to split the diagram to a desired number of pages when printing.

1. Click the **Fit to** drop-down menu and select pages.
2. Click the **Multiple Page Mode** button on the toolbar. The page selector will appear.



Select multiple pages

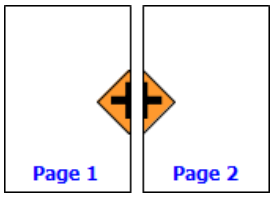
3. Click the row-column combination to select it (note that you can click and drag on the page selector to extend the selection). The diagram will be split into multiple pages by the rows and columns that you have selected.



Fit to page

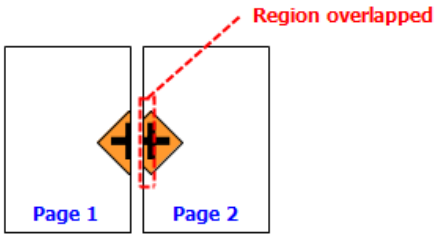
Setting the diagram overlap percentage

Overlapping is used when users want the diagrams to be overlapped at the boundaries between pages. This is particularly useful when you have a large diagram that span multiple pages and you want to stick the pages of the printout together; the overlapping area can then be used as a hint when sticking the pages.



Multiple page without overlap

1. Input the overlapping percentage and press **Enter** in **Overlapping** text field.
2. The printing area near the boundaries of the pages will be duplicated through the input value of overlapping percentage.



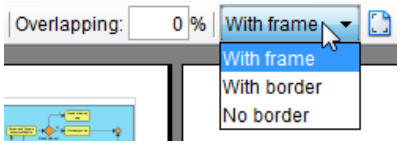
Multiple page with overlap

Printing with frame/Border option

You can print your diagram with a frame or border. There are three options available:

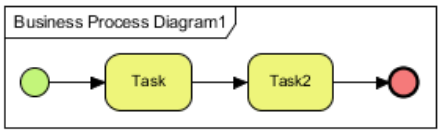
- With frame
- With border
- No border

Select **With frame**/ **With border**/ **No border** option from the drop-down menu.



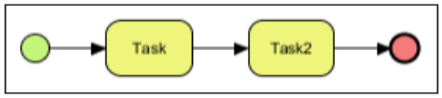
Select an option from drop-down menu

Output of printing with frame:



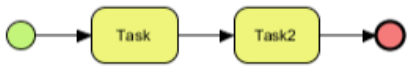
Printing with frame

Output of printing with border:



Printing with border

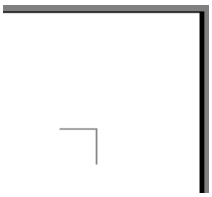
Output of printing with no border:



Printing with no border

Showing/Hiding clip marks on page

Clip marks act as an indication of the boundary of a page.

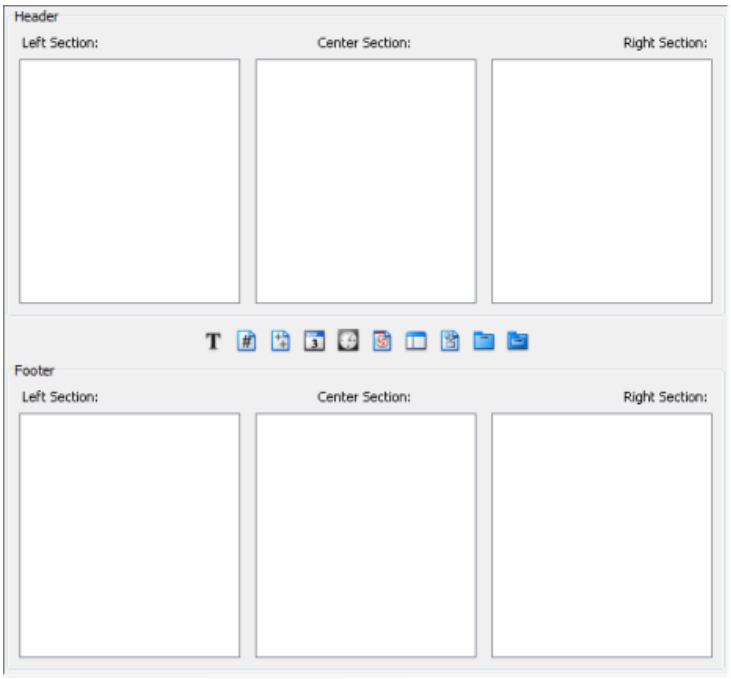


Clip marks

To show clip marks on the printout, click the **Show Clip Marks on Page** button . The boundaries of the pages will be surrounded by clip marks. To hide the clip marks, click the **Hide Clip Marks on Page** button again.

Editing header/footer of the pages

To edit the header/ footer of the printout, click the **Edit Header/Footer** button on the toolbar. You will then switch to the edit header/footer pane.



Editing header/footer of pages

You can edit the header and the footer in the Header panel and the Footer panel respectively. Each of the panel consists of the Left Section, Center Section and the Right Section, which represents the position that the content will be located in the header/footer.

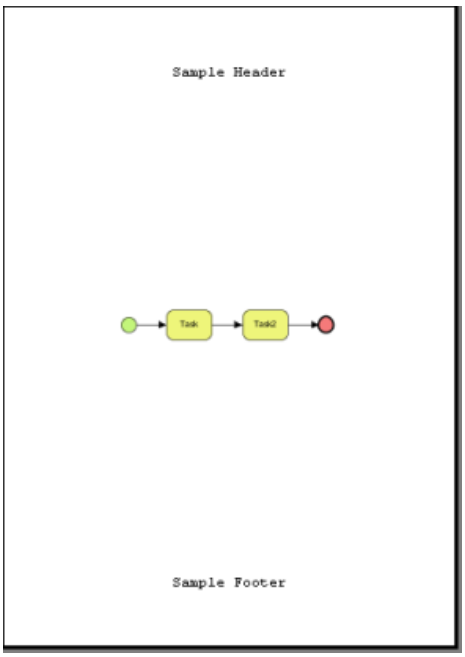
There is a toolbar between the Header panel and the Footer panel, which facilitates the editing of header/footer. The description of the buttons in the toolbar can be found in the following table:

Button	Name	Description
T	Select Font	Select the font format. Note that you have to click the section you want its font to be formatted before you start setting the font format.
#	Insert Page Number	Insert the page number. Note that you have to click the section you want page number to be inserted into before you click it.
	Insert Number of Pages	Insert the total number of pages. Note that you have to click the section you want the number of pages to be inserted into before you click it.
	Insert Date	Insert the date that the printing starts. Note that you have to click the section you want the date to be inserted into before you click it.
	Insert Time	Insert the time that the printing starts. Note that you have to click the section you want the time to be inserted into before you click it.
	Insert File Name	Insert the file name of the Agilian project. Note that you have to click the section you want the file name to be inserted into before you click it.
	Insert Project Name	Insert the name of the Agilian project. Note that you have to click the section you want the project name to be inserted into before you click it.
	Insert Diagram Name	Insert the diagram name. Note that you have to click the section you want the diagram name to be inserted into before you click it.

	Insert Parent Package	Insert the parent package. Note that you have to click the section you want the parent package to be inserted into before you click it.
	Insert Parent Hierarchy	Insert the parent hierarchy. Note that you have to click the section you want the parent hierarchy to be inserted into before you click it.

The description of editing of header/ footer toolbar

After you have finished editing the header/footer, click the **Close Edit Header/Footer** button to switch to the print preview mode. A sample page with the header and footer is shown in the figure below:

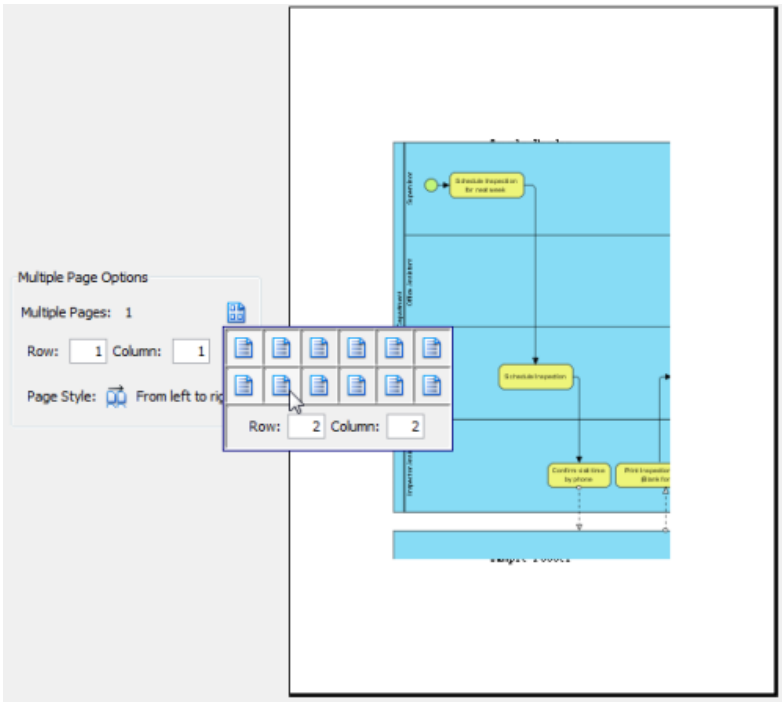


Page with header and footer

The multiple page mode

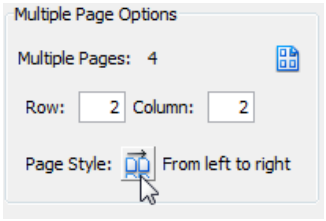
The Multiple Page Mode allows users to configure how the diagrams should be distributed in multiple pages. To switch to the Multiple Page Mode, click the **Multiple Page Mode** button on the toolbar.

Click the icon behind **Multiple Pages** will pop the page selector out, where you can select the row-column combination for the printout. Alternatively, you can type in the **Row** and **Column** text field directly.



Select multiple pages

Click the icon behind **Page Style** to change the printout order. Considering a large diagram is divided into many pages, choosing **From left to right** will arrange the printout order from the pages on the left to the pages on the right, while choosing **From top to bottom** will arrange the print order from the pages on the top to the pages on the bottom.



Distribute diagram in multiple page

After you have finished configuring the multiple page settings, click the **Close Multiple Page Mode** button to close it.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Printing a Diagram with Quick Print

In Agilian, you can print a diagram with simple and easy setting by using quick print feature. The Quick Print feature allows you to print diagrams without previewing them, hence speeding up the print job.

Printing with quick print

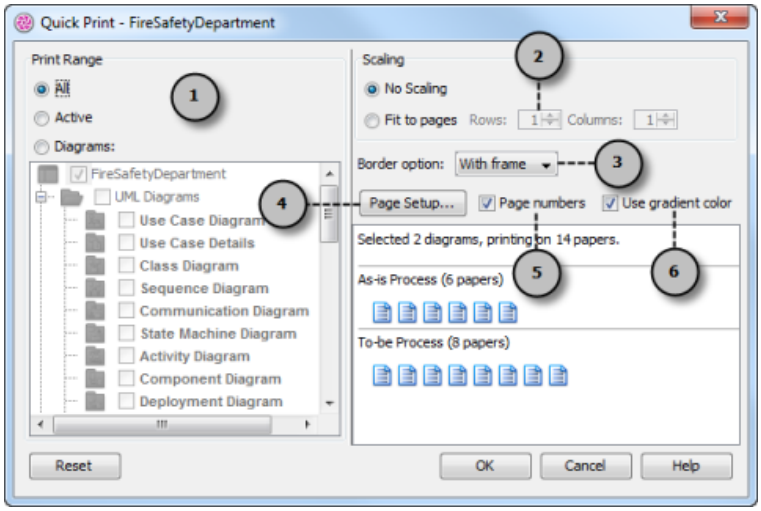
To perform quick print:

1. Select **File > Quick Print...** from main menu.

> **NOTE:** Alternatively, you can select **Print > Quick Print...** on the toolbar to launch quick print.

2. In the **Quick Print** dialog box, select printout setting for the diagram.
3. Click **OK** button to proceed printing.

The overview of Quick Print



***Quick Print** dialog box*

No.	Name	Description
1	Print Range	Click on either of the below options to specify the print range: • All - Print all the diagrams within the current project. • Active - Print only the active diagram. • Diagrams - Check from the diagram tree to select the diagram(s) for printing.
2	Scaling	Select **No scaling** to print with diagrams' original size. Numbers of pages used for each diagram are subject to the scale of diagrams. Select **Fit to pages** to print with specified number of pages per diagram with respect to the specified number of rows and columns.
3	Border option	Select border option of printout.
4	Page Setup...	It allows you to specify the page size, the orientation as well as the margins of the pages.
5	Page numbers	Select it to print diagrams with page number.
6	Use gradient color	Select it to use gradient color in printout.

Details of quick print dialog box

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Team collaboration in Agilian

Team collaboration is the practice of working in a team instead of by one person. Team members can work individually on their own specific parts of a project, and eventually combine works together to form a complete project. This chapter provides you with clear information on Agilian's team collaboration support.

Introduction to team collaboration

Describes the key concepts about team collaboration like administration, commit, update, revert, etc.

The teamwork client

Teamwork Client, where you can manage, checkout and open projects, involves all teamwork activities that you can perform with.

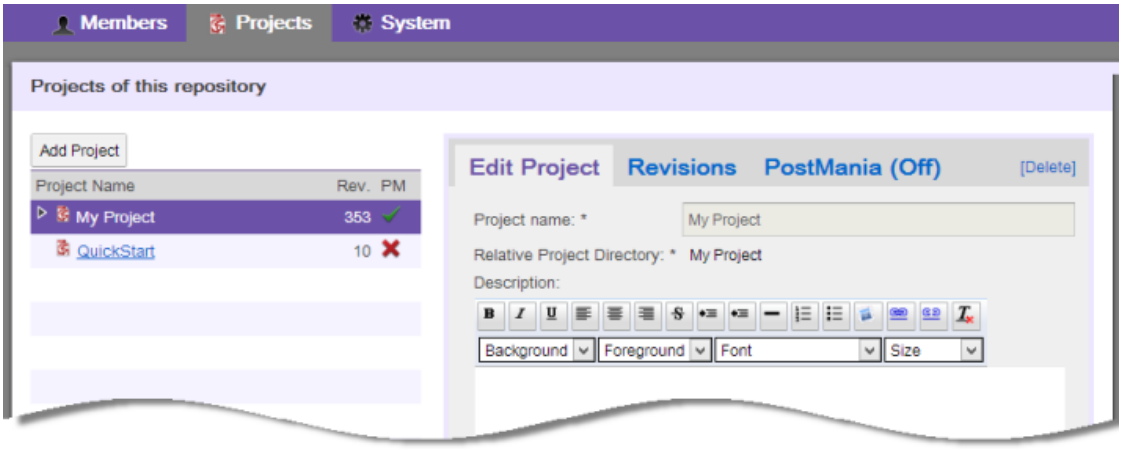
Introduction to team collaboration

Team collaboration is the practice of working in a team instead of by one person. Team members can work individually on their own specific parts of a project, and eventually combine works together to form a complete project. As a result of collaborations between members by employing the unique skills of each, works can be done more effectively and in higher quality.

Agilian's team collaboration support provides access to a central repository for managing, sharing and versioning projects. You can let team members get project from repository, start working on their own parts, let them commit (i.e. upload) their work to server, and update others' works. Visual Paradigm VPository, VP Teamwork Server, SVN, CVS, ClearCase and Perforce are the supported standards of versioning systems. This chapter outlines some of the key concepts in team collaboration in Agilian.

Administration

Administration is the process of setting up projects and members, and deciding the access rights of members against projects. Administration is a must in order to start working.



Edit project

Checkout and open project

Checkout project is a process done by team members, for getting a project from repository to start working with. Team members should login into the server, then checkout the project(s) to work with, provided that they have the permission to do so, as granted by administrator. After that, open to start working on it.

Commit

Commit is the process of uploading changes done in working copy to server. As team members make changes in a project, they can share their works by committing those changes to the server. Agilian will try to merge changes from working copy to server copy. When merging, there may be conflict when any changes a team member made cause an unresolvable contradiction with changes made by others. Team member is required to decide whether to keep his/her change (i.e. overwrite) or to take the co-worker’s change (i.e. revert). All conflicts must be resolved in order to proceed with committing.

Update

Update is the process of refreshing the working copy by merging changes that others had made and committed to server before. Similar to commit, update is a process of merging differences instead of overwriting. If your changes overlap the changes others had made, you will be asked to resolve conflict. All conflicts must be resolved in order to proceed with updating.

Conflict

Conflict is a situation that happens when committing or updating. It occurs during the merging between working and server copy of project, when a contradiction is detected between them. For example, a team member has renamed a shape from *A* to *B* and has committed the change. Then, another team member has renamed the same shape from *A* to *C* and attempt to commit. Due to difference in the name of shape, a conflict is occurred. Whenever a conflict occurred, you have to resolve it or else to abort before commit/update operation.

Branching

Branching is the process of creating a branch from trunk (i.e. main development flow), for isolating changes that are not supposed to be available on trunk, either at the moment or permanently. By working in a branch, team members can carry out half-broken or risky changes without worrying the risk of damaging the work in trunk. Once the works done in branch is examined, and confirmed alright, team member can make changes available in trunk through merging. Merging can also be done from trunk to branch to ensure that the branch is always up-to-date.

Tagging

Tagging is the process of producing a snapshot (i.e. tag) of a project in time. People often create tags for archiving releases of works. Therefore, tags are often named as release-1.0 where 1.0 is the number of version. Since a tag is a snapshot, team members can never commit under a tag.

Revision history

Every time a team member performs a commit with success, a new revision is created as a snapshot of project. More and more revisions will be created through repeated committing. A list of revisions that shows the changes of project is called the revision history. In Agilian, you can review the works done in specific revision(s) by exporting them in project files. You can identify the differences between revisions by comparing them.

Revert changes

Revert is the process of undoing changes. In Agilian's team collaboration support, there are two kinds of revert actions that you can perform. The first one is to revert locally modified changes to make the working copy back to the original state. Another revert action is for undoing changes made in revisions. Team members can undo changes made in revisions by reverting them.

Related Resources

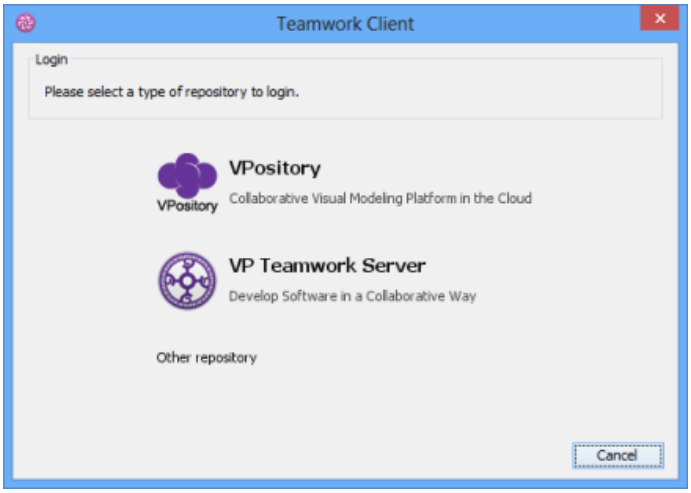
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

The Teamwork Client

Teamwork Client is where you can manage your projects. It enables you to select/de-select projects to work on with, and to provide you with access to all team operations like commit, update, branching, tagging, etc. To open **Teamwork Client**:

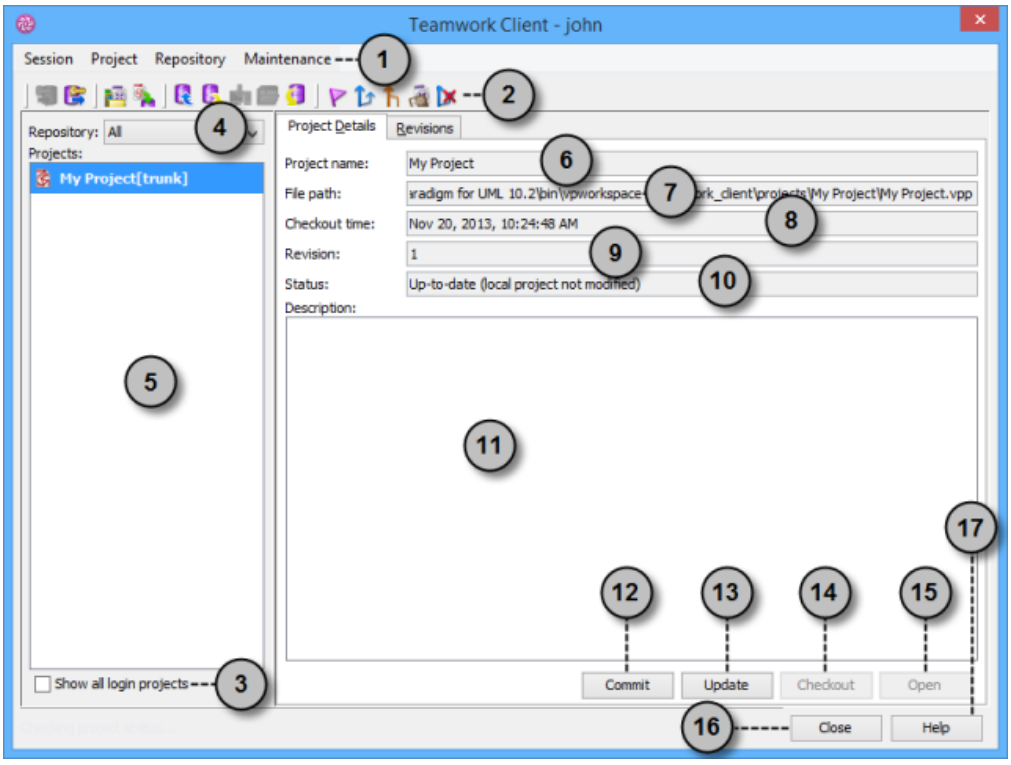
1. Select **Teamwork > Open Teamwork Client...** from the main menu of Agilian. You are then prompted to select the type of repository to login.



Select repository type

2. Here are the types of version control system and collaboration platforms supported: VPository , Visual Paradigm Teamwork Server , SVN, CVS, Perforce and ClearCase. Contact your team leader or system administrator about the type of repository you are using and then choose the right one to connect to.
3. You are then prompted to provide the connection settings. Enter the connection settings and click **Login** at bottom right to connect to the repository selected.
4. This should show the **Teamwork Client** window. But if this is the first time you login to the server, you are prompted the **Open Project** window with available projects listed. You have either or both read and update right to these projects. You can start working on a project by selecting it and click **Open** at bottom right, or simply by double clicking on the project thumbnail.

Overview of Teamwork Client window



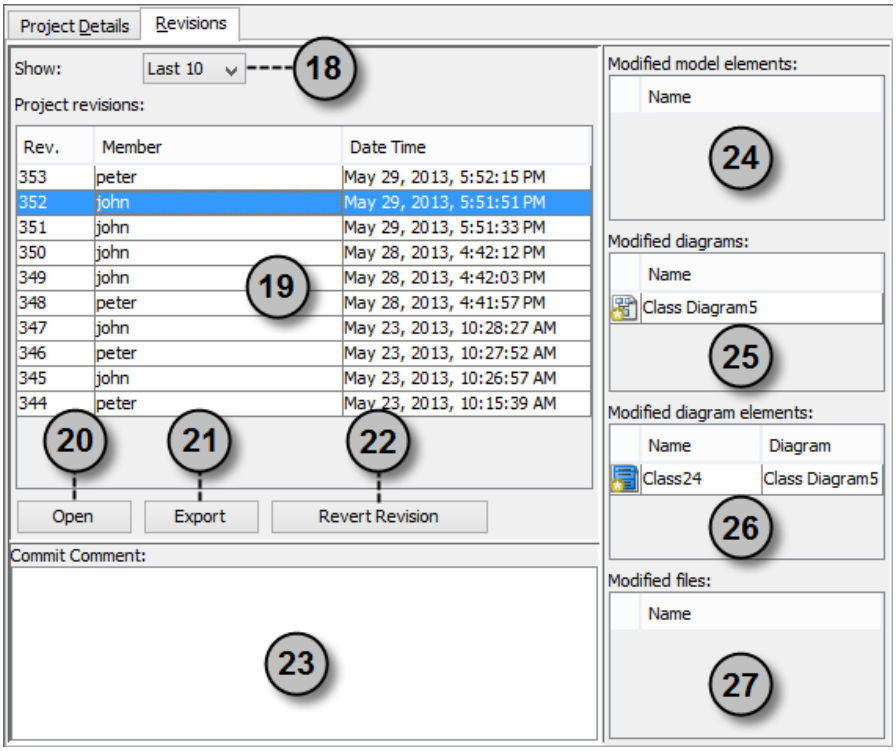
*The **Teamwork Client** window*

No.	Name	Description
1	Main menu	**Session**: It is a period of active connection with server. **Login**: Login the server. After you choose it, you will be able to execute all actions. **Logout**: Logout the server. After you choose it, you will not be able to execute any actions.

		Project: Provides an access to main functions, such as commit and update. **Manage Project**: Select a project that you get involved in. **Import Project to Repository**: Import a new project to the server. **Commit**: Commit your current modified project to the server. **Update**: Update the latest copy of project from the server to your computer. **Checkout**: Click it to checkout the project selected in Projects list. It will be disabled when the selected project has already been checked out. **Open**: Click it to open the checkout project on your computer. **Tag**: Create a new tag for your current project. It allows you to produce a static release version of project. **Branch**: Create a new branch for your current project. It becomes a duplication of project to perform isolated changes. **Merge**: Combine the selected branch(es) with the trunk (main project). When some changes made in branch, it will be made in trunk as well. **Switch**: Switch from a branch/ tag to another branch/ tag or from the trunk (main project) to a branch/ tag and vice versa. **Delete Branch**: Select a branch to delete, for preventing accidental modifications in branch. **Reset Password**: Reset your account's password. **Revert Local**: Click it to undo un-committed changes made on the local project copy. **Repository**: **Synchronize Design Pattern to Server**: Synchronize the template files stored in workspace with those stored in repository. When a conflict occurs, you will be asked which design template files to keep. **Maintenance:** All the functions under the Maintenance menu are prepared for diagnosis purposes. You should not run them unless you are requested by Visual Paradigm support team. And when you are requested to execute any maintenance function, you will be briefed.
2	Toolbar	**Login**: Log into the server. After you choose it, you will be able to execute all actions. **Logout**: Log out the server. After you choose it, you will not be able to execute any actions. **Manage Project**: Select a project that you get involved in. **Import Project to Repository**: Import a new project to the server on the list. **Update**: Update the latest copy of project from the server to your computer. **Commit**: Commit your current modified project to the server. **Checkout**: Click it to checkout the project selected in Projects list. It will be disabled when the selected project has already been checked out. **Open**: Click it to open the checkout project on your computer. **Revert Local**: Click it to undo un-committed changes made on the local project copy. **Check for Update**: Click it to check whether the project is up-to-date or not. **Tag**: Create a new tag for your current project. It allows you to produce a static release version of project. **Branch**: Create a new branch for your current project. It becomes a duplication of project to perform isolated changes. **Merge**: Combine the selected branch(es) with the trunk (main project). When some changes are made in branch, it will be made in trunk as well. **Switch**: Switch from a branch/ tag to another branch/ tag or from the trunk (main project) to a branch/ tag and vice versa. **Delete Branch**: Select a branch to delete, for preventing accidental modifications in branch.
3	Show all login projects	By checking it, projects that have been checked out before will all be displayed. By unchecking it, only projects checked out by the current member will be displayed.
4	Repository	It refers to the list of available project(s). Select **All** from the drop-down menu means all projects managed by all eligible members who have logged into server in this workspace will be listed. On the other hand, the project(s) managed by a specific member can be selected from the drop-down menu. If you uncheck **Show all login projects** and do not select the current member in **Repository**, no project will be listed.
5	Projects	It lists the project(s) you selected to manage.
6	Project name	The name of selected project.
7	File path	The path of the selected project file. It is shown only when project is checked out from the server.
8	Checkout time	It displays the date and time of your first checkout for the project.

9	Revision	It displays the revision of your local project copy. Note that the revision here does not always mean the latest revision on the server.
10	Status	It displays the status of selected project, such as "Not Checked Out" will be shown when the project has not been checked out yet.
11	Comment	It shows the textual description of selected project written by administrator when creating project.
12	Update	Update the latest project from the server to your computer.
13	Commit	Commit your current modified project to the server.
14	Checkout	Click it to checkout the selected project.
15	Open Project	Click it to open the checkout project on your computer. If the project has not checked out yet, it will perform a checkout prior to opening project.
16	Close	Click to close the **Teamwork Client**.
17	Help	Click it to get assistance from help system.

*The description of the **Teamwork Client** window*



*The **Revisions** tab of the Teamwork Client*

No.	Name	Description
18	Show drop-down menu	Select the number of latest project revision to view from the drop-down menu.
19	Project revisions	It lists all the latest project revisions. The number of revisions is in accordance with the show drop-down menu.
20	Open	Click it to open the selected revision of project.
21	Export	**Export selected revisions**: Export the selected revision(s) to a folder. **Export all revisions from repository**: Export all the projects in repository to a folder.
22	Revert Selected	Undo changes committed by the selected revisions.
23	Commit Comment	A textual description of commit given by you or your teammates before committing.
24	Modified model elements	It displays the modified model elements of the selected revision.
25	Modified diagrams	It displays the modified diagrams of the selected revision.
26	Modified diagram elements	It displays the modified diagram elements of the selected revision.
27	Modified files	It displays the modified Teamwork Files of the selected revision.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Basic Team Collaboration Features in Agilian

Check out the team collaboration features supported by Agilian.

Checkout project

Checkout project is a process that you download a managed project from repository to your computer and start working with.

Commit

Commit refers to the process of uploading local modifications to the server.

Support commit part of project to Teamwork

Other than committing the whole project, you may commit specific model elements/diagrams to Teamwork.

Update

Update is the process of refreshing your current copy by merging changes that others have made and committed previously to server.

Revert local modification

Since VP teamwork collaboration allows the feature of revert, you can undo all modifications you made in the local project copy when the changes haven't been committed yet.

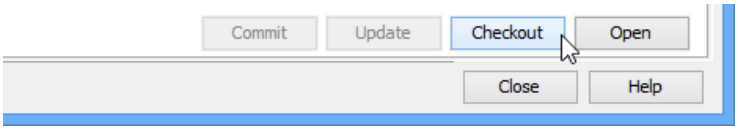
Import project

Import project refers to the process of adding project file in server so that team members can check that out and start working on it.

Checkout and open project

Checkout project is a process done by team members, for getting a project from repository to start working with. Team members should login into the server, then checkout the project(s) to work with, provided that they have the permission to do so, as granted by administrator. After that, open to start working on it.

1. Select **Teamwork > Open Teamwork Client...** from the main menu.
2. In **Teamwork Client**, select **Project > Manage Project...** from the menu.
3. When the **Manage Project** window pops out, select the project you are going to take part in and click **>** button. Click **OK** to return to **Teamwork Client**.
4. Select the project from the project list on the left hand side and click **Checkout**.



Checkout project

5. Click **Open** to open to project and start to work on it.

Checkout multiple projects

Instead of checking out a single project, you can checkout multiple project files at the same time, for those listed in the **Teamwork Client** window as managed projects.

1. Select **Teamwork > Open Teamwork Client...** from the main menu.
2. From the list on the left hand side, select the projects to checkout. You can perform a multiple selection by first pressing the **Ctrl** key.
3. Right click on the selection and select **Checkout** from the popup menu.

Related Resources

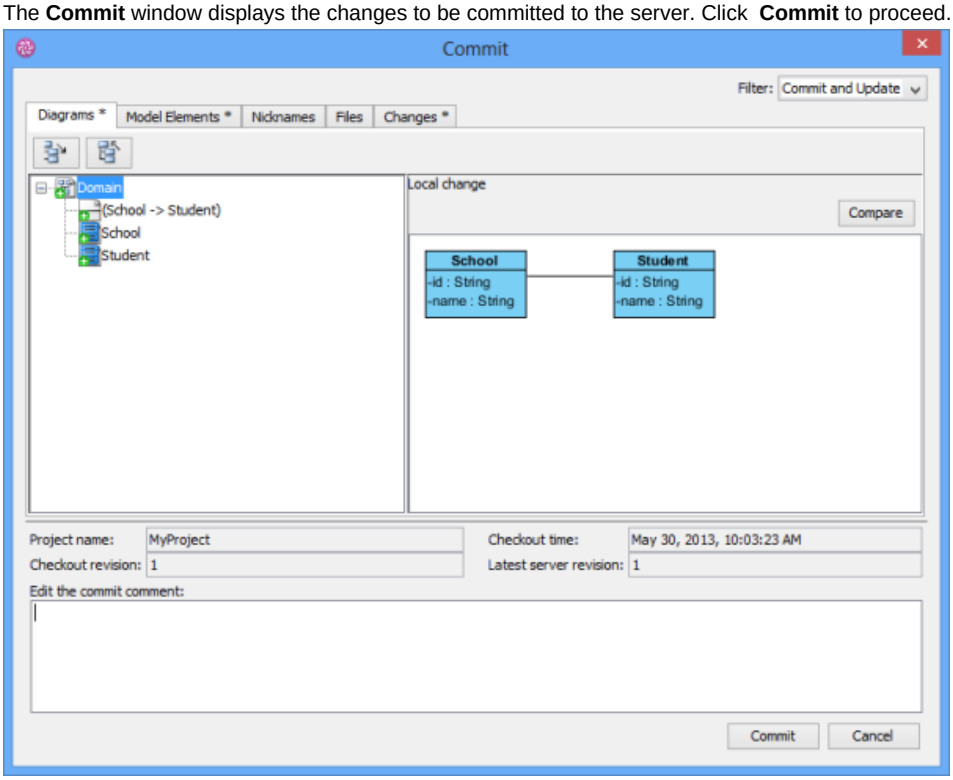
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Commit

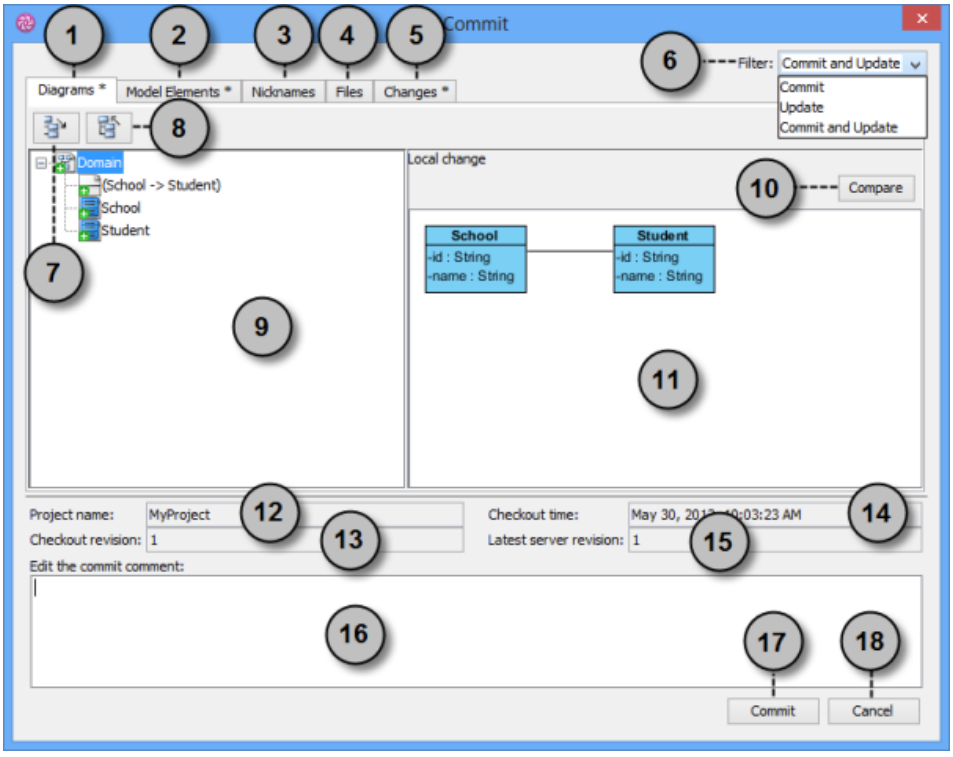
Commit refers to the process of uploading local modifications to the server. As team members make changes in a project, they can share their works by committing those changes to the server. Agilian can help to merge changes from working copy to server copy. During merging, a conflict may be caused when there is a contradiction between team members. Decision have to be made whether to keep the current modified copy (i.e. overwrite) or to accept others' copy (i.e. revert). All conflicts have to be solved before proceeding to commit. To commit changes:

1. Select **Teamwork > Commit...** from the main menu.
2. If the change you made contradicts the change made by another team member, this will result in a conflict. You must resolve all the conflicts in order to continue. For details, read the Resolving conflicts section below. Clear the conflicts, if any, and continue.
3. The **Commit** window displays the changes to be committed to the server. Click **Commit** to proceed.



*The **Commit** window*

Overview of Commit window



*The **Commit** window*

No.	Name	Description

1	Diagrams tab	The diagram level changes to be performed when you execute commit.
2	Model Elements tab	The model element level changes to be performed when you execute commit.
3	Nicknames tab	The changes of nickname to be performed when you execute commit.
4	Files tab	The file changes to be performed when you execute commit.
5	Changes tab	All the changes to be performed when you execute commit.
6	Filter	When you commit, local changes will be merged to the server copy and meanwhile, changes in server copy will be merged to the local copy, too. The filter let you filter the tree on the left hand side to list the commit changes, which are changes to perform on server copy, and/or update changes, which are changes to perform on local copy.
7	Expand All	Expand the tree nodes in the tree below.
8	Collapse All	Collapse the tree nodes in the tree below.
9	Tree	List out the changes to be performed when you execute commit.
10	Compare	Click this button to compare local and server copy side by side.
11	Preview	The preview of the element as selected in the tree on the left hand side.
12	Project name	Name of current project.
13	Checkout revision	The number of current checkout revision.
14	Checkout time	The time for latest checkout.
15	Latest server revision	The number of latest revision in the server.
16	Commit comment	You can give a comment for your current commit by typing here.
17	Commit	Proceed committed
18	Cancel	Cancel committing and close the window.

*The description of **Commit** window*

Committing multiple projects

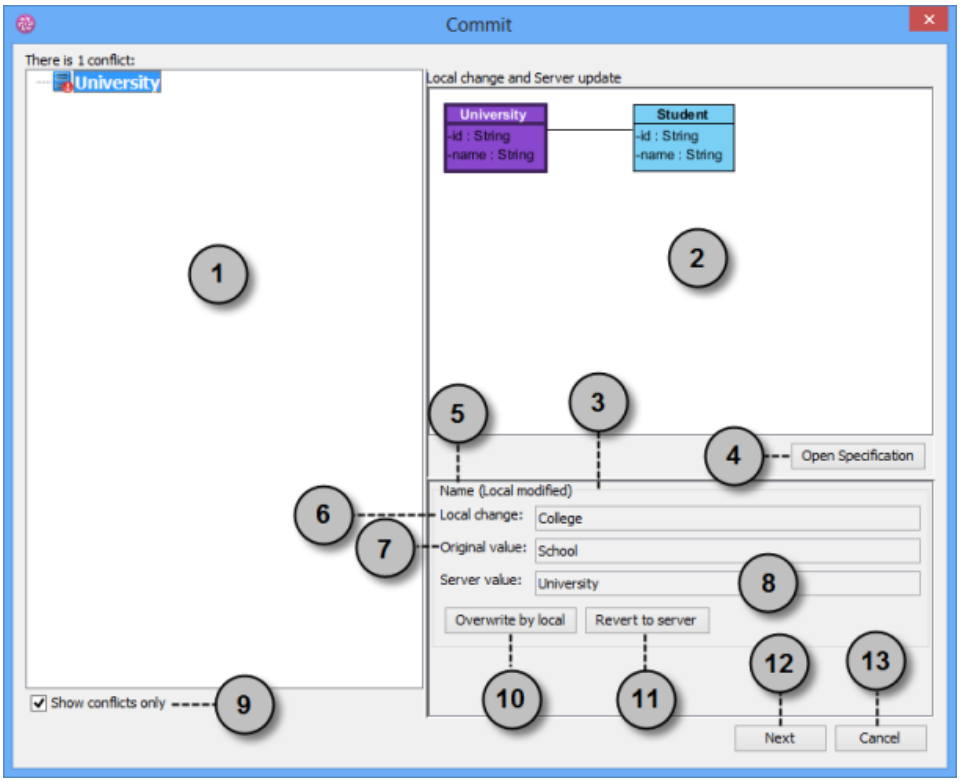
Instead of commtting a single project, you can commit multiple project files at the same time, for those listed in the **Teamwork Client** window as managed projects.

1. Select **Teamwork > Open Teamwork Client...** from the main menu.
2. From the list on the left hand side, select the projects to commit.
3. Right click on the selection and select **Commit...** from the popup menu.

Resolving conflict

If the change you made contradicts the change made by another team member, this will result in a conflict. For example, your colleague has renamed a class from School to University and performed a commit, and then you rename the same class to College and perform a commit. This produces a conflict.

When a conflict occur, you must resolve it in order to continue committing. You have to resolve conflict either by overwriting or reverting the change. Overwrite means to adopt the server copy's change, while Revert means to adopt the local's change.



Conflicts when committing

No.	Name	Description
1	List of change (action)	List of changes to be performed. Initially only conflicted changes are listed. You can uncheck **Show conflicts only** to list all changes.
2	Preview	The preview of the element as selected in the tree on the left hand side.
3	Conflicted properties	The properties that cause conflicts are listed in this panel.
4	Open Specification	Click on this button to open the specification of element selected in the tree on the left hand side.
5	Property name	The names of conflicted properties.
6	Local change	The value of property in local project copy.
7	Original value	The value of property before changed. In other words, it is the value in checkout copy.
8	Server value	The value of property in server project copy.
9	Show conflicts only	Check it to list only conflicted changes in the tree on the left hand side. Uncheck it to list all changes.
10	Overwrite by local	Click on this button to adopt the source copy.
11	Revert to server	Click on this button to adopt the target copy.
12	Preview	Click this button to continue committing.
13	Cancel	Cancel committing and close the window.

*The description of **Commit** window when have conflicts*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Commit Part of a Project

In addition to committing all the changes you have made within the session, you can commit change(s) made on specific model element(s), shape(s) or diagram(s). This way of committing is called a *partial commit*. To work with partial commit, simply right click on the project data you want to commit and **Teamwork > Commit...** from the popup menu. You may select multiple items (e.g. two shapes) and commit them at once.

Related Resources

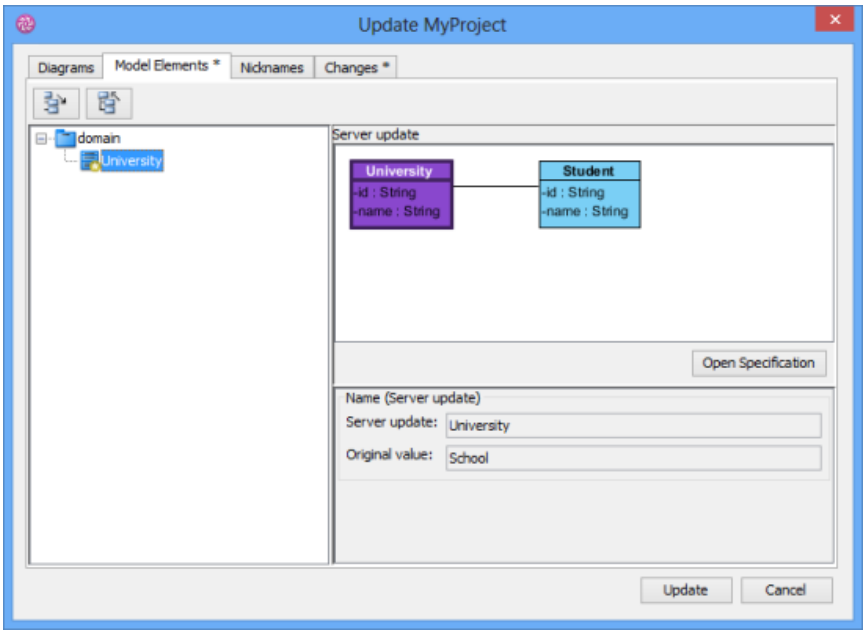
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Update

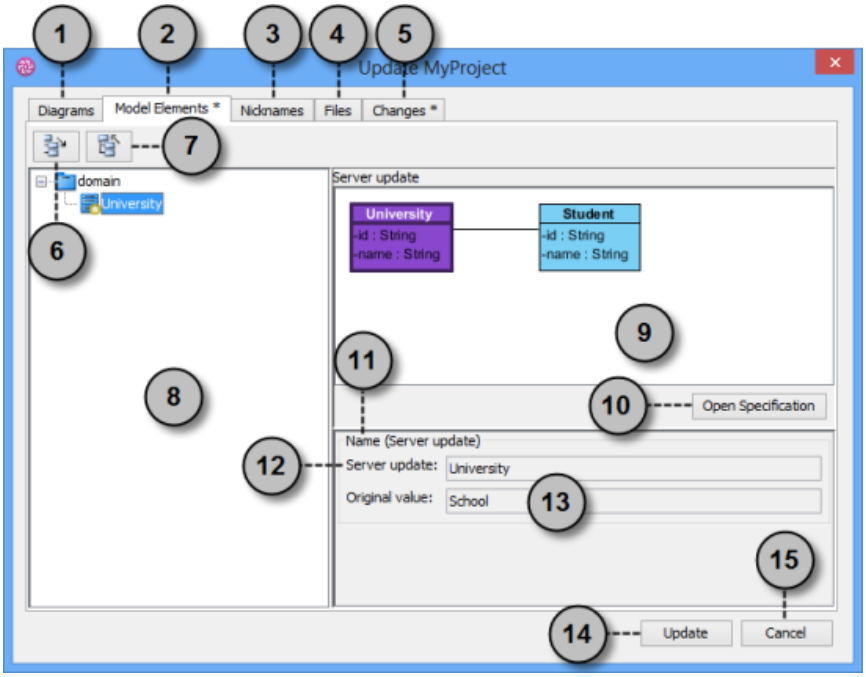
Update is the process of refreshing your current copy by merging changes that others have made and committed previously to server. Similar to commit, update is a process of merging differences instead of overwriting. When any of your changes contradicts the changes others have made, you will be asked to resolve conflict. All conflicts have to be resolved before proceeding with updating. To update:

1. Select **Teamwork > Update** from the main menu.
2. If the change you made contradicts the change made by another team member, this will result in a conflict. You must resolve all the conflicts in order to continue. For details, read the Resolving conflicts section below. Clear the conflicts, if any, and continue.
3. The **Update** window displays the changes to be made upon updating. Click **Update** to proceed.



*The **Update** window*

Overview of Update window



*The **Update** window*

No.	Name	Description
1	Diagrams tab	The diagram level changes to be performed when you execute update.
2	Model Elements tab	The model element level changes to be performed when you execute update.
3	Nicknames tab	The changes of nickname to be performed when you execute update.
4	Files tab	The file changes to be performed when you execute update.

5	Changes tab	All the changes to be performed when you execute update.
6	Expand All	Expand the tree nodes in the tree below.
7	Collapse All	Collapse the tree nodes in the tree below.
8	Tree	List out the changes to be performed when you execute update.
9	Preview	The preview of the element as selected in the tree on the left hand side.
10	Open Specification	Click on this button to open the specification of element selected in the tree on the left hand side.
11	Property name	The name of conflicted property.
12	Server update	The value of property in server project copy.
13	Original value	The current value of property (before updating).
14	Update	Proceed updating
15	Cancel	Cancel updating and close the window.

*The description of **Update** window*

Updating multiple projects

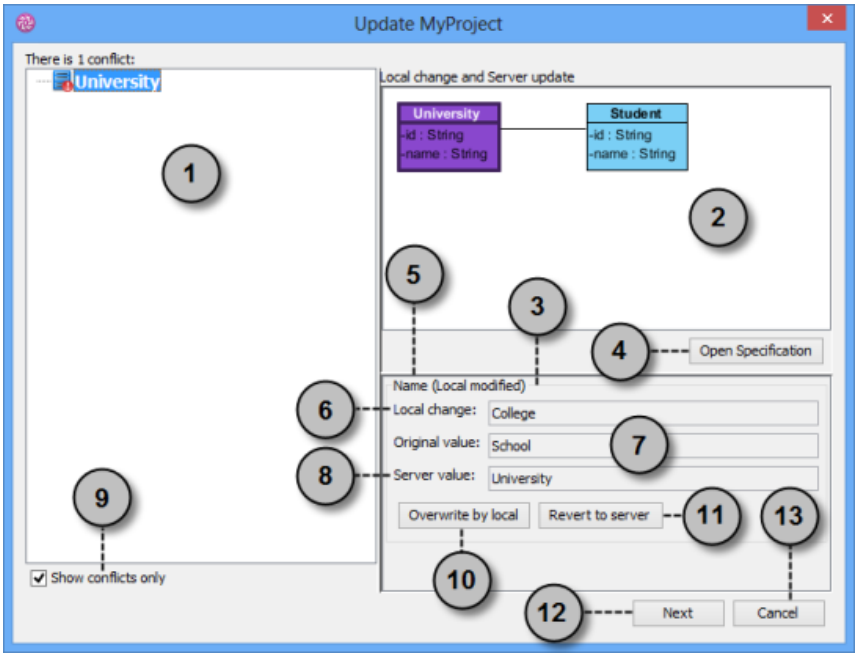
Instead of updating a single project, you can update multiple project files at the same time, for those listed in the **Teamwork Client** window as managed projects.

1. Select **Teamwork > Open Teamwork Client...** from the main menu.
2. From the project list on the left hand side, select the projects to update.
3. Right click on the selection and select **Update...** from the popup menu.

Resolving conflict

If the change you made contradicts the change made by another team member, this will result in a conflict. For example, your colleague has renamed a class from School to University and performed a commit, and then you rename the same class to College and perform an update. This produces a conflict.

When a conflict occur, you must resolve it in order to continue updating . You have to resolve conflict by overwriting or reverting the conflicted change. Overwrite means to adopt the server copy's change, while Revert means to adopt the local's change.



Conflicts when updating

No.	Name	Description
1	List of change (action)	List of changes to be performed. Initially only conflicted changes are listed. You can uncheck **Show conflicts only** to list all changes.
2	Preview	The preview of the element as selected in the tree on the left hand side.
3	Conflicted properties	The properties that cause conflicts are listed in this panel.

4	Open Specification	Click on this button to open the specification of element selected in the tree on the left hand side.
5	Property name	The name of conflicted property.
6	Local change	The value of property in local project copy.
7	Original value	The value of property before changed. In other words, it is the value in checkout copy.
8	Server value	The value of property in server project copy.
9	Show conflicts only	Check it to list only conflicted changes in the tree on the left hand side. Uncheck it to list all changes.
10	Overwrite by local	Click on this button to adopt the source copy.
11	Revert to server	Click on this button to adopt the target copy.
12	Next	Click this button to continue updating.
13	Cancel	Cancel updating and close the window.

*The description of **Update** window when have conflicts*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Revert local modifications

You can undo all the non-committed modification you made in the local project copy. The operation of giving up non-committed modifications is called *Revert*. To revert:

1. Select **Teamwork > Revert Local** from the main menu.
2. Click **Yes** when you are asked for confirmation.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Import project to server

Import project refers to the process of adding project file in server so that team members can check that out and start working on it.

Ways to import project

There are multiple possible ways you can take to import project, depending on the type of repository you use. For Teamwork Server users, project can be imported both in the server, via its Web interface, and through the client. For the other servers, import can only be done through client. Note that you cannot manually place a project file in server to imitate the import process. You must import the project step by step, as described in this page. Take a look at the following table to see the supported ways of imrporting project.

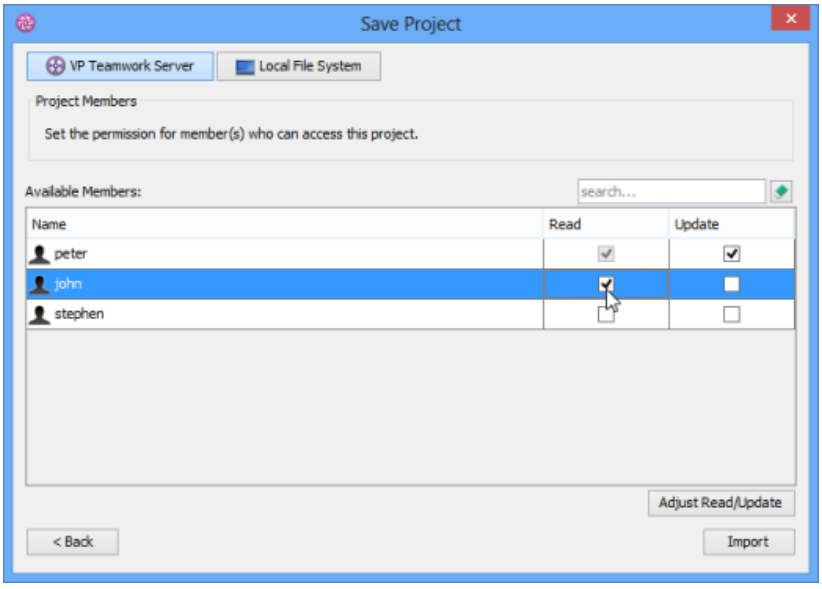
Server type	VPository	VP Teamwork Server	Other (SVN/CVS/Perforce/ClearCase)
Import via project saving	●	●	
Import via Web interface	●	●	
Import via Teamwork Client	●	●	●

Supported ways of importing project

Import project via project saving

This methods allow you to save the currently opening project into the server. Perform the steps below to import the opening project to server.

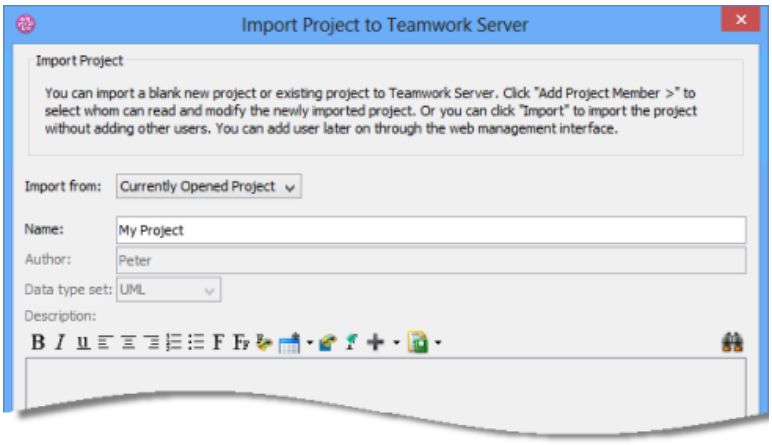
1. Select **File > Save Project as...** from the main menu.
2. In the **Save Project** window, keep **VPository/ VP Teamwork Server** selected at the top of the window. If necessary, change the project name.
3. The **Enable PostMania** option allows you to enable the social networking platform bundled by VPository/VP Teamwork Server. If you want to let you and your teammates view and comment on your design via PostMania, keep the option enabled.
4. Click **Import** to immediately create the opening project in server. By doing so you will be the only one who can access the project. But of course, you or the manager can add other members to the project later on, via the web interface.
5. If you want to assign members to your project now, click **Add Project Member>**. In the **Project Members** screen, grant either or both the **read** and **update** rights to the other members/groups. The **Read** permission means that member can only checkout the project and read its content. The **Update** permission means that member can both read the project content and commit changes to server. When ready, click **Import** to continue.



Granting member the read right to a new project

Import project via Teamwork Client (For VPository/Teamwork Server users)

1. Select **Teamwork > Open Teamwork Client...** from the main menu.
2. In the **Teamwork Client** window, select **Project > Import Project to Repository** from the main menu.
3. In the **Import Project window**, select the source of project file:
   Currently opened project - This will import the opening project to server as a new project.
   Blank new project - This will import a blank new project as a new project. If you select this option, enter/modify the author, data type set and description of project.
   Import existing project - This will start with an existing .vpp project file. If you select this option, click **...** and choose the .vpp file to import to server.

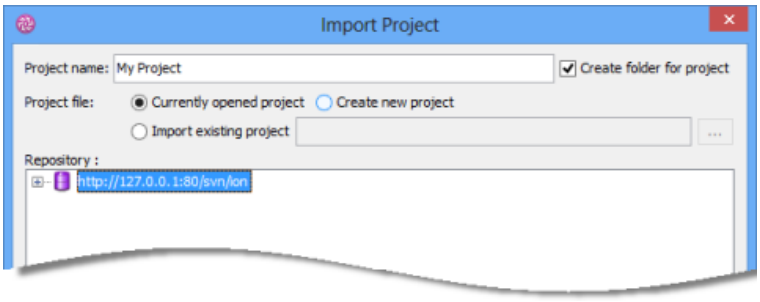


Import project

4. Enter/modify the project name.
5. Click **Import** to immediately create the opening project in server. By doing so you will be the only one who can access the project. But of course, you or the manager can add other members to the project later on, via the web interface.
6. If you want to assign members to your project now, click **Add Project Member>**. In the **Project Members** screen, grant either or both the **read** and **update** rights to the other members/groups. The **Read** permission means that member can only checkout the project and read its content. The **Update** permission means that member can both read the project content and commit changes to server. When ready, click **Import** to continue.

Import project via Teamwork Client (For SVN/CVS/Perforce/ClearCase users)

1. Select **Teamwork > Open Teamwork Client...** from the main menu.
2. In the **Teamwork Client** window, select **Project > Import Project to Repository** from the main menu.
3. In the **Import Project window**, enter the project name.
4. Next to the name field you can see the option **Create folder for project**. When checked, a folder with same name as project will be created to containthe project.
5. Select the source of project file:
   Currently opened project - This will import the opening project to server as a new project.
   Create new project - This will import a blank new project as a new project.
   Import existing project - This will start with an existing .vpp project file. If you select this option, click **...** and choose the .vpp file to import to server.



Project sources

6. In the **Repository** pane, select the folder where the project file will be stored. Some servers (e.g. SVN) allows you to create new folder in repository. You can create folder through the popup menu.
7. You can optionally describe the import action by entering your comment in the **Comment** field.
8. Click **OK**. This will import the

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Advanced Team Collaboration Features in Agilian

Check out the advanced team collaboration features supported by Agilian.

Branching

Branch is defined as a copy of work derived from a certain point in the trunk. It sets up an extra space for users to work on and make modifications without disturbing the trunk. As soon as the modifications in branch are done, you can merge it back to the trunk.

Tagging

Tag refers to a named version of your work at a point of time on the trunk.

Roll back past revision changes

Undo changes made in specific revision(s).

Export revision

Export revisions to project files.

Managing Teamwork Files

Add, commit teamwork files in Teamwork Files pane

Diagram protection

You can prevent a diagram from being edited by another team member by locking it. In this chapter, you will see how to protect diagram by locking.

See the evolution of design using Visual History

See the evolution of design. Visual History features a side-by-side viewer that allows the comparison of diagram over revisions.

Design recovery with Visual History

Recover your design by restoring your diagram from old revision. Run Visual History. Get changes back in an instant.

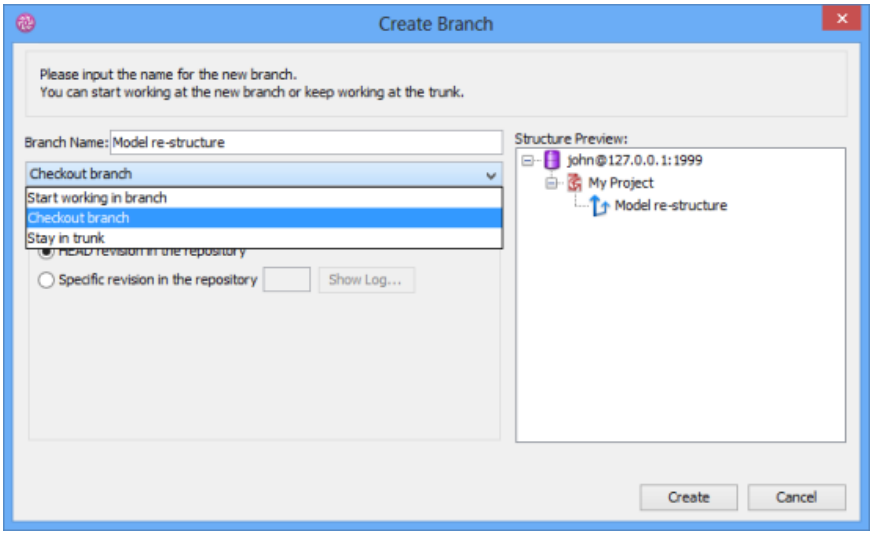
Branching

In terms of team collaboration, a trunk refers to the main stream of development. If you create a new project, you will spend the majority of your time making changes and committing them to the trunk of your repository.

Branch is defined as a copy of work derived from a certain point in the trunk. It sets up an extra space for users to work on and make modifications without disturbing the trunk. As soon as the modifications in branch are done, you can merge it back to the trunk.

Creating a branch

1. Select **Teamwork > Branch...** from the main menu.
2. When the **Create Branch** window appear, enter its name in the **Branch Name** field for your new branch and select whether to start working on the branch or to stay in trunk. Click **OK** to confirm.



*The **Create Branch** window*

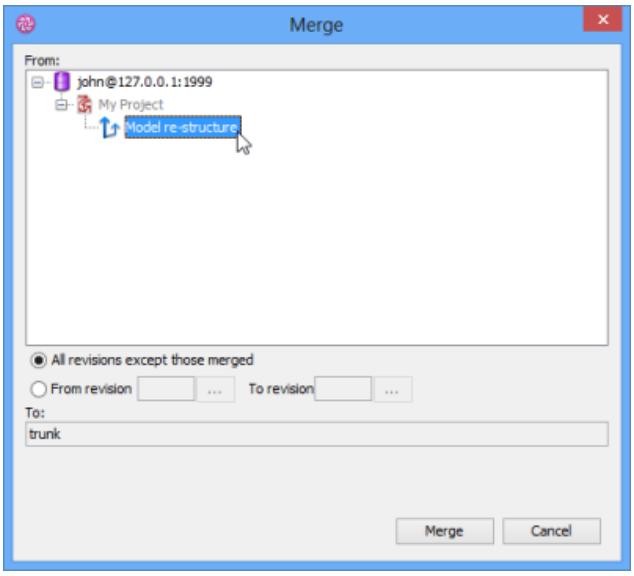
The three options in the drop-down menu are **Start working in branch**, **Checkout branch** and **Stay in trunk**. Selecting **Start working in branch** means you create, checkout and open a new branch and then start working on it. Selecting **Checkout branch** means you create and checkout a new branch, but just keep it in repository for working on it later on. Selecting **Stay in trunk** mean you create a new branch, but do not check it out and do not show it in repository either.

> **NOTE:** For Teamwork Server, SVN and Perforce user, you can create branch from a specific revision of trunk by selecting **Specific revision in the repository** and entering the revision number. On the other hand, select **HEAD revision in the repository** if you want to create a branch from the latest revision in trunk. Finally, click **OK** button to confirm.

Merging a branch

When the development activity of branch has been completed, you can optionally merge the branch back to trunk. To merge:

1. Work in trunk. Select **Teamwork > Merge...** from the main menu.
2. When the **Merge** window appear, select the branch you want to merge to the working trunk. A specified revision of project can be merged both from/to by selecting **From Revision** and entering the revision number of from and/or to revision. You can also click on the **Show Log...** button to popup a window with revisions listed and then select the revision from the list. Click **OK** to continue.



*The **Merge** window*

Related Resources

The following resources may help you learn more about the topic discussed in this page.

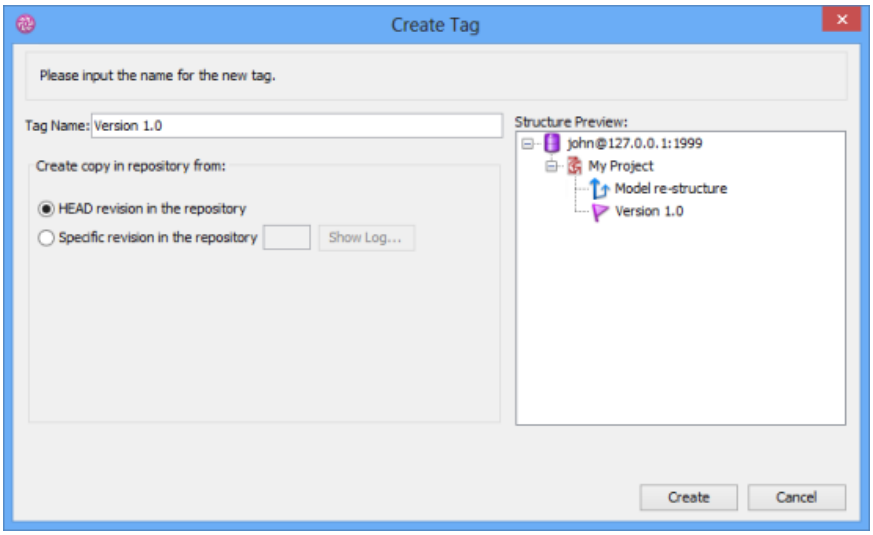
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Tagging

Tag refers to a named version of your work at a point of time on the trunk. The best application of a tag is to create it for every major release or milestone. Note that you cannot merge a tag to the trunk nor commit it to the server.

To create a tag:

1. Select **Teamwork > Tag...** from the main menu.
2. In the **Create Tag** window, enter the name of the new tag in the **Tag Name** field. Click **OK** to confirm.



*The **Create Tag** window*

> **NOTE:** For Teamwork Server, SVN and Perforce user, check **HEAD revision in the repository** if you want to create a tag from the latest revision of trunk while check **Specific revision in the repository** if you want to create a tag from a specific revision.

Related Resources

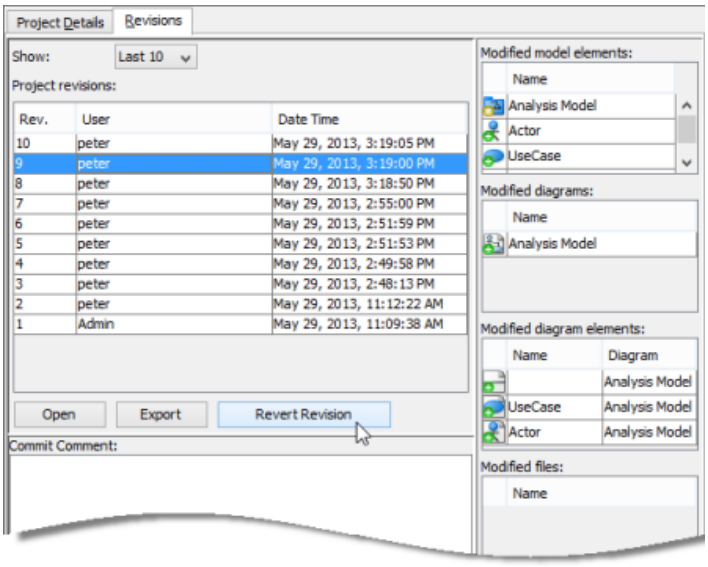
The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Roll back past revisions changes

Agilian maintains the histories of changes team members made. If changes were made by mistake, you can undo it by performing a revert on the revision where the change was made in.

Select **Teamwork > Open Teamwork Client...** from the main menu to open the **Teamwork Client**. In **Teamwork Client**, all your project's revisions are shown in **Project revisions** under **Revisions** tab. Select the number of project's revisions you want to list from the drop-down menu **Show**. Select the revision(s) that contain the unwanted changes and click **Revert Revisions.**



Revert a revision

A new revision will be created for the result of reverting. Therefore, you are asked to commit here. Sometimes, the undo action may cause a conflict with a more recent revision, or revisions. Just like the normal commit process, you must resolve all the conflicts in order to continue committing. You have to resolve conflict either by overwriting or reverting the change. Overwrite means to adopt the server copy's change, while Revert means to adopt the local's change.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

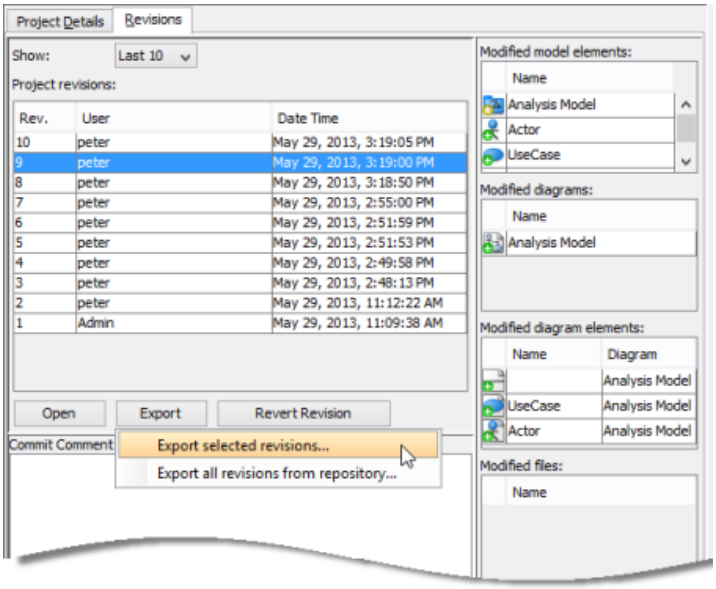
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Export revision

You can export revision(s) from server to your machine for checking the project content of certain phase of development. You can export specific revision, or all revisions.

Export the selected revision

1. Select **Teamwork > Open Teamwork Client...** from the main menu.
2. In the **Teamwork Client** window, open the **Revisions** tab.
3. Select the revisions(s) you want to export under **Project revisions**. Click **Export** and choose **Export selected revisions...** from the pop-up menu.



Export selected revision

4. In **Select Directory** window, select an existing folder or make a new folder for storing the selected revision(s). Click **OK** to proceed.

Export all revisions from repository

1. Select **Teamwork > Open Teamwork Client...** from the main menu.
2. In the **Teamwork Client** window, open the **Revisions** tab.
3. Click **Export** and choose **Export all revisions from repository...** from the pop-up menu.
4. In **Select Directory** window, select an existing folder or make a new folder for storing all revisions. Click **OK** to proceed.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Managing Teamwork Files

When modeling, there may be external resources you want to attach to a model which help describe it in detail or include data that cannot be modeled within the tool, like a text document. For example, you may want to attach a scanned image of a transaction receipt to a diagram that describes the transaction process so that the analyst can design the new system based on the image. Or maybe an image file showing the user's expectation of the user interface.

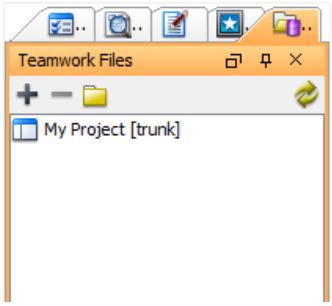
The reference function enables you to add file references to model elements and diagrams. You, as a user who work in a team-based environment with Teamwork Server do not need to copy any referenced files for other team members to open. Instead, you could commit your model along with the referenced files to the server by referencing a **teamwork file**. Teammates can then get the referenced files from server and open them in their environment.

In VP, there is a folder under the workspace for storing files that are readily be committed to Teamwork Server for versioning. Those files are called **teamwork files**. The file revision will follow the model, i.e. whenever you open a particular revision of work from server, the file of that revision will be obtained and thus there is always a consistency between the file and the design.

NOTE: "Teamwork Files" is supported by Teamwork Server Corporate Edition.

The Teamwork Files pane

Teamwork Files pane is a user interface component where you can manage and see the teamwork files. If you are running in the default perspective, the Teamwork Files pane is placed within the pane group at bottom left of the user interface. You may show it by selecting **View > Panes > Teamwork Files** from the main menu, or press **Ctrl-Shift-F**.

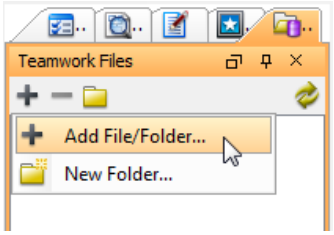


The Teamwork Files pane

Adding teamwork files/folder

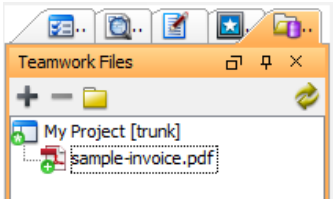
Adding a teamwork file/folder will copy that file/folder to under the workspace folder. Once added, the teamwork file/folder can be versioned by Teamwork Server. To add a teamwork file:

1. Select the node in **Teamwork Files** pane for creating file/folder. You may select the project root node or a folder node.
2. Click on the plus button in **Teamwork Files** pane and select **Add Files/Folder...** from the popup menu.



Add file/folder

3. In the **Add** window, select the file or folder to add and click **Open**.
4. You are prompted to confirm copying the file or folder to workspace. Click **OK**.



Teamwork file added

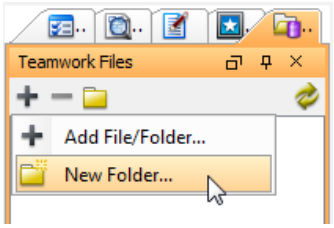
Opening the containing folder of file

You can open the containing folder of a teamwork file/folder so that you can access it with file explorer provided by the operating system. Right click on the file/folder and select **Show in Folder** from the popup menu.

Organizing teamwork files with folder

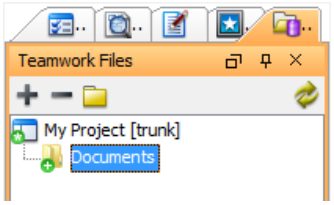
You may organize the added physical files/folder by adding folders. Note that when you create a folder, a physical folder will be created in your workspace folder. To add folder:

1. Select the node in **Teamwork Files** pane for creating folder. You may select the project root node or a folder node.
2. Click on the plus button in **Teamwork Files** pane and select **New Folder...** from the popup menu.



Create a folder

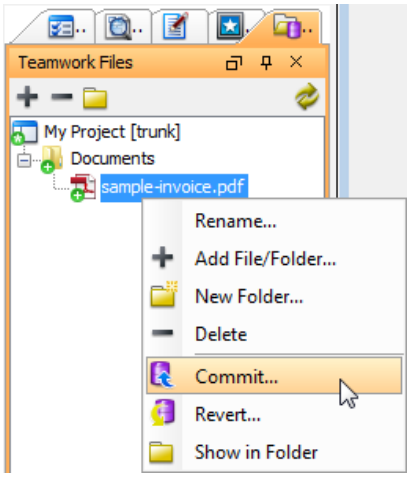
3. Enter the name of folder in the popup dialog box. Click **OK**. Once a folder is created, you may add files/folder in it, or drag existing teamwork files/folders into it.



Folder added

Committing teamwork files

You may make use of the commit function to commit teamwork files to Teamwork Server. When you perform a commit globally (i.e. commit the whole project), teamwork files will get committed. If you want to perform commit on specific teamwork file/folder, right click on it and select **Commit...** from the popup menu.



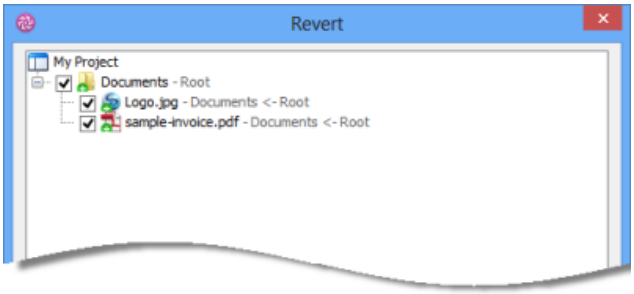
Commit teamwork files to server

Removing teamwork files

To remove a teamwork file from Teamwork Files pane, select the file/folder to remove and press **Delete**. Alternatively, right click on the file/folder and select **Delete** from the popup menu.

Revert changes

Revert is the process of discarding non-committed modifications. To revert a file/folder, right click on the file/folder and select **Revert** from the popup menu. In the Revert window, select the files to revert.

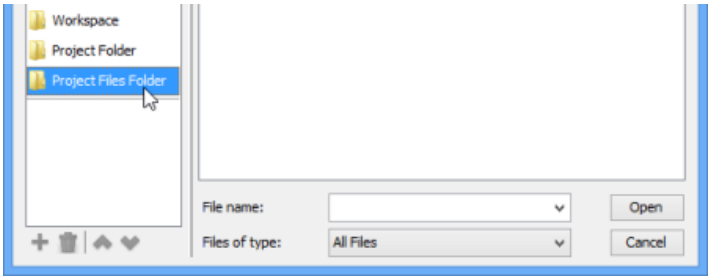


Revert changes of teamwork files

Referencing a teamwork file

When you try to add a file reference to a teamwork file, you may manually navigate through the folders to locate the teamwork file to add. Or to speed things up, Agilian lets you create a shortcut that brings you to the folder where the teamwork files are stored. In order to have such shortcut, you need to define a user path of project files to create a corresponding shortcut in file chooser. After that, you may add the reference easily.

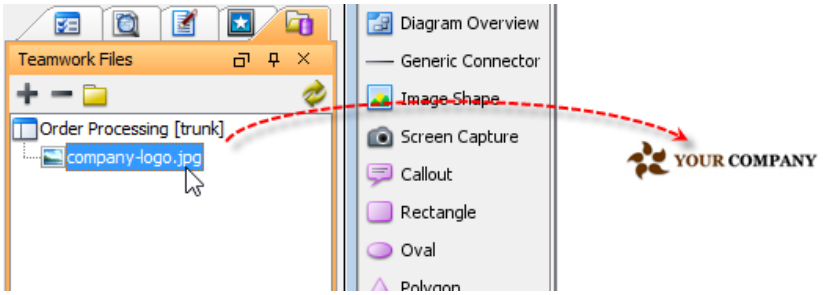
1. Select **Tools > Application Options...** from the main menu to open the Options window.
2. In the **Options** window, select **User Path** on the left hand side.
3. On the right hand side, click **Add...** and select **Project Files Path** from the popup menu.
4. Click **OK** at the bottom of the **Options** window to close it.
5. From now on, when you try to add file reference to model elements, you can select the shortcut Project Files Folder in file chooser to jump to the folder where the teamwork files are listed.



Project files folder

Creating image shape from a teamwork file

You can show a teamwork file in diagram if that file is in one of the supported images types - .jpg, .jpeg, .gif, .png, .bmp). To achieve this, simply drag and drop the image file from **Teamwork Files** pane to the diagram.



Dragging image file from Teamwork Files pane to diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagram protection

When you are working on a project with your team members, you probably don't want your work be modified by others by mistake. To protect your work, you can lock your current diagram with password to make it uneditable by others.

Locking diagram

1. To lock a diagram which you don't want it to be modified by others, right click on the diagram's background and select **Diagram Content > Lock ...** from the pop-up menu.

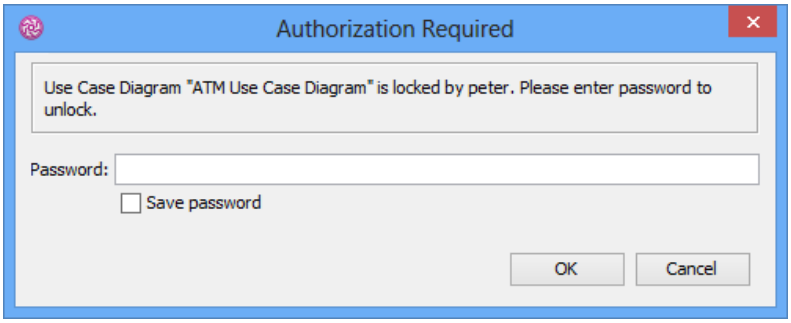
2. Enter your password in the pop-up **Lock** window and then click **OK** button. For your safety, always keep your password secret.

Enter password

Your diagram is then locked. Here is a detailed description of the options in the Lock window:
Lock by password: If you check this option, you need to set password for locking diagram. If you uncheck it, you don't have to enter password but you can still lock the diagram.
Save password: You can save your password in the project under your user account by checking this option. Thereafter, you don't have to enter password when you edit the diagram. Otherwise, you need to enter password to edit the diagram after you uncheck it.
Show password: Check this option to reveal the password. The password will be hidden by asteriks when you keep it unchecked.

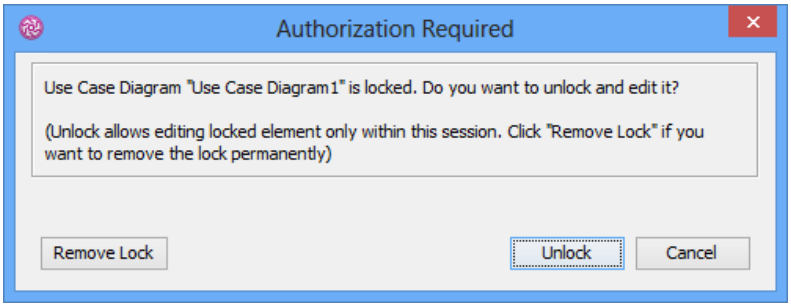
Trying to edit a locked diagram

If your team members want to modify any model elements of the locked diagram, they need to provide the password set by you in the pop-up **Authorization Required** window in order to proceed modification.



Require to enter password to proceed editing

If you try to modify any model elements of the diagram locked by you, an **Authorization Required** window will prompt out. Click **Remove Lock** button if you want to unlock the diagram permanently while click **Unlock** button if you want to unlock it temporarily.



Unlock the diagram

NOTE: If you uncheck **Save password** when you lock the diagram, you will be asked to enter password to edit the locked diagram.

Moreover, you can also unlock the diagram permanently in **Configure Lock** window. Right click on the diagram's background and select **Diagram Content > Configure Lock...** from the pop-up menu. Choose **Remove lock** and enter password.

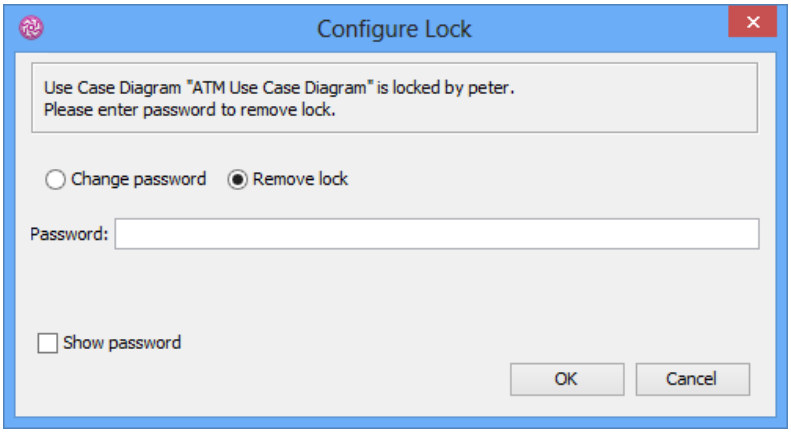
Removing lock

You can remove the lock or change the password after you lock the diagram.

1. Right click on the diagram's background and select **Diagram Content > Configure Lock...** from the pop-up menu.

> **NOTE:** **Configure Lock...** option is available on the pop-up menu only after you have locked the diagram.

2. In **Configure Lock** window, choose **Change password** if you want to change the current password and set a new password; otherwise, choose **Remove lock** if you want to remove the lock permanently. Click **OK** button to confirm.



Remove lock

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

See the evolution of design using Visual History

There are many reasons why you want to know how a diagram looked like at some point in time. First, you may find the latest design not feasible and you want to see how the feasible one looked like in the past. Second, you may spot a flaw/mistake and you want to recover it by referencing the correct design did before. And for diagrams like business process diagram, ArchiMate and business motivation model, you can observe their changes to determine the changes the business has made over the years. You can easily tell what and why the design, hence the business was changed. Such information can help you plan the future with minimized risks.

While seeing the evolution of diagram is pretty much the most common use case, you can also see the evolution of specific shape, or even a model element, say, to know what a package contained at some point in time.

Visual History enables you to see the evolution of your design. It features a side-by-side diagram viewer that allows the comparison of diagram over revisions. With the revision slider, you can easily slide across revisions to compare the diagram in the latest revision and a specific revision in the past.

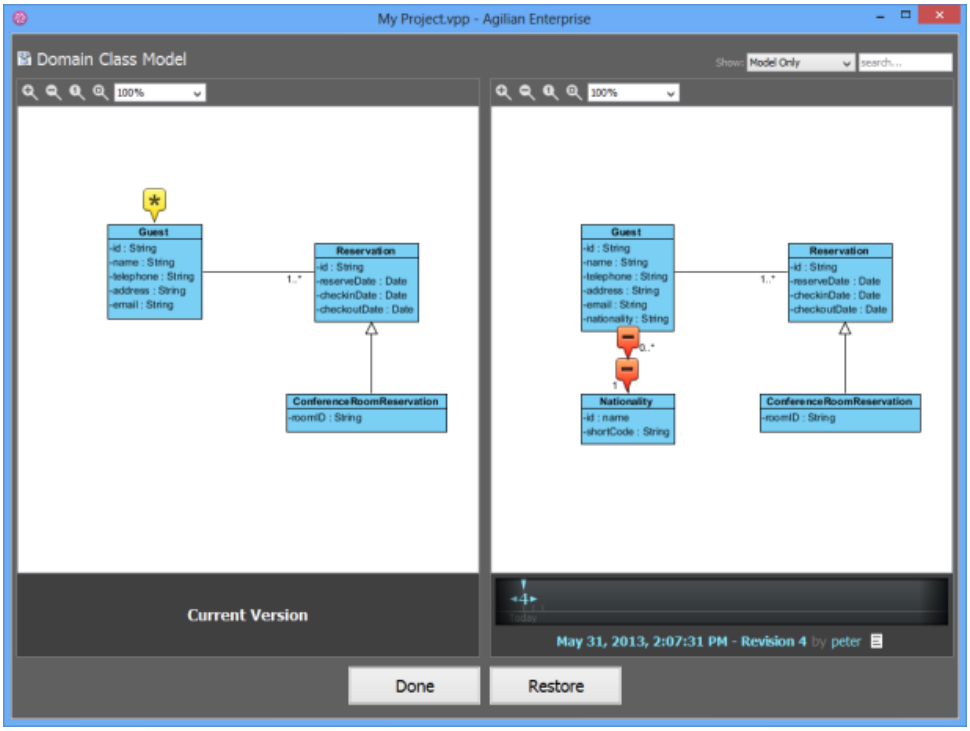
Another important function of Visual History is to help recover your work by restoring from an old revision. You can read the next chapter for details.

Browse a diagram in old revision with Visual History

To see how a diagram looked like in the past, perform the steps below.

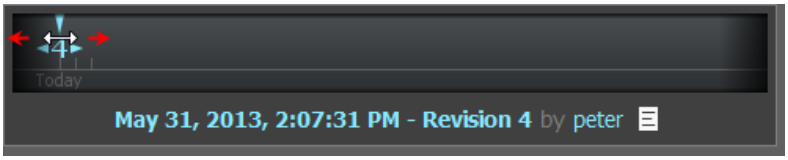
1. Take any of the following steps to open **Visual History**.
   - Right click on a diagram node in **Diagram Navigator**/ **Model Explorer** and select **Visual History...** from the popup menu.
   - Right click on the background of a opening diagram and select **Visual History...** from the popup menu.

2. This opens the **Visual History** window. At the top of the window you can see the diagram name as well as the drop down menu **Show**, which controls the kind of difference you want to focus on when comparing revisions. We will talk about it in the coming sections. In the middle you see two diagram viewers. The left hand side is the diagram in latest revision, while the right hand side is the same diagram in an old revision. You can drag the viewers to pan the diagrams. If you are comparing a model element or a shape instead of a diagram, you will see the properties of that element/shape listed here.



The Visual History window

3. Under the diagrm viewer on right hand side you see the time line. Press on the revision number and drag right or left to move to older or more recent revisions. The diagram viewer will be updated to show the diagram in the chosen revision. Note that only the revisions that reflect a change in diagram are listed on the time line. You can slide between revisions to see the evolution of diagram, and to stop at certain revision to analyse and compare that revision with the latest revision, as shown in the viewer on left hand side.

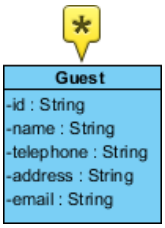


The time line

Compare diagrams with the help of visual indicator

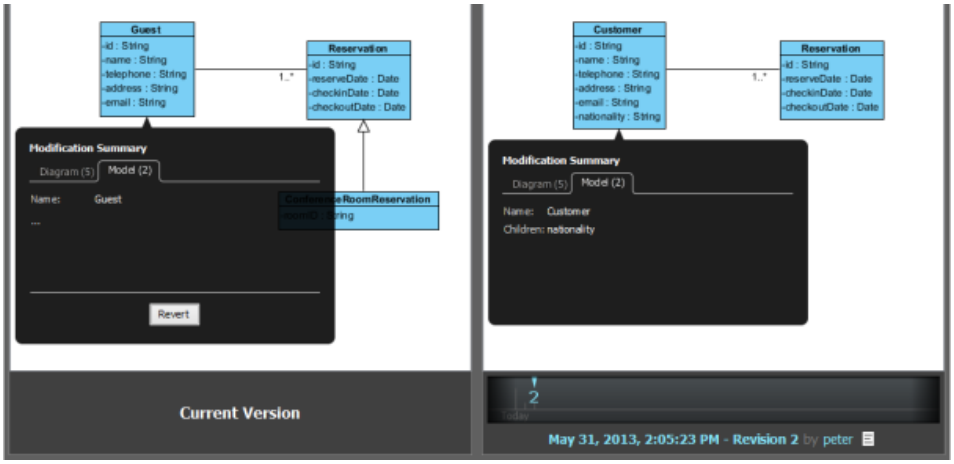
While visual difference of diagrams can be identified easily by bare eyes, you are unable to tell the changes of shape/model element properties that are not presented on diagram, such as the modification of element documentation. In order to know this kind of change, you need to check the visual indicators that appear on the diagram.

Each indicator represents a difference found when comparing the latest revision and the revision as selected in the time line. The pointer of indicator tells you where the difference is.



A modified class

By clicking on an indicator, you can see a list of modified properties, categorized by diagram level property (e.g. shape position, color, etc) and model level property (e.g. name, documentation of model element, etc).

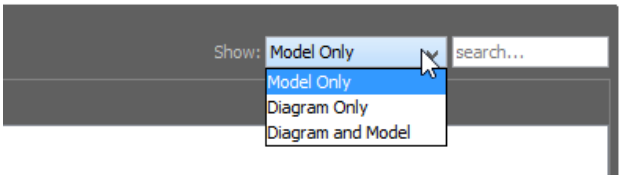


The 'Guest' class was named 'Customer' before, and owned an attribute nationality

Adjust the type of difference to indicate on diagram

There are two kinds of differences **Visual History** can identify when comparing diagrams - model and diagram. Model refers to all sort of model element properties change, such as the renaming of element, addition of file reference, modification of documentation, etc. Diagram refers to all sort of shape properties change, such as shape position, shape color, etc.

If you want to focus on model element properties rather than the view level properties, you can click on the drop down menu **Show** at the top right of the **Visual History** window, and select **Model Only**. There are also options like **Diagram Only** and **Diagram and Model** to suit different needs. The diagram viewers will be updated to show or hide away indicators, after considering or ignoring the differences base on your choice.



Show only model element properties changes

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Design recovery with Visual History

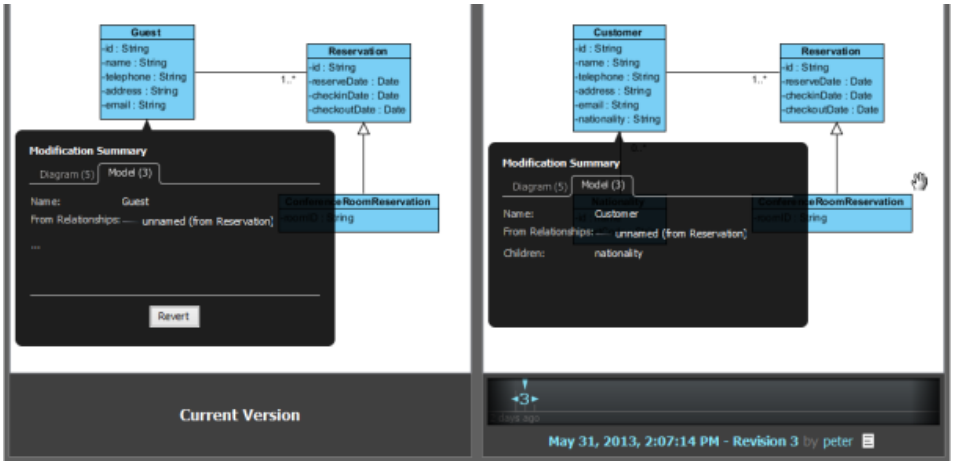
Aside from browsing a designs in old revision, **Visual History** also allows you to restore a diagram from old revision. Let's say you have deleted a shape by mistake and you want to get it back. You can locate in Visual History the revision that has the shape intact, and get it back by clicking the **Revert** button (You will learn how in this page). Besides restoring changes one by one, you can restore the whole diagram. By doing so the whole diagram will be overwritten by the old version.

Revert modification

1. In the **Visual History** window, when you are comparing diagrams, there are indicators that appear on diagram showing the areas of changes. Click on an indicator to popup the modification summary.

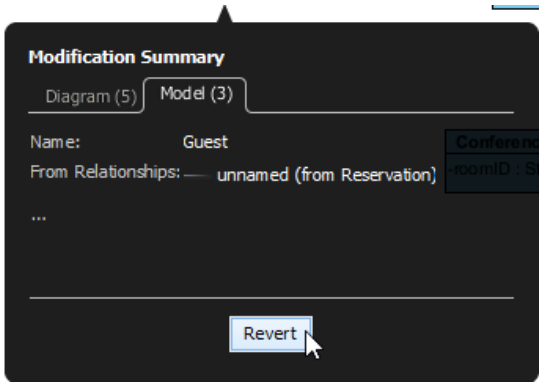
To review modifications

2. Review the modifications.



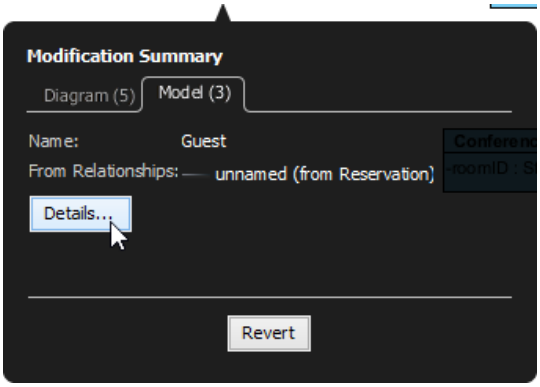
Review modifications

3. If you want to revert **ALL** the modifications listed in the modification summary, click the **Revert** button at the bottom of the summary popup.



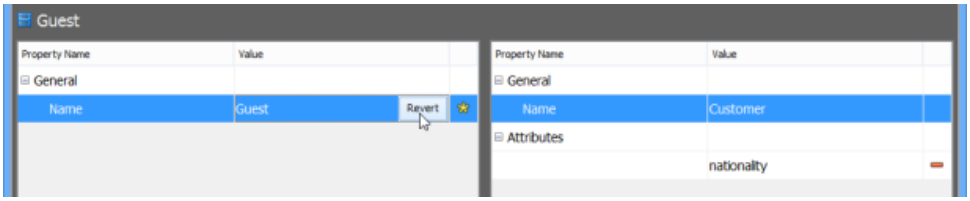
Revert modifications

4. If you want to revert a specific modification, move the mouse point below the last modification to show the **Details...** button. Click on it.



Review modification details

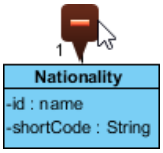
5. The modifications are listed in in a table. You can click on individual modification and then click the **Revert** button to revert it.



Revert a modification

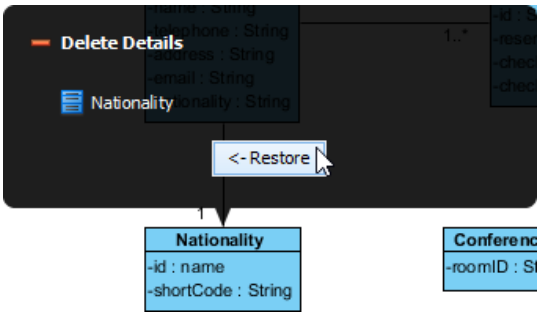
Restore deleted shape

1. In the **Visual History** window, when comparing diagrams, click on the indicator that appear on top of the deleted shape.



To get deleted shape back

2. Click the **Restore** button in the **Delete Details** popup to get the shape back.



Restore a deleted shape

Restore the whole diagram

If you want to restore the entire diagram from an old revision, simply click Restore at the bottom of the **Visual History** window.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Using PostMania

PostMania Overview

Learn what PostMania is.

The PostMania Page

Learn how to start using PostMania.

Posting Discussion Topic in PostMania

Learn how to post a discussion topic.

Replying in PostMania

Learn how to reply a post.

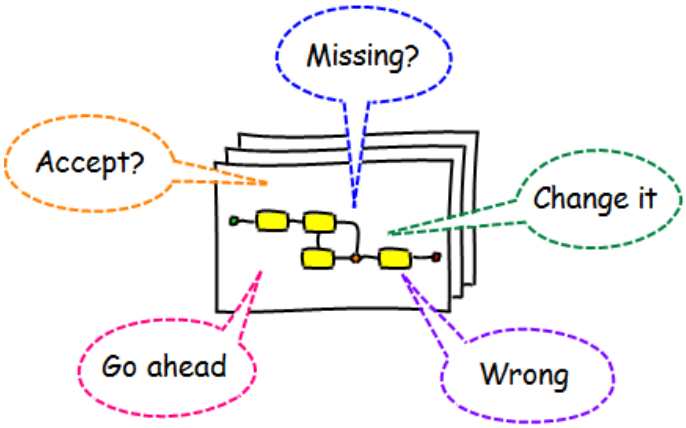
Watching Diagram

Learn how to get notified by changes.

PostMania Overview

As long as your design is not made for yourself, you must need to share it with others, like your boss, your teammates, your client, etc. You need to communicate with them for confirmation of work, for clarifying needs, for notification, etc. All these are what PostMania offers.

PostMania is a social networking tool that lets people both inside and outside the development team to see the designs, discuss the designs and keep track of design changes. Stakeholders can review process designs from time to time to see if the process behaves as expected. Developers can get up-to-date system specification and design guidelines. Everybody can discuss and debate the designs for clearing doubts, requesting for improvements and exchange ideas.



PostMania overview

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

The PostMania Page

PostMania page provides a central place for you to access the discussion topics in diagrams and projects.

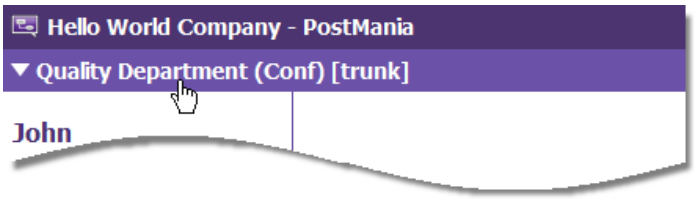
Enabling PostMania Page

To enable the PostMania page, select **Teamwork > PostMania Page** from the main menu.

Opening Project in PostMania page

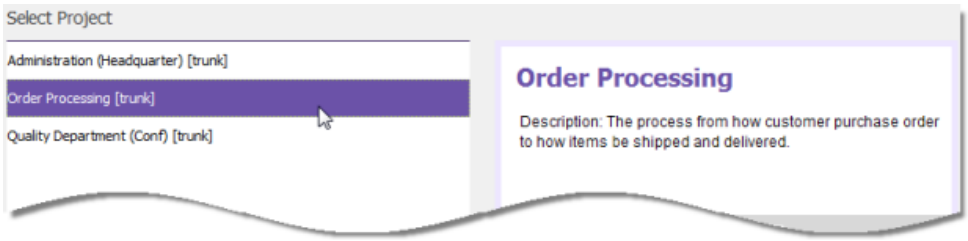
In order to view diagrams of a project or to post, read and reply discussions for diagrams in that project, you need to open the project first. You may open project in PostMania page. Note that only projects that have PostMania enabled in server can be opened via PostMania page.

1. On the top left of PostMania page, click on the name of project you currently work on.



Click on the working project

2. You are prompted to select the project that you want to review and/or discuss. Select the project on the left hand side. Its description, if written, is presented on the right hand side.



Select project

3. Click **Open Project** to open it.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

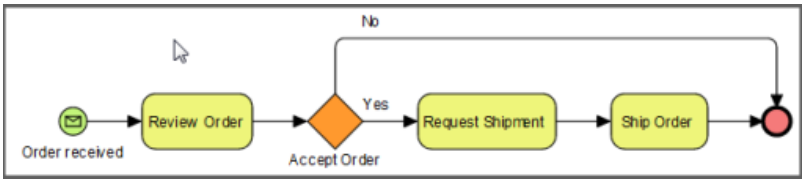
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Posting Discussion Topic in PostMania

Any people who want to say something about a diagram or its content, no matter for what purpose, can post a discussion topic. Discussion topic can be posted for a diagram or a shape. Viewers can use this function to report status ("I have approved."), to ask for action ("Please implement this."), ask whatever questions ("When do you need this?"), seek for confirmation ("May I remove this from the model?"), request changes ("Fix it please"), etc.

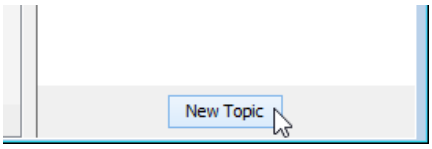
Posting Discussion Topic

1. Browse and open the diagram that you want to post discussion.
2. The diagram must exist in the server in order for you to post discussion. If the diagram is newly created, commit it first. You can commit the entire project or just commit the diagram partially.
3. You can post discussion topic for a diagram or for specific shape(s) (including connector) in diagram. Click on the background of diagram or the shape to discuss. If you want to discuss multiple shapes, press the **Ctrl** key and click on the shapes you want to discuss.



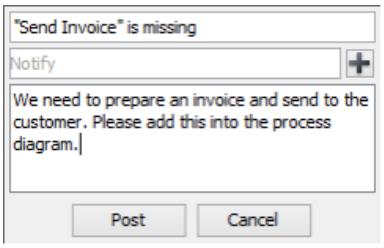
Click on a diagram to add discussion topic for it

4. In the **Topics Pane** on the right hand side, click **New Topic**.



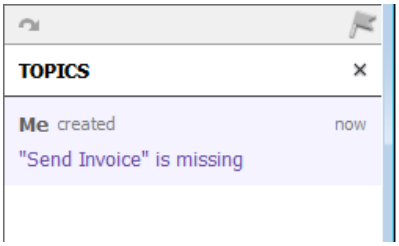
Create a new topic

5. Enter the title and content.



Enter the content of a topic

6. If you want specific team member be informed about your post, click the **+** button next to the **Notify** field and choose the team member(s) to notify. Members who are set notified will receive notification under the **Notify me** section in the home page of PostMania.
7. Click **Post**. This creates a topic and the topic is listed in the **Topic** pane.



Created topic is listed in Topic pane

Related Resources

The following resources may help you learn more about the topic discussed in this page.

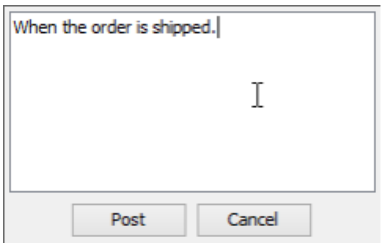
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Replying in PostMania

When someone has posted a discussion topic, you can give him a reply if you have any comment about what he/she says. Reply can be made to a topic, which gives a general reply that is not responding any previous reply. Besides, you can also reply to a specific reply.

Replying a Discussion Topic

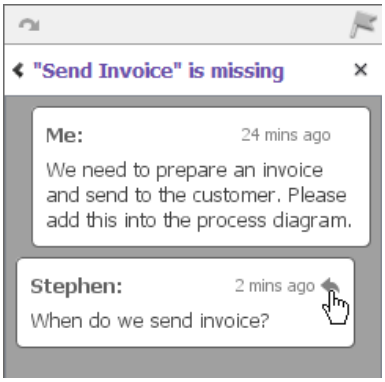
1. Open the discussion topic you want to reply.
2. Click **Reply**.
3. Enter the message and click **Post**.



Enter reply content

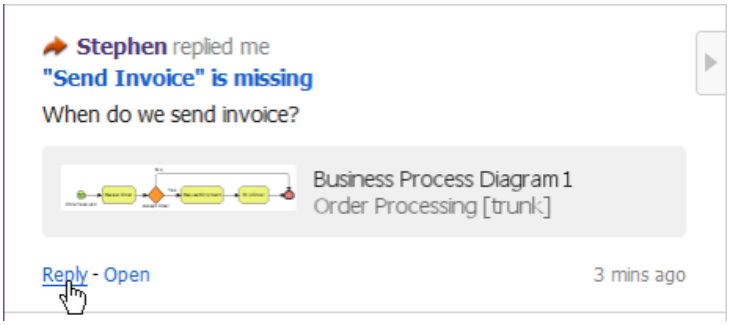
Replying a Specific Reply

1. You can reply someone's post instead of giving a general reply to the whole topic. There are two ways you can do this. First, open a discussion topic and click on the tiny reply button next to the reply you want to reply.



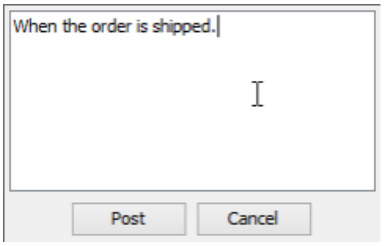
Reply a specific reply

Alternatively, in main page click the **Reply** link on the post you want to reply.



Reply in main page

2. Enter the message and click **Post**.



Enter reply content

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian

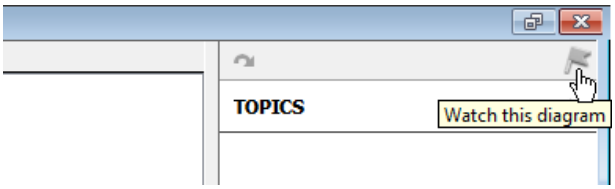
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Watching Diagram

If you want PostMania inform you that certain diagram has discussion topic posted/replied, or has been modified, you can watch that diagram. By watching a diagram, if someone has posted a discussion/reply, or when someone has committed changes made in that diagram, you will be informed in the main page.

Watching a Diagram

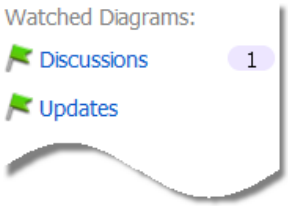
1. Browse and open the diagram that you want to post discussion.
2. On top of the **Topics Pane**, click on the gray flag to make it green. A green flag indicates that the diagram is being watched.



Watch the diagram

About Notification

Once you have set yourself watching a diagram, when someone has posted a discussion/reply, or when someone has committed changes made in that diagram, you will be informed in the main page. On the left hand side of the main page, check the **Watch Diagrams** section. The **Discussions** flag reports the number of new discussions made to the watched diagram(s). The **Updates** flag reports the number of modification made to the watched diagram(s). You can click on them to see the list of updated diagrams.



Notification of updates

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

General Options

Project Options

Option Name	Description
Default author for non-teamwork project	The name of person who edit the project. This will be filled in as the author within project management properties.
Auto save project	Save the opening project as backup periodically.
Auto save interval (mins)	Time needed to wait until the next auto saving.
Delete model element when deleting its view	Action that happen when all views of a model element are being deleted: • Ask - (default) Prompt if you want to delete the model element as well. • Delete without ask - Delete the model element as well. • Don't delete - Do not delete the model element.
Open exported image file	Action that happen after exporting images • Ask - (default) Prompt if you want to open the exported image file. • Yes - Open image file (when export single image) or image folder (when export multiple images) directly. • No - Do not take any action.
Backup level	The number of files that will be saved as backup. When reached the limit, the next backup will overwrite the earliest one.
Confirm diagram deletion	(default true) Prompt if you really want to close diagram.
Confirm shape deletion	(default true) Prompt if you really want to delete a shape on diagram.
Never delete model elements with views	(default false) If a model element has multiple views, delete any of them will never cause the model element to be deleted, even if you are deleting a master view.
Open last project on startup	(default true) If checked, it will open the last opened project immediately when starting VP-UML. If unchecked, it will open new project.
Export as image with frame	• None - (default) Do not surround the diagram with neither frame nor border. • Export with frame - Add a frame around image to show a border with the name of diagram appear at top left of diagram. • Export with border - Add a thin border around image.
Show password in lock dialog pane	
Lock by password	

Project Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Referenced Project Options

Option Name	Description
Duplicate Model from Linked Project	Duplicating a model element from linked project will cause a local copy to be created. • Prompt - (default) Prompt if you really want to duplicate. • Yes - Confirm duplication. A local copy of selected shape(s) will be created. • No - Discard duplication.
Show warning when failed to add child to model from linked project	You cannot place a shape inside a linked shape, such as a linked package. If you try to do this you will be warned by default. This option determines whether the warning will appear or not.

Caption Here

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Appearance Options

Option Name	Description
Look and feel	It controls the appearance of application screen.
Theme (for Office 2003 L&F only)	The tone of the application screen.
User Language	Language being applied on the user interface. This affects the text in menus, tooltips, dialog content, report content, etc.
Change application font	Select the font family and size of font to apply for the application user interface.
Date Format	Format of date values that appears in the application.
Date Sample	The sample of date is shown after you select the format of date.
Time Format	Format of time values that appears in the application.
Time Sample	The sample of time is shown after you select the format of time.
Measurement Unit	• inch - (default) Set the measurement unit to be inch • cm - Set the measurement unit to be centimeters

Appearence Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Connection Options

Option Name	Description
E-mail	Enter the Email field to specify your email address.
Use proxy	(default false) To check/uncheck in order to use a proxy server for connecting to the Internet.
Host	The host of the proxy server.
Port	The port of the proxy server.
Login name	The user name of the proxy server (if the proxy server required the user to login).
Password	The password of the proxy server (if the proxy server required the user to login).
ElaborView server Connection - Connection Name	A name to describe the connection of ElaborView server.
ElaborView server Connection - Hostname	The host of the ElaborView server.
ElaborView server Connection - Port	The port of the ElaborView server.
ElaborView server Connection - Email	The Email to log into the ElaborView server.
ElaborView server Connection - Password	The Password to log into the ElaborView server.
ElaborView server Connection - Add	VP can store multiple ElaborView connection. You may add a new one by clicking Add, and specify the connection properties after added.
ElaborView server Connection - Remove	Click to remove the current connection.
ElaborView server Connection - Test Connection	Click to test if the connection is valid or not.

Connection Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Printing Options

Option Name	Description
Use gradient color when print diagrams	Control whether to use gradient or solid color for shapes and diagrams when printing.
Print diagram background when printing diagrams	Control whether to print the diagram background color when printing diagram.

Printing Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Edition Options

Option Name	Description
Enable non-supported feature	(default true) When checked, executing features that are not available in the running edition will be prompted, asking whether you want to advance to higher edition in order to use the feature. When unchecked, those non supported features will be disabled.

Edition Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Teamwork Options

Option Name	Description
Show Export/Import Teamwork project	(default false) Determines whether the export/import Teamwork project menus will appear in the Projects menu of Teamwork Client dialog box

Teamwork Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Update Options

Option Name	Description
Auto update	Let the application checks for available updates with respect to the running build/version, and notify you to perform update whenever possible. • Never - Do not inform product update • On every start - Check for updates everytime when starting VP-UML • Daily - Check for updates daily • Weekly - (default) Check for updates weekly • Monthly - Check for updates monthly

Update Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagramming Options

Appearance Options

Option Name	Description
Global Pallet Option - Show name	(default true) Determines whether the name of items will be shown in the pallet.
Global Pallet Option - Expand group	(default false) Determines whether the group will be expanded to display all items.
Show Hidden Layer Indicator	• Yes - When move the mouse over a shape, the hidden layer indicator is shown • No - The hidden layer indicator isn't shown even when move the mouse over a shape • Prompt -Ask if users want to show hidden layer indicator when move the mouse over a shape
Highlight Glossary Terms	Underline the text in shape name or documentation if it is a term.

Appearance Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Environment Options

Option Name	Description
Alignment Guide - Show diagram alignment guide	(default true) Show alignment guide which appear when moving a shape on a diagram.
Alignment Guide	• Show edges - (default) Show guides at edges of the closest shape • Show center - Show a guide that lies on the center of the closest shape
Shape Selection Detection	• Inside selection area - (default) When selecting a range of shape, only shapes that are completely inside the selection range are included in selection • Overlapped with selection area - When selecting a range of shape, shapes that are partly or completely covered by the selection range are included in selection
Copy as XML with RTF style	• Yes - When copy Use Case, the rich text format of Use Case Details will also be copied (size of copied content will increase considerably) • No - When copy Use Case, Use Case Details will be copied as plain text • Prompt - Ask if user want to copy rich text for Use Case Details when copying XML
Delay until showing Quick Preview in Diagram Navigator (seconds)	• Never show - Never show Quick Preview when moving mouse cursor over diagram node in Diagram Navigator • 1.0 - 3.5 - The number of seconds that a Quick Preview will disappear after moving the mouse cursor out of a diagram
Show "Copy to Clipboard as OLE" in menu	(default false) Determines whether the Copy to Clipboard as OLE menu is available or not
Default Copy cction	• Within VP-UML - (default) When triggering the hotkey for Copy (Ctrl-C, by default), it will perform copying within VP-UML • Copy to Clipboard as Image (JPG) - When triggering the hotkey for Copy (Ctrl-C, by default), it will perform copying as JPG image • Copy to Clipboard as Image (EMF) - When triggering the hotkey for Copy (Ctrl-C, by default), it will perform copying as EMF image
Copy as image with frame	• Unspecified - (default) Prompt for adding a frame or a border when copying shapes as image • None - Do not add border nor frame to image when copy shapes as image • Copy with frame - Add a frame around image to show a border with the name of diagram appear at top left of diagram • Copy with border - Add a thin border around image
Show shape content when dragging	(default true) Show the shape content such as shape name when dragging shape.
Prompt for clearing undo history before applying design pattern	(default true) Prompt for applying design pattern even when there are remaining undo or redo due to the undo and redo records will be cleared after applying design pattern.
Number of stereotypes shown in popup menu	You can assign stereotypes to a shape through its popup menu. This option determines the number of stereotypes to display in that popup menu.
Auto expand diagram borders	Sometimes, the diagram border may be wrongly calculated. You may click the Reset to default button to reset the range of diagram border.
Enable Mouse Gesture	Mouse gesutre enables you to create and connect elements by forming a gesture with the right mouse button.
Show Logical View selector when naming new diagram	When there is at least one logical view in your project, you can optionally select the logical view to store a diagram during the creation of diagram. If you want to turn this option off, uncheck here.
Show diagram element tooltip	Show the tooltip of shape when moving mouse pointer over shape

Environment Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Model Generation Options

Option Name	Description
Generate Sequence Diagram from Use Case Description Overwrite Existing Diagram	• Yes - Automatically overwrite Sequence Diagram generated from Use Case Details when generate again • Prompt - (default) Prompt for overwriting Sequence Diagram generated from Use Case Details when generate again
Generate Diagram from Scenario Overwrite Existing Diagram	• Yes - Automatically overwrite diagram generated from Scenario when generate again • Prompt (default) Prompt for overwriting diagram generated from Scenario when generate again
Overwrite Flow of Events when Synchronize from Sequence Diagram	• Ask - (default) Prompt for overwriting Flow of Events in Use Case Details when Synchronize from Sequence Diagram • Yes - Overwrite Flow of Events in Use Case Details automatically when Synchronize from Sequence Diagram

Model Generation Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Shape Options

Option Name	Description
Initial Shape Size When Set to Display Image	• Fit shape to image - Resize the image placeholder to fit the selected image • Fit image to shape - (default) Resize the image to fit into the image placeholder
Add Covered Shapes as Children After Resize	• Yes - Automatic contain the covered shapes after resizing a container to cover shapes • No - Do not contain covered shapes after resizing a container to cover shapes • Prompt - (default) Ask if you want to contain covered shapes after resizing a container to cover shapes
Show Added Member in Other View	When you try to create a new member, like a column of an entity, on a shape, and if there is another view of the model element, you can set whether to show the member on other view as well. • Yes - Show the member on other view(s) • No - Do not show the member on other view(s) • Prompt - (default) Ask if you want to show the member on other view(s)
Auto Expand Parent When Moving Child	Determines whether to expand the parent when moving the shape around the parent's border.
Create new line key	• <Ctrl> + <Enter> - Press Ctrl-Enter to create a new line when inline editing • <Alt> + <Enter> - (default) Press Alt-Enter to create a new line when inline editing • <Enter> - Press Enter to create a new line when inline editing
Show unlimited level in preview shape (May slower the diagram display)	(default false) Determines whether a diagram overview will show contents for all nested diagrams.
Show rich text content in preview shape	(default false) Determines whether diagram overview will show rich text content.
Name completion (Classes, Entities and Requirements)	(default true) When creating a class/ requirement/ an entity, you can select an existing class/ requirement/entity from a list so that a view of the selected class/requirement/entity can be created. You can enable or disable the popup of such list.
Resize shape when add member	(default true) For shapes that have member, like class and entitiy, when you add a member, it will or will not resize the owner against the length of member, base on this option.
Auto resize parent when resizing child	(default true) When you resize a shape that makes it touches the boundary of parent, this option determines whether the parent will be expanded accordingly.

Shape Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Connector Options

Option Name	Description
Default pin from connection point	(default false) Automatically pin connector's from end when connector is being created.
Default pin to connection point	(default false) Automatically pin connector's to end when connector is being created.
Show relationship connectors for dropped models	(default true) Show connectors when dragging and dropping inter-related model elements/views from tree to diagram.
Auto relocate connector when overlapped with other shapes	(default false) Auto relocate connector when connector is being overlapped by another shape.
Scroll connector delay (second)	• No delay - Immediately scroll to the other side of connector • 1 - 9 - Time provided for scrolling to the other side of connector
Highlight selected connector	(default true) Increase the thickness of selected connector(s).

Connector Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Resource Centric Options

Option Name	Description
Show resources	(default true) Show resource icons around shapes.
Show group resources	(default true) Show group resources that appear when selecting multiple shapes.
Show extra resources	(default false) Show also uncommon resource icons.
Show generic resources only	(default false) Show generic resource but hide other resource icons.
Show resources delay (second)	0 - 2 - Time needed to wait from having mouse cursor hover on shape till the resource icons appear.
Auto hide resource delay (second)	Time needed to wait the resource icons to disappear when mouse cursor is moved out of a shape.
Always show model element indicators	(default true) Always show the reference, subdiagram, transitor, documentation resource icon at the bottom of shape no matter whether the shape has reference/sub-diagram/transitor/ documentation added/defined.

Resource Centric Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Class Options

Option Name	Description
Show parent style	• UML - (default) Show tooltip of child class in UML style like *Package::Class* • Java/C# - Show tooltip of child class in Java/C# style like *Package.Class*
Show fully qualified class and package name in tooltip	(default true) Enable to show fully qualified class and package name in tooltip like *Package::Class*/ *Package.Class* (depending on the setting of **Show parent style**). Disable to show only the hovering class or package name and type like *Class : Class*.

Class Options details

Auto Attribute Type

Option Name	Description
Name	When an attribute entered matches with the name defined here, the type and default value will be automatically filled.
Type	Automatically set attribute type value when the name user entered for an attribute matches with name specified in Name.
Default Value	Automatically set default value when the name user entered for an attribute matches with name specified in Name.

Auto Attribute Type of Class Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Generalization Options

Option Name	Description
Direction of creation	• from General to Specific - (default) When creating a generalization, the arrow head will appear at the mouse release side • from Specific to General - When creaNessageting a generalization, the arrow head will appear at the firstly selected shape

Generalization Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

ERD & ORM Options

Auto Column Type

Option Name	Description
Name	When a column entered matches with the name defined here, the type and default value will be automatically filled.
Type	Automatically set column type value when the name user entered for a column matches with name specified in Name.
Nullable	Automatically set column nullable when the name user entered for a column matches with name specified in Name.
Default Value	Automatically set default value when the name user entered for a column matches with name specified in Name.

Auto Column Type of Entity Options details

Behavior & Presentation

Option Name	Description
Show table record editor	(default true) Show the table record editor at the bottom of ERD for adding default table records to database tables.

Behavior & Presentation of Entity Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Interaction Options

Option Name	Description
Show sequence diagram text editor	You can edit sequence diagram through the text editor pane that appears at the bottom of diagram. This option controls the visibility of editor.
Auto fit message completion size	(default true) Fit message completion box's size everytime you activate it. If disabdeterminsled, size adjusted manually won't be remembered.
Show operation documentation in message completion	(default true) When selecting an Operation in the message completion box, the documentation of operation will appear next to the completion box.

Interaction Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Business Process Options

Option Name	Description
Invalid Connection Handling	• Ignore all - Ignore all invalid actions related to connecting shapes • Cancel move - Cancel invalid actions related to connecting shapes • Prompt - (default) Prompt for an action when an invalid actions related to connecting shapes is discovered
Show Lane Handle	• Auto - (default) Show horizontal/vertical Lane header only when horizontal/vertical Lane exist • Always Show - Always show both horizontal and vertical Lane headers even when Lane does not exist • Always Hide - Always hide Lane headers
Show convert Sub-Process/Task warning	(default true) Show warning when trying to convert between Sub-Process and Task.

Business Process Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Report Options

Option Name	Description
When insert custom cover page, show disable generate cover page warning.	In a report in report composer, there are two ways for you to add a cover page. One is to add a custom cover page, and another one is to fill in the details of cover page in Report Inspector. You can choose either way to add a cover page, not both. By checking this option, you will be prompted to disable the report inspector option when you add a custom cover page.

Report Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

View Options

View Options

Option Name	Description
Diagram Navigator Sort Type	• Sort by name - (default) Sort tree nodes in Diagram Navigator by their names • Sort by type - Sort tree nodes in Diagram Navigator by their types
Model Explorer Sort Type	• No sort - Do not sort tree nodes in Model Explorer • Sort by name - (default) Sort tree nodes in Model Explorer by their names • Sort by type - Sort tree nodes in Model Explorer by their types
Show data types	(default false) Show Data Types node in trees.
Show relationships	(default false) Show relationships in trees.
Show sub diagrams	(default true) Show subdiagrams in trees.
Show Activation in Diagram Navigator	(default true) Show activations in Diagram Navigator.
Show project path in Diagram Navigator	(default false) Show project path in Diagram Navigator.
Sort elements in tree with case sensitive	(default false) Make sorting of tree nodes case sensitive (consider the case).
Show carriage return character	(default true) Show carriage return character for line breaks of shape names that are in multiple lines.

View Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Office Exchange Options

Office Exchange Options

Option Name	Description
Remember Import Decision	VP detects changes made in exported document, and will suggest you to import changes back to your project. This option determines whether the import will be on or off. • Yes - Enable the import option. • No - Disable the import option. • Not Specified - (default) You will be asked if you want to import the changes from document to VP whenever changes are detected.
Launch viewer	(default true) Open the document after export.

Office Exchange Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

User Path Options

User Path Options

A user path is a variable that refers to a base path in user’s computer. You can add a reference to local file using user path so that the reference refers to a file relative to a user path, instead of an absolute path. This means you can move references files to a different location, or even to a different computer, and can still open them as long as the user path value is up-to-date.

Option Name	Description
Show user path	(default false) Select to show user paths in references, instead of displaying resolved absolute paths. A user path is displayed with its name enclosed by ${ }.
Prompt to specify user path	(default false) When adding a reference comprises a path that is not defined as a user path, you will be prompted to add path as user path.

User Path Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

File Types Options

File Types Options

Model Reference lets you add reference(s) to external file or URL into diagram element. You can open the referenced file or URL
to get more information of the model in later stage. In the Options dialog box, you can configure to use specific application or command to open different types of file and specify your favorite web browser to open a URL. The system default handling method will be used if you have not configure the application or command to handle a particular file type.

Configure application/Command for file types

To configure application/command for file types:

1. Press on the upper **Add...** button. This shows a dialog box where you can add file extension.
2. Specify the **Extension**. Any file reference with this extension will be opened by the particular application or command. Note that for a valid extension a dot is required to put in front of the name of that extension, such as *.doc*.
3. Specify the **Application/Command**. The application or command for opening a file reference with file extension same as that defined in the **Extension** field.
   A command can be entered directly to the text field, and can include application arguments, while an application an be chosen from a file chooser by pressing **...** next to the text field.
4. Specify the **Name** of this application or command. This is an optional field for identifying this file extension.
5. Click **OK** to close the dialog box.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Spell Checking Options

Spell Checking Options

The Spell Checking feature supports spell checking in all inline editing, as well as in Textual Analysis. We support in-place editing of misspelled words, simply by right-clicking your mouse instead of using the complex spell-check box. Spell-check provides intelligent suggestions for words, and you can add your own words into your personal dictionary.

Options

Option Name	Description
Enable spell checking	(default true) Enable spell checking.
Dictionary	The choose of dictionary affects the judgment of correctness of words. • American - (default) Perform spell checking using an American dictionary. • British - Perform spell checking using an British dictionary. • Canadian - Perform spell checking using an Canadian dictionary.
Check spelling as you type	(default true) Check spelling when typing.
Ignore words in UPPERCASE	(default true) Do not classify the use of upper case in a word as a spelling mistake(unless the spelling is wrong).
Ignore words with numbers	(default true) Do not classify the inclusion of number in word as a spelling mistake (unless the spelling is wrong).
Ignore Internet and file address	(default true) Do not classify Internet and file address as a spelling mistake.

Spell Checking Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Keys Options

Keys Options

Customizable program shortcuts

Commands can be invoked by pressing certain keys in the keyboard, as shortcuts. For example, holding down the Ctrl modifier key with the 'S' key invokes the save command. Now, key bindings, which is the assignment of keys to commands, can be customized. This permits you to use the familiar keystroke for invoking commands in Agilian.

To assign/re-assign a key:

1. Double-click on the binding cell of the desired action.
2. Click on the **Binding** field at the bottom of dialog box.
3. Press the key for invoking the command selected. The binding field will be updated accordingly.
4. Press **OK** button to confirm the updates. You will be prompted to restart the application in order to make the changes take effect. By restarting, you can invoke commands using the key defined.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Import/Export Options

XMI Options

Option Name	Description
Enable business process diagram for enterprise architect (experimental feature)	Check this to include BPD when import and export XMI.

XMI options

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Diagramming Options

Appearance

Environment

Model Generation

Shape

Connector

Class

Association

Generalization

ERD & ORM

Interaction

Use Case Diagram

Activity and State

Component Diagram

Deployment Diagram

Business Process

Requirement Diagram

DFD

Communication Diagram

Textual Analysis

Appearance Options

Option Name	Description
Grid - Show grid	(default false) Show grid lines on diagram
Grid - Color	Color of grid lines.
Grid - Width	Determines the horizontal spaces between grid lines.
Grid - Height	Determines the vertical spaces between grid lines.
Grid - Snap to grid	(default true) When checked, shapes will be docked to the closest grid line when being created/ moved. Otherwise, shapes can be moved freely as if the grid does not exist.
Graphics anti-aliasing	(default true) Smoothen the graphics.
Text anti-aliasing	(default true) Smoothen the text.
Documentation Type	Default type of documentation • HTML - (default) HTML text that consists of formatting such as bold, italic, underline, table • Plain text - Text without formatting
Model Element Name Alignment	• Top Left - Shape name will appear at top left of shape • Top Middle - Shape name will appear at top middle of shape • Top Right - Shape name will appear at top right of shape • Middle Left - Shape name will appear at middle left of shape • Middle - (default) Shape name will appear at middle middle of shape • Middle Right - Shape name will appear at middle right of shape • Bottom Left - Shape name will appear at bottom left of shape • Bottom Middle - Shape name will appear at bottom middle of shape • Bottom Right - Shape name will appear at bottom right of shape
Diagram background	Background color of diagrams.
Enable minimum size	(default true) Determines whether shapes are restricted to a built-in minimum size.
Fractional Metrics	(default true) When checked, fit size of shape will be performed correctly. When disabled, the shape may look better but size may not fit.
Show Package Name Style	• Within Package Body - The name of package will be shown at the top of package shape • Within Package Tab - The name of package will be shown inside the package tab

Appearance Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Environment Options

Option Name	Description
Default HTML Documentation Font	The default font face, size, color, bold and italic status for HTML content in documentation pane.
Stereotype support HTML tagged value	Enables you to define tagged value in HTML format for stereotype.

Environment Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Model Generation Options

Option Name	Description
Default generate diagram type from scenario	• Sequence Diagram - (default) Take Sequence Diagram to be the type of diagram that will be generated by Scenario • Interaction Overview Diagram - Take Interaction Overview Diagram to be the type of diagram that will be generated by Scenario
ID Generator Format	Specify the ID format of various element types. If you want ID be generated to a type of model element and you find the type does not exist in the table, click **Add** to add the type manually. Then, specify the format of ID by specifying prefix, number of degits and suffix.

Model Generation Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Shape Options

Option Name	Description
Font	Default font settings for shape content, inlcuding font, size, color, bold and italic.
Shape line format	The default line format for shapes.
Shape fill format	The default fill format for shapes.
Reset Formats to Default	Click to reset formatting properties to factory-default. By clicking it, you are asked if you want also to reset the formatting of all the shapes in project. Then, you are asked if you want to apply the setting to workspace. By applying to workspace, newly created project will follow the factory-default formatting setting, too.
Auto fit size (diagram-based)	(default false) Determines whether shapes in diagrams will fit in size automatically.

Shape Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Connector Options

Option Name	Description
Font	Default font settings for connector caption.
Connector Style	• Rectilinear - Set the default connector style to be rectilinear • Round Rectilinear - Set the default connector style to be round rectilinear • Oblique - (default) Set the default connector style to be oblique • Round Oblique - Set the default connector style to be round oblique • Curve - Set the default connector style to be curve
Connection Point Style	• Round the shape - (default) Set the connector end to attach the round the shape • Follow center - Set the connector end to point to the center of attached shapes
Line Jumps	• Off - (default) Disable line jump • Arc - Show connectors' intersections as an arcs • Gap - Show connectors' intersections as a gaps • Square - Show connectors' intersections as a squares
Line jump size	• Normal • Large • Extra large
Caption orientation	• Horizontal only - Enforce connector caption to appear horizontally regardless of connector angle • Horizontal or Vertical only - Enforce connector caption to appear either horizontally or vertically, depending on the connector angle • Follow Connector Angle - Enforce connector caption to appear an the same horizontal level as the connector • Follow Connector Angle and Keep Text Upright - Enforce connector caption to appear an the same horizontal level as the connector, but keep the text upright
Foreground	Foreground color of connector.
Background	Background color of connector.
Paint connector through label	(default true) Captions' background will become transparent so that connectors can show completely without having part of it covered by opaque caption.

Connector Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Class Options

Option Name	Description
Auto-synchronize role name	(default true) Rename role when the owner class is being renamed.
Auto-generate role name	(default false) Auto generate role names for a relationship when the relationship is between created.
Support multiple-line attribute	(default true) Allow to enter attribute name in multiple lines by pressing the new line key defined in **Diagramming > Shape**.
Support multiple-line class name	(default true) Allow to enter class name in multiple lines by pressing the new line key defined in **Diagramming > Shape.**
Update constructor after class renamed	• Auto rename - Automatic update constructor name when the class name is being updated • Do not rename - Do not update constructor name when the class name is being updated • Prompt - (default)
Synchronize documentation of Interface to sub-class	• Always synchronize • Do not synchronize • Prompt
Show row grid line within compartment of Classes in Class Diagram (diagram type-based)	(default false) Show a horizontal line between each attribute or operation in class.
Default parameter direction	• in - When creating a parameter in operation, the direction will be in • out - When creating a parameter in operation, the direction will be out • inout - (default) When creating a parameter in operation, the direction will be inout • return - When creating a parameter in operation, the direction will be return
Default Visibility - Class	• Unspecified - A new class will take Unspecified as visibility • private - A new class will take private as visibility • protected - A new class will take protected as visibility • package - A new class will take package as visibility • public - (default) A new class will take public as visibility • protected internal (.NET only) - A new class will take protected internal as visibility when programming language is set to be .NET • internal (.NET only) - A new class will take internal as visibility when programming language is set to be .NET
Default Visibility - Attribute	• Unspecified - A new attribute will take Unspecified as visibility • private - (default) A new attribute will take private as visibility • protected - A new attribute will take protected as visibility • package - A new attribute will take package as visibility • public - A new attribute will take public as visibility • protected internal (.NET only) - A new attribute will take protected internal as visibility when programming language is set to be .NET • internal (.NET only) - A new attribute will take internal as visibility when programming language is set to be .NET
Default Visibility - Operation	• Unspecified - A new operation will take Unspecified as visibility • private - A new operation will take private as visibility • protected - A new operation will take protected as visibility • package - A new operation will take package as visibility • public - (default) A new operation will take public as visibility • protected internal (.NET only) - A new operation will take protected internal as visibility when programming language is set to be .NET • internal (.NET only) - A new operation will take internal as visibility when programming language is set to be .NET
Default Attribute Value - Multiplicity	Specify the multiplicity of attribute apply when creating a new attribute.

Default Attribute Value - Ordered	Specify the ordered property is checked or not when creating a new attribute.
Default Attribute Value - Unique	Specify the unique property is checked or not when creating a new attribute.

Class Options details

Presentation

Option Name	Description
Show attribute option	• Show all - (default) Show all attributes in Classes • Show public only - Show all public attributes in Classes • Hide all - Hide all attributes in Classes
Show type option	• Fully-qualified - Show attribute type, operation return type and parameter type as full qualified class name • Name only - (default) Show attribute type, operation return type and parameter type as class name • Relative - Show attribute type, operation return type and parameter type as relative class name
Show operation option	• Show all - (default) Show all operations in Classes • Show public only - Show all public operations in Classes • Hide all - Hide all operations in Classes
Visibility style	• Icon - Show icons for representing class members' visibilities • UML - (default) Show icons for representing class members' visibilities such as + for public, minus for private • None - Do not display visibilities
Show attribute initial value	(default true) Show initial value of attribute after its name.
Show attribute multiplicity	(default false) Show multiplicity of attribute after its name.
Show attribute getter/setter	(default false) Show getter and setter symbol for attribute, in front of attribute name.
Show operation signature	(default true) Show operation signature.
Show class member stereotype	(default true) Show the stereotypes set to attributes and operations.
Wrap class member	(default false) Automatic wrap class member against the class's width.
Show owner of class/package	(default false) Show the owner of class or package in class shape.
Show template parameter	(default true) Show template parameter of class.
Display as Robustness Analysis icon	(default true) Display class as robustness analysis icon for classes stereotyped as boundary/ control/entity.
Display as stereotype icon	(default false) Display stereotyped class as stereotype icon.
Show operation parameter name	(default true) Show operation parameter name. When disabled, only parameter type, if defined, would be shown.
Show empty compartments	(default false) Show compartments even when no members are defined.

Presentation of Class Options details

Auto Attribute Type

Option Name	Description
Default attribute type	Define attribute type that will be applied to newly created attributes.
Auto set attribute type by name	(default true) Automatically set attribute type and default value when the name user entered for an attribute matches with one of those listed in the table followed.

Auto Attribute Type of Class Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Association Options

Option Name	Description
Show association stereotype	(default true) Show the stereotypes assigned to an association.
Show from role name	(default true) Show the role name of the from end of association.
Show to role name	(default true) Show the role name of the to end of association.
Show from role visibility	(default true) Show the role visibility of the from end of association.
Show to role visibility	(default true) Show the role visibility of the to end of association.
Show from multiplicity	(default true) Show the role multiplicity me of the from end of association.
Show to multiplicity	(default true) Show the role multiplicity of the to end of association.
Show multiplicity constraints	(default false) Show multiplicity constraint such as {unique} for roles.
Show direction	(default false) Show a triangle mark on association for indicating direction.
Show association role stereotypes	(default true) Show stereotypes assigned to role.
Default Association End Navigable	• Unspecified - A new association will set Navigable as Unspecified for both ends • True - (default) A new association will set Navigable as True for both ends • False - A new association will set Navigable as False for both ends
Default Association End Visibility	• Unspecified - (default) A new association will set Visibility as Unspecified for both ends • private - A new association will set Visibility as private for both ends • protected - A new association will set Visibility as protected for both ends • package - A new association will set Visibility as package for both ends • public - A new association will set Visibility as public for both ends • protected internal (.NET only) - A new association will set Visibility as protected internal for both ends when programming language is set to be .NET • internal (.NET only) - A new association will set Visibility as internal for both ends when programming language is set to be .NET
Suppress implied "1" multiplicity for attribute and association end	(default false) Suppress implied "1" multiplicity for attribute and association end.
Synchronize association name with association class	• Prompt - Prompt if you want to make the name of association follow the association class when creating the association class • Yes - Make the name of association follow the association class when creating the association class • No - Do not make the name of association follow the association class when creating the association class

Association Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Generalization Options

Option Name	Description
Generalization set notation	• One Shape per Generalization - One generalization set shape per each Generalization relationship • Common Generalization Arrowhead - (default) Combine Generalization relationships' arrow head for the same set • Dashed Line

Generalization Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

ERD & ORM Options

Option Name	Description
Show column type	(default true) Show data type of column.
Foreign key connector end points to associated column	(default false) Attach foreign key connector end points to the column associated.
Align column properties	(default true) Align the column properties so that columns in entity will appear tidier.
Show extra column properties	(default true) Show extra column properties such as Nullable.
Show row grid line within compartment of Entities/Views in ERD (diagram type-based)	(default true) Show grid lines between row within Entities and Database Views in ERD.
Show row grid line within compartment of Entities/Views/Classes in ORM Diagram (diagram type-based)	(default true) Show grid lines between row within Entities, Database Views and Classes in ORM.
Warning on create ORM-Persistable Class in default package	(default true) Warn when creating ORM Persistable class at root.
Show schema name in ERD: ${name}.${tableName}	(default true) Show schema name, if defined, for entities.
Show table record editor	
Column Constraints Presentation	• Symbol • Text • Icon
Foreign key arrow head size	• Very Small • Small • Medium (default) • Large • Extra Large • Jumbo • Colossal
Primary Key Pattern	Pattern of primary keys that will be applied when synchronizing Class Diagram to Entity Relationship Diagram, which may create primary key.
Primary Key Constraint Patterm	Pattern of primary key constraint.
Foreign Key Pattern	Pattern of foreign key.
Foreign Key Relationship Pattern	Pattern of foreign key relationship pattern.
Index Pattern	Pattern of index.

ERD & ORM Options details

Auto Column

Option Name	Description
Default column type	Define column type that will be applied to newly created columns.
Default column nullable	Specify if a newly created column is nullable.
Auto set column type by name	(default true) Automatically set column type and default value when the name user entered for a column matches with one of those listed in the table followed.

Auto Column Type of ERD & ORM Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Interaction Options

Option Name	Description
Mark target lifeline stopped when attached by destroy message	The effect when attaching a Destroy Message to a Lifeline, whether the Lifeline will be marked stopped or not • Yes - Lifeline will mark as stopped • No - Lifeline will not mark as stopped • Prompt - (default) Ask if you want to mark the Lifeline as stopped
Show sequence number in Communication Diagram	(default true) Show numbering on Sequence Message in Communication Diagram.
Show sequence number in Sequence Diagram	(default true) Show numbering on Sequence Message in Sequence Diagram.
Show messages operation signature in Sequence Diagram and Communication Diagram (diagram-based)	(default false) Show Sequence Messages' operation signatures in Sequence Diagram and Communication Diagram.
Show stereotype of message in Sequence Diagram and Communication Diagram (diagram-based)	(default true) Show Sequence Messages' stereotypes in Sequence Diagram and Communication Diagram.
Show activations in Sequence Diagram (diagram-based)	(default true) Show Activations in Sequence Diagram. If unchecked, sequence message will be attached to Lifeline instead of Activations.
Display as Robustness Analysis icon in Sequence Diagram	(default true) Display Lifeline as robustness analysis icon for Lifelines stereotyped as boundary/ control/entity.
Show associated diagram name of Interaction	(default false)

Interaction Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Use Case Diagram Options

Option Name	Description
Show Use Case Extension Points	(default true) Show Use Case Extension Points within Use Case shapes.
Show Use Case ID	(default false) Show the ID of use cases.
Rename Extension Point to Follow Extend Use Case	• Yes - Rename Extension Points to follow extend use case automatically when the name of extend use case is changed. • No - Even when the name of extend use case is changed, the name of Extension Points will not change to follow extend use case. • Prompt (default) - Ask if rename Extension Points to follow extend use case when the name of extend use case is changed.

Use Case Diagram Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Activity and State Options

Activity Diagram

Option Name	Description
Show Caption	• Initial Node - (default false) Show caption for Initial Node • Expansion Node - (default false) Show caption for Expansion Node • Activity Final Node - (default false) Show caption for Activity Final Node • Flow Final Node - (default false) Show caption for Flow Final Node • Decision Node - (default false) Show caption for Decision Node • Merge Node - (default false) Show caption for Merge NodeArtifactat • Fork Node - (default false) Show caption for Fork Node • Join Node - (default false) Show caption for Join Node
Show Partition Header	• Auto - (default) Show horizontal and/or vertical Partition headers if there is Partitions in that orientation • Always Show - Always show Partition headers regardless of the orientation of Partitions, even if there is no Partition • Always Hide - Always hide Partition headers
Decision/Merge Node connection point style	Determines how connector connects to decision/merge node. • Default (default) • Connect to vertex
Show Object Node Type	(default true) Show the type of object node inside the object node shape.

Activity Diagram Options details

State Machine Diagram

Option Name	Description
Show Caption (State Machine Diagram)	• Shallow History - (default false) Show caption for Shallow History • Deep History - (default false) Show caption for Deep History • Initial Pseudo State - (default false) Show caption for Initial Pseudo State • Junction - (default false) Show caption for Junction • Final State - (default false) Show caption for Final State • Terminate - (default false) Show caption for Terminate • Fork - (default false) Show caption for Fork • Join - (default false) Show caption for Join
Auto create Initial State on State Diagram	(default true) Automatic create an initial state when creating a State Machine Diagram.
Default location (in pixel)	Position of Initial State create by default.
Use state name tab	(default false) Name tab is a a tiny rectangle that appear on top of a state and at the left hand side, displaying the name of a state. Use state name tab is to enable such tab.
Show transition trigger	Triggers can be added to a Transition relationship. This option determines the visibility of Triggers. • Show - (default) Show Triggers information on a Transition connector • Hide - Do not show Triggers information on a Transition connector
Show precondition, postcondition and body of internal activities in State	(default true) Show the precondition, postcondition and body of internal activities in state.

State Machine Diagram Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Component Diagram Options

Option Name	Description
Show Component Option	• Keyword - Show only the keyword <<component>> at the top of Component • Icon - Show only an icon representing a Component at the top right of Component • Keyword and Icon - (default) Show both keyword and icon for a Component • None - Do not show keyword and icon for a Component

Component Diagram Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Deployment Diagram Options

Option Name	Description
Show Artifact Option	• Keyword - Show only the keyword <<artifact>> at the top of Artifact • Icon - Show only an icon representing an Artifact at the top right of Artifact • Keyword and Icon - (default) Show both keyword and icon for a Artifact • None - Do not show keyword and icon for a Artifact

Component Diagram Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Business Process Options

Option Name	Description
Connect Gateway with Flow Object in Different Pool	• Prompt - (default) Prompt if user want to change the Message to Message Flow, Sequence Flow or cancel the action • Connect with Message Flow - Change or keep the relationship as Message Flow • Connect with Sequence Flow - Change or keep the relationship as Sequence Flow
Auto stretch pools	(default true) Stretch Pools automatically to the reach diagram bound.
Default Type of Sub-Process	The default type of newly created sub-process. • Unspecified (default) • Embedded • Reusable • Reference • Event
Business Process Diagram default language	The default value of language property of BPD
ID Generator Format - Show ID option	• Not Show - (default true) Do not display ID • Show Below Caption - Display ID as part of the caption, under the name • Show as Label - Display ID as a label attaching to shape • Show as Customized - Set the way to show ID per element type

Business Process Options details

Presentation

Option Name	Description
Connection Point Style	• Round the shape - Set the connection point style to Round the shape • Follow center - (default) Set the connection point style to Follow center
Connector Style	• Rectilinear - Set the connector style to Rectilinear • Round Rectilinear - (default) Set the connector style to Round Rectilinear • Oblique - Set the connector style to Oblique • Round Oblique - Set the connector style to Round Oblique • Curve - Set the connector style to Curve
Show Activities type icon (diagram-based)	(default true) Show icons that represent the type of Task and Sub-Process.
Data-based gateway markers visible	(default false) Show a cross in BPMN gateway shape when the gateway has a "Data-Based Exclusive Decision/Merge (XOR)" type.
Draw text annotation open rectangle folow connector end	(default true) Make the open rectangle of text annotation follow the position of end of the attached connector.

Business Process Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Block Definition Diagram

Option Name	Description
Wrap compartment text	Wrap the text in compartments like part, reference, value and flow property in SysML block

Block Definition Diagram Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Requirement Diagram Options

Option Name	Description
Show Attributes	• Show All Attributes - (default) Show all Requirement attributes • Show Non-empty Attributes - Show only Requirement attributes that have values defined • Hide All Attributes - Hide all Requirement attributes
Wrap member	(default false) Wrap the Requirement members' content.
Support HTML Attribute	(default false) Allow to fill in attributes with rich text format.

Requirement Diagram Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

DFD Options

Option Name	Description
Add Data Stores and External Entities to Decomposed DFD	• Yes - When decompose a DFD, Data Stores and External Entities on the current diagram will be copied to the decompose diagram • No - When decompose a DFD, Data Stores and External Entities on the current diagram will not be copied to the decompose diagram • Prompt - (default) When decompose a DFD, prompt if user want the Data Stores and External Entities on the current diagram to be copied to the decompose diagram

DFD Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Communication Diagram Options

Option Name	Description
Use one message for one direction	On a link between lifeline, you can choose whether to group message arrows that follow the same direction into a single arrow, or show as separate message arrows.

Communication Diagram Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Textual Analysis Options

Option Name	Description
Highlight Option	• Case insensitive - (default) Words which are the same as the entered word, even in different cases, are highlighted. • Case sensitive - Words which are the same as the entered word or/and with same case are highlighted.
Generate Requirement Text from Candidate Option	• Extracted text - (default) When create requirement from a candidate requirement, its text property will be filled by the candidate's extracted text. • Class description - When create requirement from a candidate requirement, its text property will be filled by the candidate's class description.

Textual Analysis options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

ORM Options

Synchronization Options

Option Name	Description
Entity => Class Name - Prefix	Append characters/words in front of name
Entity => Class Name - Class name	• Capitalize - (default) The first character of each word become uppercase • Decapitalize - The first character of each word become lowercase • Upper case - All characters become uppercase • Lower case - All characters become lowercase • Upper camel case - words are joined without underscore ("_") and are capitalized • Lower camel case - Same as upper camel case except that the first character is lower case • Reverse camel case - each upper case character are considered as word separator, words are joined with underscore ("_") and are lower case • Reverse camel to upper case - each upper case character are considered as word separator, words are joined with underscore ("_") and are upper case • Don't change - Keep name unchanged
Entity => Class Name - Suffix	Append characters/words after name.
Column => Attribute Name - Prefix	Append characters/words in front of name.
Column => Attribute Name - Attribute Name	• Capitalize - The first character of each word become uppercase • Decapitalize - (default) The first character of each word become lowercase • Upper case - All characters become uppercase • Lower case - All characters become lowercase • Upper camel case - words are joined without underscore ("_") and are capitalized • Lower camel case - Same as upper camel case except that the first character is lower case • Reverse camel case - each upper case character are considered as word separator, words are joined with underscore ("_") and are lower case • Reverse camel to upper case - each upper case character are considered as word separator, words are joined with underscore ("_") and are upper case • Don't change - Keep name unchanged
Column => Attribute Name - Suffix	Append characters/words after name.
Class => Entity Name - Prefix	Append characters/words in front of name.
Class => Entity Name - Table name	• Capitalize - The first character of each word become uppercase • Decapitalize - The first character of each word become lowercase • Upper case - All characters become uppercase • Lower case - All characters become lowercase • Upper camel case - words are joined without underscore ("_") and are capitalized • Lower camel case - Same as upper camel case except that the first character is lower case • Reverse camel case - each upper case character are considered as word separator, words are joined with underscore ("_") and are lower case • Reverse camel to upper case - each upper case character are considered as word separator, words are joined with underscore ("_") and are upper case • Don't change - (default) Keep name unchanged
Class => Entity Name - Suffix	Append characters/words after name.
Attribute => Column Name - Prefix	Append characters/words in front of name.
Attribute => Column Name - Column name	• Capitalize - (default) The first character of each word become uppercase • Decapitalize - The first character of each word become lowercase • Upper case - All characters become uppercase • Lower case - All characters become lowercase • Upper camel case - words are joined without underscore ("_") and are capitalized • Lower camel case - Same as upper camel case except that the first character is lower case • Reverse camel case - each upper case character are considered as word separator, words are joined with underscore ("_") and are lower case

	• Reverse camel to upper case - each upper case character are considered as word separator, words are joined with underscore ("_") and are upper case • Don't change - Keep name unchanged
Attribute => Column Name - Suffix	Append characters/words after name.
Synchronize Name	• Yes • No • Prompt
Table per subclass FK Mapping	• Top Base Class • Super Class

Synchronization Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Data Type Options

Data type options

UML is theoretically a modeling language independent to particular programming language(s). Yet, it is possible to transform between UML models to a software applications or systems. While the pre-defined data-type set works well in the UML world, there is enormous need to ensure the design can be applied to programming source code. Problems comes from the fact that programming languages, by nature, are unlikely to share the same set of data-types suggested by UML. A typical example is about the use of boolean. ‘boolean’, ‘bool’ and ‘Boolean’ are adopted by UML and Java, C# and VB.NET respectively. But they are all referring to the same thing – boolean.

Visual Paradigm lets you choose a programming language that your UML project should be based on. When modeling, you can easily select a data-type that is allowed for the chosen language, without typing it. Besides, new languages and data types can be added, which increase the flexibility of working under different domains.

Configure programming language

1. In **Diagram Navigator / Model Explorer/ Class Repository**, right click on the project root node and select **Configure Programming Language...** from the pop-up menu.
2. In the **Programming Language** dialog box, select the language to switch to. The way how data-type will be mapped from the current language to the chosen language is listed in the table following the data-type definition of that language.

Customizing programming language and data types

By default, there are six types of predefined (programming) languages. Each of them consists of a set of supported data types. Besides working with those default languages and types, you can add your own languages and add data types. To do so:

1. Open the **Project Options** window by selecting **Tools > Project Options** from the main menu.
2. Choose **Data Type** from the list on the left hand side of the **Project Options** window.
3. In the **Data Type** page, click the **plus** button under **Languages:** to add a language.
4. Enter its name, and press **OK** button to confirm.
5. Press **Add...** button to add a data-type to the chosen language.
6. Enter its name, and press **OK** to confirm. From now on, once you have set your own language as the language for your project, you can pickup the associated data-types as attribute type, operation return type and parameter type.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Model Quality Options

Model Quality Options

Option Name	Description
Enable model quality checking	Model quality checking is the ability to check project data for potential flaws. The chekcing is done automatically. Whenever a problem is detected, shapes will be underlined directly in diagram. You may turn this function off by unchecking this option.

Model Quality Options details

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Automatic update

Automatic update refers to the ability to detect and download possible updates from the Internet. This chapters shows you how to work with the automatic update feature.

Updating Agilian

How to perform manual update.

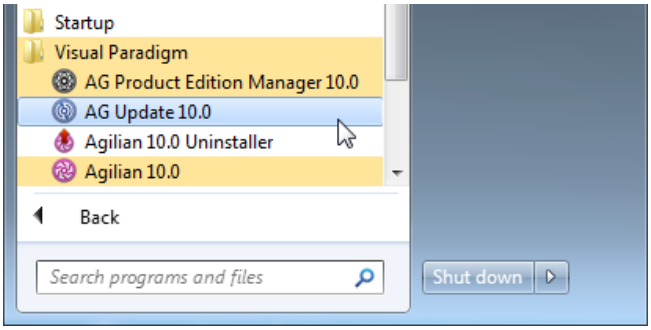
Automatic update notification

How automatic update works and how to react to update prompt.

Updating Agilian to Latest Hotfix Build

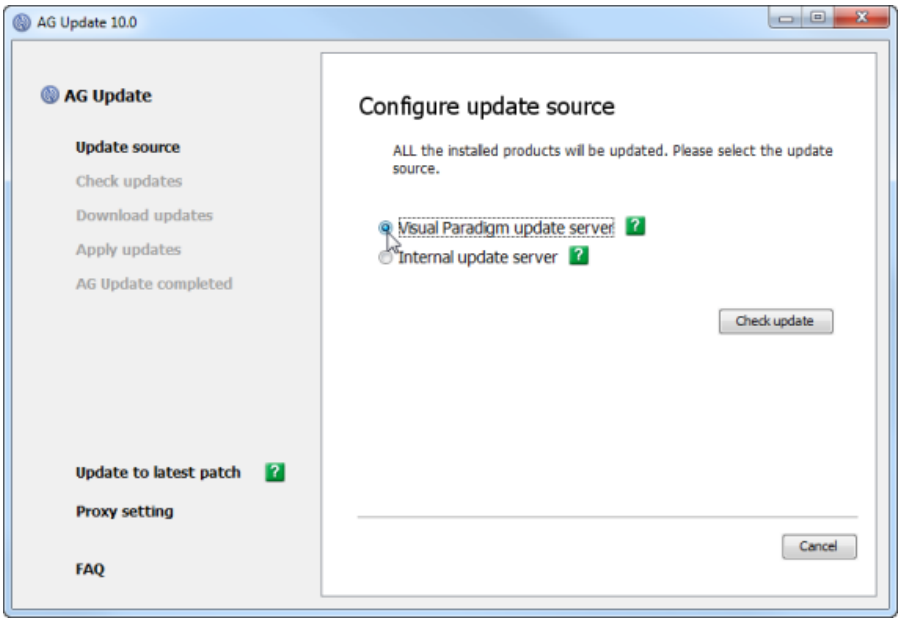
Visual Paradigm releases hotfixes from time to time. It is recommended to run the update program once a week to make sure the installation is up-to-date. With an up-to-date program, latest bug fixes and minor enhancements can be obtained.

1. Run the Agilian update program. You can run it via the Start Menu or by executing **Agilian Update** under **%Agilian_Install_DIR%\bin**.



Run Agilian update

2. This shows the update program. Select the place where the update program can look for the update files. If you want to update Agilian to the latest hotfix build, select **Visual Paradigm update server** and click **Check update**. If you have a specific place where the update file is stored, select **Internal update server** and fill in the URL. Click **Check update** button.



Select Visual Paradigm update server

> **NOTE:** If you need to configure a proxy server for connection, click **Proxy setting** at bottom left.

> **NOTE:** A patch is special build made that contains specific bug fix/enhancement, made for specific users.
> When and only when you are asked by Visual Paradigm to update to the latest patch build, click **Update to latest patch**.

3. Let the program download and update the files for you. When a file is found modified both in the latest build and the installation, you need to select whether to keep the local copy, by clicking **Ignore update** or to apply the latest version by clicking **Overwrite**. When a file is found removed in the latest build but exist in the installation, it will be listed in red. Since the file is obsolete and has already been removed in the latest version, neither **Overwrite** button nor **Ignore update** button will be shown on screen. It will be removed without choice.

4. Click **Complete** when finish.

Related Resources

The following resources may help you learn more about the topic discussed in this page.

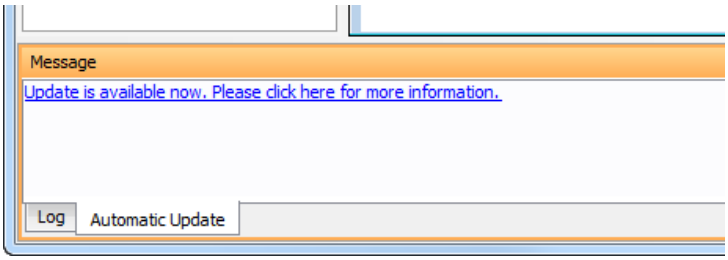
- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Automatic Update Notification

When you run Agilian, checking of possible updates is done in background. If there are available updates, you will be notified through the message pane. Then, you can perform an update to advance to the latest build.

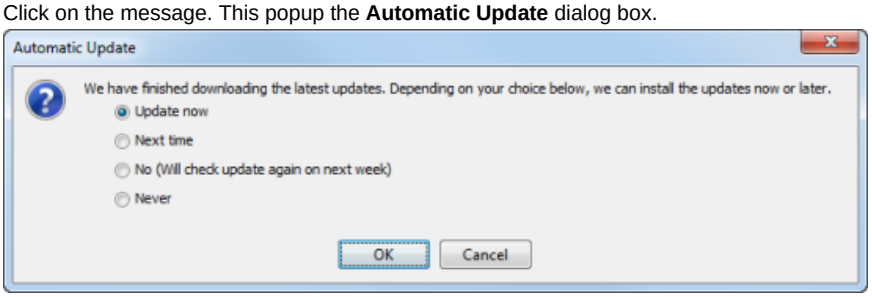
Updating when running Agilian

1. When running Agilian, a message " **Update is available now. Please click here for more information**" may popup in the Message pane. This message indicates that there is a build newer than the one that you are running, and you are recommended to perform an update to advance to the latest build.



Notification of available updates

2. Click on the message. This popup the **Automatic Update** dialog box.



Automatic update options

Here are the available options in the dialog box.

Option	Description
Update now	Run the product update now.
Next time	Check for product updates next time when starting Agilian.
No	Check for product updates after a certain period of time (The invertal can be defined in the Options dialog box. Refer to the section below for details).
Never	Do not check for product updates anymore.

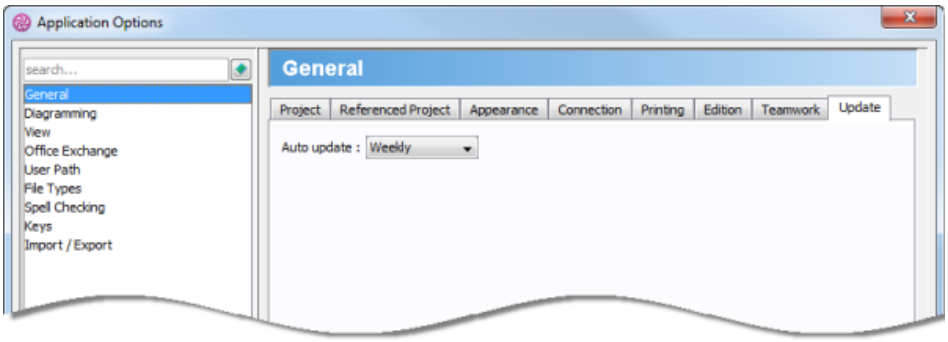
Available options for automatic update

3. Select **Update** now and click **OK** to proceed. This popup the **Agilian Update**. In order to update, close all the running VP application and continue.
4. Let the program update the files for you. When a file is found modified both in the latest build and the installation, you need to select whether to keep the local copy, by clicking **Ignore update** or to apply the latest version by clicking **Overwrite**. When a file is found removed in the latest build but exist in the installation, it will be listed in red. Since the file is obsolete and has already been removed in the latest version, neither **Overwrite** button nor **Ignore update** button will be shown on screen. It will be removed without choice.
5. Click **Complete** when finish.

Setting the interval of checking updates

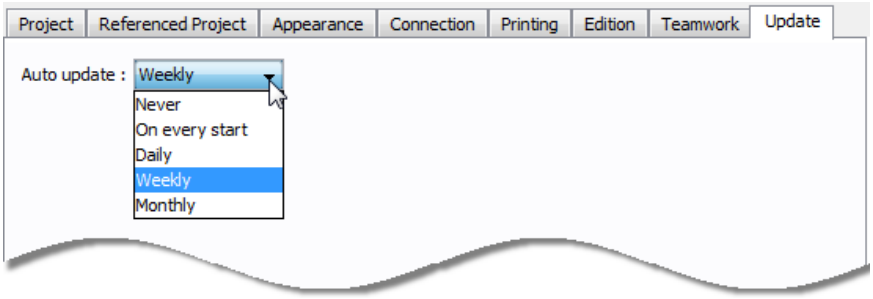
By default, update is checked weekly when starting Agilian. You can change the interval of checking updates through the **Application Options** window. To change:

1. Open the **Application Options** window by selecting **Tools > Application Options...** from the main menu.
2. In the **Application Options** window, select **General** from the list at the left hand side, then open the **Update** tab.



__Update__ page in __Application Options__ window

3. Select the interval of performing auto update from the **Auto update** drop down menu.



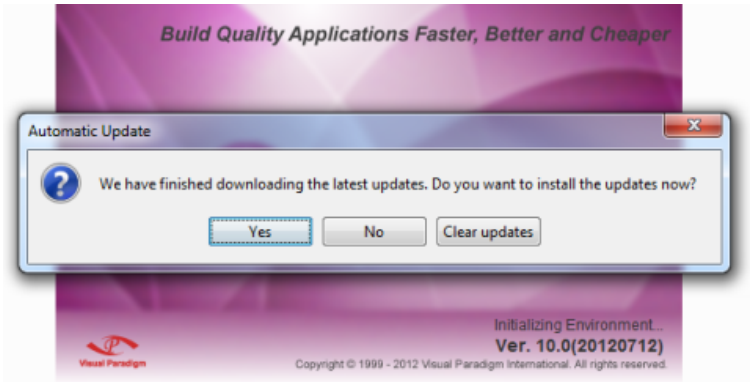
Selecting update interval

Here are the available options:

Option	Description
Never	Do not check for product updates anymore
On every start	Check for product updates everytime when starting Agilian
Daily	Check for product updates everyday, when starting Agilian
Weekly	Check for product updates every week, when starting Agilian
Monthly	Check for product updates every month, when starting Agilian

Available update interval

Click **OK** to confirm updating. From now on, once the interval elapsed, and if there are available updates, you will see the **Automatic Update** dialog box, letting you to update to the latest build.



Prompting for update when starting Agilian

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Connection rules

Model elements can be linked together using connectors. There are connection rules to control the type of model elements a connector support.

Use case diagram connection rules

Connection rules for shapes in use case diagram

Class diagram connection rules

Connection rules for shapes in class diagram

Sequence diagram connection rules

Connection rules for shapes in sequence diagram

Communication diagram connection rules

Connection rules for shapes in communication diagram

State machine diagram connection rules

Connection rules for shapes in state machine diagram

Activity diagram connection rules

Connection rules for shapes in activity diagram

Component diagram connection rules

Connection rules for shapes in component diagram

Deployment diagram connection rules

Connection rules for shapes in deployment diagram

Package diagram connection rules

Connection rules for shapes in package diagram

Object diagram connection rules

Connection rules for shapes in object diagram

Composite structure diagram connection rules

Connection rules for shapes in composite structure diagram

Interaction overview diagram connection rules

Connection rules for shapes in interaction overview diagram

Requirement diagram connection rules

Connection rules for shapes in requirement diagram

Basic diagram connection rules

Connection rules for shapes in basic diagram

Entity relationship diagram connection rules

Connection rules for shapes in entity relationship diagram

ORM diagram connection rules

Connection rules for shapes in ORM diagram

Business process diagram connection rules

Connection rules for shapes in business process diagram

Conversation diagram connection rules

Connection rules for shapes in conversation diagram

Data flow diagram connection rules

Connection rules for shapes in data flow diagram

EPC diagram connection rules

Connection rules for shapes in EPC diagram

Process map diagram connection rules

Connection rules for shapes in process map diagram

Organization chart diagram connection rules

Connection rules for shapes in organization chart diagram

Archimate diagram connection rules

Connection rules for shapes in archimate diagram

EJB diagram connection rules

Connection rules for shapes in EJB diagram

Overviewview diagram connection rules

Connection rules for shapes in overview diagram

Mind mapping diagram connection rules

Connection rules for shapes in mind mapping diagram

Use case diagram connection rules

	Actor	System	Collaboration	Use case
Actor	Association, Dependency, Realization, Generalization	Dependency, Realization	Dependency, Realization	Association, Dependency, Realization
System	Dependency, Realization	Dependency, Realization	Dependency, Realization	Dependency, Realization
Collaboration	Dependency, Realization	Dependency,	Dependency, Realization, Generalization	Association, Dependency, Realization
Use case	Dependency, Realization	Dependency, Realization	Association, Dependency, Realization	Association, Dependency, Include, Extend, Realization, Generalization

Connection rules in use case diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Class Diagram Connection Rules

	Class	NARY	Collaboration	Model
Class	Generalization, Realization, Usage, Association, Aggregation, Composition, Dependency, Binding Dependency, Permission, Access, Import, Merge, Instantiation, Substitution, Abstraction, Derive, Refine, Trace	Realization, Association, Aggregation, Composition, Dependency, Access, Import, Merge, Instantiation, Substitution	Realization, Dependency, Binding Dependency, Permission, Access, Import, Merge, Instantiation, Substitution, Abstraction, Derive, Refine, Trace	Realization Dependency, Binding Dependency, Permission, Access, Import, Merge, Instantiation, Substitution, Abstraction, Derive, Refine, Trace
NARY	Realization, Association, Aggregation, Composition, Dependency, Access, Import, Merge, Instantiation, Substitution	Realization, Dependency, Access, Import, Merge, Instantiation, Substitution	Realization, Dependency, Access, Import, Merge, Instantiation, Substitution,	Realization, Dependency, Access, Import, Merge, Instantiation, Substitution,
Collaboration	Realization, Dependency, Binding Dependency, Permission, Access, Import, Merge, Instantiation, Substitution, Abstraction, Derive, Refine, Trace	Realization, Dependency, Access, Import, Merge, Instantiation, Substitution	Generalization, Realization, Usage, Dependency, Binding Dependency, Permission, Access, Import, Merge, Instantiation, Substitution	Realization, Dependency, Binding Dependency, Access, Import, Merge, Instantiation, Substitution, Abstraction, Derive, Refine, Trace
Model	Realization, Dependency, Binding Dependency, Permission, Access, Import, Merge, Instantiation, Substitution, Abstraction, Derive, Refine, Trace	Realization, Dependency, Access, Import, Merge, Instantiation, Substitution	Realization, Dependency, Binding Dependency, Access, Import, Merge, Instantiation, Substitution, Abstraction, Derive, Refine, Trace	Generalization, Realization, Usage, Dependency, Binding Dependency, Access, Import, Merge, Instantiation, Substitution, Abstraction, Derive, Refine, Trace

Connection rules in class diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Sequence diagram connection rules

	Lifeline	Combined fragment	Interaction use	Frame	Actor	Concurrent	Continuation	Gate
Lifeline	Message				Message	Message		Message
Combined fragment								
Interaction use								
Frame								
Actor								
Concurrent	Message				Message			Message
Continuation								
Gate	Message				Message			

Connection rules in sequence diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Communication diagram connection rules

	Lifeline	Actor
Lifeline	Line, Dependency	Link, Dependency
Actor	Link, Dependency	Link, Generalization, Dependency

Connection rules in communication diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

State machine diagram connection rules

	State	Submachine state	Initial pspeudo state	Shallow history	Deep history	Choice	Junction	Fork	Join	Entry point	Exit point	Terminate	Final state
State	Transition	Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Submachine state	Transition	Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Initial pseudo state	Transition	Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Shallow history		Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Deep history		Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Choice		Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Junction		Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Fork		Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Join	Transition	Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Entry point	Transition	Transition		Transition	Transition	Transition	Transition	Transition	Transition		Transition	Transition	Transition
Exit point	Transition	Transition		Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition	Transition
Terminate													
Final state													

Connection rules in state machine diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Activity diagram connection rules

	Activity	Activity parameter node	Action	Accept event action	Send signal action	Decision node	Merge node	Fork node	Join node	Initial node	Activity final node	Flow final node	Input pin	Output pin
Activity	Control Flow	Object Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow	
Activity parameter node	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	
Action	Control Flow	Object Flow	Control Flow, Exception handler	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow, Exception handler	Exception handler
Accept event action	Control Flow	Object Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow, Exception handler	Exception handler
Send signal action	Control Flow	Object Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow, Exception handler	Exception handler
Decision node	Control Flow	Object Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow	
Merge node	Control Flow	Object Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow	
Fork node	Control Flow	Object Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow	
Join node	Control Flow	Object Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow	
Initial node	Control Flow	Object Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Object Flow	Control Flow	Control Flow	Object Flow	
Activity final node														
Flow final node														
Input pin														
Output pin		Object Flow				Object Flow	Object Flow	Object Flow	Object Flow		Object Flow	Object Flow		
Value pin														
Object node	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow		Object Flow	Object Flow	Object Flow	
Central buffer node	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow		Object Flow	Object Flow	Object Flow	
Data store node	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow		Object Flow	Object Flow	Object Flow	
Interruptible activity region														
Expansion region													Exception handler	Exception handler
Expansion node	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow		Object Flow	Object Flow	Object Flow	

Swimlane														
Structured Activity node	Control Flow		Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow		Control Flow	Control Flow	Exception handler	Exception handler
Conditional node	Control Flow		Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow		Control Flow	Control Flow	Exception handler	Exception handler
Loop node	Control Flow		Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow		Control Flow	Control Flow	Exception handler	Exception handler
Sequence node	Control Flow		Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow	Control Flow		Control Flow	Control Flow	Exception handler	Exception handler

Connection rules in activity diagram A

	Value pin	**Object node**	**Central buffer node**	**Data store node**	**Interruptible activity region**	**Expansion region**	**Expansion node**	**Swimlane**	**Structured Activity node**	**Conditional node**	**Loop node**	**Sequence node**
Activity		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Activity parameter node		Object Flow	Object Flow	Object Flow			Object Flow		Object Flow	Object Flow	Object Flow	Object Flow
Action		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Accept event action		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Send signal action		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Decision node		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Merge node		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Fork node		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Join node		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Initial node		Object Flow	Object Flow	Object Flow			Object Flow		Control Flow	Control Flow	Control Flow	Control Flow
Activity final node		Object Flow	Object Flow	Object Flow			Object Flow					
Flow final node		Object Flow	Object Flow	Object Flow			Object Flow					
Input pin												
Output pin							Object Flow					
Value pin												
Object node		Object Flow	Object Flow	Object Flow			Object Flow		Object Flow	Object Flow	Object Flow	Object Flow
Central buffer node		Object Flow	Object Flow	Object Flow			Object Flow		Object Flow	Object Flow	Object Flow	Object Flow
Data store node		Object Flow	Object Flow	Object Flow			Object Flow		Object Flow	Object Flow	Object Flow	Object Flow

Interruptible activity region									
Expansion region	Exception handler								
Expansion node		Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow	Object Flow
Swimlane									
Structured Activity node	Exception handler					Control Flow	Control Flow	Control Flow	Control Flow
Conditional node	Exception handler					Control Flow	Control Flow	Control Flow	Control Flow
Loop node	Exception handler					Control Flow	Control Flow	Control Flow	Control Flow
Sequence node	Exception handler					Control Flow	Control Flow	Control Flow	Control Flow

Connection rules in activity diagram B

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Component diagram connection rules

	Component	Interface	Port	Instance specification
Component	Dependency, Generalization, Realization, Association, Aggregation, Composition, Usage	Dependency, Association, Aggregation, Composition, Usage	Dependency, Generalization, Realization, Association, Aggregation, Composition, Usage	Dependency, Realization
Interface	Dependency, Realization, Association, Aggregation, Composition, Usage	Dependency, Generalization, Realization, Association, Aggregation, Composition, Usage	Dependency, Generalization, Realization, Association, Aggregation, Composition	Dependency, Generalization, Realization, Association, Aggregation, Composition
Port	Dependency, Realization, Association, Aggregation, Composition, Usage	Dependency, Association, Aggregation, Composition, Usage	Dependency, Realization, Association, Aggregation, Composition	Dependency, Realization
Instance specification	Dependency, Realization	Dependency, Generalization, Realization, Association, Aggregation, Composition	Dependency	Dependency, Generalization, Realization, Link

Connection rules in component diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Deployment diagram connection rules

	Node	Artifact	Deployment specification	Component	Interface	Port	Instance specification
Node	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Deployment, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Deployment, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation	Dependency, Manifestation, Realization	Dependency, Manifestation, Realization
Artifact	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Dependency, Manifestation	Dependency, Manifestation, Realization	Dependency, Manifestation, Realization
Deployment specification	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Dependency, Manifestation	Dependency, Manifestation, Realization	Dependency, Manifestation, Realization
Component	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization	Dependency, Manifestation, Realization
Interface	Association, Aggregation, Composition, Dependency, Manifestation, Realization	Dependency, Manifestation, Realization	Dependency, Manifestation, Realization	Association, Aggregation, Composition, Dependency, Manifestation, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization
Port	Dependency, Manifestation, Generalization, Realization	Dependency, Manifestation, Generalization, Realization	Dependency, Manifestation, Generalization, Realization	Association, Aggregation, Composition, Dependency, Manifestation, Generalization, Realization, Usage	Association, Aggregation, Composition, Dependency, Manifestation, Usage	Association, Aggregation, Composition, Dependency, Generalization, Realization	Dependency, Manifestation, Generalization, Realization
Instance specification	Dependency, Manifestation, Realization	Dependency, Manifestation, Realization	Dependency, Manifestation, Realization	Dependency, Manifestation, Realization	Association, Aggregation, Composition, Dependency, Manifestation, Generalization	Dependency, Manifestation, Realization	Link, Dependency, Manifestation, Deployment, Generalization, Realization

Connection rules in deployment diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems

- Contact us if you need any help or have any suggestion

Package diagram connection rules

	Package	Subsystem
Package	Dependency, Import, Access, Generalization, Realization, Merge	Dependency, Import, Access, Realization, Merge
Subsystem	Dependency, Import, Access, Realization, Merge	Dependency, Import, Access, Generalization, Realization, Merge

Connection rules in package diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Object diagram connection rules

	Instance specification	Class
Instance specification	Link, Dependency, Generalization, Realization	Association, Aggregation, Composition, Dependency, Generalization, Realization
Class	Association, Aggregation, Composition, Dependency, Generalization, Realization	Association, Aggregation, Composition, Dependency, Generalization, Realization

Connection rules in object diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Composite structure diagram connection rules

	Class	Part	Property	Interface	Port	Collaboration	Collaboration Use
Class	Dependency, Represents, Occurrence, Generalization, Realization	Dependency, Represents, Occurrence, Realization	Dependency, Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Generalization	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Generalization, Realization	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Realization
Part	Dependency, Represents, Occurrence,	Connector, Dependency, Represents, Occurrence	Connector, Dependency, Represents, Occurrence	Dependency, Represents, Occurrence	Connector, Dependency, Represents, Occurrence	Dependency, Represents, Occurrence	Dependency, Represents, Occurrence,
Property	Dependency, Represents, Occurrence	Connector, Dependency, Represents, Occurrence	Connector, Dependency, Represents, Occurrence	Dependency, Represents, Occurrence	Connector, Dependency, Represents, Occurrence	Dependency, Represents, Occurrence	Dependency, Represents, Occurrence
Interface	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Generalization, Realization	Dependency, Represents, Occurrence, Realization	Dependency, Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Generalization, Realization	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Generalization, Realization	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Realization
Port	Association, Aggregation, Composition, Dependency, Represents, Occurrence, Realization	Connector, Dependency, Represents, Occurrence, Realization	Connector, Dependency, Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency, Represents, Occurrence	Connector, Association, Aggregation, Composition, Dependency, Represents, Occurrence, Realization	Dependency, Represents, Occurrence, Realization	Dependency, Represents, Occurrence, Realization
Collaboration	Association, Aggregation, Composition, Dependency Represents, Occurrence, Realization	Dependency Represents, Occurrence, Realization	Dependency Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency Represents, Occurrence	Association, Aggregation, Composition, Dependency Represents, Occurrence, Realization	Dependency Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency Represents, Occurrence, Realization
Collaboration Use	Association, Aggregation, Composition, Dependency Represents, Occurrence, Realization	Dependency Represents, Occurrence, Realization	Dependency Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency Represents, Occurrence	Dependency Represents, Occurrence, Realization	Association, Aggregation, Composition, Dependency Represents, Occurrence, Realization	Dependency Represents, Occurrence, Realization

Connection rules in composite structure diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Interaction overview diagram connection rules

	Interaction	Interaction use	Decision node	Merge node	Fork node	Join node	Initial node	Activity final
Interaction	Control flow	Control flow	Control flow	Control flow	Control flow	Control flow		Control flow
Interaction use	Control flow	Control flow	Control flow	Control flow	Control flow	Control flow		Control flow
Decision node	Control flow	Control flow	Control flow	Control flow	Control flow	Control flow		Control flow
Merge node	Control flow	Control flow	Control flow	Control flow	Control flow	Control flow		Control flow
Fork node	Control flow	Control flow	Control flow	Control flow	Control flow	Control flow		Control flow
Join node	Control flow	Control flow	Control flow	Control flow	Control flow	Control flow		Control flow
Initial node	Control flow	Control flow	Control flow	Control flow	Control flow	Control flow		Control flow
Activity final								

Connection rules in interaction overview diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Requirement diagram connection rules

	Requirement	Model	Testcase
Requirement	Composition, Derive, Trace		
Model	Satisfy, Refine		
Testcase	Verify,		

Connection rules in requirement diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Basic diagram connection rules

	Process	Decision	Business actor	Document
Process	Connector	Connector	Connector	Connector
Decision	Connector	Connector	Connector	Connector
Business actor	Connector	Connector	Connector	Connector
Document	Connector	Connector	Connector	Connector

Connection rules in basic diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Entity relationship diagram connection rules

	Entity	View	Sequence	Stored procedures	Stored procedure resultset	Triggers
Entity	One-to-one relationship, One-to-many relationship, Many-to-many relationship	One-to-one relationship, One-to-many relationship, Many-to-many relationship				
View	One-to-one relationship, One-to-many relationship, Many-to-many relationship	One-to-one relationship, One-to-many relationship, Many-to-many relationship				
Sequence						
Stored procedures						
Stored procedure resultset						
Triggers						

Connection rules in entity relationship diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

ORM diagram connection rules

	Class	Entity	View
Class	Association, Aggregation, Composition	Class-entity mapping	Class-entity mapping
Entity	Class-entity mapping	One-to-one relationship, One-to-many relationship, Many-to-many relationship	One-to-one relationship, One-to-many relationship, Many-to-many relationship
View	Class-entity mapping	One-to-one relationship, One-to-many relationship, Many-to-many relationship	One-to-one relationship, One-to-many relationship, Many-to-many relationship

Connection rules in orm diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Business process diagram connection rules

	Task	Sub-process	Start event	Intermediate event	End event	Gateway	Choreography task	Choreography Sub-process	Call activity	Text annotation	Data Object	Pool/ Lane
Task	Sequence flow, Message flow	Sequence flow, Message flow	Message flow	Sequence flow, Message flow	Sequence flow,	Sequence flow, Message flow	Sequence flow,	Sequence flow,	Sequence flow, Message flow	Association, To-Direction Association	Association, To-Direction Association, Data Association	
Sub-process	Sequence flow, Message flow	Sequence flow, Message flow	Message flow	Sequence flow, Message flow	Sequence flow,	Sequence flow, Message flow	Sequence flow,	Sequence flow,	Sequence flow, Message flow	Association, To-Direction Association	Association, To-Direction Association, Data Association	
Start event	Sequence flow,	Sequence flow,		Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,	Association, To-Direction Association	Association, To-Direction Association	
Intermediate event	Sequence flow,	Sequence flow,		Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,	Association, To-Direction Association	Association, To-Direction Association	
End event	Message flow	Message flow	Message flow	Message flow					Message flow	Association, To-Direction Association	Association, To-Direction Association	
Gateway	Sequence flow, Message flow	Sequence flow, Message flow		Sequence flow, Message flow	Sequence flow,	Sequence flow, Message flow	Sequence flow,	Sequence flow,	Sequence flow, Message flow	Association, To-Direction Association	Association, To-Direction Association	
Choreography task	Sequence flow,	Sequence flow,		Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,		Association, To-Direction Association		
Choreography sub-process	Sequence flow,	Sequence flow,		Sequence flow,	Sequence flow,	Sequence flow,	Sequence flow,			Association, To-Direction Association		
Call activity	Sequence flow, Message flow	Sequence flow, Message flow	Message flow	Sequence flow, Message flow	Sequence flow,	Sequence flow, Message flow			Sequence flow, Message flow	Association, To-Direction Association	Association, To-Direction Association, Data Association	
Text annotation	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association		Association, To-Direction Association	Association, To-Direction Association
Data object	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association	Association, To-Direction Association			Association, To-Direction Association	Association, To-Direction Association	Data Association	
Pool/ lane										Association, To-Direction Association		

Connection rules in business process diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian

- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Conversation diagram connection rules

	Participant	Text annotation
Participant	Conversation link, Sub-conversation link, Call conversation link	Association
Text annotation	Association	

Connection rules in conversation diagram diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Data flow diagram connection rules

	Process	External entity	Data store
Process	Data Flow, Bidirectional data flow	Data Flow, Bidirectional data flow	Data Flow, Bidirectional data flow
External entity	Data Flow, Bidirectional data flow		
Data store	Data Flow, Bidirectional data flow		

Connection rules in data flow diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

EPC diagram connection rules

	Event	Function	And operator	Or operator	XOR operator	Organization unit	Process path	Information resource
Event		Control flow	Control flow	Control flow	Control flow		Control flow	
Function	Control flow	Control flow	Control flow	Control flow	Control flow			Information flow
And operator	Control flow	Control flow	Control flow	Control flow	Control flow			
Or operator	Control flow	Control flow	Control flow	Control flow	Control flow			
XOR operator	Control flow	Control flow	Control flow	Control flow	Control flow			
Organization unit		Organization unit assignment						
Process path	Control flow							
Information resource		Information flow						

Connection rules in EPC diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Process map diagram connection rules

	Process	Send	Receive
Process	Process link	Process link	Process link
Send	Process link	Process link	Process link
Receive	Process link	Process link	Process link

Connection rules in process map diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Organization chart connection rules

	Organization unit
Organization unit	Line

Connection rules in organization chart

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Fact diagram connection rules

	Term	Fact Type
Term	Fact Association, Generalization	Term-Fact Type Association
Fact Type		

Connection rules in fact diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Business motivation model diagram connection rules

	End	Vision	Goal	Objective	Means	Mission	Strategy	Tactic	Business Policy	Business Rule
End										
Vision										
Goal		Amplify	Composition							
Objective			Quantity	Composition						
Means										
Mission		Make Operative								
Strategy			Channel Efforts, Define, Require	Channel Efforts, Define, Require		Component of Plan for, Define, Require	Enable, Formulated Based on, Require, Composition		Define, Require	Define, Require
Tactic			Channel Efforts, Define, Require	Channel Efforts, Define, Require				Enable, Formulated Based on, Require, Composition	Define, Require	Define, Effect Enforcement Level, Require
Business Policy			Act as Regulation, Support Achievement	Act as Regulation, Support Achievement			Act as Regulation, Govern	Act as Regulation, Govern	Composition	Act as Regulation, Basis for
Business Rule			Act as Regulation,	Act as Regulation,			Act as Regulation, Govern	Act as Regulation, Govern		
Internal Influencer										
External Influencer										
Regulation										
Assessment	Affect Achievement, Use	Affect Achievement, Use	Affect Achievement, Use	Affect Achievement, Use	Affect Employment, Use	Affect Employment, Use	Affect Employment, Use	Affect Employment, Use	Affect Employment, Provide Impetus, Use	Affect Employment, Provide Impetus, Use
Risk									Provide Impetus	Provide Impetus
Potential Reward									Provide Impetus	Provide Impetus
Organization Unit	Define	Define	Define	Define	Define, Establish	Define, Establish	Define, Establish	Define, Establish	Define, Establish	Define, Establish
Business Process							Deliver, Realize	Deliver, Realize		
Asset										
Liability										

Connection rules in business motivation model diagram A

	Business Rule	Internal Influencer	External Influencer	Regulation	Assessment	Risk	Potential Reward	Organization Unit	Business Process	Asset	Liability
End											
Vision											

Goal											
Objective											
Means											
Mission											
Strategy	Define, Require							Define, Determine, Require		Define (Offerings), Require (Resources)	Define, Discharge, Require
Tactic	Define, Effect Enforcement Level, Require									Define,	Define, Discharge,
Business Policy	Act as Regulation, Basis for		Act as Regulation					Act as Regulation,	Act as Regulation, Govern	Act as Regulation, Govern	
Business Rule			Act as Regulation					Act as Regulation,	Act as Regulation, Govern, Guide	Act as Regulation, Govern	
Internal Influencer											
External Influencer											
Regulation											
Assessment	Affect Employment, Provide Impetus, Use	Judge, Use	Judge, Use	Judge, Use	Use	Identity, Use	Identity, Use				
Risk	Provide Impetus										
Potential Reward	Provide Impetus										
Organization Unit	Define, Establish	Define, Recognize, Source	Define, Recognize, Source	Define	Define, Make				Define, Responsible	Define (Offerings), Responsible	Define, Responsible
Business Process										Deliver (Offerings), Manage	
Asset											
Liability										Claim (Resources Only)	

Connection rules in business motivation model diagram B

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Archimate diagram connection rules

	Business actor	Business role	Business collaboration	Business process	Business function	Business interaction	Business event	Junction	Business service
Business actor	Aggregation, Composition, Specialization	Assignment							Realization
Business role	Assignment	Aggregation, Composition, Specialization		Assignment	Assignment	Assignment			
Business collaboration		Aggregation	Aggregation, Composition, Specialization			Assignment			
Business process		Assignment		Triggering, Aggregation, Composition, Specialization	Triggering	Triggering	Triggering	Triggering	Realization
Business function		Assignment		Triggering, Composition	Triggering, Aggregation, Composition, Specialization	Triggering	Triggering	Triggering	Realization
Business interaction		Assignment	Assignment	Triggering	Triggering	Triggering, Aggregation, Composition, Specialization	Triggering	Triggering	Realization
Business event				Triggering	Triggering	Triggering	Aggregation, Composition, Specialization		
Junction				Triggering	Triggering	Triggering	Triggering		
Business service				Used by	Used by	Used by			Aggregation, Composition, Specialization
Business interface		Association	Association						Assignment
Business object				Read access	Read access	Read access	Read access		Read access
Product	Used by								Aggregation
Contract									Read access
Representation									
Meaning									
Value									Association
Application collaboration						Assignment			
Application component				Assignment	Assignment	Assignment			
Application service				Used by	Used by	Used by			Used by
Application function									
Application interaction									
Application interface		Used by							
Data object									

Node
Device
System software
Infrastructure interface
Infrastructure service
Artifact

Connection rules in archimate diagram A

	Business interface	**Business object**	**Product**	**Contract**	**Representation**	**Meaning**	**Value**
Business actor	Line						
Business role	Association						
Business collaboration	Association						
Business process		Read access, Write access,					
Business function		Read access, Write access					
Business interaction		Read access, Write access					
Business event		Read access, Write access					
Junction							
Business service	Assignment	Read access, Write access		Read access, Write access			Association
Business interface	Aggregation, Specialization						
Business object		Association, Aggregation, Specialization			Association		
Product			Aggregation, Specialization	Aggregation			Association
Contract				Aggregation, Specialization			
Representation		Realization, Association			Aggregation, Specialization	Association	
Meaning					Association	Aggregation, Specialization	
Value			Association				Aggregation, Specialization
Application collaboration							
Application component							
Application service							
Application							

function	
Application interaction	
Application interface	
Data object	Realization
Node	
Device	
System software	
Infrastructure interface	
Infrastructure service	
Artifact	

Connection rules in archimate diagram B

	Application collaboration	**Application component**	**Application service**	**Application function**	**Application interaction**	**Application interface**	**Data object**
Business actor							
Business role						Require	
Business collaboration							
Business process		Assignment					
Business function							
Business interaction	Assignment	Assignment					
Business event							
Junction							
Business service				Used by	Used by		
Business interface		Used by					
Business object							
Product							
Contract							
Representation							
Meaning							
Value							
Application collaboration	Aggregation, Composition, Specialization	Association, Aggregation			Assignment	Provide, Require	
Application component	Association, Composition	Aggregation, Composition, Specialization	Realization	Assignment, Composition		Association, Provide, Require,	

						Composition	
Application service			Aggregation, Composition, Specialization		Assignment		Read access, Write access
Application function		Assignment	Realization	Aggregation, Composition, Specialization			Read access, Write access
Application interaction	Assignment		Realization		Aggregation, Composition, Specialization		Read access, Write access
Application interface		Association	Assignment			Aggregation, Composition, Specialization	
Data object			Read access, Write access	Read access, Write access	Read access, Write access		Association, Aggregation, Composition, Specialization
Node							
Device							
System software							
Infrastructure interface							
Infrastructure service							
Artifact		Realization					Realization

Connection rules in archimate diagram C

	Node	**Device**	**System software**	**Infrastructure interface**	**Infrastructure service**	**Artifact**
Business actor						
Business role						
Business collaboration						
Business process						
Business function						
Business interaction						
Business event						
Junction						
Business service						
Business interface						
Business object						

Product						
Contract						
Representation						
Meaning						
Value						
Application collaboration						
Application component						
Application service						
Application function						
Application interaction						
Application interface						
Data object						
Node	Communication path, Aggregation, Composition, Specialization	Composition	Composition	Provide, Require	Realization	Assignment
Device	Network	Aggregation, Composition, Specialization	Assignment	Provide, Require	Realization	Assignment
System software		Assignment	Aggregation, Composition, Specialization	Provide, Require	Realization	Assignment
Infrastructure interface				Aggregation, Composition, Specialization	Assignment	
Infrastructure service				Assignment	Aggregation, Composition, Specialization	Read access, Write access
Artifact	Assignment	Assignment	Assignment		Read access, Write access	Aggregation, Composition, Specialization

Connection rules in archimate diagram D

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Overview diagram connection rules

	Diagram overview
Diagram overview	Diagram containment, Directional generic connector

Connection rules in overview diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion

Mind mapping diagram connection rules

	Node
Node	Branch, Link, From link, To link

Connection rules in mind mapping diagram

Related Resources

The following resources may help you learn more about the topic discussed in this page.

- New to Agilian? We have a lot of enterprise architecture tutorials written to help you get started with Agilian
- Visual Paradigm on YouTube
- Visual Paradigm Know-How - Tips and tricks, Q&A, solutions to users' problems
- Contact us if you need any help or have any suggestion