Table of Contents

About Visual Paradigm for UML

Visual Paradigm for UML (VP-UML) is a powerful, cross-platform and yet the most easy-to-use visual UML modeling and CASE tool. VP-UML provides software developers the cutting edge development platform to build quality applications faster, better and cheaper! It facilitates excellent interoperability with other CASE tools and most of the leading IDEs which excels your entire Model-Code-Deploy development process in this one-stop-shopping solution.

UML Modeling

You can draw all kinds of UML 2.x diagrams in VP-UML, 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

Requirement Modeling

Capture requirements with SysML Requirement Diagram, Use Case Modeling, Textual Analysis, CRC Cards, and create screen mock-up with User Interface designer.

Database Modeling

You can draw the following kinds of diagrams to aid in database modeling:

- Entity Relationship Diagram
- ORM Diagram (visualize the mapping between object model and data model)

You can model not only database table, but also stored procedure, triggers, sequence and database view in an ERD.

Besides drawing a diagram from scratch, you can reverse engineer a diagram from an existing database.

Apart from diagramming, you can also synchronize between class diagram and entity relationship diagram to maintain the consistency between them.

SQL generation ane execution feature is available for producing and executing SQL statement from model instantly.

Business Process Modeling

You can draw the following kinds of diagrams to aid in business process modeling:

- Business process diagram
- Data flow diagram
- Event-drive process chain diagram
- Process map diagram
- Organization Chart

You can also export Business process diagram to BPEL.

Object-Relational Mapping

Object-Relational Mapping enables you to access relational database in an object relational approach when coding. VP-UML generates object-relational mapping layer which incorporates features such as transaction support, pluggable cache layer, connection pool and customizable SQL statement.

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

Interoperability

The interoperability support allows you to exchange model data with other tools. The following tools are supported:

	Import	Export
Telelogic Modeler	✓	
Rational Rose	✓	
ERwin Data Modeler project	✓	
Rational Software Architect	✓	
Rational DNX	✓	
NetBeans 6.x UML diagrams	✓	
Visio drawing	✓	
BPEL for Oracle workflow engine		✓
BPEL for JBoss workflow engine		✓
Diagram (JPG, PNG, SVG, EMF, PDF)		✓
Microsoft Excel	✓	✓
EMF based UML2 model	✓	✓
XMI (1.0, 1.2, 2.1)	✓	✓
XML (native)	✓	✓
VP project	✓	✓
Microsoft Word document (for use case model)	✓	✓

Table 1-1 *List of standard or tools that are covered by interoperability support*

Code Engineering

Instant Reverse and Instant Generator are provided for reversing engineering and forward engineering. In addition, the Java Round-Trip engineering support lets you keep code and model in-sync. Below are the type of source code that can be reversed or generated through Instant Reverse and Instant Generator.

	Instant Reverse	Instant Generator
Java	✓	✓
C++	✓	✓
XML Schema	✓	✓
PHP	✓	✓
Python Source	✓	✓
Objective-C	✓	✓
CORBA IDL Source	✓	✓
.NET dll or .exe files (binary)	✓	
CORBA IDL Source	✓	
XML (structure)	✓	
JDBC	✓	
Hibernate	✓	
C#		✓
VB.NET		✓
ODL		✓
ActionScript		✓
Delphi		✓
Perl		✓
Ada95		✓
Ruby		✓

Table 1-2 *List of souce code supported by Instant Reverse and/or Generator*

IDE Integration

Support full software development life-cycle, from analysis to design, and from design to implementation with your most favorite IDE. Below are the supported IDEs:

- Eclipse
- JBuilder
- NetBeans/Sun ONE
- IntelliJ IDEA
- JDeveloper
- WebLogic Workshop

Documentation Generation

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

- HTML (report generation)
- HTML (project publisher)

- PDF
- Word

Editions

Belows are the kinds of features that can be found in each edition of VP-UML. For the support of specific features, please refer to the Features section.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
UML Support	✓	✓	✓	✓	✓	✓	#
Requirements Management	✓	✓	✓	✓	✓	✓	#
User Interface Designer	✓						#
Business Process Modeling	✓						#
Animacian	✓						
Mind Mapping	✓						
Entity Relationship Diagram	✓	✓	✓	✓	✓	✓	#
ORM Diagram	✓	✓	✓	✓	✓	✓	#
Database Modeling	✓	✓					
Object-Relational Mapping (Java)	✓	✓					
Object-Relational Mapping (.NET)	✓						
General Visual Modeling	✓	✓	✓	✓	✓	✓	
Project Referencing	✓	✓	✓				
Visual Diff	✓	✓	✓				
Design Pattern Support	✓	✓	✓				
Model Element Nicknaming	✓	✓	✓				
Model Transitor	✓	✓	✓				
Style and Formatting	✓	✓	✓	✓	✓	✓	
Team Collaboration with VP Teamwork Server	✓	✓	✓	✓			
Team Collaboration with CVS Repository	✓	✓	✓				
Team Collaboration with Subversion Repository	✓	✓	✓				
Team Collaboration with Perforce	✓	✓	✓				
PDF and HTML report generation	✓	✓	✓	✓			
Word report generation	✓	✓	✓				
Project Publisher	✓	✓	✓				
Ad Hoc report creation	✓	✓	✓				
Printing	✓	✓	✓	✓	+	*	
IDE Integrations	✓	✓					
Reverse Engineering	✓	✓	✓				
Code Generation	✓	✓	✓				
State Machine Diagram Code Generation	✓	✓					
Java Round-Trip Engineering	✓	✓					

+ This feature is available but the single watermark will be shown.

* The feature is available but the single watermark will be shown when the project has one diagram per diagram type. And the pattern watermark will be shown when the project has more than one diagram per diagram type.

The feature is available but unable to make changes.

Features

UML Support

Improve modeling efficiency with this easy-to-use, feature-rich and reliable UML 2.1 modeling tool.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
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	✓	✓	✓	✓	✓	✓	#
Use case detail editor	✓	✓	✓	✓	✓	✓	#
Use case flow-of-events listing	✓	✓	✓	✓	✓	✓	#
Generate sequence diagrams from flow of events lists	✓	✓	✓	✓	✓	✓	
Select attribute's getter or setter as call message's action	✓	✓	✓	✓	✓	✓	
Business model use case support	✓	✓	✓	✓	✓	✓	

The feature is available but unable to make changes.

Table 1-4 *UML Supports in different editions of VP-UML*

Requirements Management

Capture requirements with SysML Requirement Diagram, Use Case Modeling, Textual Analysis, CRC and more!

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Requirement diagram	✓	✓	✓	✓	✓	✓	#
Textual analysis	✓	✓	✓	✓	✓	✓	#
CRC Cards	✓	✓	✓	✓	✓	✓	#
User interface designer	✓						#
Identify candidate activity and action by textual analysis	✓	✓	✓	✓	✓	✓	
Define and customize requirement types	✓	✓	✓				
Display full set of requirements in tabular view	✓	✓	✓	✓	✓	✓	
Support generating ID for Requirements	✓	✓	✓				

The feature is available but unable to make changes.

Table 1-5 *Requirements management support in different editions of VP-UML*

Business Process Modeling

Visualize, understand and improve your business process with the most comprehensive BPMN tool.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Business process diagram	✓						#
Data flow diagram	✓						#
Event-driven process chain diagram	✓						#
Process map diagram	✓						#
Export business process diagrams to BPEL	✓						
Identify candidate business process elements using textual analysis	✓						
Automatically stretch pools and lanes to fit diagram	✓						
Smart routing for connecting objects	✓						
Extend business process model with stereotype and tagged value	✓						
Set state for data object	✓						
Organization chart	✓						#
Relocate a branch of unit through drag and drop	✓						
Nested lanes support	✓						#

The feature is available but unable to make changes.

Table 1-6 *Business Process modeling support in different editions of VP-UML*

Animacian

A set of tools to help animating paths in a diagram, or to export animation to flash for further analysis.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Animate business process using Animacian	✓						
Animate sequence diagram using Animacian	✓						
Animate activity diagram using Animacian	✓						
Travel along execution path with slider	✓						
Step forward and backward in an animation	✓						
Auto validate execution path	✓						
Auto calculate all possible execution paths	✓						
Eliminate paths using filter	✓						
Dim non-execution path	✓						
Support Full and Compact Mode	✓						
Protect diagram when animating	✓						#
Export selected path to flash format	✓						
Animate and navigate animation in exported flash	✓						#

The feature is available but unable to make changes.

Table 1-7 *Animacian support in different editions of VP-UML*

Mind Mapping

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Mind map diagram	✓						
Create link relationships between nodes	✓						
Smart layout for mind mapping nodes and diagram	✓						

Table 1-8 *Mind Mapping support in different editions of VP-UML*

Database Modeling

Enhance database documentation quality with sophisticated ERD and Object-Relational Mapping diagrams.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Entity relationship diagram	✓	✓	✓	✓	✓	✓	#
ORM diagrams	✓	✓	✓	✓	✓	✓	#
Reverse engineer existing databases to entity relationship diagrams (ERDs)	✓	✓					
Generate and execute database schema (DDLs)	✓	✓					
Conceptual, logical and physical ERD support	✓	✓	✓	✓	✓	✓	
Reverse engineer stored procedures to ERDs	✓	✓					
Foreign key auto-naming	✓	✓					
Define PK naming pattern	✓	✓					
Define FK relationship naming pattern	✓	✓					
Model primary key in object model by using the <<PK>> stereotype	✓	✓					
Download database drivers automatically	✓	✓					#
Display database architecture in object-relational mapping (ORM) pane	✓	✓					
Customizable SQL generation	✓	✓					#
Generate class diagrams from ERDs	✓	✓					
Generate ERDs from class diagrams	✓	✓					
Select target diagram when first synchronized between class diagram and ERD	✓	✓					
Jump between ORM class and entity	✓	✓					
Generate and reverse engineer database support for Oracle schema	✓	✓					
Generate or reverse engineer user-defined database types	✓	✓					
Database trigger and stored procedure modeling	✓	✓					
Reverse engineer DDL models to ERD models	✓	✓					
Database trigger and stored procedure generation and reversal	✓	✓					
Unique and Index support for entities	✓	✓					
Convert a normal association to ORM association	✓	✓					
Support AbstractPersistable class for generating non ORM super class attributes	✓	✓					
Configurable generated SQL statements in upper or lower case	✓	✓					
Model and generate DB Sequence (Oracle and DB2 only)	✓	✓					

The feature is available but unable to make changes.

***Table 1-9** Database modeling support in different editions of VP-UML*

Object-Relational Mapping

Know how to manipulate objects in Java, .NET and PHP? Then you now know how to access relational databases. No more tedious database programming! Just leave it to us.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Java ORM code generation	✓	✓					
.NET ORM code generation	✓						
PHP ORM code generation	✓						
Lazy collection fetching	✓	✓					
Database view support	✓	✓					
Custom ID generator support	✓	✓					
Map single classes to multiple tables	✓	✓					
Custom query support	✓	✓					
Automatic array table generation	✓	✓					
Criteria class generation	✓	✓					
DAO code generation	✓	✓					
Factory code generation	✓	✓					
POJO code generation	✓	✓					
Optional library selection	✓	✓					
Hibernate annotation support in ORM persistence	✓	✓					
Generate Hibernate version tags for optimistic concurrency control	✓	✓					
Formula support for ORM attribute	✓	✓					
User defined exception support in error handling	✓	✓					
Define default type mapping between database column and class attribute	✓	✓					
Configurable naming pattern for synchronization between Class Diagram and ERD	✓	✓					

Table 1-10 *Object Relational Mapping support in different editions of VP-UML*

Visual Modeling

Improve the user-friendlyness of drawing diagrams, thus speed up the time need to spend on diagramming.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Reference other projects' model elements	✓	✓	✓				
Freehand shape support	✓	✓	✓	✓	✓	✓	
Justify shape name	✓	✓	✓	✓	✓	✓	
Support enable or disable minimum shape size	✓	✓	✓	✓	✓	✓	
Layer support to show/hide set of shape with single click	✓	✓	✓	✓	✓	✓	
Support showing line jump in Arc, Square, Skip or normal	✓	✓	✓	✓	✓	✓	
Align Connector Caption Base on Connector Orientation	✓	✓	✓	✓	✓	✓	
Annotation with callout shapes	✓	✓	✓	✓	✓	✓	
Annotation with freehand shapes	✓	✓	✓	✓	✓	✓	
Organize model elements and diagrams using Model Explorer	✓	✓	✓	✓	✓	✓	
Bookmark support	✓	✓	✓	✓	✓	✓	#
Handi-Selection tool	✓	✓	✓	✓	✓	✓	
Customizable data types for use with multiple programming languages	✓	✓	✓	✓	✓	✓	
Overview diagrams	✓	✓	✓	✓	✓	✓	#
Resource-centric interface	✓	✓	✓	✓	✓	✓	
Model sharing	✓	✓	✓	✓	✓	✓	
Cut, copy and paste	✓	✓	✓	✓	✓	✓	
Copy diagrams as images for use in other applications	✓	✓	✓	✓	✓	✓	
Undo and redo options	✓	✓	✓	✓	✓	✓	
Mouse gestures	✓	✓	✓	✓	✓	✓	
Reverse connector direction	✓	✓	✓	✓	✓	✓	
Group creation support	✓	✓	✓	✓	✓	✓	
Jump to feature for selecting a particular shape or model	✓	✓	✓	✓	✓	✓	✓
Quick connect feature	✓	✓	✓	✓	✓	✓	
Easy navigation to connected elements	✓	✓	✓	✓	✓	✓	
Model commenting	✓	✓	✓	✓	✓	✓	#
Duplicate shapes and models	✓	✓	✓	✓	✓	✓	
Selectable/non-selectable toggling for shapes	✓	✓	✓	✓	✓	✓	
Diagram locking	✓	✓	✓	✓	✓	✓	
Reference to any type of artifact	✓	✓	✓	✓	✓	✓	#
Advanced file and directory selector	✓	✓	✓	✓	✓	✓	✓

Visual Diff

A set of tools that help to compare diagrams, and identify their differences.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Compare and show difference between diagrams visually	✓	✓	✓				
Support compare by name, id and transitor options	✓	✓	✓				
Compare view, model element or both kinds of properties	✓	✓	✓				
Filter result by displaying all, addition, modification or deletion of items	✓	✓	✓				

Table 1-12 *Support of Visual Diff in different editions of VP-UML*

Design Pattern Support

Define a pattern and save it for future use, or share with team members for making the design consistent.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Design pattern support for class diagrams	✓	✓	✓				
Full Gang of Four (GOF) design patterns support	✓	✓	✓				
Define pattern with a configurable number of subclasses	✓	✓	✓				
Define pattern with a configurable number of attributes and operations	✓	✓	✓				
Define pattern with implementation hierarchy	✓	✓	✓				
Define pattern with automatically named association end	✓	✓	✓				
Automatically rename elements in pattern	✓	✓	✓				
Design pattern support for sequence diagrams	✓	✓	✓				
Design pattern support for all UML diagram types	✓	✓	✓				
Design pattern support for entity relationship diagrams	✓	✓	✓				
Design pattern support for business process diagrams	✓						
Sharing Design Patterns through team collaboration support	✓	✓	✓				

Table 1-13 *Design Pattern support in different editions of VP-UML*

Model Element Nicknaming

Define nickname for a set of models.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Assign nicknames to model elements	✓	✓	✓				

Table 1-14 *Model element nicknaming in different editions of VP-UML*

Model Transitor

Setup and maintain transition between models through the Model Transitor

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Generate and link model elements	✓	✓	✓				
Trace the origin of model elements (model traceability)	✓	✓	✓				
Trace the origin of model elements (model traceability)	✓	✓	✓				
Navigate between operation and sequence diagram	✓	✓	✓				
Transit Entity relationship diagram from Conceptual to Logical to Physical	✓	✓	✓				

Table 1-15 *Model Transitor support in different editions of VP-UML*

Style and Formatting

Define and apply style and formatting to shapes.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Manage and apply styles within project	✓	✓	✓	✓	✓	✓	
Customizable shape style and formatting	✓	✓	✓	✓	✓	✓	
Image incorporation in diagrams	✓	✓	✓	✓	✓	✓	#
Stereotyped element appearance	✓	✓	✓	✓	✓	✓	#
Rich text documentation	✓	✓	✓	✓	✓	✓	#
Add rich text elements to diagrams	✓	✓	✓	✓	✓	✓	
Shape format copier	✓	✓	✓	✓	✓	✓	
Enrich model documentation with images	✓	✓	✓	✓	✓	✓	#
Save/load template for model documentation	✓	✓	✓	✓	✓	✓	
Display stereotyped model element as image icon	✓	✓	✓	✓	✓	✓	#

The feature is available but unable to make changes.

Table 1-16 *Style and Formatting support in different editions of VP-UML*

Team Collaboration with VP Teamwork Server

Perform modeling collaboratively and concurrently with the VP Teamwork Server.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Concurrent and collaborative modeling with VP Teamwork Server	✓	✓	✓	✓			
Import projects to VP Teamwork Server	✓	✓	✓	✓			
View projects from VP Teamwork Server	✓	✓	✓	✓			#
Commit project changes to VP Teamwork Server	✓	✓	✓	✓			
Update local project copy using VP Teamwork Server	✓	✓	✓	✓			
Review past revisions using VP Teamwork Server	✓	✓	✓	✓			#
Compare past revisions using VP Teamwork Server	✓	✓	✓	✓			#
Undo committed changes by reverting revisions	✓	✓	✓	✓			
Detect and resolve conflicts using VP Teamwork Server	✓	✓	✓	✓			
Branch and tag projects using VP Teamwork Server	✓	✓	✓	✓			
Merge branch changes using VP Teamwork Server	✓	✓	✓	✓			
Export multiple revisions from VP Teamwork Server	✓	✓					
Run VP Teamwork Server on common Java web servers							
Element based revision history							
Preview local and server changes before commit or update							

The feature is available but unable to make changes.

Table 1-17 *Support of Teamwork Collaboration with Teamwork Server in different editions of VP-UML*

Team Collaboration with CVS Repository

Perform modeling collaboratively and concurrently with CVS.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Concurrent and collaborative modeling with CVS Repository							
Import projects to CVS Repository							
View projects from CVS Repository							
Commit project changes to CVS Repository							
Update local project copy using CVS Repository							
Review past revisions using CVS Repository							
Compare past revisions using CVS Repository							
Undo committed changes by reverting revisions							
Detect and resolve conflicts using CVS Repository							
Branch and tag projects using CVS Repository							
Merge branch changes using CVS Repository							
Export multiple revisions from CVS Repository							
Element based revision history							
Preview local and server changes before commit or update							

The feature is available but unable to make changes.

Table 1-18 *Support of Teamwork Collaboration with CVS Repository in different editions of VP-UML*

Team Collaboration with Subversion Repository

Perform modeling collaboratively and concurrently with Subversion.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Concurrent and collaborative modeling with Subversion Repository							
Import projects to Subversion Repository							
View projects from Subversion Repository							
Commit project changes to Subversion Repository							
Update local project copy using Subversion Repository							
Review past revisions using Subversion Repository							
Compare past revisions using Subversion Repository							
Undo committed changes by reverting revisions							
Detect and resolve conflicts using Subversion Repository							
Branch and tag projects using Subversion Repository							
Merge branch changes using Subversion Repository							
Export multiple revisions from Subversion Repository							
Element based revision history							
Preview local and server changes before commit or update							

The feature is available but unable to make changes.

Table 1-19 *Support of Teamwork Collaboration with Subversion Repository in different editions of VP-UML*

Team Collaboration with Perforce

Perform modeling collaboratively and concurrently with the Perforce.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Concurrent and collaborative modeling with Perforce Repository							
Import projects to Perforce Repository							
View projects from Perforce Repository							
Commit project changes to Perforce Repository							
Update local project copy using Perforce Repository							
Review past revisions using Perforce Repository							
Compare past revisions using Perforce Repository							
Undo committed changes by reverting revisions							
Detect and resolve conflicts using Perforce Repository							
Branch and tag projects using Perforce Repository							
Merge branch changes using Perforce Repository							
Export multiple revisions from Perforce Repository							
Element based revision history							
Preview local and server changes before commit or update							

The feature is available but unable to make changes.

Table 1-20 *Support of Teamwork Collaboration with Perforce in different editions of VP-UML*

Documentation Generation

Share your design with your customers in popular document formats (PDF, HTML and MS Word).

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
PDF report generation							
MS Word report generation [1]							
HTML report generation							
Project publisher							
Ad Hoc report creation [1]							
Intelligent element sorting during report generation							

[1] This feature is only available for the Windows platform.

Table 1-21 *Documentation generation support in different editions of VP-UML*

Printing

Print out diagrams using the Print features.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Print multiple diagrams							
Preview printable pages							
Print clip marks							
Page margin, size and orientation alteration support							
Fit-to-pages option							
Fit-to-ratio option							
Customizable page header and footer							
Project name and diagram name display in header or footer							
Print with frame or border support							
Toggle gradient color printing							
Quick print support							

This feature is available but the single watermark will be shown.

The feature is available but the single watermark will be shown when the project has one diagram per diagram type. And the pattern watermark will be shown when the project has more than one diagram per diagram type.

Table 1-22 *Printing support in different editions of VP-UML*

IDE Integrations

Support full software development life-cycle, from analysis to design, and from design to implementation with your most favorite IDE.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Activate full UML environment from your favorite IDE							
Automatic code and model synchronization							
Simple integration of any IDE							
Import existing VP-UML project to IDE							
Integration with Eclipse							
Integration with NetBeans							
Integration with IntelliJ IDEA							
Integration with WebLogic Workshop							
Integration with Borland JBuilder							
Integration with Oracle JDeveloper							
Multilingual support in IDE integration							

Table 1-23 *IDE integration support in different editions of VP-UML*

Reverse Engineering

Reverse engineer UML models from source code.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Reverse engineer source code or executables to class diagrams using Instant Reverse feature							
Reverse engineer Java source code, classes and .jar files							
Reverse engineer C++ source code							
Reverse engineer .NET .dll and .exe files							
Reverse engineer CORBA IDL source code							
Reverse engineer Ada 9x source code							
Reverse engineer XML							
Reverse engineer XML Schema							
Reverse engineer databases with JDBC							
Reverse engineer Hibernate mapping files							
Reverse engineer PHP 5.0 source code							
Reverse engineer Python							
On-demand Java reverse engineering							
Template parameter support							

***Table 1-24** Reverse Engineering support in different editions of VP-UML*

Code Generation

Generate various kind of source code from class models.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Instantly generate source code from class diagrams							
Generate Java source code							
Generate C# source code							
Generate VB.NET source code							
Generate PHP 5.0 source code							
Generate Object Definition Language source code							
Generate Flash ActionScript 3.0 source code							
Generate IDL source code							
Generate C++ source code							
Generate Delphi source code							
Generate Perl source code							
Generate XML Schema source code							
Generate Python source code							
Generate Objective-C source code							
Generate Ada source code							
Generate Ruby source code							
Template parameter support							
Template support for generating source code with generic constructs							
Customizable source code generation							

Table 1-25 *Code generation support in different editions of VP-UML*

State Machine Diagram Code Generation

Generate code from a state machine diagram for state machine.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Generate Java source code from state diagram							
Generate C++ source code from state diagram							
Generate C# source code from state diagram							
Generate VB.NET source code from state diagram							

Table 1-26 *State Machine Diagram code generation support in different editions of VP-UML*

Java Round-Trip Engineering

Update between UML class models and Java source code with the round-trip engineering functionalities.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Reverse engineer Java source code to class diagram							
Update Java source code based on class diagram							

Table 1-27 *Java Round-Trip support in different editions of VP-UML*

Shape Editor

Create domain specific shapes with Shape Editor to use in diagrams.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Design arbitrary shapes							
Import SVG shapes							
Incorporate different shapes into UML diagram							
Organize shapes by gallery, category and stencil							
Advanced shape design capabilities							

Table 1-28 *Shape Editor support in different editions of VP-UML*

Layout Facilities

Tidy up diagram content by performing layout with several mouse actions.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Automatic diagram layouts							
Shape alignment and centering							
Uniform shape width and height maintenance							
Automatic shape distribution							

Table 1-29 *Layout support in different editions of VP-UML*

Interoperability

Exchange UML diagrams and models with other tools with industrial standard representation.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Import Telelogic Modeler project files							
Import and export EMF based UML2 model							
Command-line operations							
Import and export XMI 1.0, 1.2 and 2.1							
Import and export XML							
Import and export VP project file format							
Import and export use case model to MS Word							
Import Rational Rose project files							
Import ERwin data modeler project files [2]							
Generate BPEL code for Oracle workflow engine							
Generate BPEL code for JBoss workflow engine							
Export diagrams as JPG, PNG, SVG and EMF image files							
Export diagrams as PDFs							
Slice exported diagrams into smaller segments							
Import Rational Software Architect files							
Import Rational DNX							
Import NetBeans 6.x UML diagrams							
Copy diagram elements as XML							
Import Visio drawings into Visual Paradigm							
Import and export Microsoft Excel file for use case diagram							
Import and export Microsoft Excel file for class diagram							
Import and export Microsoft Excel file for sequence diagram							
Import and export Microsoft Excel file for communication diagram							
Import and export Microsoft Excel file for state machine diagram							
Import and export Microsoft Excel file for activity diagram							
Import and export Microsoft Excel file for component diagram							
Import and export Microsoft Excel file for deployment diagram							
Import and export Microsoft Excel file for package diagram							
Import and export Microsoft Excel file for object diagram							
Import and export Microsoft Excel file for composite structure diagram							
Import and export Microsoft Excel file for interaction overview diagram							
Import and export Microsoft Excel file for requirement diagram							
Import and export Microsoft Excel file for basic diagram							
Import and export Microsoft Excel file for crc card diagram							
Import and export Microsoft Excel file for entity relationship diagram							
Import and export Microsoft Excel file for orm diagram							
Import and export Microsoft Excel file for business process diagram							
Import and export Microsoft Excel file for data flow diagram							
Import and export Microsoft Excel file for epc diagram							
Import and export Microsoft Excel file for process map diagram							
Import and export Microsoft Excel file for organization chart							

Intuitive User Interface

User friendly user interface to let you achieve tasks effortlessly.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Group diagrams by category							
Advanced property pane							
Dockable user interface windows							
New project generation using predefined templates							
Easy-to-use "New Diagram" dialog							
Flexible zooming							
Palette-style toolbar							
Collapsible toolbar							
Display tool names for toolbar buttons							
Expand grouped toolbar buttons							
Display various diagram categories in toolbar							
Display stereotyped model types in toolbar							
Numerous looks-and-feels							
Import user preferences from existing workspaces							
Multilingual support							
Searchable options							
Customizable program shortcuts							

The feature is available but unable to make changes.

Table 1-31 *Support of 'intuitive user interface' in different editions of VP-UML*

Automatic Updates

Keep VP-UML up-to-date with the automatic updates features.

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Automatic online updating							
Maintain local update server with update synchronizer							

Table 1-32 *Support of 'automatic updates' in different editions of VP-UML*

Open Architecture

Extend VP-UML's functionalities using plug-in

	Enterprise	Professional	Standard	Modeler	Personal	Community	Viewer
Plug-in support (Java)							
VP model and XML interaction							

Table 1-33 *Support of 'open architect' in different editions of VP-UML*

Licensing

Vp-UML need to run with a valid license key. A license key of a higher edition can be used on a lower edition. For example, you can run Standard Edition of VP-UML with Professional Edition of VP-UML key.

VP-UML also provides a 30-days evaluation key for trial. After the evaluation key expires, you can purchase the full license from our website or resellers, or un-install the program.

The license key for Community Edition will not expire. However, Community Edition cannot be used for commercial purposes.

Importing Single-Seat License Key

A Visual Paradigm's Single Seat (developer based) license allows a licensee to install the software on more than one machine, such as desktop and notebook, which belong to the licensee only. As the license is developer based, the software must be used by the licensee only, without running more for than one instance concurrently. When you run VP-UML the first time, the **License Key Manager** pops up to let you import the license key.

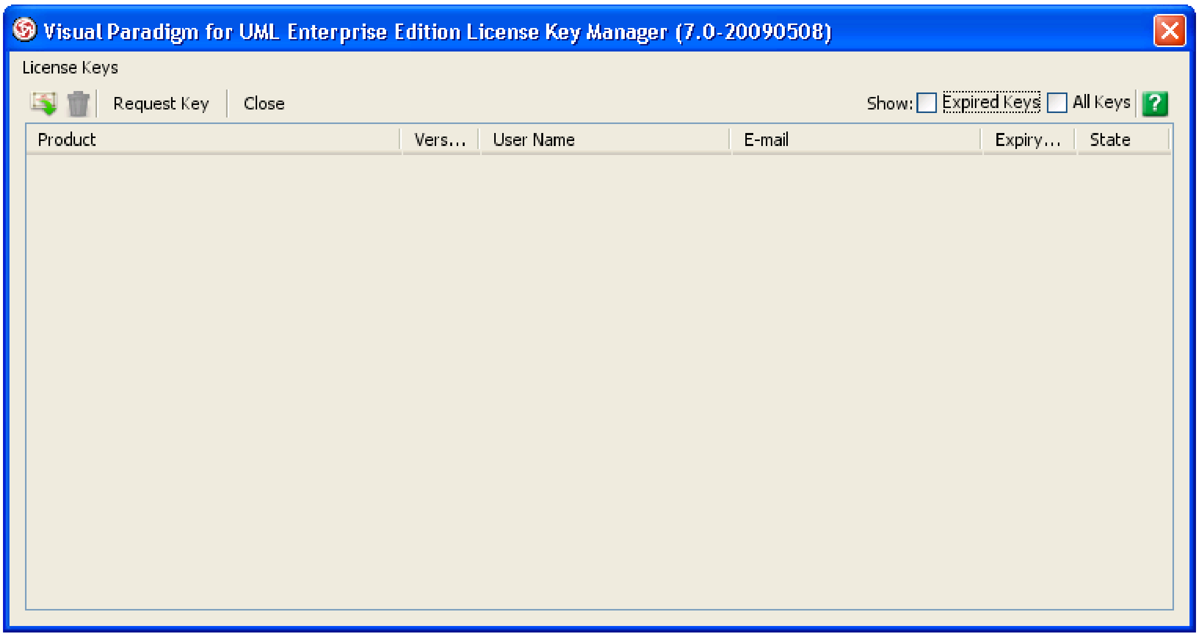
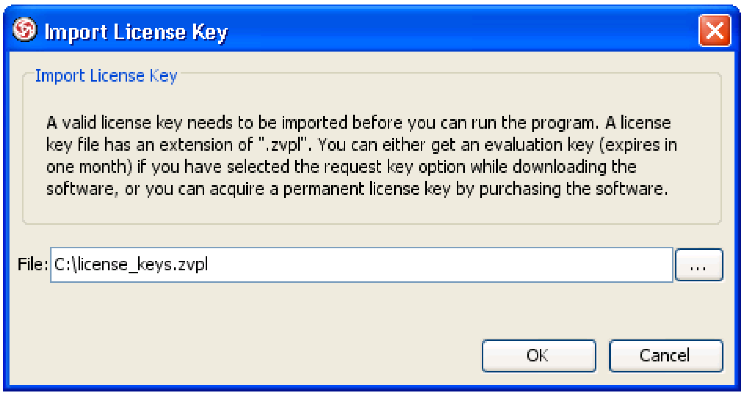

Visual Paradigm for UML Enterprise Edition License Key Manager (7.0-20090508)
License Keys
Request Key
Close
Show: □ Expired Keys □ All Keys
Product
Vers...
User Name
E-mail
Expiry...
State

Figure 1-1 *License Key Manager*

NOTE: To start License Key Manager in VP-UML, select **Tools > Key Manager...** from the main menu.

To import a single-seat key, select **License Keys > Import...** from the menu. In the **Import License Key** dialog box, specify the filepath of the key and press **OK** to confirm.

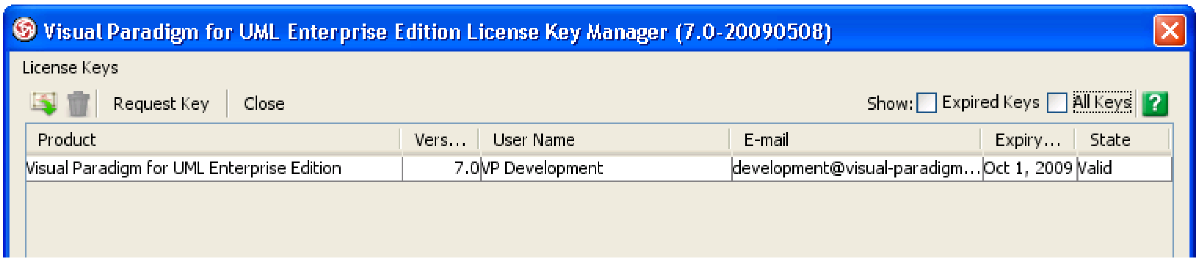


A valid license key needs to be imported before you can run the program. A license key file has an extension of ".zvpl". You can either get an evaluation key (expires in one month) if you have selected the request key option while downloading the software, or you can acquire a permanent license key by purchasing the software.

File: C:\license_keys.zvpl

Figure 1-2 *Entering the filepath of license key*

If the imported key is valid, information of key will appear in the **License Key Manager**.



Visual Paradigm for UML Enterprise Edition License Key Manager (7.0-20090508)

Product	Vers...	User Name	E-mail	Expiry...	State
Visual Paradigm for UML Enterprise Edition	7.0	VP Development	development@visual-paradigm...	Oct 1, 2009	Valid

Figure 1-3 *License key is imported*

Working with Floating License Server

Before you use VP-UML with a floating license key, your machine need to access to the license server via LAN to acquire a license key first. For more details, you can refer to the Floating License Server Installation Guide about floating license server installation for more details.

Configuring Floating License Server

Floating license need to be acquired from server, through the **License Key Manager**. When you start VP-UML the first time, the **License Key Manager** will show up automatically. From the **License Key Manager**, select **License Keys > Floating License > Configure...** in the menu.

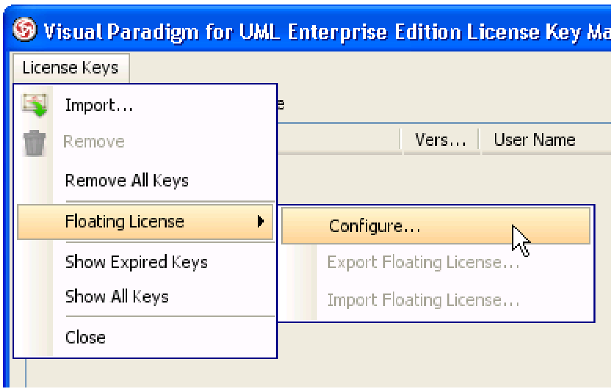
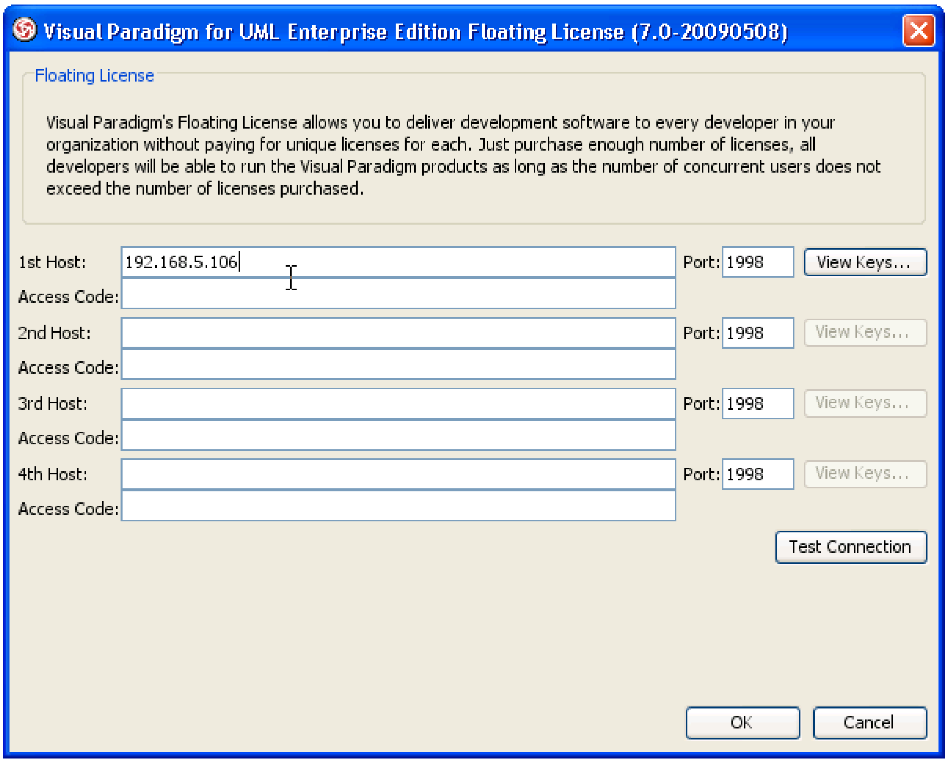


Figure 1-4 *To confirm floating license*

NOTE: To start License Key Manager in VP-UML, select **Tools > Key Manager...** from the main menu.

In the configuration page, specify the host IP and adjust the port (if necessary). You may need to specify the access code if it has been defined in floating license server. You should contact your administrator about this matter.



Visual Paradigm for UML Enterprise Edition Floating License (7.0-20090508)

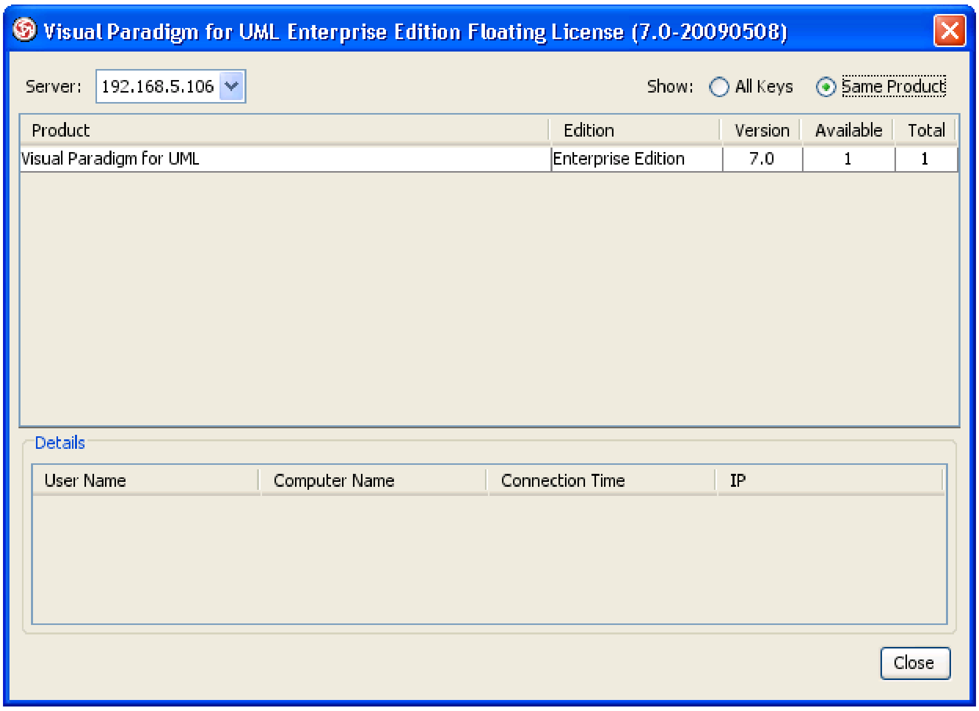
Floating License

Visual Paradigm's Floating License allows you to deliver development software to every developer in your organization without paying for unique licenses for each. Just purchase enough number of licenses, all developers will be able to run the Visual Paradigm products as long as the number of concurrent users does not exceed the number of licenses purchased.

1st Host: 192.168.5.106 Port: 1998
Access Code:
2nd Host: Port: 1998
Access Code:
3rd Host: Port: 1998
Access Code:
4th Host: Port: 1998
Access Code:

Figure 1-5 *Specifying host of license server*

In order to verify the connection, press **Test Connection**. Alternatively, you may click on the **View Keys** button to see how many keys are available in server.



Visual Paradigm for UML Enterprise Edition Floating License (7.0-20090508)

Server: 192.168.5.106 Show: ○ All Keys ● Same Product

Product	Edition	Version	Available	Total
Visual Paradigm for UML	Enterprise Edition	7.0	1	1

Details

User Name	Computer Name	Connection Time	IP

Figure 1-6 *Viewing license in license server*

Once the configuration is done correctly, license key will be acquired from license server. By closing the **License Key Manager**, VP-UML will start.

Exporting Floating License Key from Server

In order to acquire a floating license key, you must have access to the floating license server machine through network. If you need to use VP-UML offline, which means, at a location that cannot reach the server machine (e.g. when having a meeting outside the office), you can export a floating license keys to your laptop for running VP-UML with a local license key. Once a key has been exported from server, the number of keys in the server will decrease by one. The volume will reset once you have imported the licence key back to the server.

To export floating license key, start the **License Key Manager** by selecting **Tools > Key Manager...** in the main menu of VP-UML, and select **License Keys > Floating License > Export Floating License...** in the menu of **License Key Manager**.

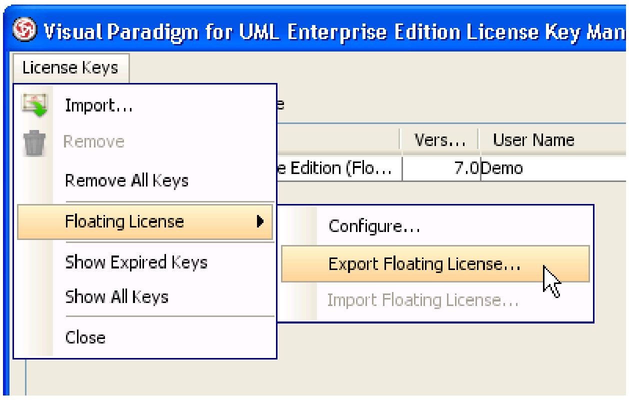
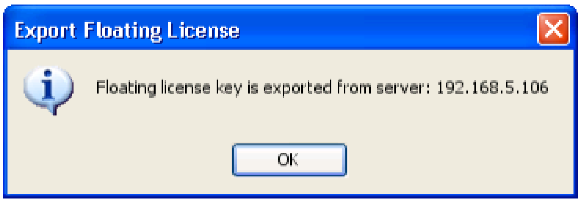


Figure 1-7 *To export floating license key*

A message appears informing you about the export of license key.


Export Floating License

Floating license key is exported from server: 192.168.5.106

Figure 1-8 *Floating license key exported*

From the moment the key is exported from server, you do not need to connect to the license server anymore until the moment you import the key back to the server.

Importing Floating License Key to Server

To import an exported floating license key back to license server, start the License Key Manager by selecting **Tools > Key Manager...** in the main menu of VP-UML, and select **License Keys > Floating License > Import Floating License...** in the menu of **License Key Manager**.

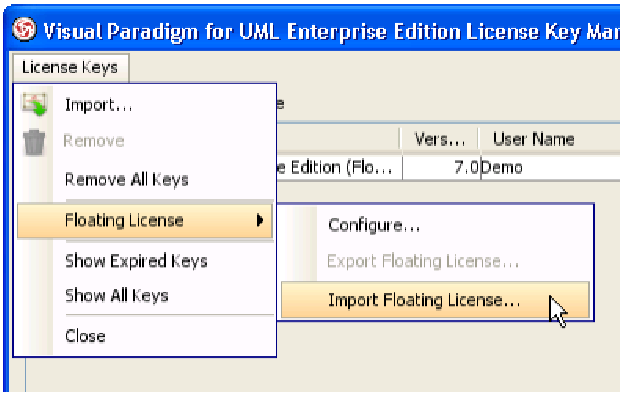
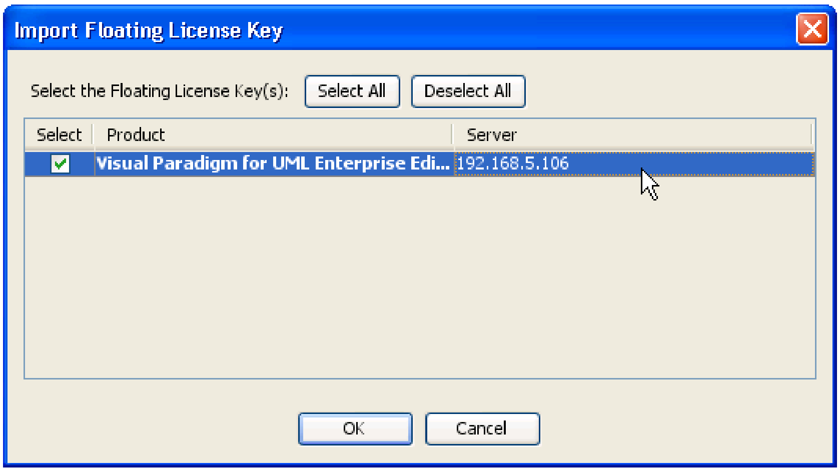


Figure 1-9 *To import floating license key*

In the **Import Floating License Key** dialog box, select the key you need to import and click **OK** to confirm.

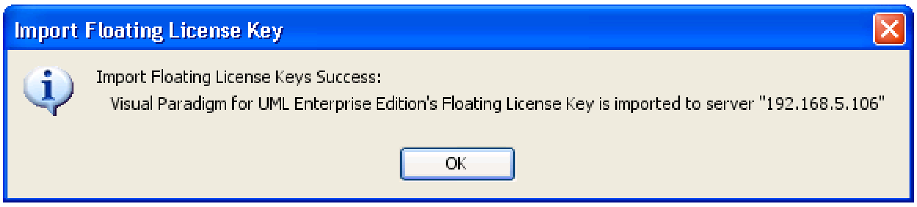


Select the Floating License Key(s):

Select	Product	Server
☑	Visual Paradigm for UML Enterprise Edi...	192.168.5.106

Figure 1-10 *Select the license key to import*

This finished importing the key back to server.



Import Floating License Keys Success:
Visual Paradigm for UML Enterprise Edition's Floating License Key is imported to server "192.168.5.106"

Figure 1-11 *Floating license key is imported*

Switching from Evaluation License Key to Purchased License Key

An evaluation key is required for evaluating VP-UML. But once you have purchased, you will receive a purchased license key from Visual Paradigm. In order to make the purchased key take effect, you need to remove the evaluation keys, then import the purchased license key. To switching from an evaluation license key to a purchased license key:

1. Start VP-UML.
2. Start the **License Key Manager** by selecting **Tools > Key Manager...** from the main menu.

> **NOTE:** If your evaluation has expired, you can skip this step as the **License Key Manager** was shown when starting VP-UML (step 1) due to no valid license key was found.

3. Remove all license keys by selecting **License Keys > Remove All Keys** from the menu of **License Key Manager**.

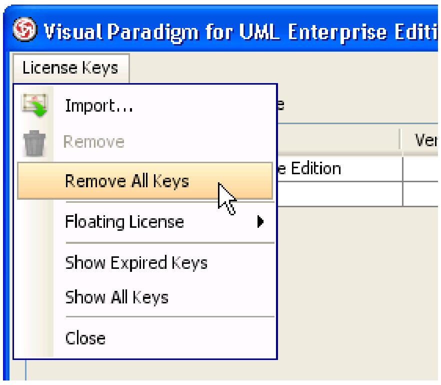
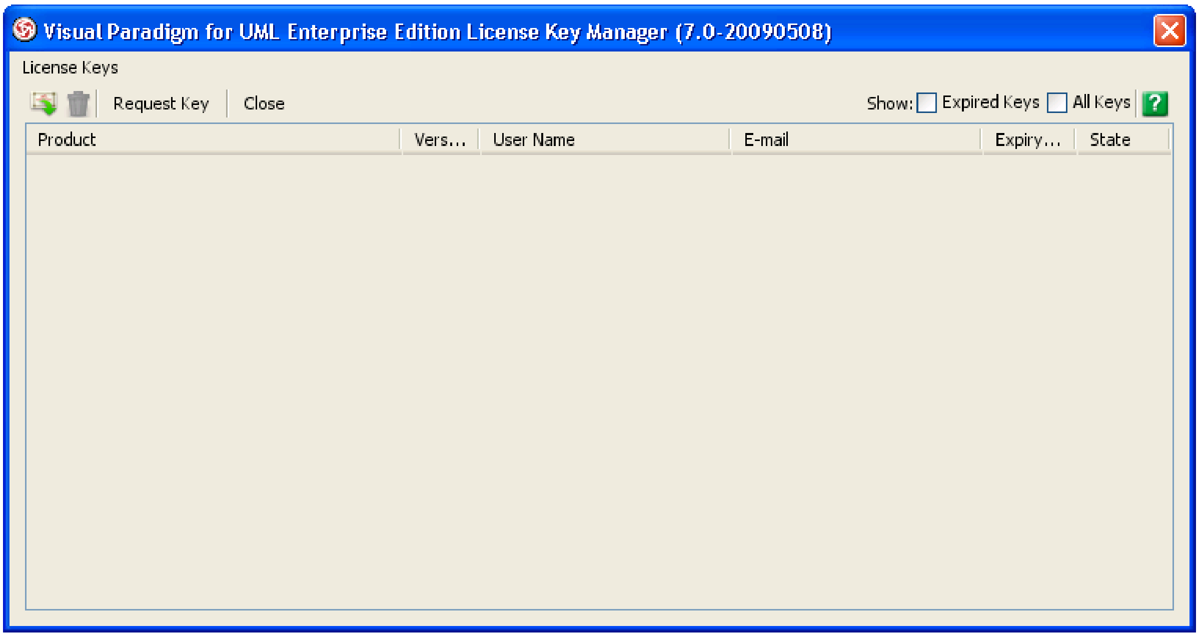


Figure 1-12 *To remove all keys in License Key Manager*

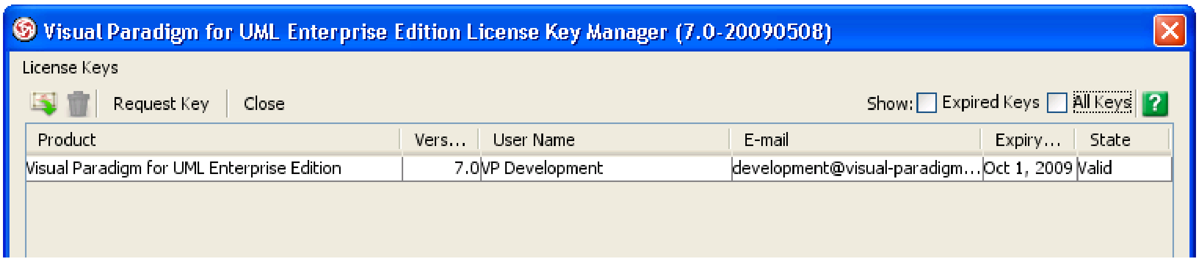
This cause all the keys to be removed from the **License Key Manager**. Therefore, you should not see any entry in the manager.


Visual Paradigm for UML Enterprise Edition License Key Manager (7.0-20090508)
License Keys
Request Key
Close
Show: □ Expired Keys □ All Keys
?

Product	Vers...	User Name	E-mail	Expiry...	State

Figure 1-13 *Keys are removed in License Key Manager*

4. You can import the purchased license key now. For single-seat key user, select **License Keys > Import...**, then specify the filepath of the purchased license key. For floating license user, configure the license server connection by selecting **License Keys > Floating License > Configure...** from the menu. For details, refer to the previous section.


Visual Paradigm for UML Enterprise Edition License Key Manager (7.0-20090508)
License Keys
Request Key
Close
Show: □ Expired Keys □ All Keys
?

Product	Vers...	User Name	E-mail	Expiry...	State
Visual Paradigm for UML Enterprise Edition	7.0	VP Development	development@visual-paradigm...	Oct 1, 2009	Valid

Figure 1-14 *Purchased license key is imported*

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.

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.

In order to work with the floating license, installation of a floating license 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 .

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/ .

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

From time to time, Visual Paradigm releases new versions of its software. Normally there are two or three versions released every year, with each new version having around five to ten distinct new features as well as a number of improvements. The easiest way to get the latest version from Visual Paradigm is to purchase the software maintenance package. Both minor upgrade (e.g., 6.1 to 6.1 SP1) and major upgrade (e.g., 7.0 to 7.1 or 8.0) of Visual Paradigm products are included in the software maintenance package, thus entitling you to get all of the version upgrades issued within the 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.

LIVE Help

Our Technical Support Team uses a real-time, web-based chat program to discuss the problem with you directly.

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 in year-long terms (e.g., June 20, 2007 to June 19, 2008).

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	$279.5	$419.5
Professional	$139.5	$209.5
Standard	$59.5	$89.5
Modeler	$19.5	$29.5
Personal	$11.5	$17.5
Community	not applicable	not applicable
Viewer	not applicable	not applicable

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

Table 1-34 *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	$363.5	$545.5
Professional	$181.5	$272.5
Standard	$77.5	$116.5
Modeler	$25.5	$38.5
Personal	not applicable	not applicable
Community	not applicable	not applicable
Viewer	not applicable	not applicable

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

Table 1-35 *Price for floating license*

System Requirements

Hardware Requirements

- Intel Pentium III Compatible Processor at 1.0 GHz or higher.
- Minimum 512MB RAM, but 1.0 GB is recommended.
- Minimum 400MB disk space.
- Microsoft Windows (98, 2000, XP, or Vista), Linux, Mac OS X, Solaris or all other Java-enabled platforms.

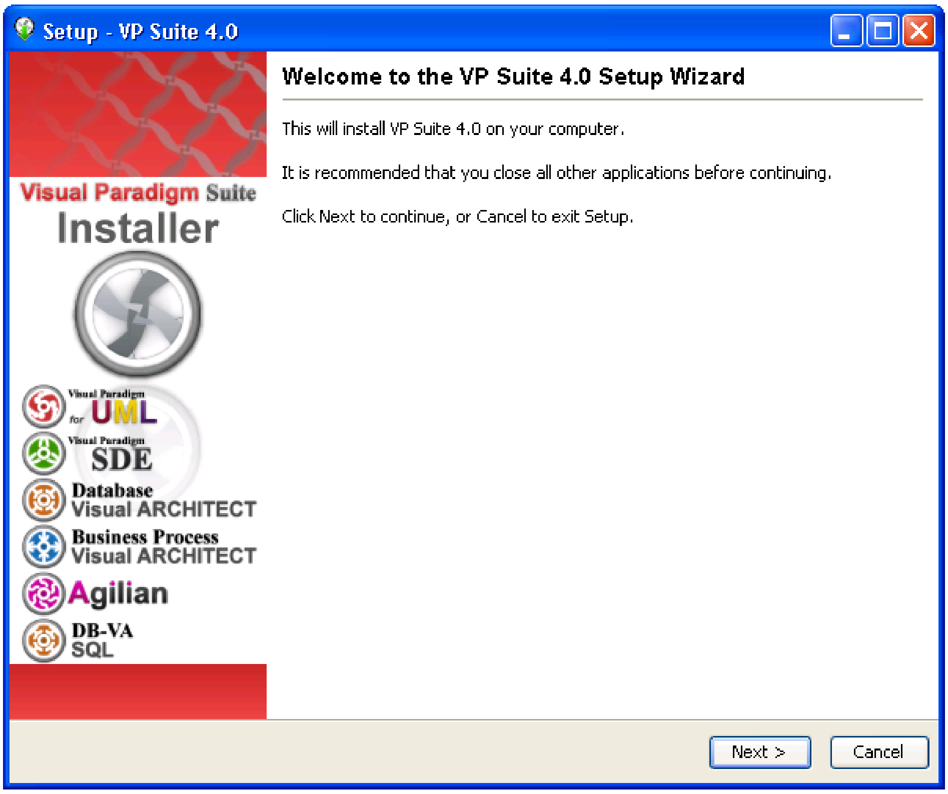
IDE Requirements (for IDE integration)

- Eclipse 3 or above
- IntelliJ IDEA 4 or above (7.0 ready)
- JBuilder 9 or above
- JDeveloper 10g (10.1.2)
- NetBeans 4.0 or above (5.0 ready)
- Sun Studio Enterprise
- WebLogic Workshop 8.1

Installing Visual Paradigm for UML on Windows 2000/NT/2003/XP/Vista

Using Installer (.exe)

1. Execute the downloaded VP Suite installer file. The setup wizard appear as below.



Welcome to the VP Suite 4.0 Setup Wizard

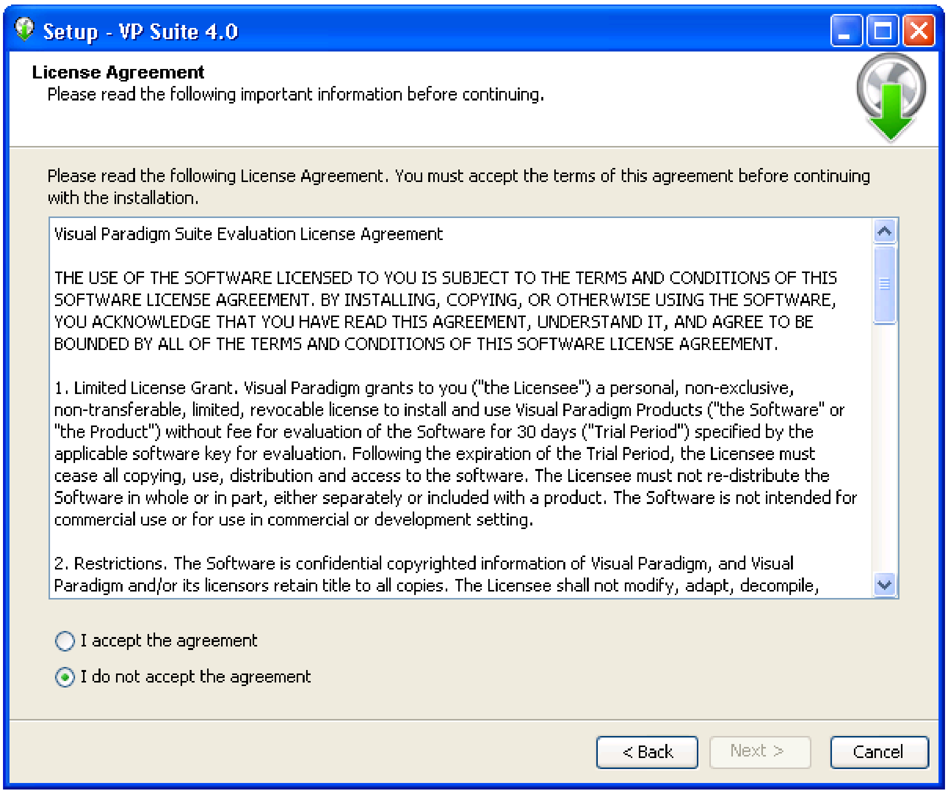
This will install VP Suite 4.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 2-1 VP Suite 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.



License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Visual Paradigm Suite Evaluation License Agreement

THE USE OF THE SOFTWARE LICENSED TO YOU IS SUBJECT TO THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT. BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE TO BE BOUNDED BY ALL OF THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT.

1. Limited License Grant. Visual Paradigm grants to you ("the Licensee") a personal, non-exclusive, non-transferable, limited, revocable license to install and use Visual Paradigm Products ("the Software" or "the Product") without fee for evaluation of the Software for 30 days ("Trial Period") specified by the applicable software key for evaluation. Following the expiration of the Trial Period, the Licensee must cease all copying, use, distribution and access to the software. The Licensee must not re-distribute the Software in whole or in part, either separately or included with a product. The Software is not intended for commercial use or for use in commercial or development setting.

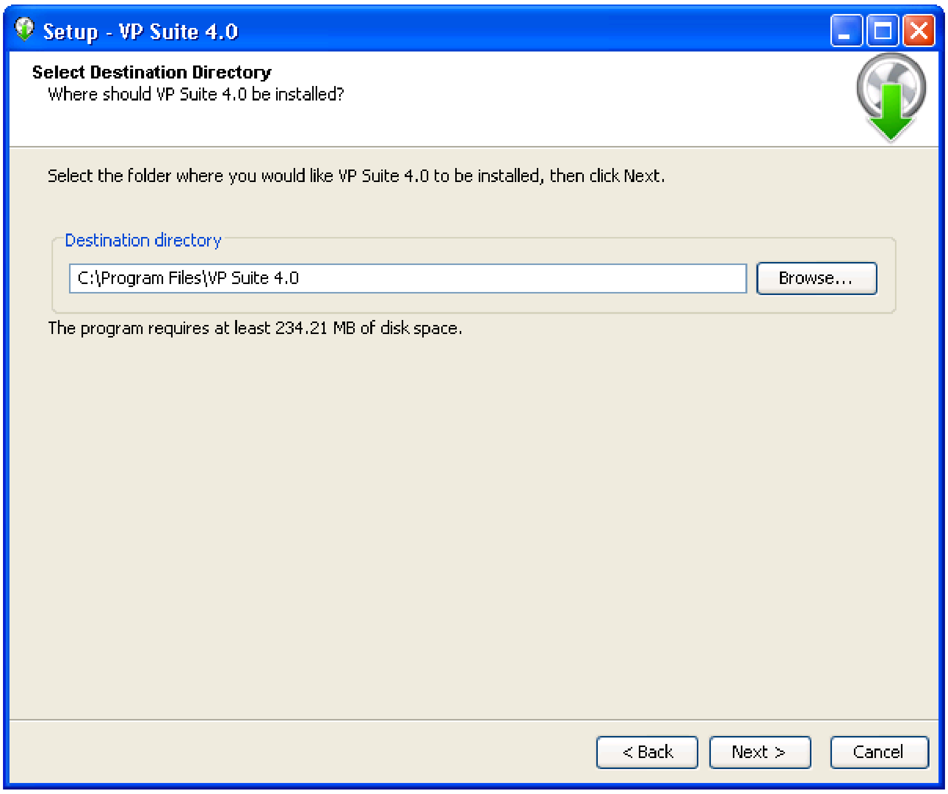
2. Restrictions. The Software is confidential copyrighted information of Visual Paradigm, and Visual Paradigm and/or its licensors retain title to all copies. The Licensee shall not modify, adapt, decompile,

○ I accept the agreement

● I do not accept the agreement

Figure 2-2 The License Agreement

4. Specify the directory for installing VP Suite. Click **Next >** to proceed to the next page.


Setup - VP Suite 4.0

Select Destination Directory

Where should VP Suite 4.0 be installed?

Select the folder where you would like VP Suite 4.0 to be installed, then click Next.

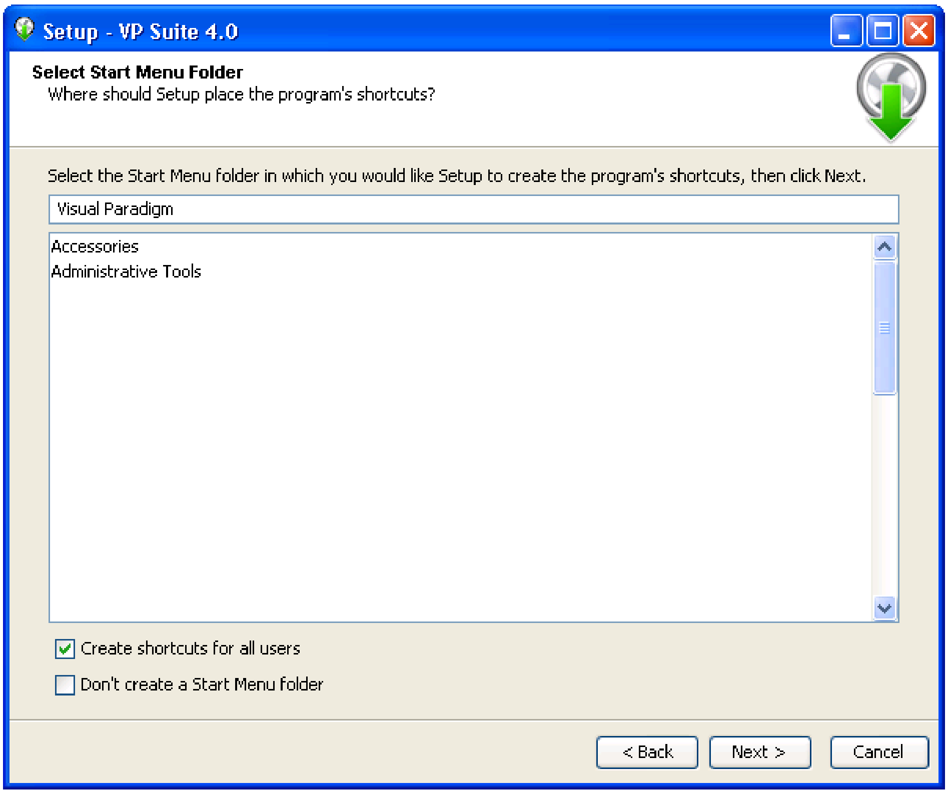
Destination directory

C:\Program Files\VP Suite 4.0

The program requires at least 234.21 MB of disk space.

Figure 2-3 *The Select Destination Directory 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.


Setup - VP Suite 4.0

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Select the Start Menu folder in which you would like Setup to create the program's shortcuts, then click Next.

Visual Paradigm

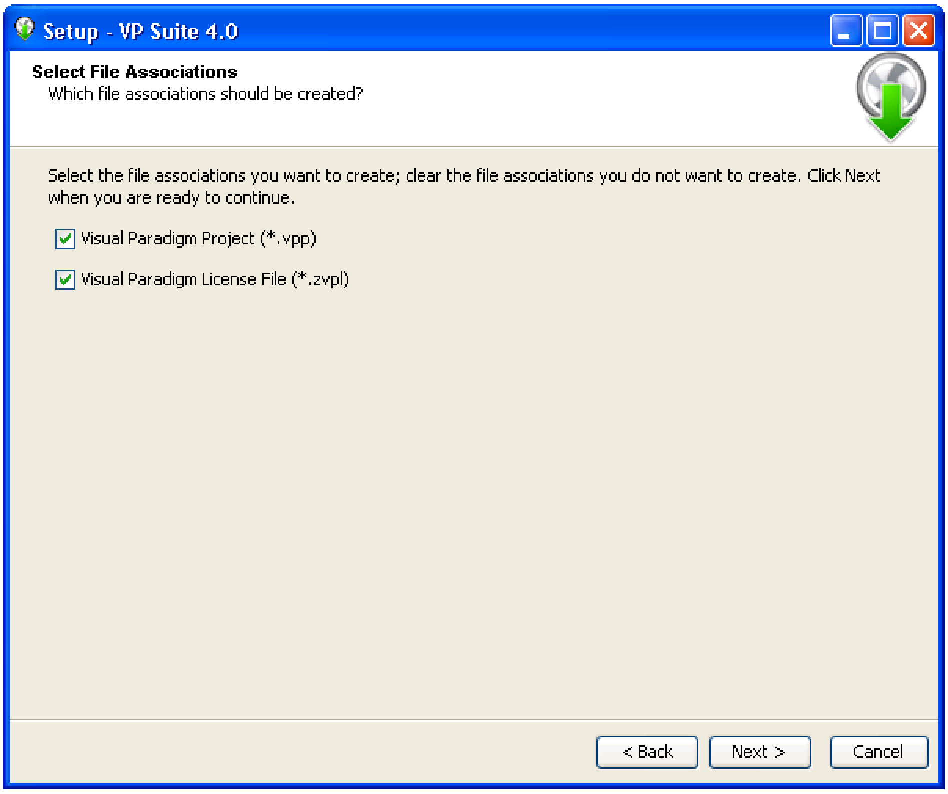
Accessories
Administrative Tools

☑ Create shortcuts for all users

□ Don't create a Start Menu folder

Figure 2-4 *The Select Start Menu Folder page*

6. In the File Association page, keep **Visual Paradigm Project (*.vpp)** and **Visual Paradigm License File (*.zvpl)** checked if you want your system able to open the project file and the license key file. Click **Next >** to proceed to the product selection page.



Setup - VP Suite 4.0

Select File Associations

Which file associations should be created?

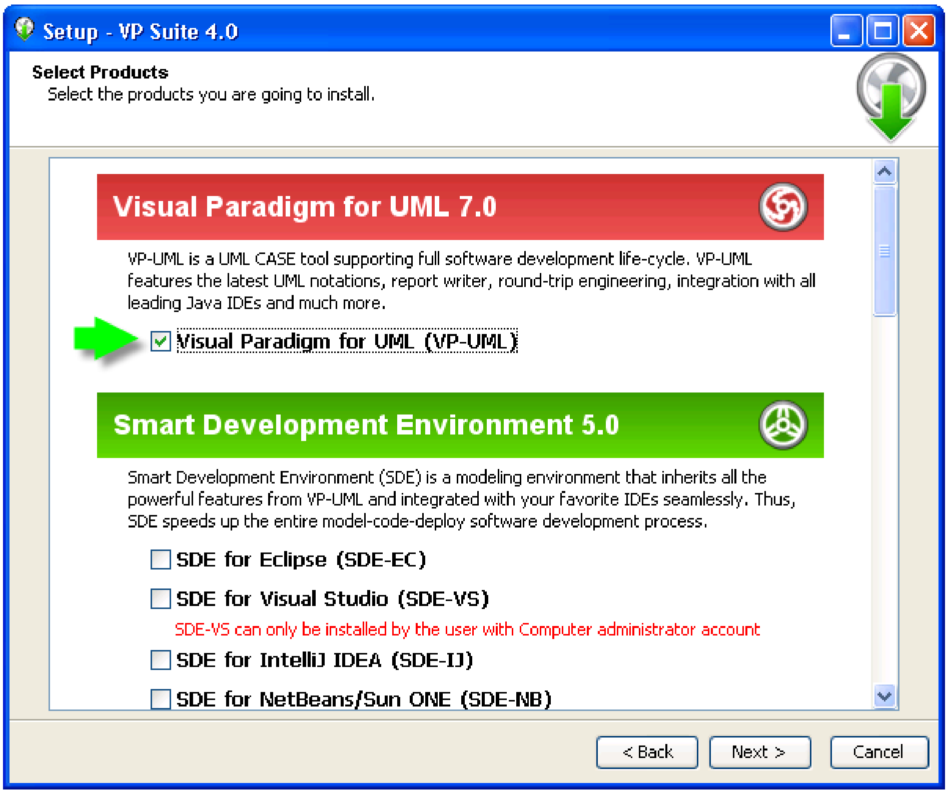
Select the file associations you want to create; clear the file associations you do not want to create. Click Next when you are ready to continue.

☑ Visual Paradigm Project (*.vpp)

☑ Visual Paradigm License File (*.zvpl)

Figure 2-5 *The Select File Associations page*

7. Select **Visual Paradigm for UML**. Click **Next >** to proceed to the product configuration page.



Setup - VP Suite 4.0

Select Products

Select the products you are going to install.

Visual Paradigm for UML 7.0

VP-UML is a UML CASE tool supporting full software development life-cycle. VP-UML features the latest UML notations, report writer, round-trip engineering, integration with all leading Java IDEs and much more.

☑ Visual Paradigm for UML (VP-UML)

Smart Development Environment 5.0

Smart Development Environment (SDE) is a modeling environment that inherits all the powerful features from VP-UML and integrated with your favorite IDEs seamlessly. Thus, SDE speeds up the entire model-code-deploy software development process.

□ SDE for Eclipse (SDE-EC)

□ SDE for Visual Studio (SDE-VS)

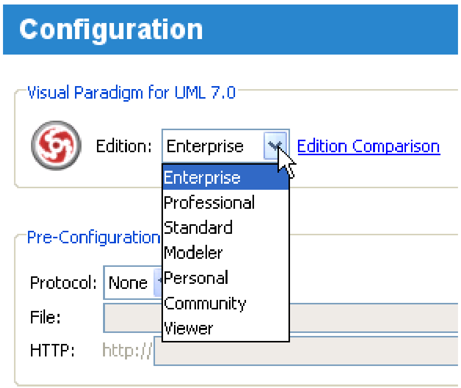
SDE-VS can only be installed by the user with Computer administrator account

□ SDE for IntelliJ IDEA (SDE-IJ)

□ SDE for NetBeans/Sun ONE (SDE-NB)

Figure 2-6 *The product selection page*

8. VP-UML features vary by product edition. For more details on the features supported by different editions, click the hyperlink **Edition Comparison** which brings you to the web page of VP-UML feature comparison for different editions.



Configuration

Visual Paradigm for UML 7.0

Edition: Enterprise

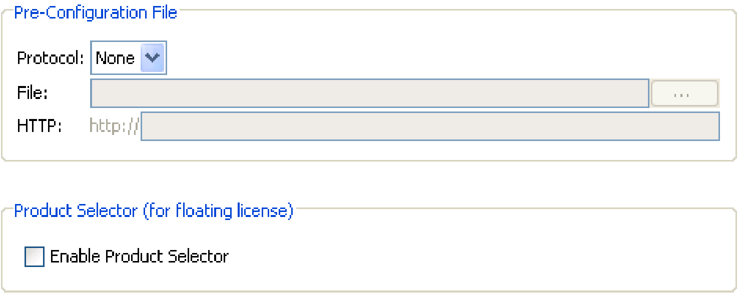
Protocol: None

__Figure 2-7__ Select product edition

9. For users who are going to run VP-UML with floating license, the Pre-Configuration File section enables you to configure the connection(s) to floating license server by providing a configuration file. The pre configuration is optional. If you do not define the connection here, you will need to do so when starting VP-UML the first time. Below is the pre-configuration file file content:

```
?xml version="1.0" encoding="UTF-8"?>
<LicenseServer>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
</LicenseServer>
```

Selecting **Enable Product Selector** for Product Selection will result in creating a shortcut under the Start Menu for starting the Product Selector, a utility that lets you realize the installed products with available keys in the floating license server.



Pre-Configuration File

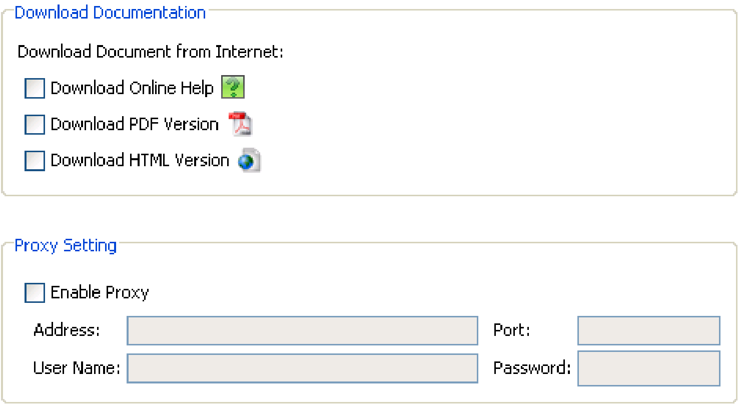
Protocol: None

Product Selector (for floating license)

☐ Enable Product Selector

__Figure 2-8__ Pre-configure floating license server connection and enable product selector for floating license user

10. Select **Download Online Help** if you want to be able to access the Help contents from within the tool. Select **Download PDF/HTML Version** if you wish to read the documentation in this two types of formats. Press **Next >** to proceed to the license import page.



Download Documentation

Download Document from Internet:

☐ Download Online Help

☐ Download PDF Version

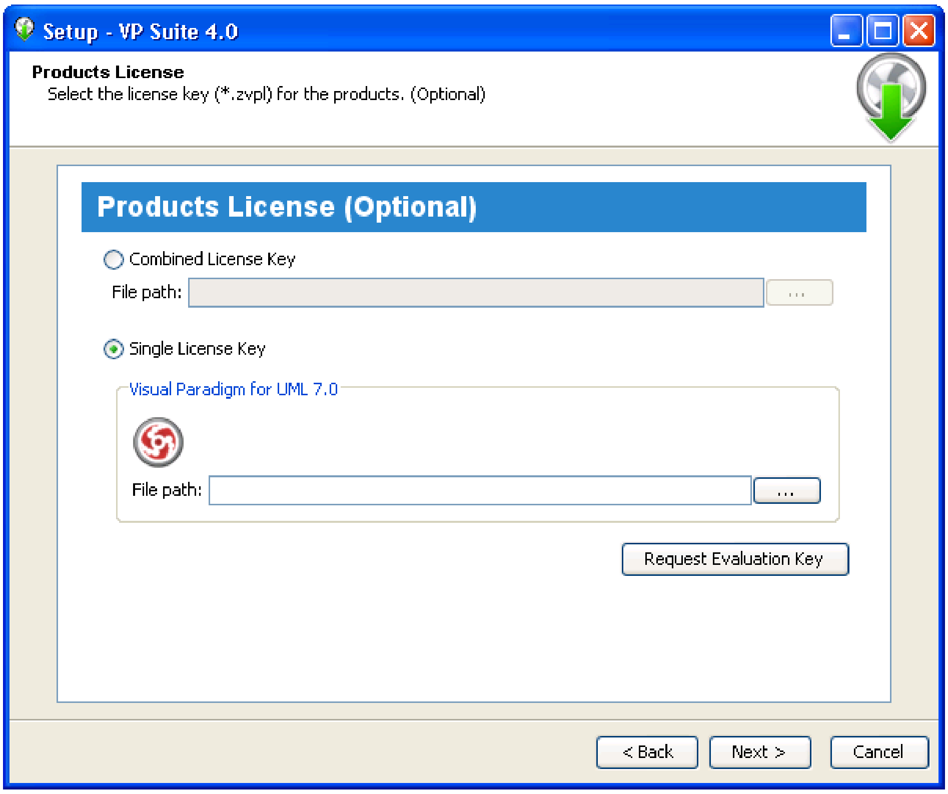
☐ Download HTML Version

Proxy Setting

☐ Enable Proxy

__Figure 2-9__ Download documentation

11. A combined license key is a key file which allows unlocking multiple products. To import a combined license key, select Combined License Key and specify the file path of the key.
A single license key is a key file which allows unlocking only one product. To import a single license key, select Single License Key and specify the file path of the key.
The import of license key is optional at this point, because you can import it when you run VP-UML. Click **Next >** to proceed with copying files.


Setup - VP Suite 4.0

Products License
Select the license key (*.zvpl) for the products. (Optional)

Products License (Optional)

○ Combined License Key
File path:

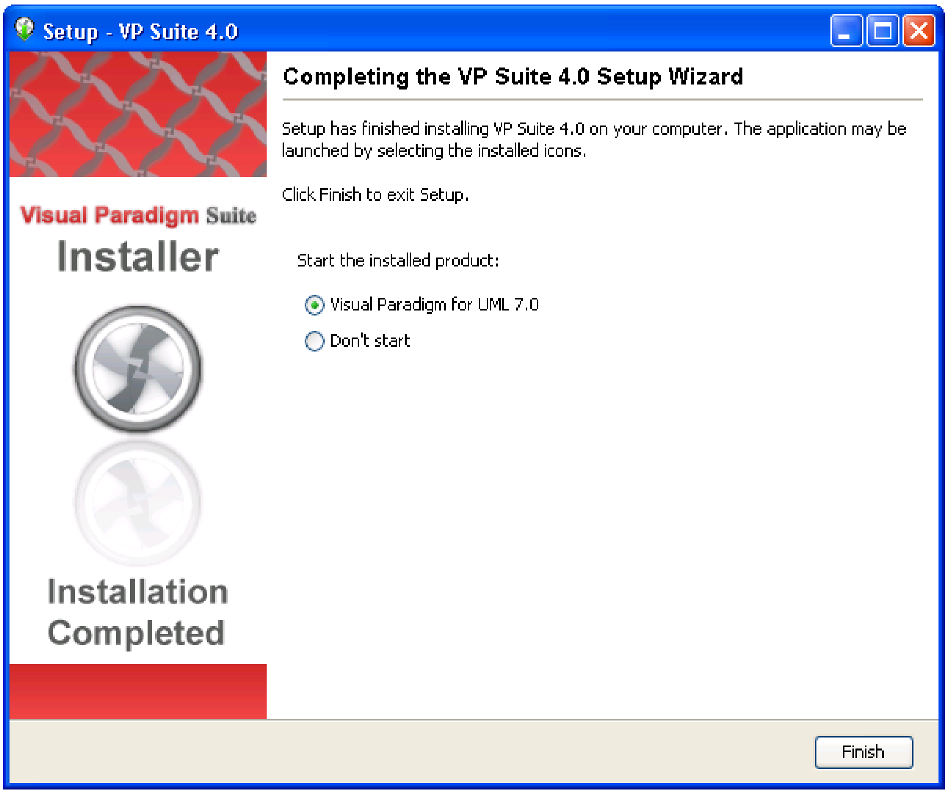
● Single License Key

Visual Paradigm for UML 7.0
File path:

Figure 2-10 *The license key selection page*

NOTE: For evaluation, the key should be sent to your email box (unless you have chosen not to receive it before download). If you have not received the email with key yet, and if you have selected to receive the key as attachment, the email might be treated as spam by mistake. Click on **Request Evaluation Key** to ask for another one. This time, try not to select sending the key as attachment.

12. Upon finishing, you can select whether to start VP-UML or not. Keep **Visual Paradigm for UML** selected and click **Finish** will run VP-UML right away.



Setup - VP Suite 4.0

Completing the VP Suite 4.0 Setup Wizard

Setup has finished installing VP Suite 4.0 on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

Start the installed product:

● Visual Paradigm for UML 7.0

○ Don't start

***Figure 2-11** Installation completed screen*

Using No Install Version (.zip)

1. Decompress the downloaded zip file into a directory. This should create a subdirectory named &ldquo;VP Suite 4.0&rdquo; where 4.0 is the version number.

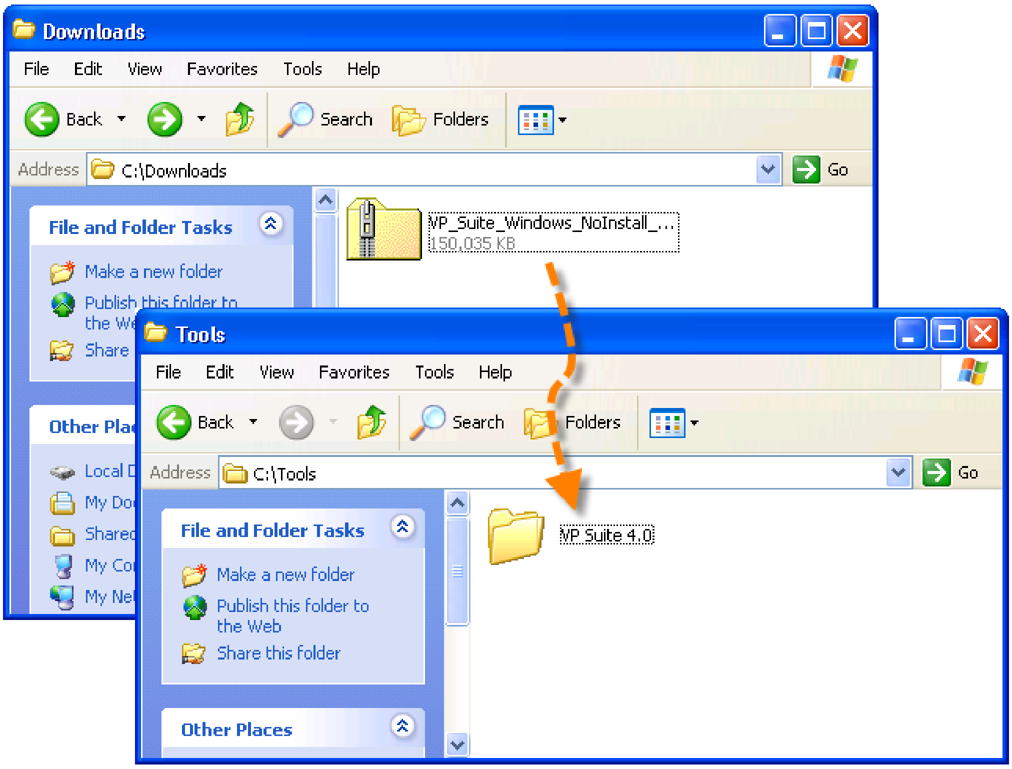
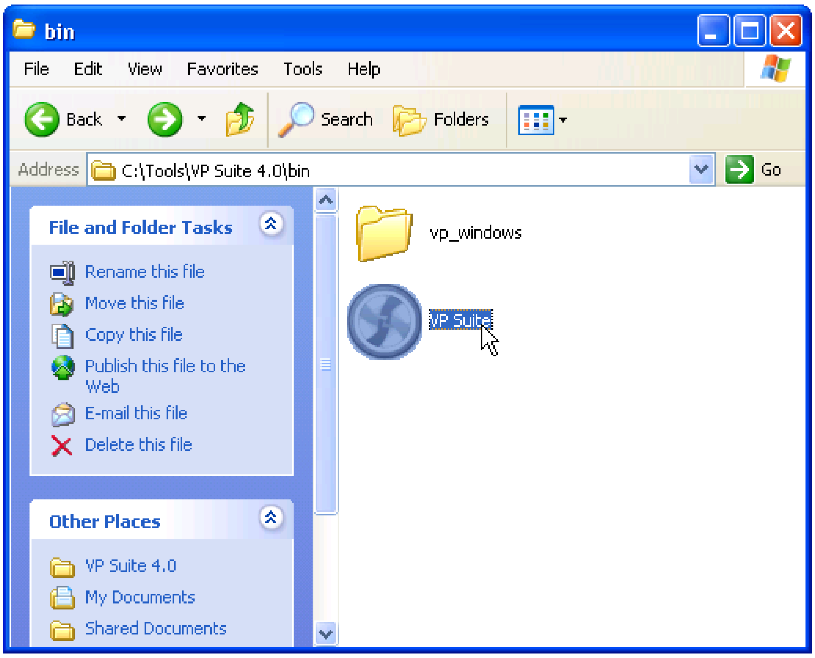


***Figure 2-12** Extracting No-Install zip file*

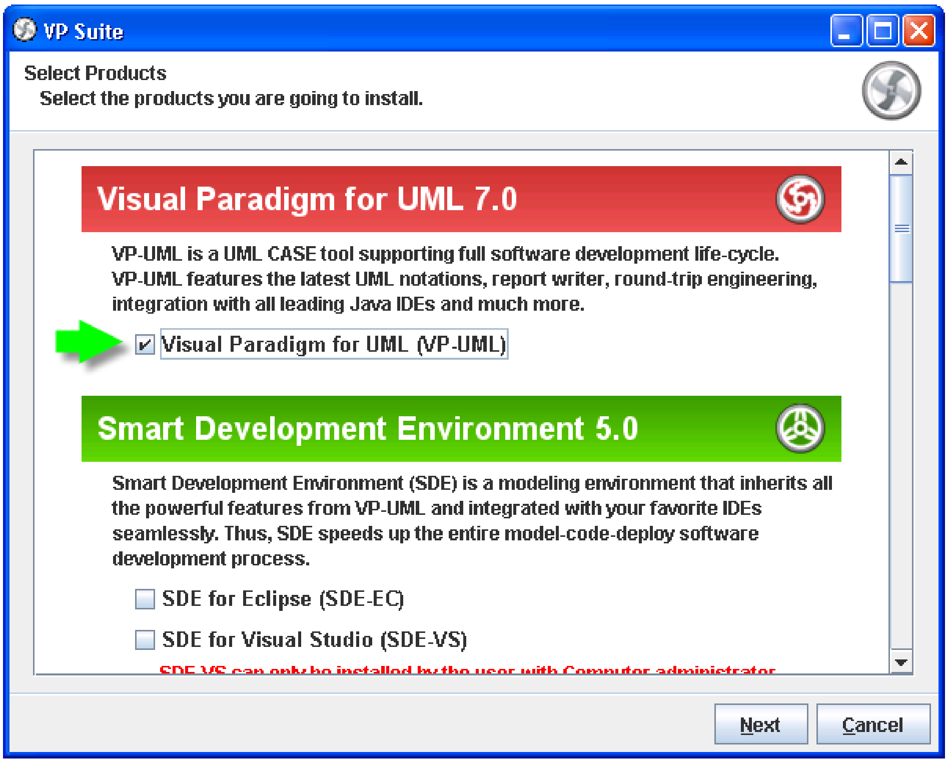
2. Change directory to &ldquo;VP Suite 4.0\bin&rdquo; and execute VP Suite in it.



Address C:\Tools\VP Suite 4.0\bin

__Figure 2-13__ Launching VP Suite

3. Select **Visual Paradigm for UML**. Click **Next >** to proceed to the product configuration page.



Select Products

Select the products you are going to install.

Visual Paradigm for UML 7.0

VP-UML is a UML CASE tool supporting full software development life-cycle. VP-UML features the latest UML notations, report writer, round-trip engineering, integration with all leading Java IDEs and much more.

☑ Visual Paradigm for UML (VP-UML)

Smart Development Environment 5.0

Smart Development Environment (SDE) is a modeling environment that inherits all the powerful features from VP-UML and integrated with your favorite IDEs seamlessly. Thus, SDE speeds up the entire model-code-deploy software development process.

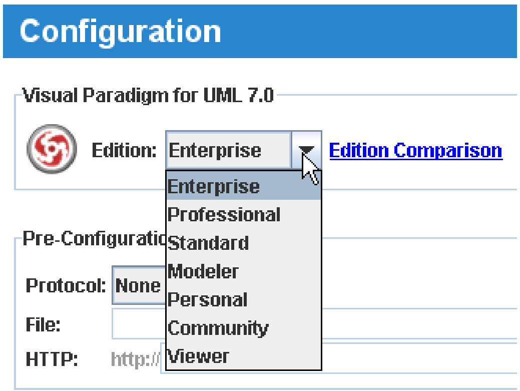
☐ SDE for Eclipse (SDE-EC)

☐ SDE for Visual Studio (SDE-VS)

SDE-VS can only be installed by the user with Computer administrator

__Figure 2-14__ The product selection page

4. VP-UML features vary by product edition. For more details on the features supported by different editions, click the hyperlink **Edition Comparison** which brings you to the web page of VP-UML feature comparison for different editions.



Configuration

Visual Paradigm for UML 7.0

Edition: Enterprise ▼ Edition Comparison

Enterprise
Professional
Standard
Modeler
Personal
Community
Viewer

Pre-Configuratio

Protocol: None

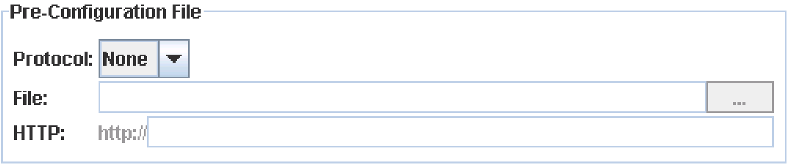
File:

HTTP: http://

Figure 2-15 *Select product edition*

5. For users who are going to run VP-UML with floating license, the Pre-Configuration File section enables you to configure the connection(s) to floating license server by providing a configuration file. The pre configuration is optional. If you do not define the connection here, you will need to do so when starting VP-UML the first time. Below is the pre-configuration file file content:

```
?xml version="1.0" encoding="UTF-8"?>
<LicenseServer>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
</LicenseServer>
```



Pre-Configuration File

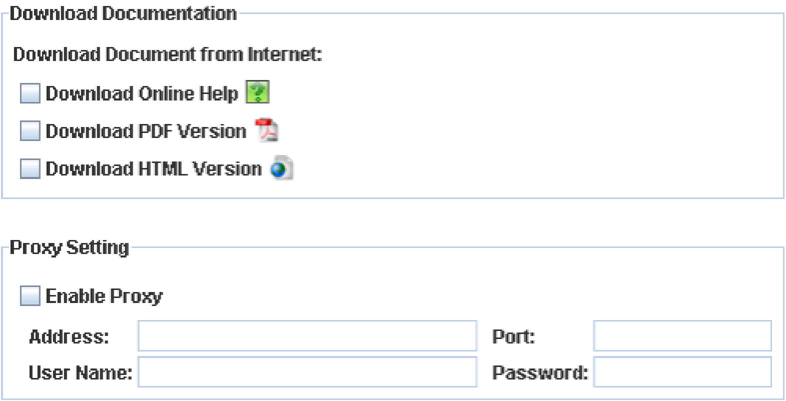
Protocol: None ▼

File: ...

HTTP: http://

Figure 2-16 *Pre-configure floating license server connection for floating license user*

6. Select **Download Online Help** if you want to be able to access the Help contents from within the tool. Select **Download PDF/HTML Version** if you wish to read the documentation in this two types of formats. Press **Next >** to proceed to the license import page.



Download Documentation

Download Document from Internet:

□ Download Online Help

□ Download PDF Version

□ Download HTML Version

Proxy Setting

□ Enable Proxy

Address: Port:

User Name: Password:

Figure 2-17 *Download documentation*

7. A combined license key is a key file which allows unlocking multiple products. To import a combined license key, select Combined License Key and specify the file path of the key.
A single license key is a key file which allows unlocking only one product. To import a single license key, select Single License Key and specify the file path of the key.
The import of license key is optional at this point, because you can import it when you run VP-UML. Click **Next >** to proceed with copying files.

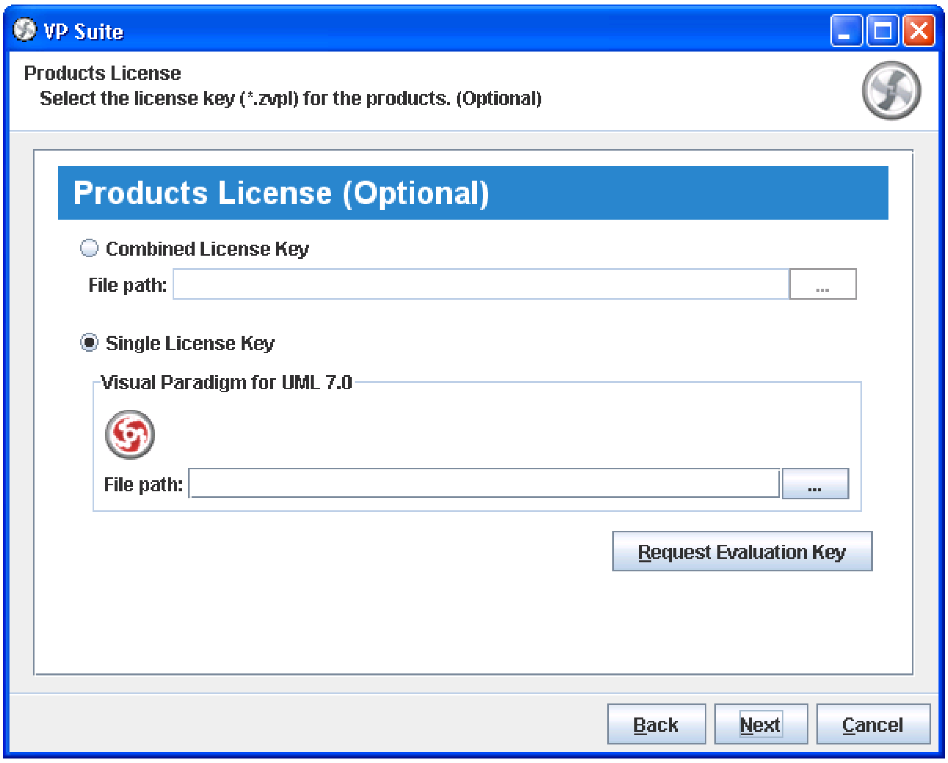


Figure 2-18 *The license key selection page*

NOTE: For evaluation, the key should be sent to your email box (unless you have chosen not to receive it before download). If you have not received the email with key yet, and if you have selected to receive the key as attachment, the email might be treated as spam by mistake. Click on **Request Evaluation Key** to ask for another one. This time, try not to select sending the key as attachment.

8. Upon finishing, you can select whether to start VP-UML or not. Keep **Visual Paradigm for UML** selected and click **Finish** will run VP-UML right away.

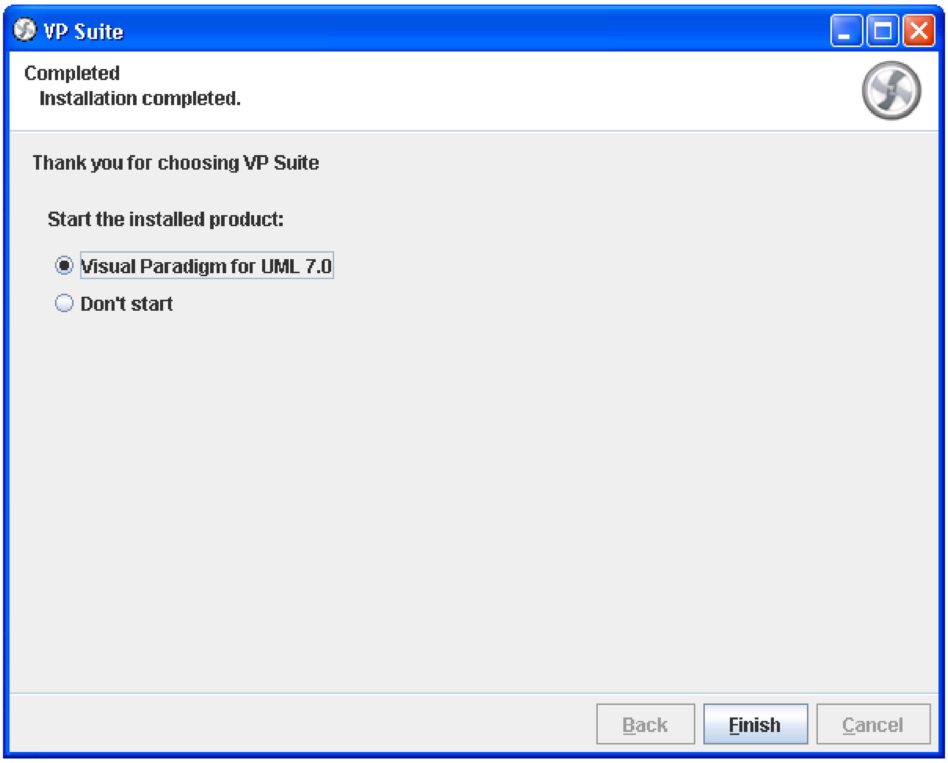

VP Suite
Completed
Installation completed.
Thank you for choosing VP Suite
Start the installed product:
● Visual Paradigm for UML 7.0
○ Don't start
Back
Finish
Cancel

Figure 2-19 *Installation completed screen*

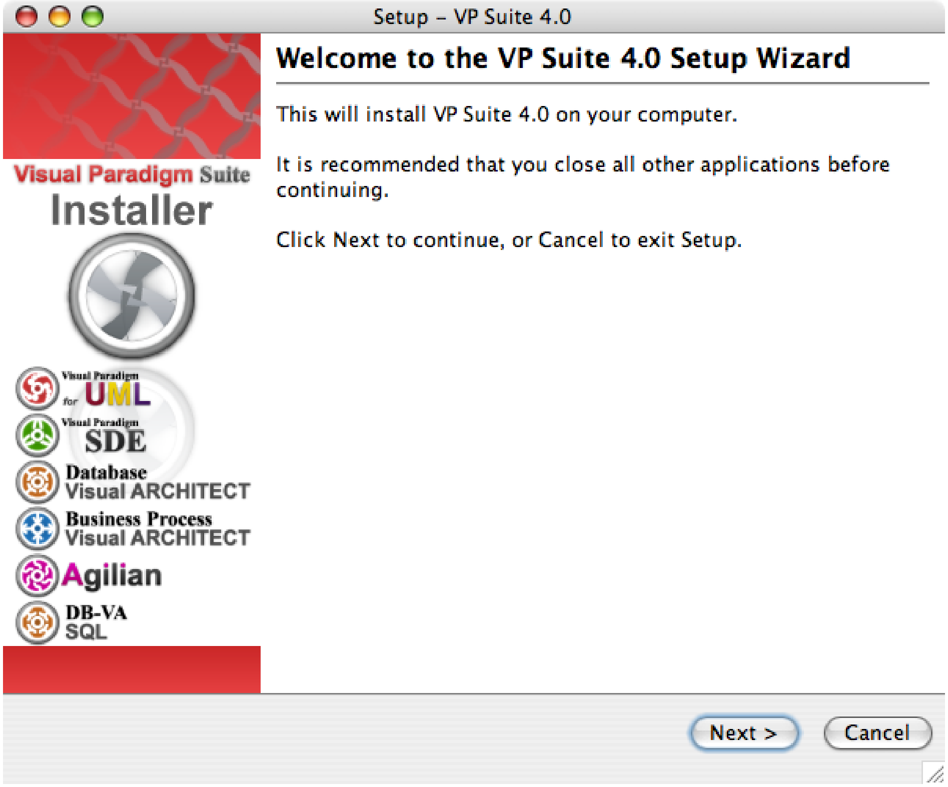
Installation FAQ

Question: What is the difference between Installer and &ldquo;Not Install Version&rdquo;?
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 vp.log file, which can be found at the bin folder of VP Suite installation directory, for our team to diagnose in further.
Question: I don't have administrator right, can I install the software?
Answer: Yes, you can.
Question: Can I change the Edition without re-install the software?
Answer: Yes, you can. Product edition can be changed by running VP Suite Product Edition Manager under the bin folder of VP Suite installation directory. Change of edition takes effect after the restart of affected products. For more details, please read the section Switching Edition a few pages later.
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.

Installing Visual Paradigm for UML on Mac OSX

Using Installer (.dmg)

1. Execute the downloaded VP Suite installer file:
   bash ./%VP-SUITE-INSTALLER-FILENAME% (e.g. bash ./VP_Suite_Linux_7_0.sh)
   The setup wizard appear as below.



Setup - VP Suite 4.0

Welcome to the VP Suite 4.0 Setup Wizard

This will install VP Suite 4.0 on your computer.

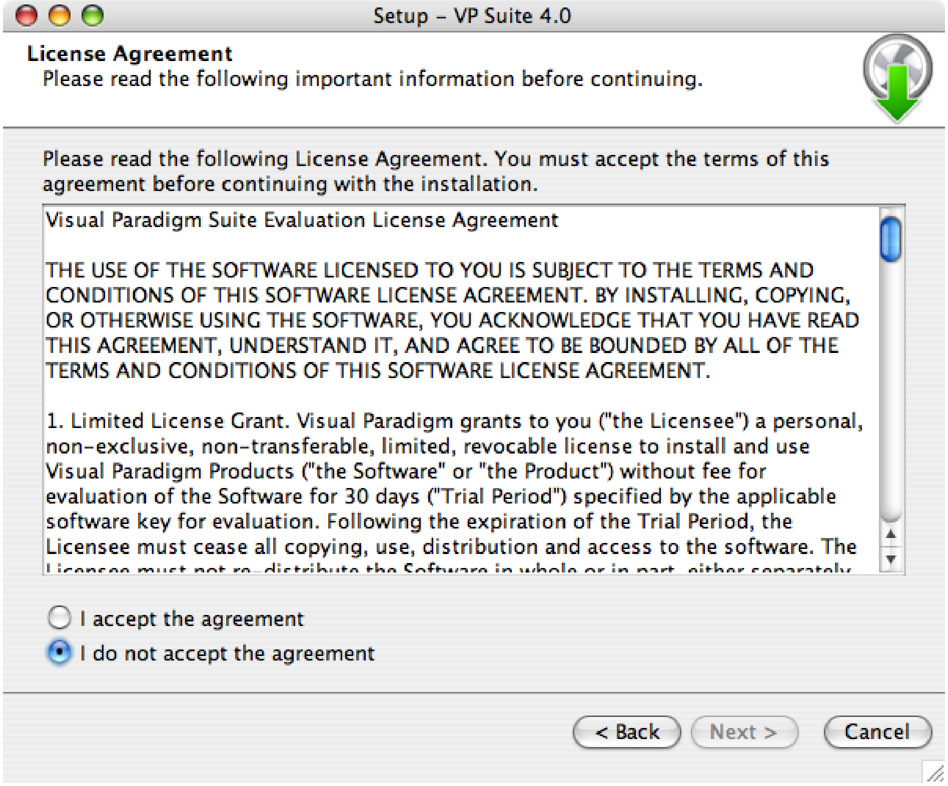
It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 2-20 *VP Suite 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.



Setup – VP Suite 4.0

License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Visual Paradigm Suite Evaluation License Agreement

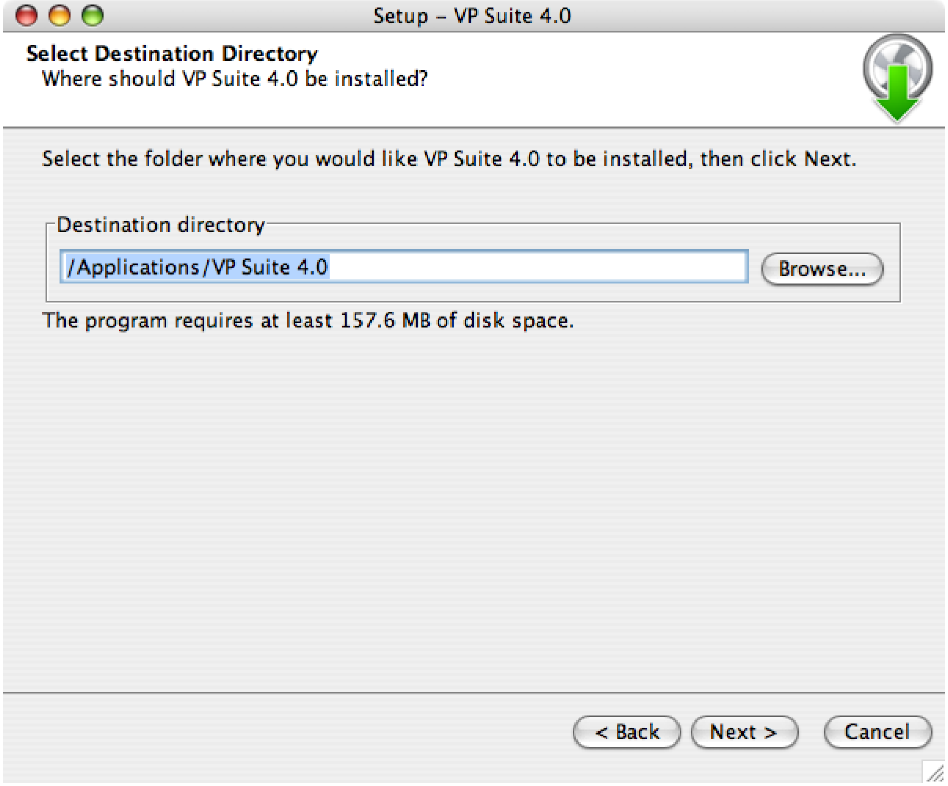
THE USE OF THE SOFTWARE LICENSED TO YOU IS SUBJECT TO THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT. BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE TO BE BOUNDED BY ALL OF THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT.

1. Limited License Grant. Visual Paradigm grants to you ("the Licensee") a personal, non-exclusive, non-transferable, limited, revocable license to install and use Visual Paradigm Products ("the Software" or "the Product") without fee for evaluation of the Software for 30 days ("Trial Period") specified by the applicable software key for evaluation. Following the expiration of the Trial Period, the Licensee must cease all copying, use, distribution and access to the software. The Licensee must not re-distribute the Software in whole or in part, either separately

○ I accept the agreement
● I do not accept the agreement

Figure 2-21 The License Agreement

4. Specify the directory for installing VP Suite. Click **Next >** to proceed to the next page.



Setup – VP Suite 4.0

Select Destination Directory
Where should VP Suite 4.0 be installed?

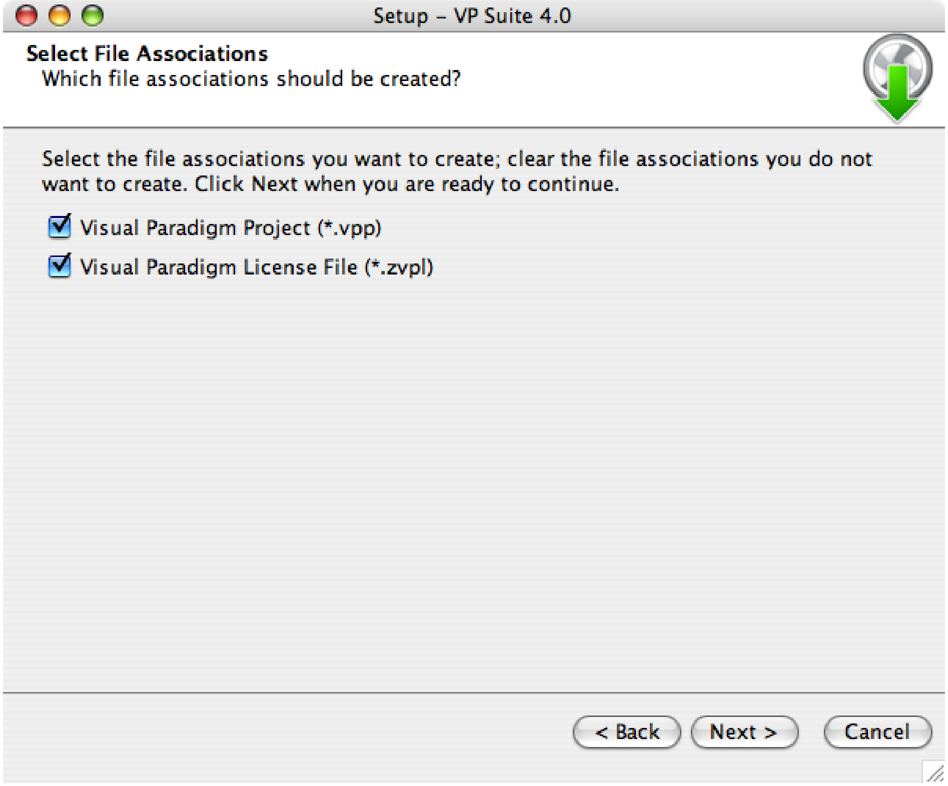
Select the folder where you would like VP Suite 4.0 to be installed, then click Next.

Destination directory: /Applications/VP Suite 4.0

The program requires at least 157.6 MB of disk space.

Figure 2-22 The Select Destination Directory page

5. In the File Association page, keep **Visual Paradigm Project (*.vpp)** and **Visual Paradigm License File (*.zvpl)** checked if you want your system able to open the project file and the license key file. Click **Next >** to proceed to the product selection page.



Setup - VP Suite 4.0

Select File Associations

Which file associations should be created?

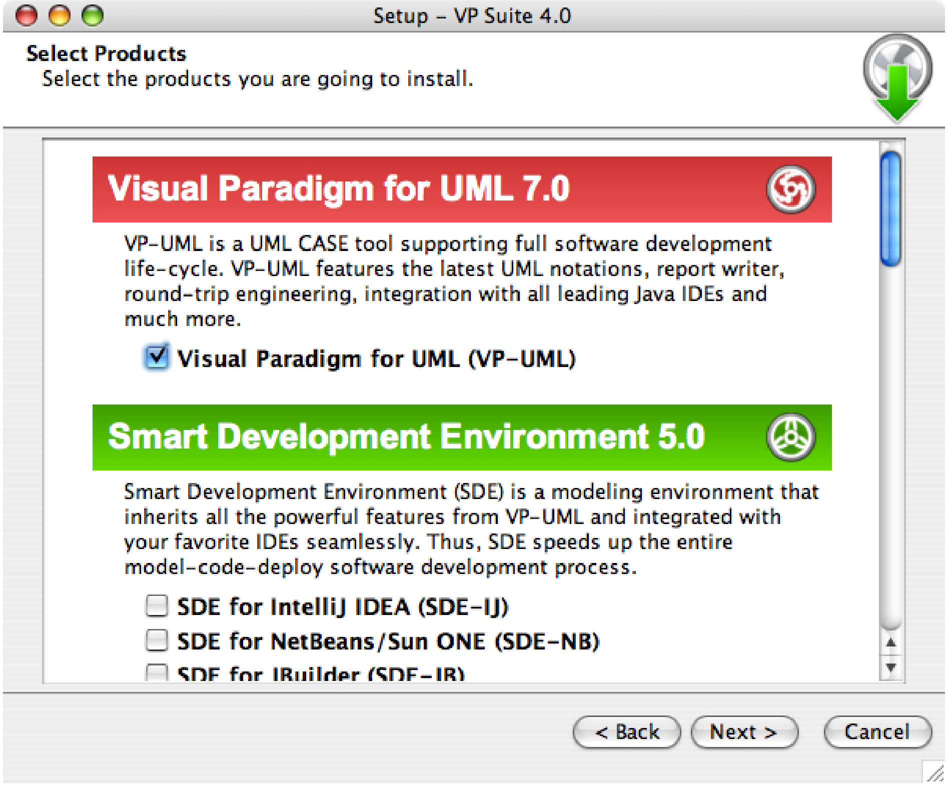
Select the file associations you want to create; clear the file associations you do not want to create. Click Next when you are ready to continue.

☑ Visual Paradigm Project (*.vpp)

☑ Visual Paradigm License File (*.zvpl)

Figure 2-23 *The Select File Associations page*

6. Select **Visual Paradigm for UML**. Click **Next >** to proceed to the product configuration page.



Setup - VP Suite 4.0

Select Products

Select the products you are going to install.

Visual Paradigm for UML 7.0

VP-UML is a UML CASE tool supporting full software development life-cycle. VP-UML features the latest UML notations, report writer, round-trip engineering, integration with all leading Java IDEs and much more.

☑ **Visual Paradigm for UML (VP-UML)**

Smart Development Environment 5.0

Smart Development Environment (SDE) is a modeling environment that inherits all the powerful features from VP-UML and integrated with your favorite IDEs seamlessly. Thus, SDE speeds up the entire model-code-deploy software development process.

□ **SDE for IntelliJ IDEA (SDE-IJ)**

□ **SDE for NetBeans/Sun ONE (SDE-NB)**

□ **SDE for JBuilder (SDE-JB)**

Figure 2-24 *The product selection page*

7. VP-UML features vary by product edition. For more details on the features supported by different editions, click the hyperlink **Edition Comparison** which brings you to the web page of VP-UML feature comparison for different editions.

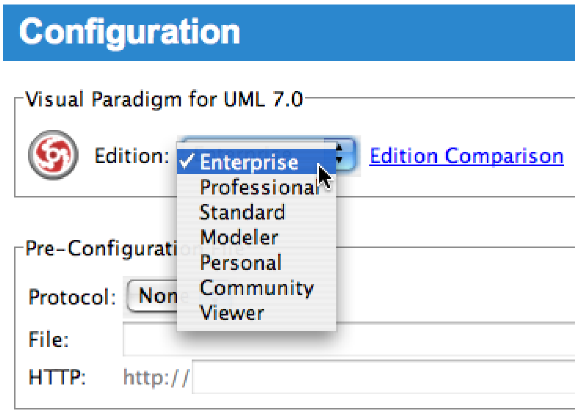


Figure 2-25 *Select product edition*

8. For users who are going to run VP-UML with floating license, the Pre-Configuration File section enables you to configure the connection(s) to floating license server by providing a configuration file. The pre configuration is optional. If you do not define the connection here, you will need to do so when starting VP-UML the first time. Below is the pre-configuration file file content:

```
?xml version="1.0" encoding="UTF-8"?>
<LicenseServer>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
</LicenseServer>
```

Selecting **Enable Product Selector** for Product Selection will result in creating a shortcut under the Start Menu for starting the Product Selector, a utility that lets you realize the installed products with available keys in the floating license server.

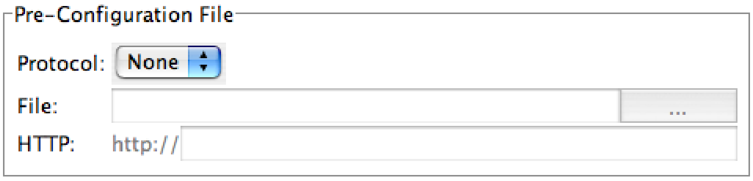


Figure 2-26 *Pre-configure floating license server connection for floating license user*

9. Select **Download Online Help** if you want to be able to access the Help contents from within the tool. Select **Download PDF/HTML Version** if you wish to read the documentation in this two types of formats. Press **Next >** to proceed to the license import page.

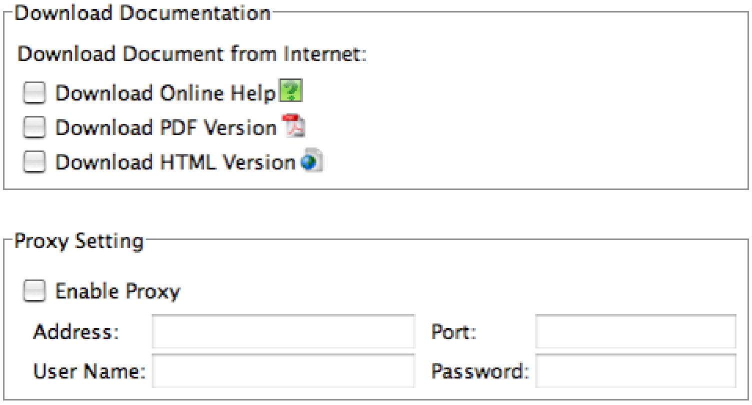
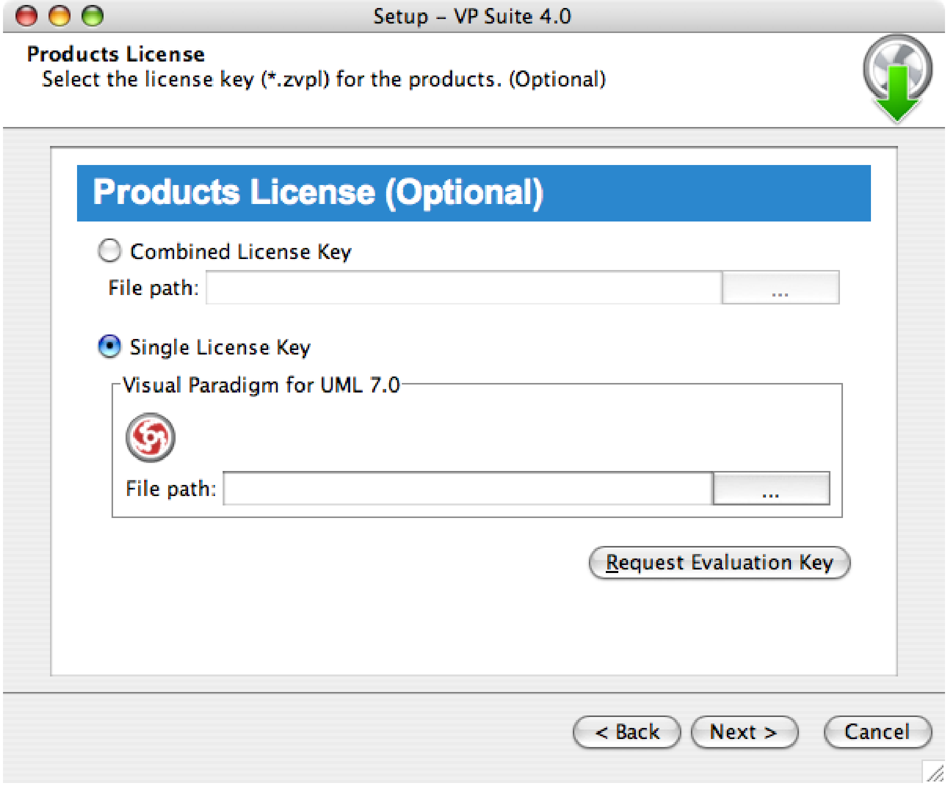


Figure 2-27 *Download documentation*

10. A combined license key is a key file which allows unlocking multiple products. To import a combined license key, select Combined License Key and specify the file path of the key.
A single license key is a key file which allows unlocking only one product. To import a single license key, select Single License Key and specify the file path of the key.
The import of license key is optional at this point, because you can import it when you run VP-UML. Click **Next >** to proceed with copying files.



Setup – VP Suite 4.0

Products License
Select the license key (*.zvpl) for the products. (Optional)

Products License (Optional)

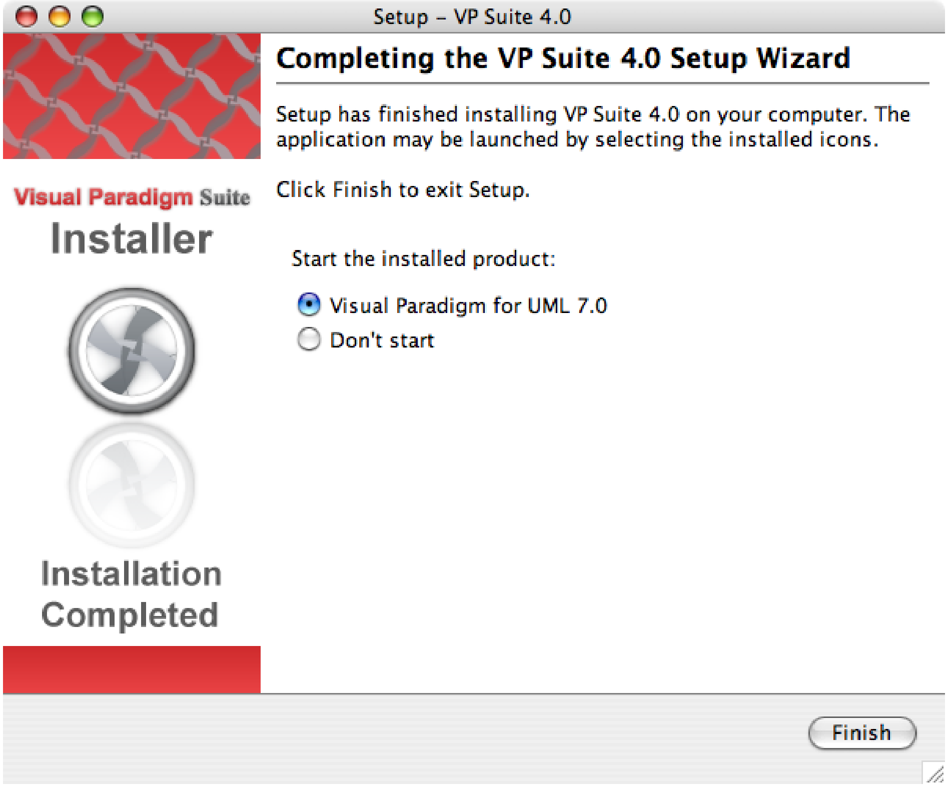
○ Combined License Key
File path:

● Single License Key
Visual Paradigm for UML 7.0
File path:

Figure 2-28 *The license key selection page*

NOTE: For evaluation, the key should be sent to your email box (unless you have chosen not to receive it before download). If you have not received the email with key yet, and if you have selected to receive the key as attachment, the email might be treated as spam by mistake. Click on Request Evaluation Key to ask for another one. This time, try not to select sending the key as attachment.

11. Upon finishing, you can select whether to start VP-UML or not. Keep **Visual Paradigm for UML** selected and click **Finish** will run VP-UML right away.



Completing the VP Suite 4.0 Setup Wizard

Setup has finished installing VP Suite 4.0 on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

Start the installed product:

● Visual Paradigm for UML 7.0

○ Don't start

Figure 2-29 Installation completed screen

Using No Install Version (.tgz)

1. Decompress the downloaded zip file into a directory.
tar -zxf %NO-INSTALL-FILE.tar.gz% -C %DESTINATION-FOLDER%
This should create a subdirectory named &ldquo;VP Suite 4.0&rdquo; where 4.0 is the version number.

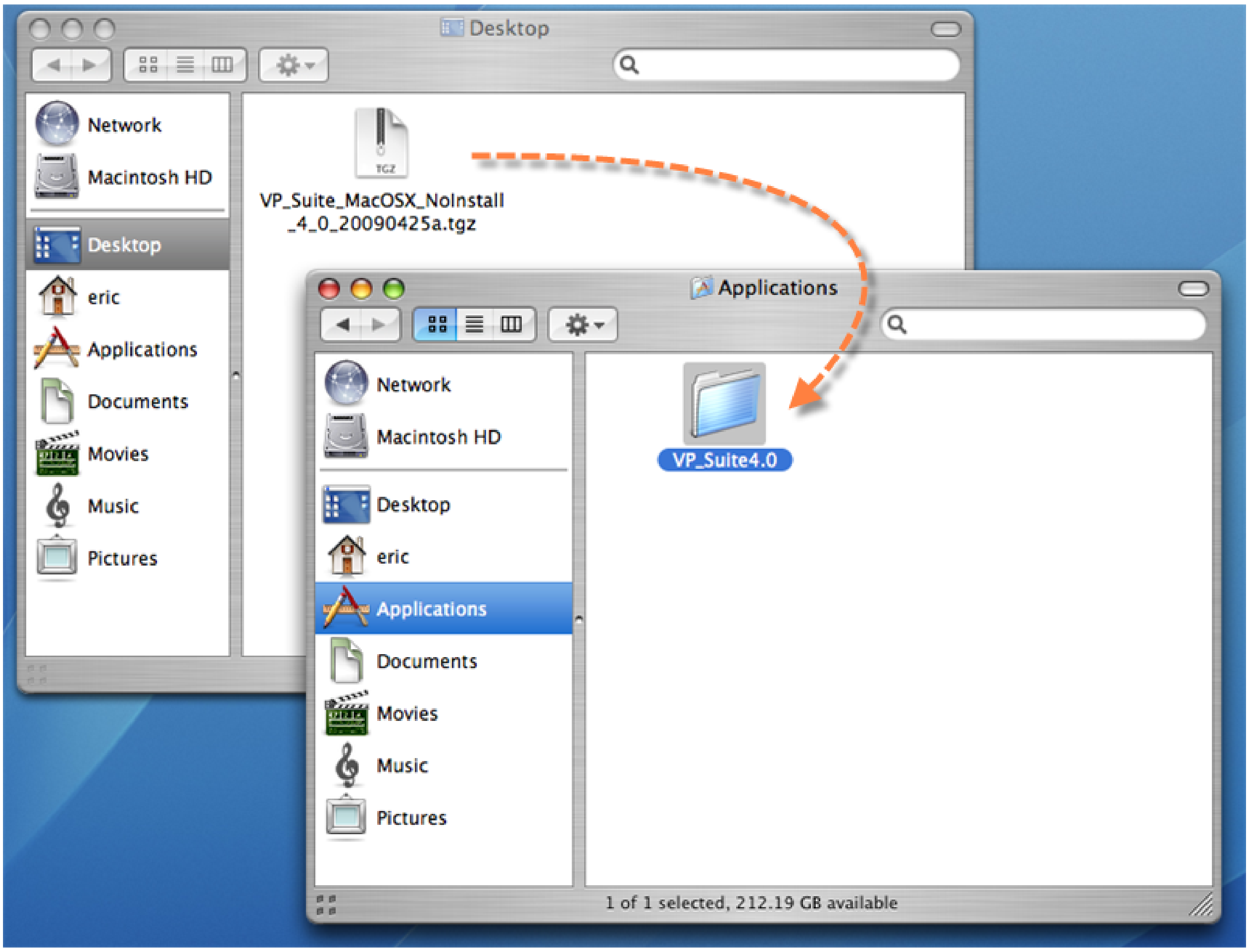


***Figure 2-30** Extracting No-Install zip file*

2. Change directory to &ldquo;VP Suite 4.0/bin&rdquo; and execute VP Suite in it.

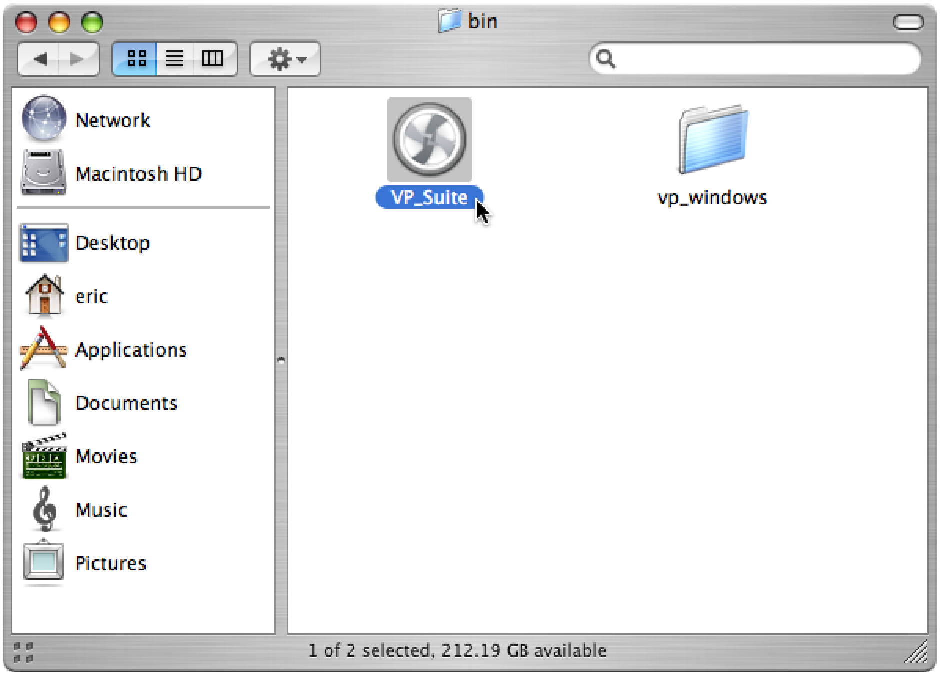
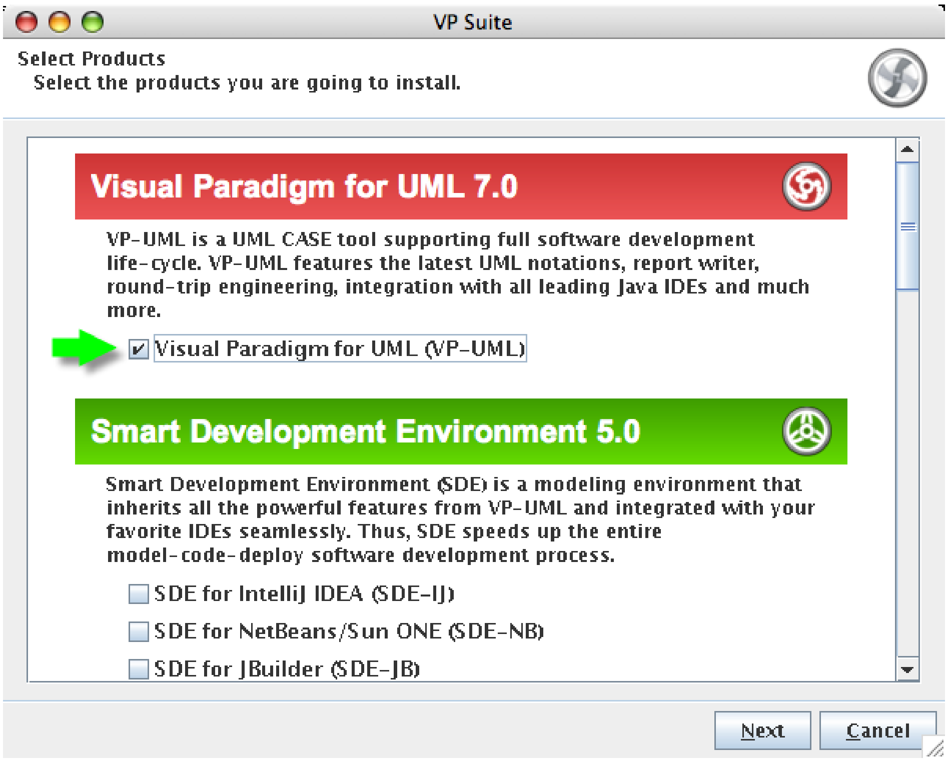


***Figure 2-31** Launching VP Suite*

3. Select **Visual Paradigm for UML**. Click **Next >** to proceed to the product configuration page.



VP Suite

Select Products
Select the products you are going to install.

Visual Paradigm for UML 7.0

VP-UML is a UML CASE tool supporting full software development life-cycle. VP-UML features the latest UML notations, report writer, round-trip engineering, integration with all leading Java IDEs and much more.

☑ Visual Paradigm for UML (VP-UML)

Smart Development Environment 5.0

Smart Development Environment (SDE) is a modeling environment that inherits all the powerful features from VP-UML and integrated with your favorite IDEs seamlessly. Thus, SDE speeds up the entire model-code-deploy software development process.

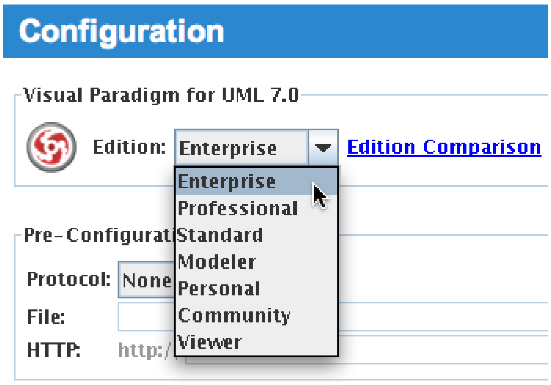
□ SDE for IntelliJ IDEA (SDE-IJ)

□ SDE for NetBeans/Sun ONE (SDE-NB)

□ SDE for JBuilder (SDE-JB)

Figure 2-32 *The product selection page*

4. VP-UML features vary by product edition. For more details on the features supported by different editions, click the hyperlink **Edition Comparison** which brings you to the web page of VP-UML feature comparison for different editions.



Configuration

Visual Paradigm for UML 7.0

Edition: Enterprise Edition Comparison

Pre-Configurati

Protocol: None

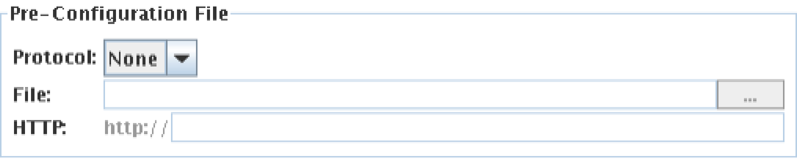
File:

HTTP: http:/

Figure 2-33 *Select product edition*

5. For users who are going to run VP-UML with floating license, the Pre-Configuration File section enables you to configure the connection(s) to floating license server by providing a configuration file. The pre configuration is optional. If you do not define the connection here, you will need to do so when starting VP-UML the first time. Below is the pre-configuration file file content:

```
?xml version="1.0" encoding="UTF-8"?>
<LicenseServer>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
</LicenseServer>
```



Pre-Configuration File

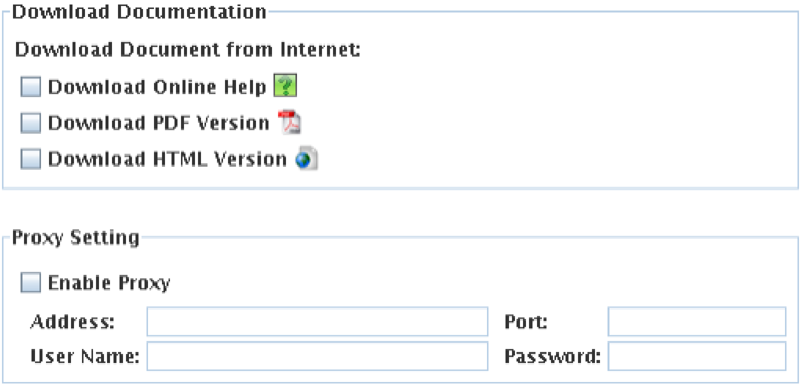
Protocol: None

File:

HTTP: http://

Figure 2-34 *Pre-configure floating license server connection for floating license user*

6. Select **Download Online Help** if you want to be able to access the Help contents from within the tool. Select **Download PDF/HTML Version** if you wish to read the documentation in this two types of formats. Press **Next >** to proceed to the license import page.



Download Documentation

Download Document from Internet:

☐ Download Online Help

☐ Download PDF Version

☐ Download HTML Version

Proxy Setting

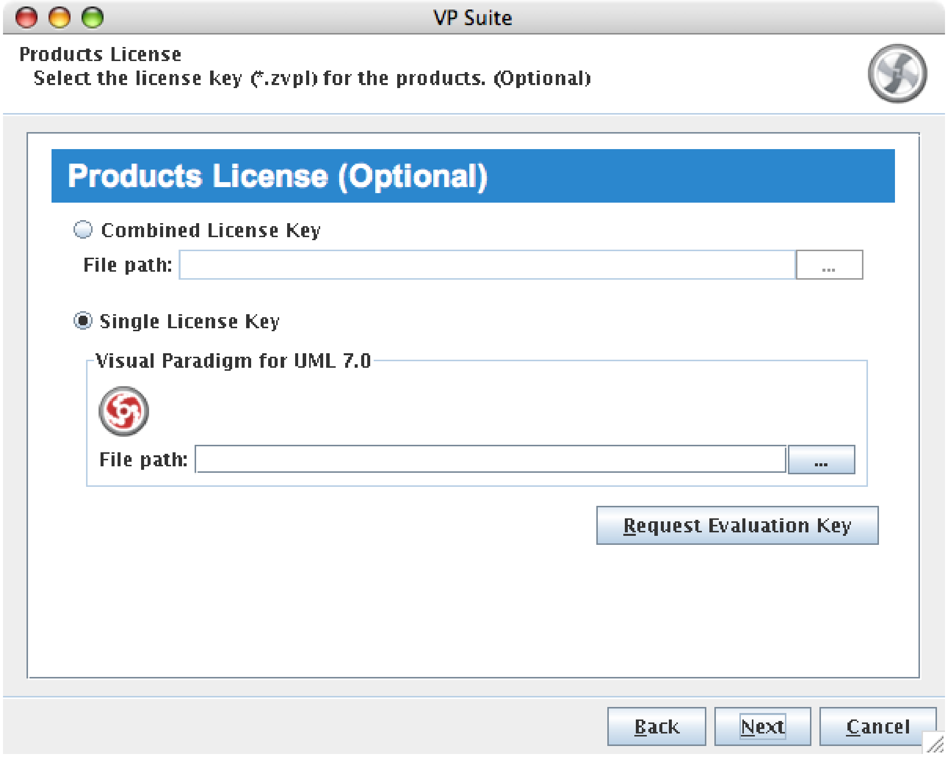
☐ Enable Proxy

Address: Port:

User Name: Password:

Figure 2-35 *Download documentation*

7. A combined license key is a key file which allows unlocking multiple products. To import a combined license key, select Combined License Key and specify the file path of the key.
A single license key is a key file which allows unlocking only one product. To import a single license key, select Single License Key and specify the file path of the key.
The import of license key is optional at this point, because you can import it when you run VP-UML. Click **Next >** to proceed with copying files.



VP Suite

Products License
Select the license key (*.zvpl) for the products. (Optional)

Products License (Optional)

○ Combined License Key

File path:

● Single License Key

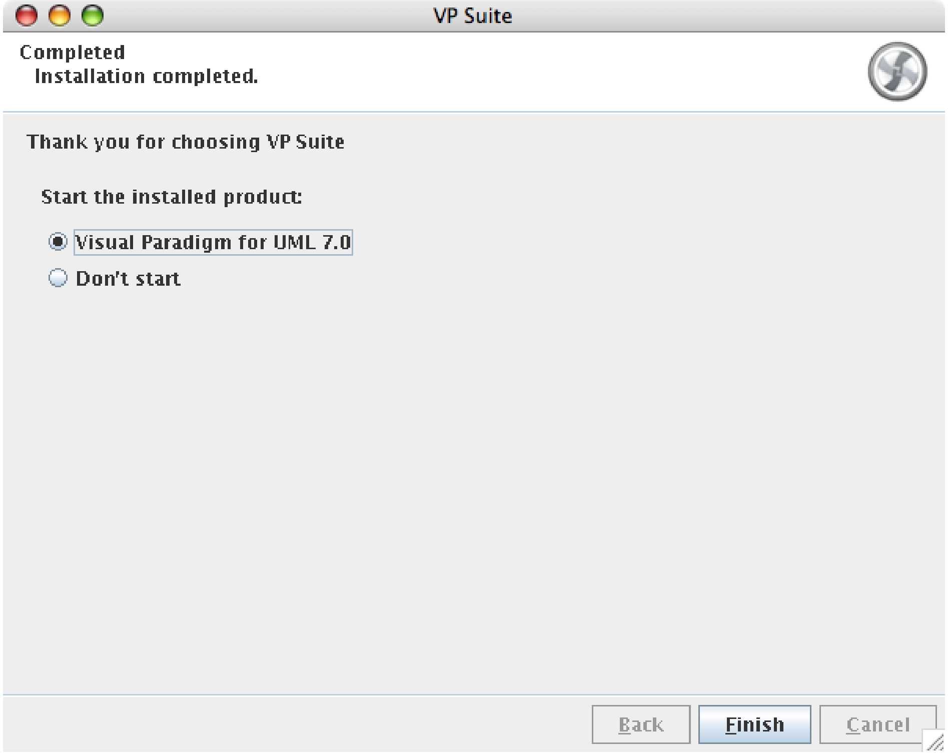
Visual Paradigm for UML 7.0

File path:

Figure 2-36 *The license key selection page*

NOTE: For evaluation, the key should be sent to your email box (unless you have chosen not to receive it before download). If you have not received the email with key yet, and if you have selected to receive the key as attachment, the email might be treated as spam by mistake. Click on **Request Evaluation Key** to ask for another one. This time, try not to select sending the key as attachment.

8. Upon finishing, you can select whether to start VP-UML or not. Keep **Visual Paradigm for UML** selected and click **Finish** will run VP-UML right away.



VP Suite

Completed
Installation completed.

Thank you for choosing VP Suite

Start the installed product:

● Visual Paradigm for UML 7.0
○ Don't start

Figure 2-37 *Installation completed screen*

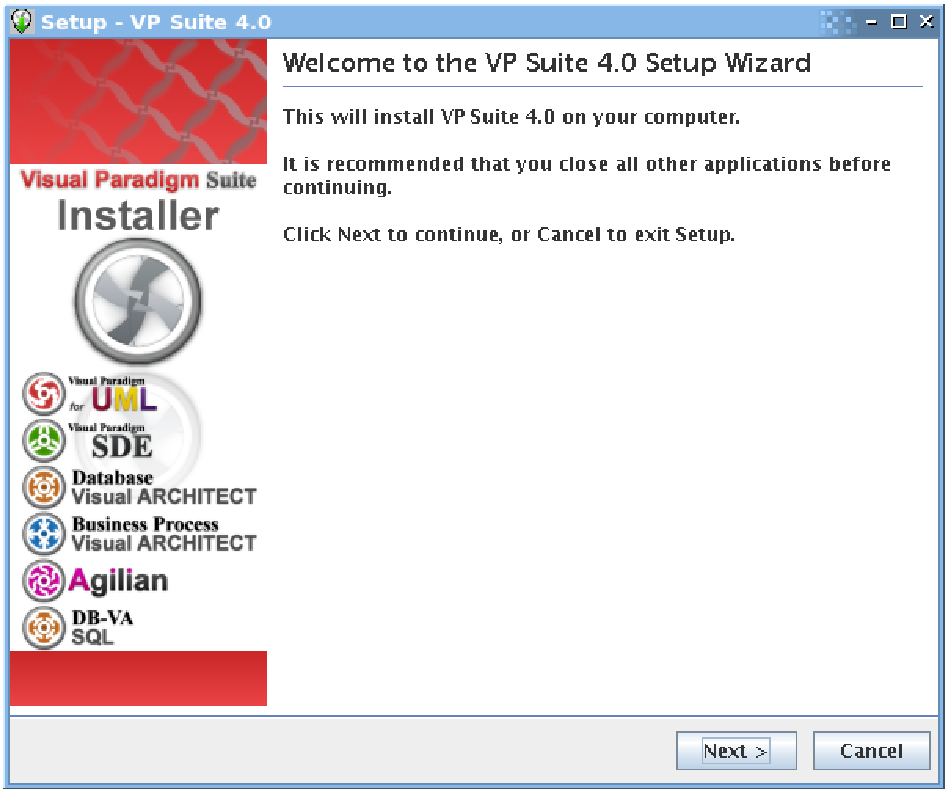
Installation FAQ

Question: What is the difference between Installer and &ldquo;Not Install Version&rdquo;?
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 vp.log file, which can be found at the bin folder of VP Suite installation directory, for our team to diagnose in further.
Question: I don't have administrator right, can I install the software?
Answer: Yes, you can.
Question: Can I change the Edition without re-install the software?
Answer: Yes, you can. Product edition can be changed by running VP Suite Product Edition Manager under the bin folder of VP Suite installation directory. Change of edition takes effect after the restart of affected products. For more details, please read the section Switching Edition a few pages later.
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.

Installing Visual Paradigm for UML on Linux and Unix

Using Installer (.sh)

1. Execute the downloaded VP Suite installer file:
   bash ./%VP-SUITE-INSTALLER-FILENAME% (e.g. bash ./VP_Suite_Linux_7_0.sh)
   The setup wizard appear as below.



Setup - VP Suite 4.0

Welcome to the VP Suite 4.0 Setup Wizard

This will install VP Suite 4.0 on your computer.

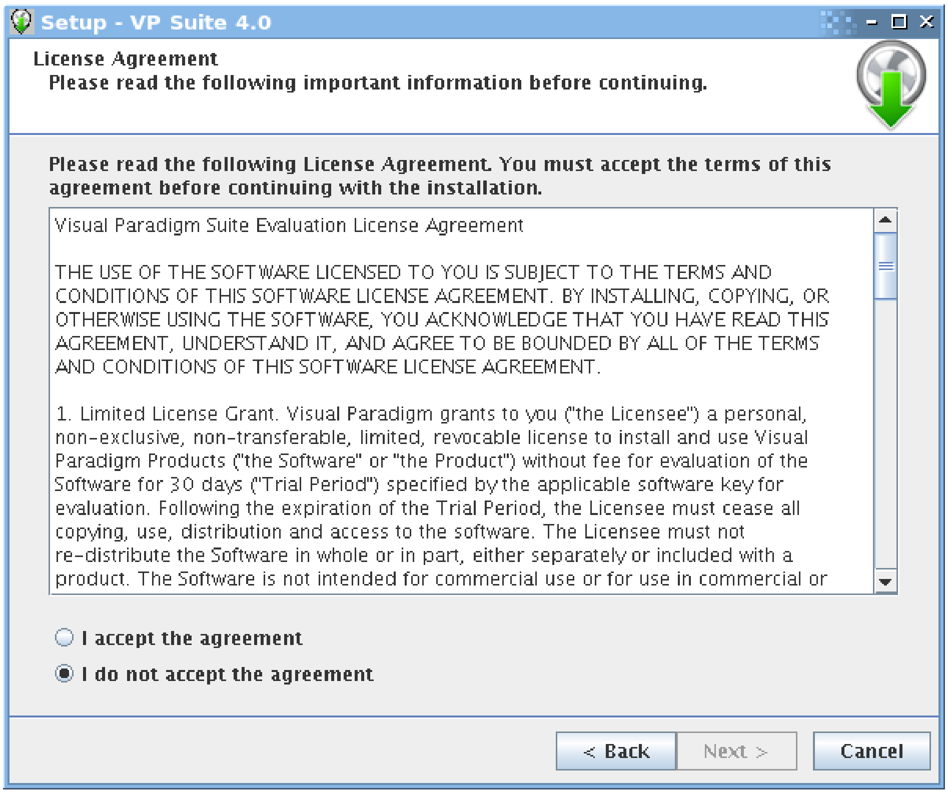
It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 2-38 *VP Suite 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.



License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Visual Paradigm Suite Evaluation License Agreement

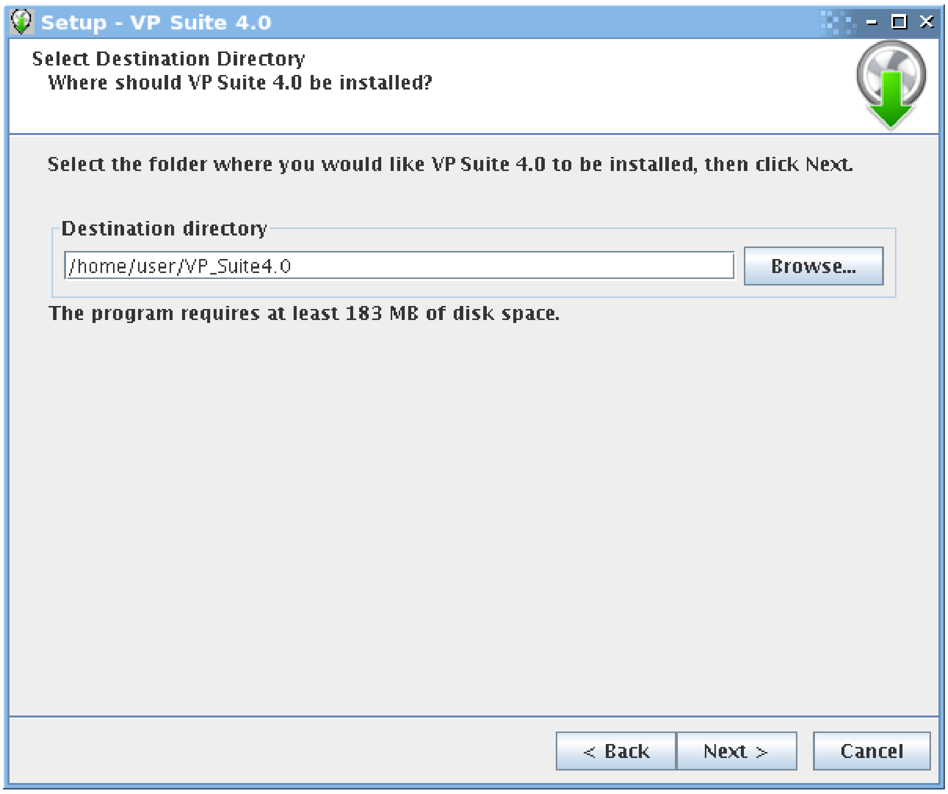
THE USE OF THE SOFTWARE LICENSED TO YOU IS SUBJECT TO THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT. BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE TO BE BOUNDED BY ALL OF THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT.

1. Limited License Grant. Visual Paradigm grants to you ("the Licensee") a personal, non-exclusive, non-transferable, limited, revocable license to install and use Visual Paradigm Products ("the Software" or "the Product") without fee for evaluation of the Software for 30 days ("Trial Period") specified by the applicable software key for evaluation. Following the expiration of the Trial Period, the Licensee must cease all copying, use, distribution and access to the software. The Licensee must not re-distribute the Software in whole or in part, either separately or included with a product. The Software is not intended for commercial use or for use in commercial or

○ I accept the agreement
● I do not accept the agreement

__Figure 2-39__ The License Agreement

4. Specify the directory for installing VP Suite. Click **Next >** to proceed to the next page.



Select Destination Directory
Where should VP Suite 4.0 be installed?

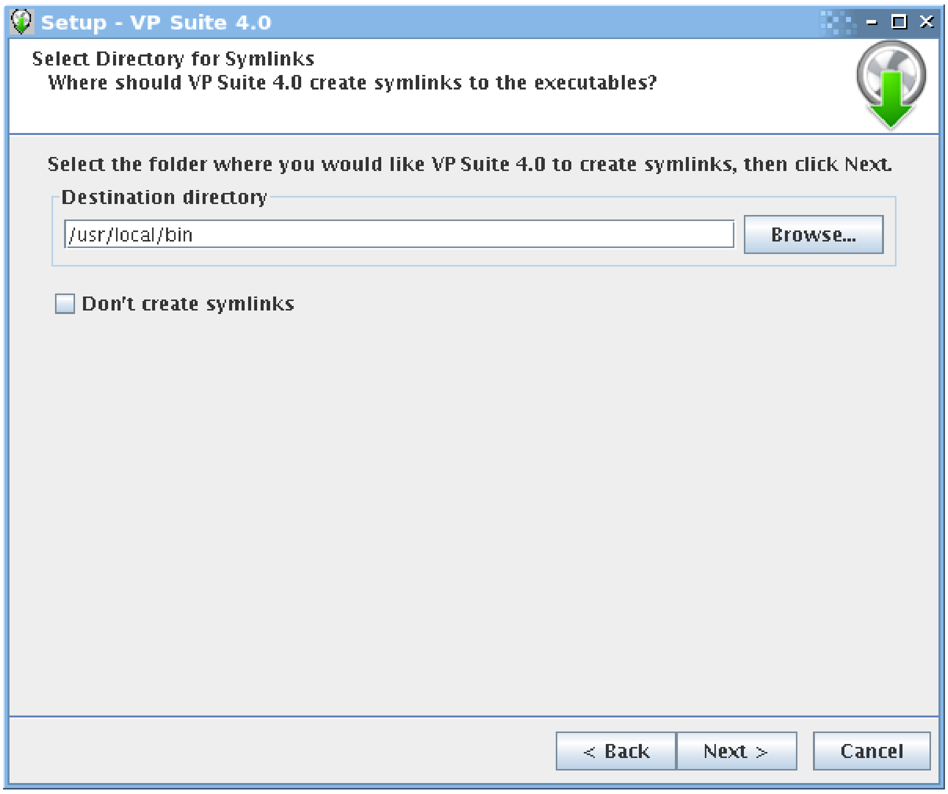
Select the folder where you would like VP Suite 4.0 to be installed, then click Next.

Destination directory
/home/user/VP_Suite4.0

The program requires at least 183 MB of disk space.

__Figure 2-40__ The Select Destination Directory page

5. select a folder for creating symlinks. You may select **Don't create symlinks** if you do not want to. Click **Next** to proceed.



Setup - VP Suite 4.0

Select Directory for Symlinks
Where should VP Suite 4.0 create symlinks to the executables?

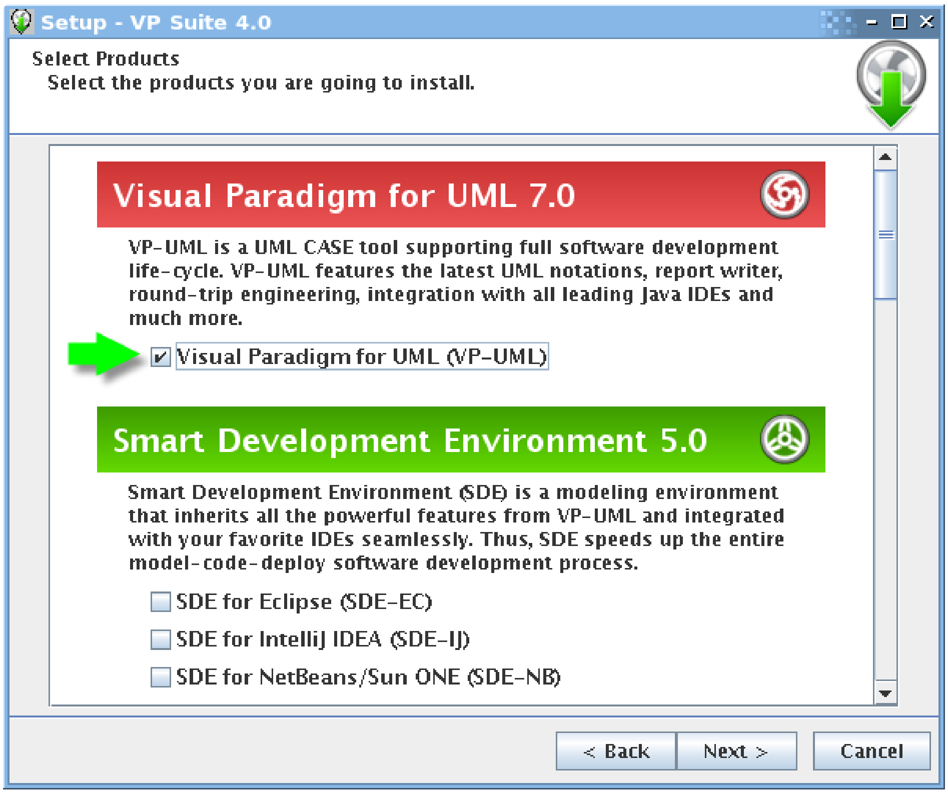
Select the folder where you would like VP Suite 4.0 to create symlinks, then click Next.

Destination directory: /usr/local/bin

□ Don't create symlinks

Figure 2-41 *The Select Symlinks Directory page*

6. Select **Visual Paradigm for UML**. Click **Next >** to proceed to the product configuration page.



Setup - VP Suite 4.0

Select Products
Select the products you are going to install.

Visual Paradigm for UML 7.0

VP-UML is a UML CASE tool supporting full software development life-cycle. VP-UML features the latest UML notations, report writer, round-trip engineering, integration with all leading Java IDEs and much more.

☑ Visual Paradigm for UML (VP-UML)

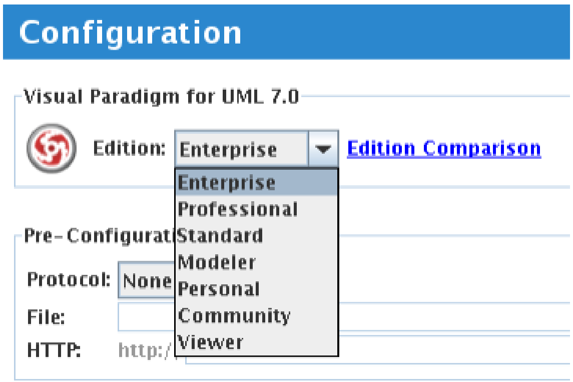
Smart Development Environment 5.0

Smart Development Environment (SDE) is a modeling environment that inherits all the powerful features from VP-UML and integrated with your favorite IDEs seamlessly. Thus, SDE speeds up the entire model-code-deploy software development process.

□ SDE for Eclipse (SDE-EC)
□ SDE for IntelliJ IDEA (SDE-IJ)
□ SDE for NetBeans/Sun ONE (SDE-NB)

Figure 2-42 *The product selection page*

7. VP-UML features vary by product edition. For more details on the features supported by different editions, click the hyperlink **Edition Comparison** which brings you to the web page of VP-UML feature comparison for different editions.



Configuration

Visual Paradigm for UML 7.0

Edition: Enterprise

Pre-Configurat

Protocol: None

File:

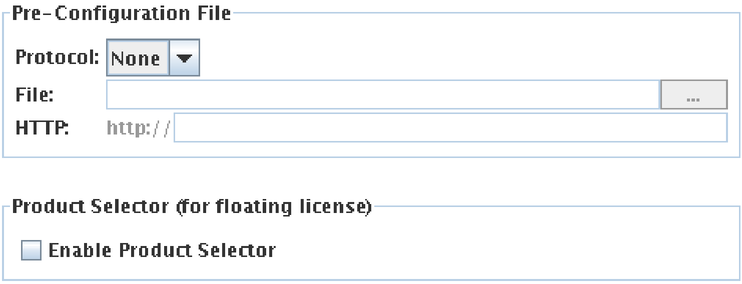
HTTP: http:/

__Figure 2-43__ Select product edition

8. For users who are going to run VP-UML with floating license, the Pre-Configuration File section enables you to configure the connection(s) to floating license server by providing a configuration file. The pre configuration is optional. If you do not define the connection here, you will need to do so when starting VP-UML the first time. Below is the pre-configuration file file content:

```
?xml version="1.0" encoding="UTF-8"?>
<LicenseServer>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
</LicenseServer>
```

Selecting **Enable Product Selector** for Product Selection will result in creating a shortcut under the Start Menu for starting the Product Selector, a utility that lets you realize the installed products with available keys in the floating license server.



Pre-Configuration File

Protocol: None

File:

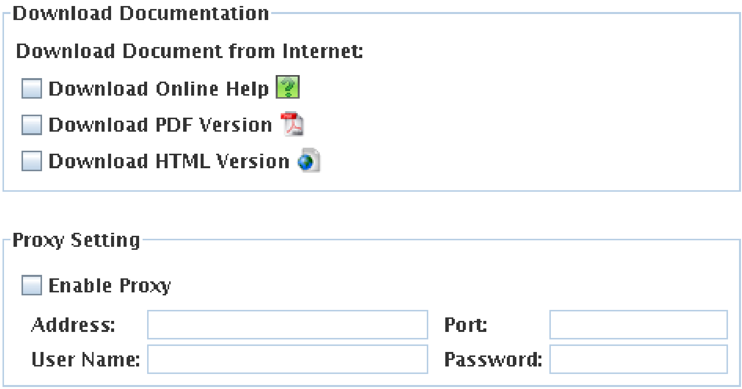
HTTP: http://

Product Selector (for floating license)

□ Enable Product Selector

__Figure 2-44__ Pre-configure floating license server connection for floating license user

9. Select **Download Online Help** if you want to be able to access the Help contents from within the tool. Select **Download PDF/HTML Version** if you wish to read the documentation in this two types of formats. Press **Next >** to proceed to the license import page.



Download Documentation

Download Document from Internet:

□ Download Online Help

□ Download PDF Version

□ Download HTML Version

Proxy Setting

□ Enable Proxy

Address:

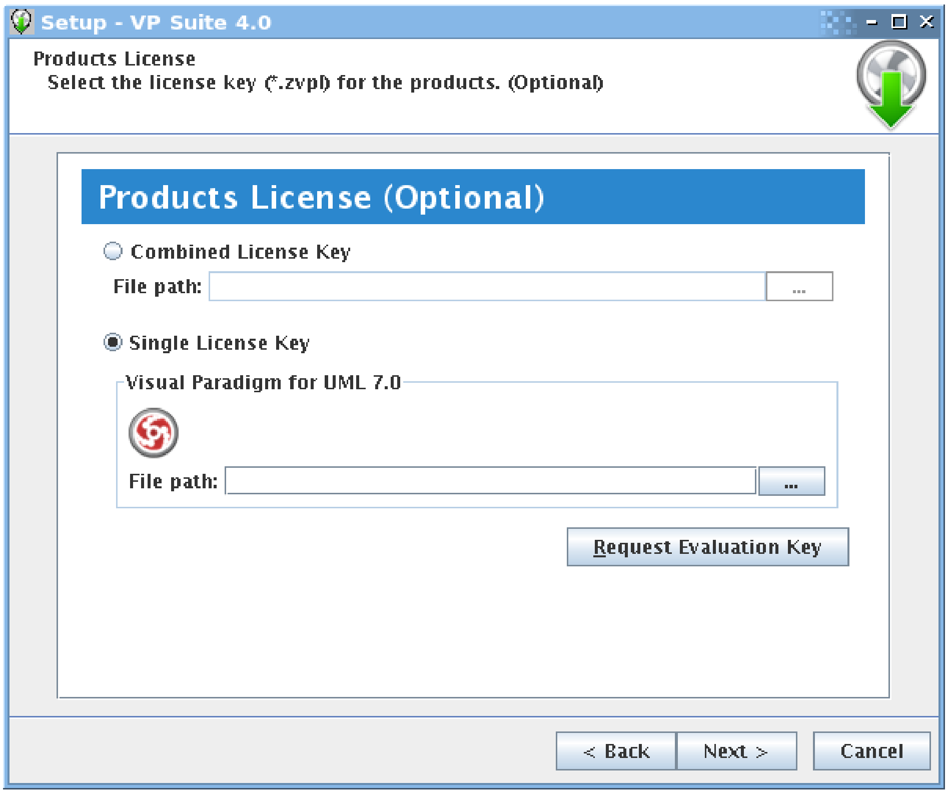
Port:

User Name:

Password:

__Figure 2-45__ Download documentation

10. A combined license key is a key file which allows unlocking multiple products. To import a combined license key, select Combined License Key and specify the file path of the key.
A single license key is a key file which allows unlocking only one product. To import a single license key, select Single License Key and specify the file path of the key.
The import of license key is optional at this point, because you can import it when you run VP-UML. Click **Next >** to proceed with copying files.



Setup - VP Suite 4.0

Products License
Select the license key (*.zvpl) for the products. (Optional)

Products License (Optional)

○ Combined License Key

File path:

● Single License Key

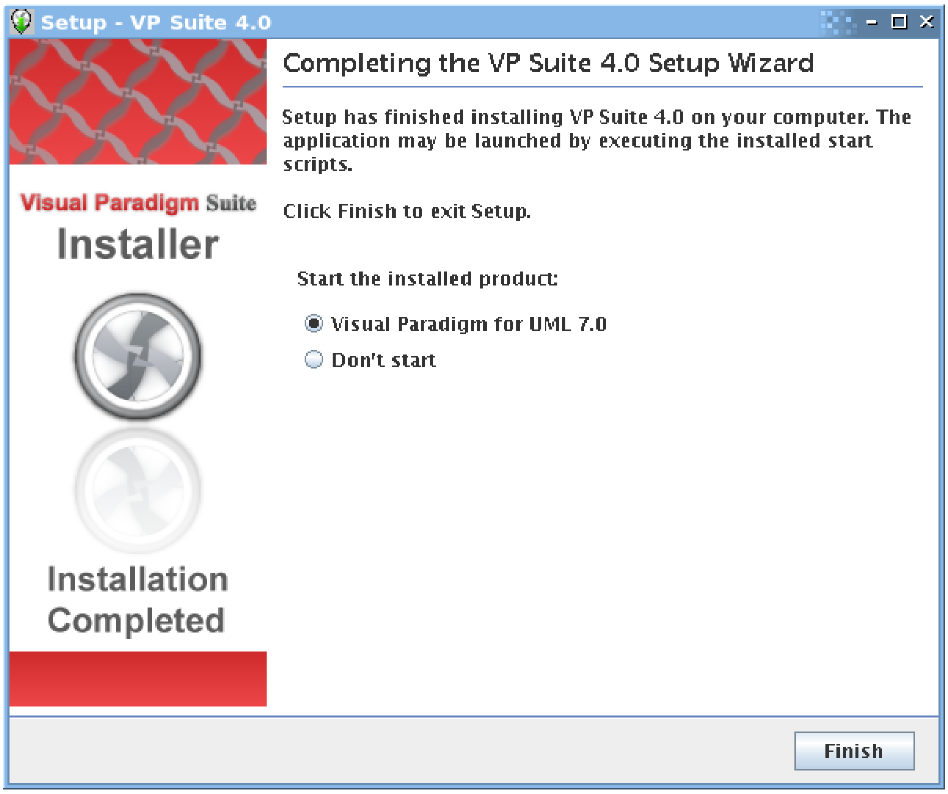
Visual Paradigm for UML 7.0

File path:

Figure 2-46 *The license key selection page*

NOTE: For evaluation, the key should be sent to your email box (unless you have chosen not to receive it before download). If you have not received the email with key yet, and if you have selected to receive the key as attachment, the email might be treated as spam by mistake. Click on Request Evaluation Key to ask for another one. This time, try not to select sending the key as attachment

11. Upon finishing, you can select whether to start VP-UML or not. Keep **Visual Paradigm for UML** selected and click **Finish** will run VP-UML right away.



Setup - VP Suite 4.0

Completing the VP Suite 4.0 Setup Wizard

Setup has finished installing VP Suite 4.0 on your computer. The application may be launched by executing the installed start scripts.

Click Finish to exit Setup.

Start the installed product:

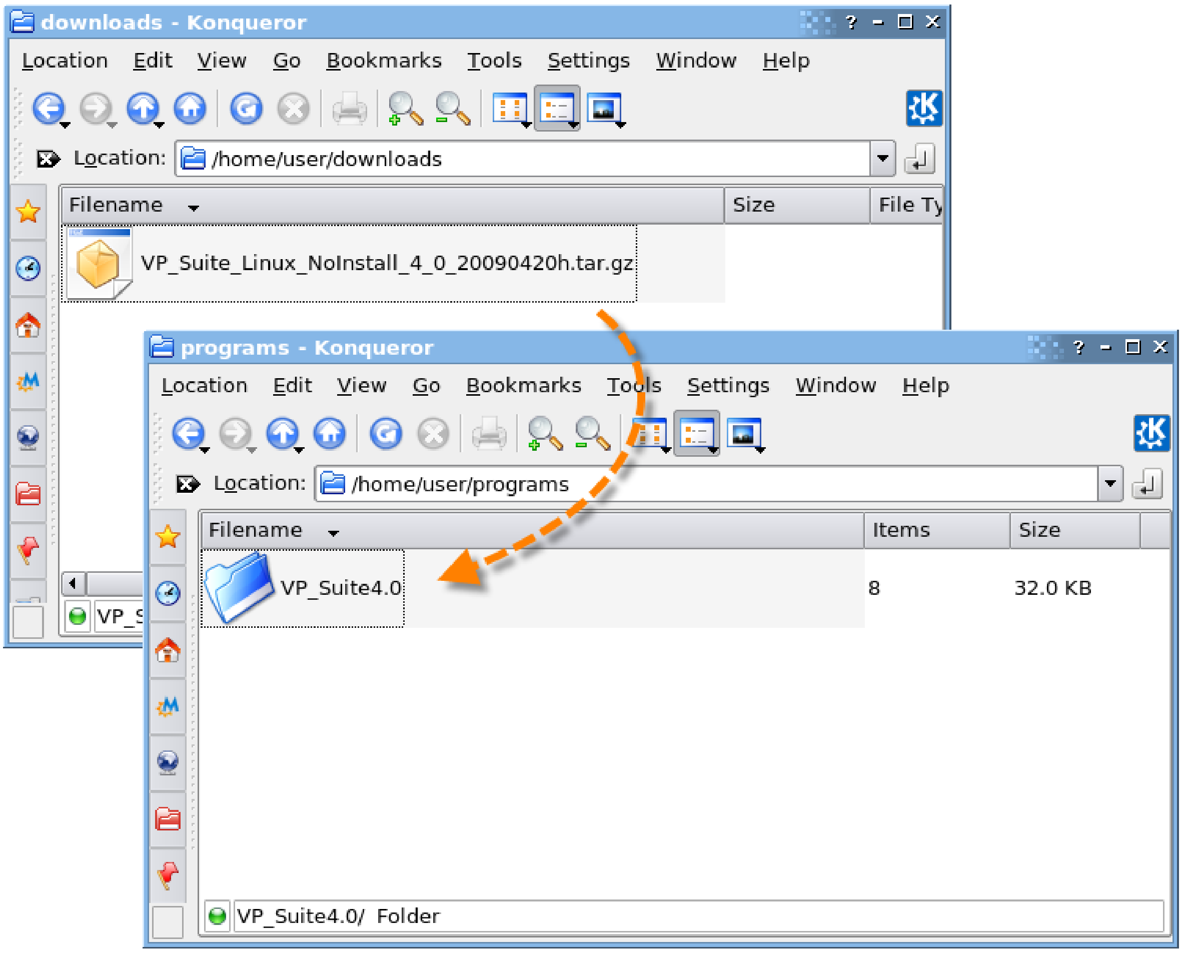
● Visual Paradigm for UML 7.0

○ Don't start

Figure 2-47 *Installation completed screen*

Using No Install Version (.tar.gz)

1. Decompress the downloaded zip file into a directory.
tar -zxf %NO-INSTALL-FILE.tar.gz% -C %DESTINATION-FOLDER%
This should create a subdirectory named &ldquo;VP Suite 4.0&rdquo; where 4.0 is the version number.


downloads - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Filename
Size
File Ty
programs - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Filename
Items
Size

Location: /home/user/downloads

VP_Suite_Linux_NoInstall_4_0_20090420h.tar.gz

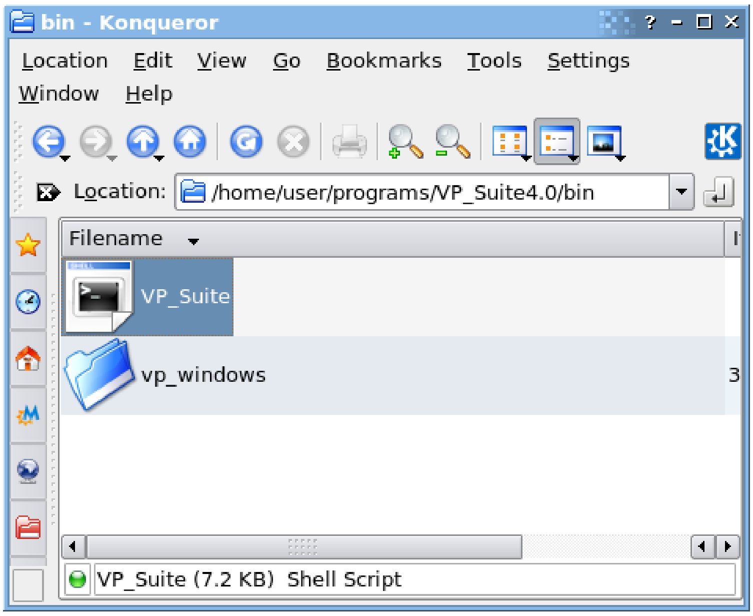
Location: /home/user/programs

Filename	Items	Size
VP_Suite4.0	8	32.0 KB

VP_Suite4.0/ Folder

Figure 2-48 *Extracting No-Install zip file*

2. Change directory to &ldquo;VP Suite 4.0/bin&rdquo; and execute VP Suite in it.



Location: /home/user/programs/VP_Suite4.0/bin

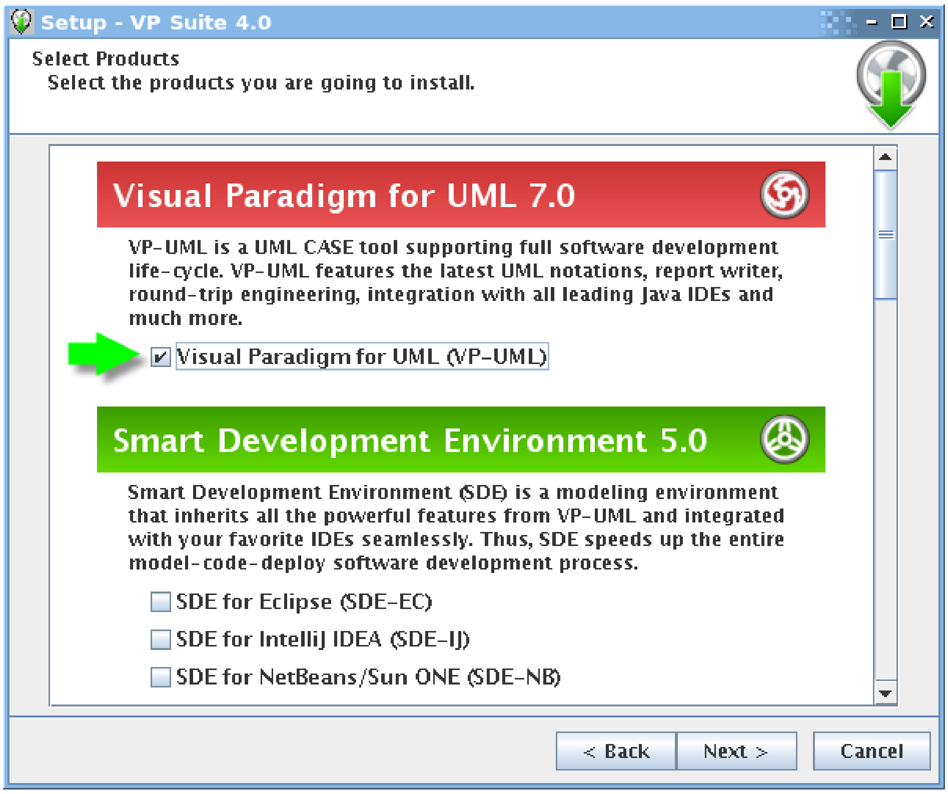
VP_Suite

vp_windows

VP_Suite (7.2 KB) Shell Script

Figure 2-49 *Launching VP Suite*

3. Select **Visual Paradigm for UML**. Click **Next >** to proceed to the product configuration page.



Select Products
Select the products you are going to install.

Visual Paradigm for UML 7.0

VP-UML is a UML CASE tool supporting full software development life-cycle. VP-UML features the latest UML notations, report writer, round-trip engineering, integration with all leading Java IDEs and much more.

☑ Visual Paradigm for UML (VP-UML)

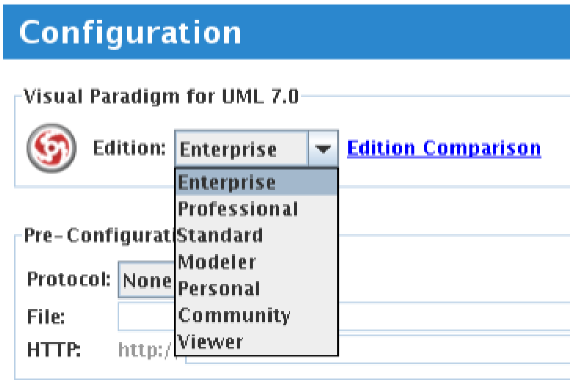
Smart Development Environment 5.0

Smart Development Environment (SDE) is a modeling environment that inherits all the powerful features from VP-UML and integrated with your favorite IDEs seamlessly. Thus, SDE speeds up the entire model-code-deploy software development process.

□ SDE for Eclipse (SDE-EC)
□ SDE for IntelliJ IDEA (SDE-IJ)
□ SDE for NetBeans/Sun ONE (SDE-NB)

Figure 2-50 *The product selection page*

4. VP-UML features vary by product edition. For more details on the features supported by different editions, click the hyperlink **Edition Comparison** which brings you to the web page of VP-UML feature comparison for different editions.



Configuration

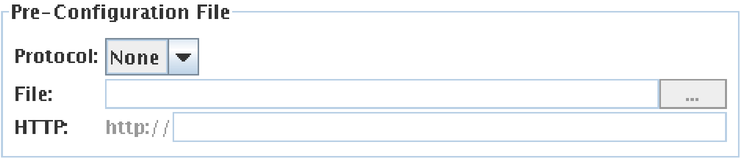
Visual Paradigm for UML 7.0

Edition: Enterprise

Figure 2-51 *Select product edition*

5. For users who are going to run VP-UML with floating license, the Pre-Configuration File section enables you to configure the connection(s) to floating license server by providing a configuration file. The pre configuration is optional. If you do not define the connection here, you will need to do so when starting VP-UML the first time. Below is the pre-configuration file file content:

```
?xml version="1.0" encoding="UTF-8"?>
<LicenseServer>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
<Server accessCode="" host="" port=""/>
</LicenseServer>
```

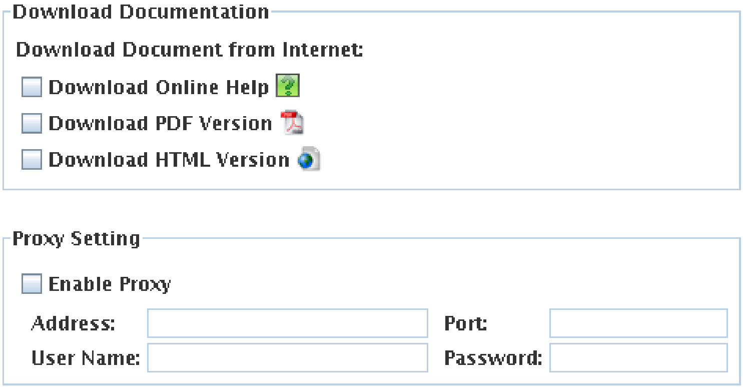


Pre-Configuration File

Protocol: None

Figure 2-52 *Pre-configure floating license server connection for floating license user*

6. Select **Download Online Help** if you want to be able to access the Help contents from within the tool. Select **Download PDF/HTML Version** if you wish to read the documentation in this two types of formats. Press **Next >** to proceed to the license import page.



Download Documentation

Download Document from Internet:

□ Download Online Help

□ Download PDF Version

□ Download HTML Version

Proxy Setting

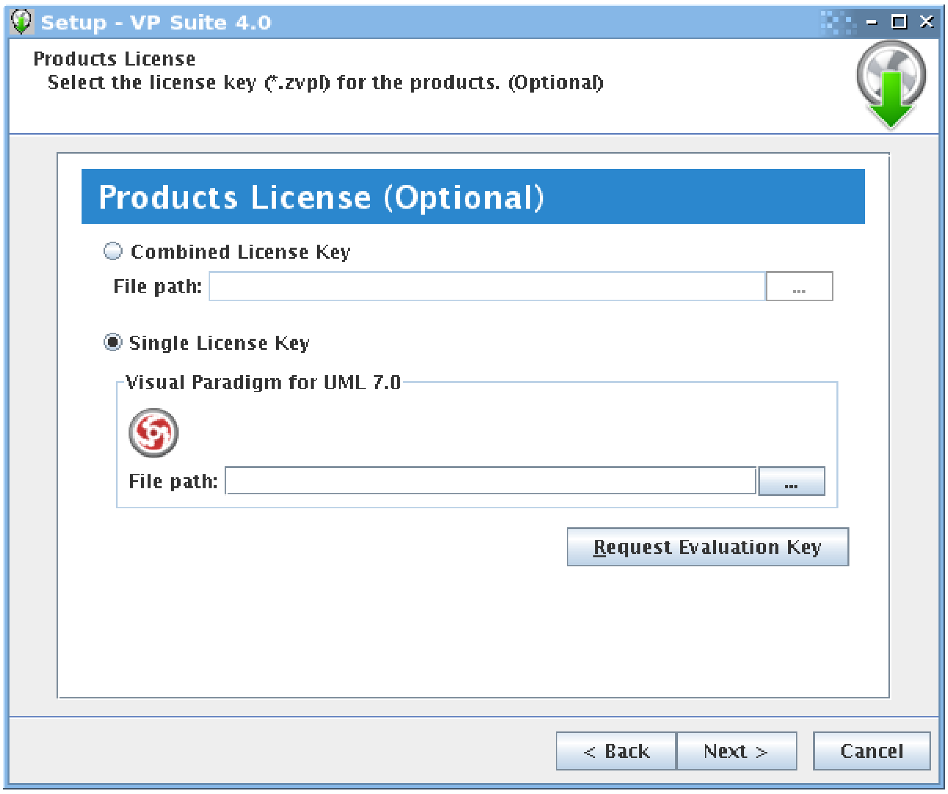
□ Enable Proxy

Address: Port:

User Name: Password:

Figure 2-53 Download documentation

7. A combined license key is a key file which allows unlocking multiple products. To import a combined license key, select Combined License Key and specify the file path of the key.
A single license key is a key file which allows unlocking only one product. To import a single license key, select Single License Key and specify the file path of the key.
The import of license key is optional at this point, because you can import it when you run VP-UML. Click **Next >** to proceed with copying files.



Setup - VP Suite 4.0

Products License
Select the license key (*.zvpl) for the products. (Optional)

Products License (Optional)

○ Combined License Key

File path:

● Single License Key

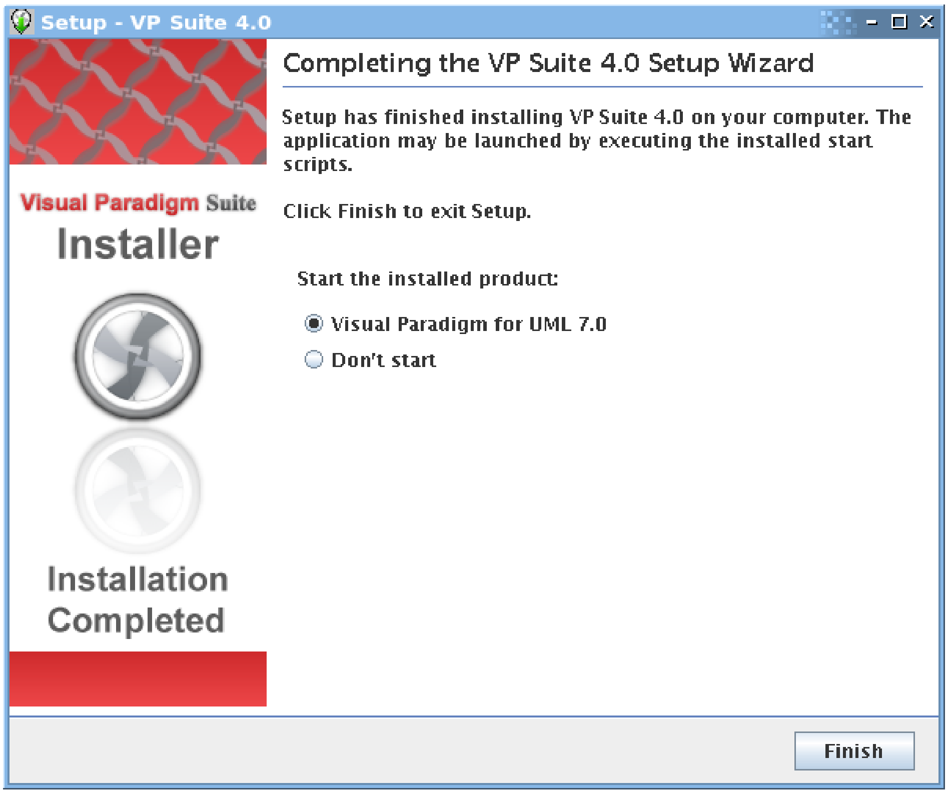
Visual Paradigm for UML 7.0

File path:

Figure 2-54 The license key selection page

NOTE: For evaluation, the key should be sent to your email box (unless you have chosen not to receive it before download). If you have not received the email with key yet, and if you have selected to receive the key as attachment, the email might be treated as spam by mistake. Click on **Request Evaluation Key** to ask for another one. This time, try not to select sending the key as attachment.

8. Upon finishing, you can select whether to start VP-UML or not. Keep **Visual Paradigm for UML** selected and click **Finish** will run VP-UML right away.



Completing the VP Suite 4.0 Setup Wizard

Setup has finished installing VP Suite 4.0 on your computer. The application may be launched by executing the installed start scripts.

Click Finish to exit Setup.

Start the installed product:

● Visual Paradigm for UML 7.0

○ Don't start

Figure 2-55 *Installation completed screen*

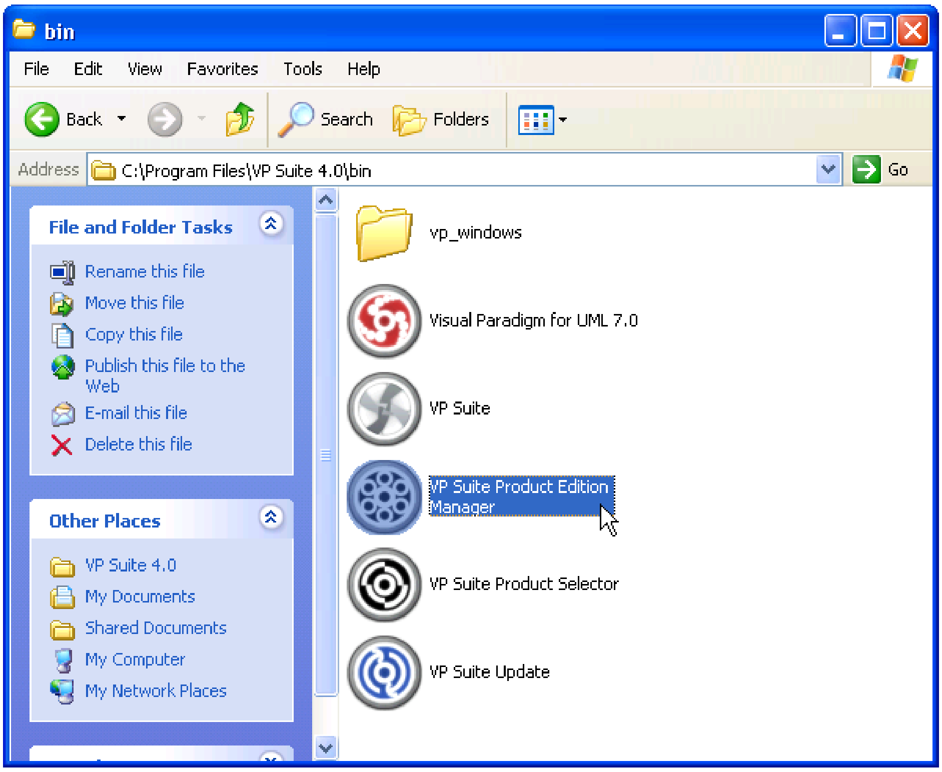
Installation FAQ

Question: What is the difference between Installer and &ldquo;Not Install Version&rdquo;?
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 vp.log file, which can be found at the bin folder of VP Suite installation directory, for our team to diagnose in further.
Question: I don't have administrator right, can I install the software?
Answer: Yes, you can.
Question: Can I change the Edition without re-install the software?
Answer: Yes, you can. Product edition can be changed by running VP Suite Product Edition Manager under the bin folder of VP Suite installation directory. Change of edition takes effect after the restart of affected products. For more details, please read the section Switching Edition a few pages later.
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.

Switching Edition

Product edition can be changed through the Product Edition Manager without the need of re-installation. When you attempt to switch to another edition, make sure you have the required key in order to run the product in new edition.

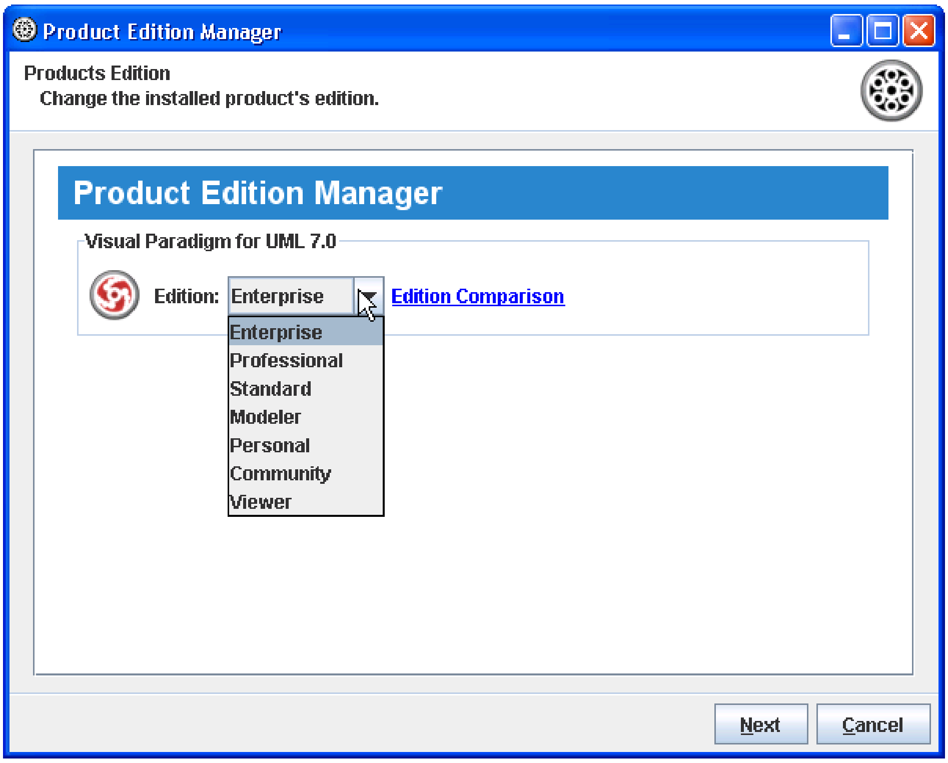
1. Launch the Product Edition Manager under the bin folder of VP Suite installation folder.



Address: C:\Program Files\VP Suite 4.0\bin

__Figure 2-56__ Launching Product Edition Manager

2. Select the edition to switch to in the edition manager.



Product Edition Manager

Products Edition

Change the installed product's edition.

Product Edition Manager

Visual Paradigm for UML 7.0

Edition: Enterprise

__Figure 2-57__ Select product edition

3. Click **Next** to confirm.

Uninstalling Visual Paradigm for UML

Uninstalling VP-UML will cause VP-UML to be removed from VP-Suite.

1. Close VP-UML if it is running.
2. Execute uninstall_vpuml under the uninstaller folder of VP Suite installation folder.

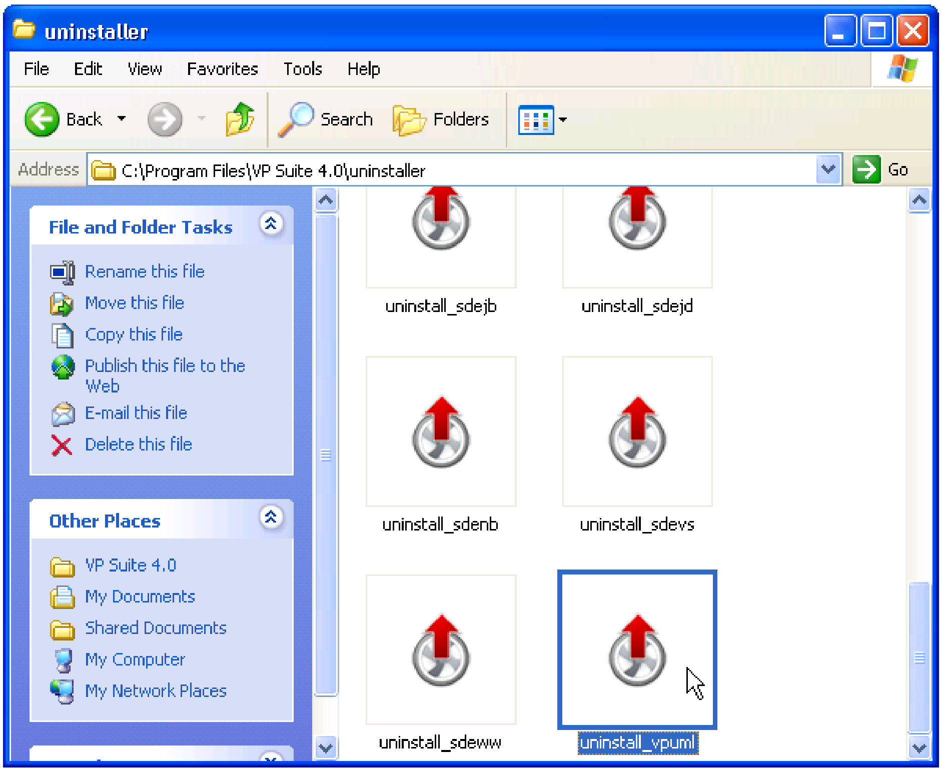


Figure 2-58 *Uninstalling VP-UML*

Starting Visual Paradigm for UML

Ways of Starting Visual Paradigm for UML

VP-UML can be launched through accessing the Start Menu.

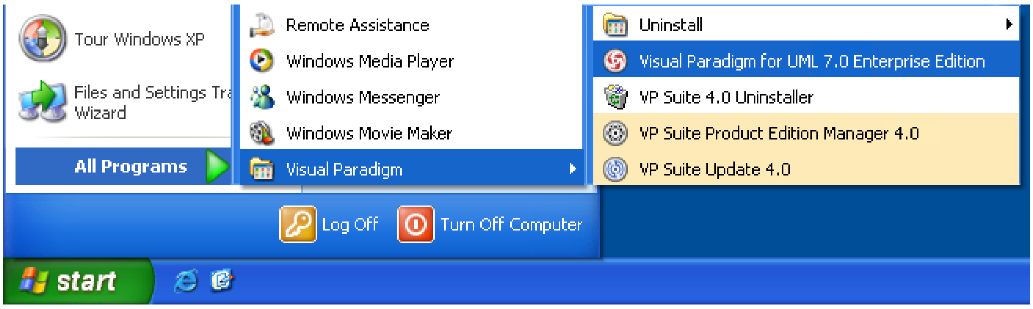


Figure 3-1 *Starting Visual Paradigm for UML via Start Menu*

Alternatively, you can start VP-UML by executing the launcher **run_vpuml** in the launcher folder of VP Suite installation directory. Users in all operating system can start VP-UML in this way.

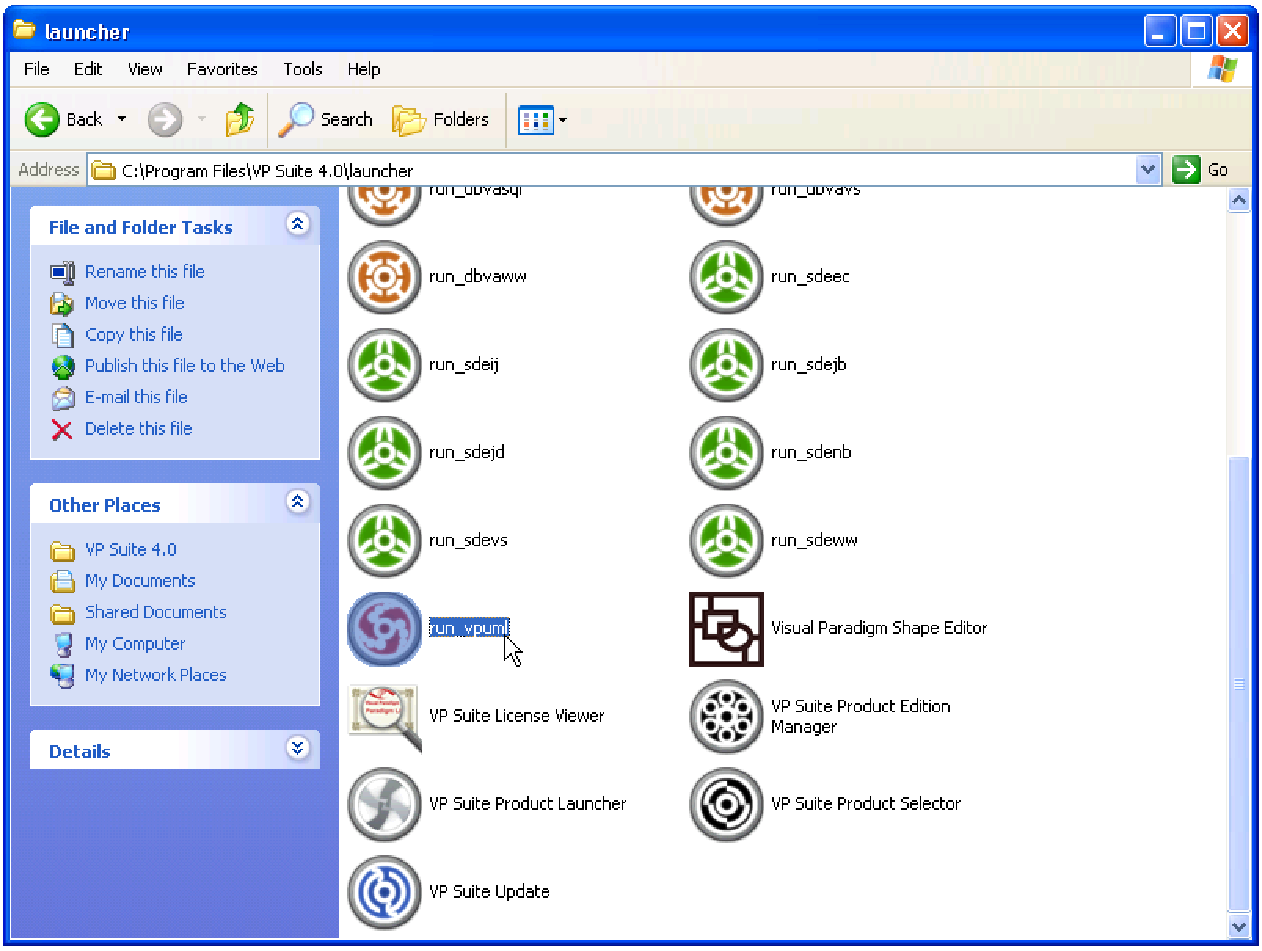


Figure 3-2 *Starting Visual Paradigm for UML with launcher*

For Linux users, VP-UML can be started through the shortcuts in desktop, created by the installer.

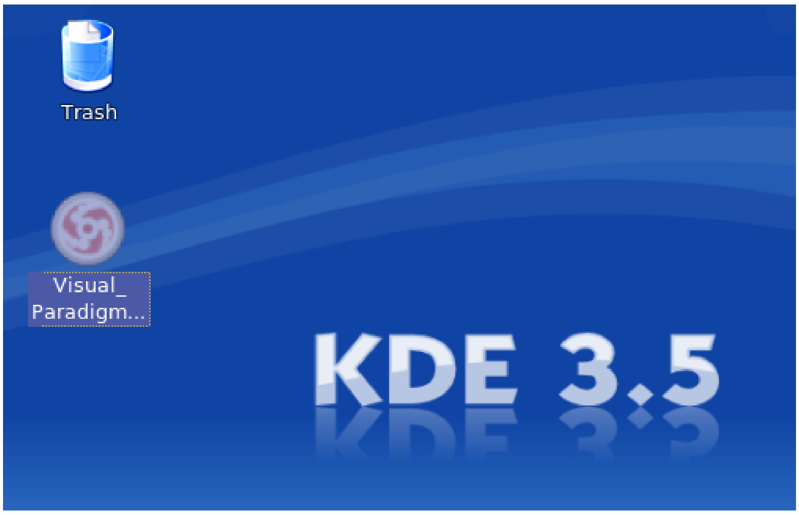
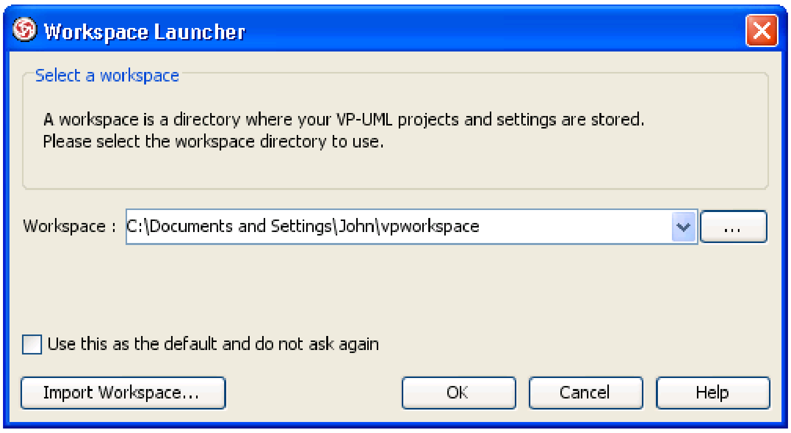


Figure 3-3 *Starting Visual Paradigm for UML in Linux using the icon in desktop*

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 > Options...** in VP-UML). 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 VP-UML. All your teamwork information and settings will then be transferred to the new computer.

The Workspace Launcher appear when running VP-UML.


Workspace Launcher

Select a workspace

A workspace is a directory where your VP-UML projects and settings are stored.
Please select the workspace directory to use.

Workspace : C:\Documents and Settings\John\vpworkspace

□ Use this as the default and do not ask again

Figure 3-4 *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 VP-UML 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.

Interface Overview

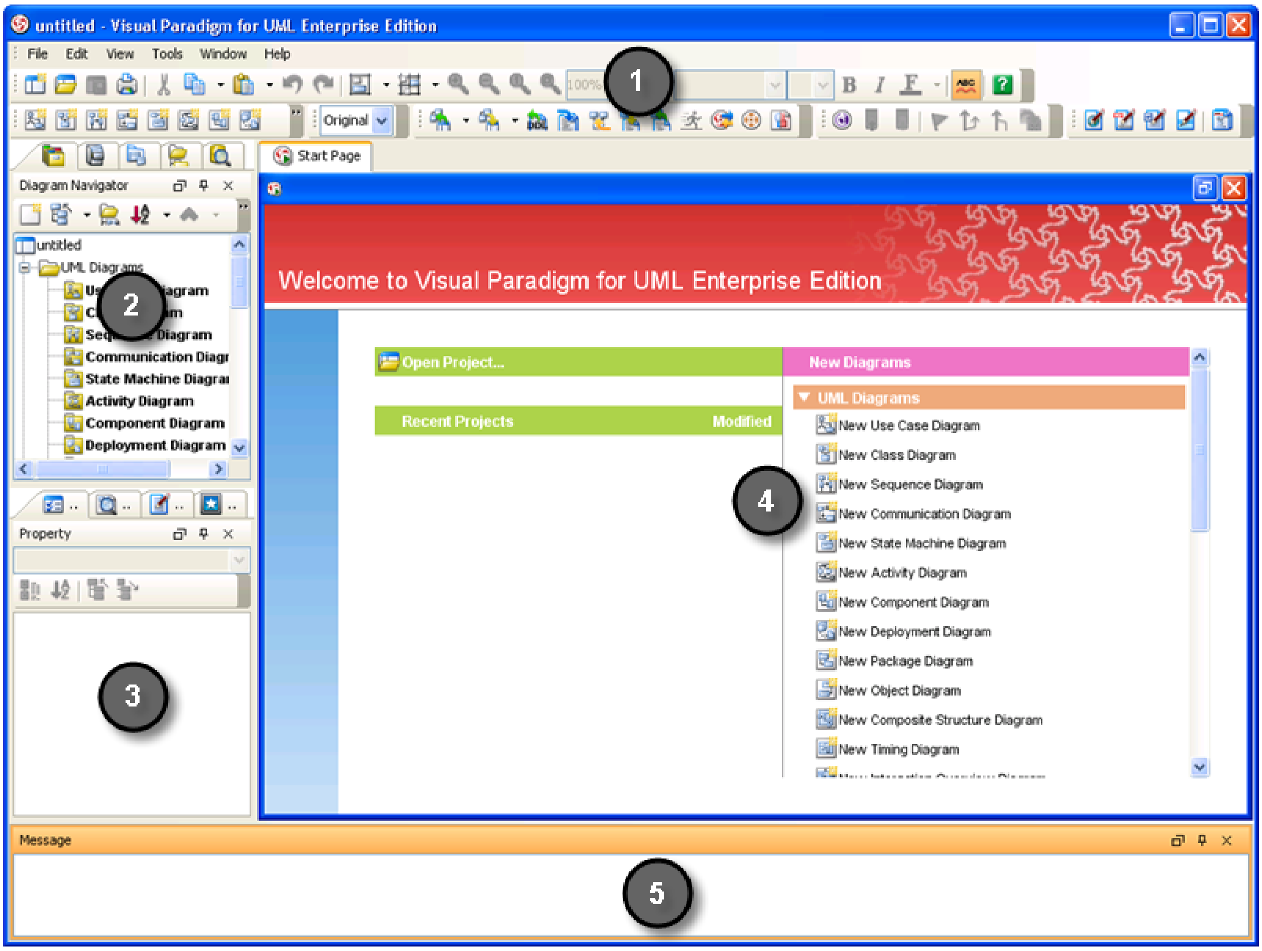


Figure 3-5 *User Interface of Visual Paradigm for UML*

No.	Name	Description
1	Menu and Toolbar	Commands for execution
2	Project Content Explorers	Various explorers are placed here for accessing project data from different angle
3	Various Auxiliary Panes	Auxiliary panes like property pane, preview pane, documentation pane and stencil pane are shown here
4	Diagram Pane	The place where a diagram is shown and edited
5	Message Pane	Display messages that are generated by certain operations

Main Menu

File Menu

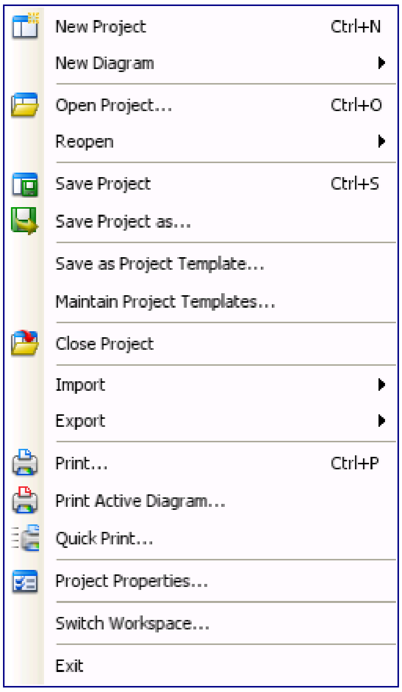


Figure 3-6 The File menu

The **File** menu enables you to:

- Create a project
- Create a diagram
- Open project
- Save project
- Save and manage project template.
- Import project data from the following media: VP-UML Project, Rose Project, XMI, XML, Erwin Project, Telelogic Phapsody Project, Telelogic System Architect, Rational Model, Rational DNX, MS Word (Use Case Model documentation exported from VP), Excel (Exported from VP), Visio, NetBeans
- Export project into the following formats: VP-UML Project, XMI, XML, MS Word (for Use Case Model), Excel
- Export images (JPG, PNG, SVG, EMF, PDF)
- Printing
- Set project properties
- Exit

Edit Menu

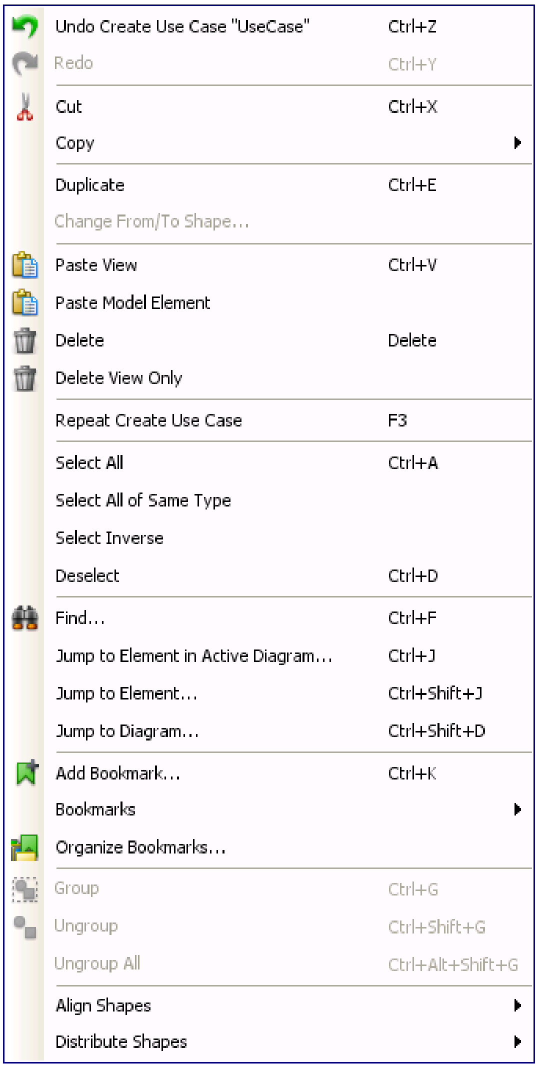


Figure 3-7 *The Edit menu*

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

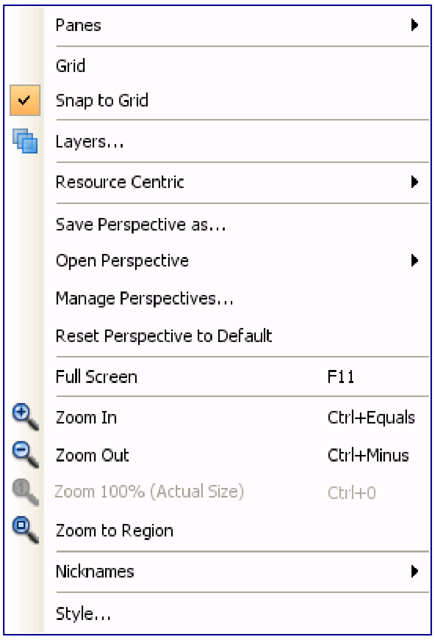


Figure 3-8 *The View menu*

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 VP-UML to show in full screen
- Zoom diagram in and out
- Manage nickname
- Manage style

Tools Menu

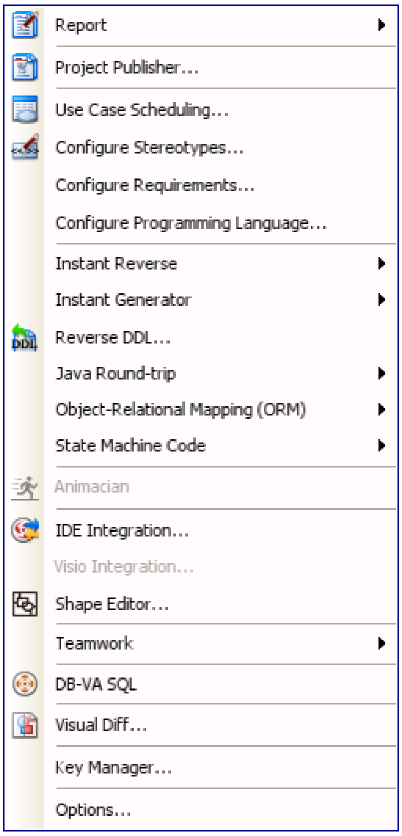


Figure 3-9 *The Tools menu*

The **Tools** menu enables you to:

- Generate report
- Publish project
- Perform use case scheduling
- Configure stereotypes
- Configure requirements
- Configure programming language
- Reverse and Forward engineering with Instant Reverse and Instant Generator
- Perform round-trip engineering
- Reverse DDL
- Perform Object Relational Mapping (ORM)
- Perform State Machine Code Generation
- Launch Animacian
- Perform IDE integration
- Perform Visio integration
- Launch Shape Editor
- Perform Teamwork operations
- Launch DB-VA SQL
- Compare diagrams with Visual Diff
- Launch License Key Manager
- Configure application options through the Options dialog box

Window Menu

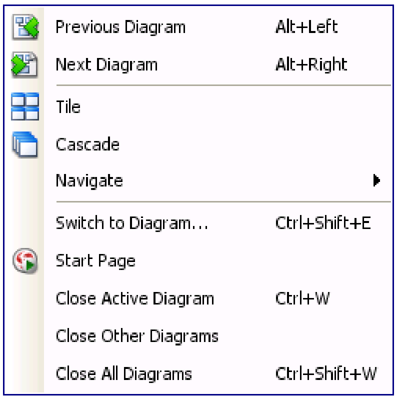

Previous Diagram Alt+Left
Next Diagram Alt+Right
Tile
Cascade
Navigate
Switch to Diagram... Ctrl+Shift+E
Start Page
Close Active Diagram Ctrl+W
Close Other Diagrams
Close All Diagrams Ctrl+Shift+W

Figure 3-10 *The Window menu*

The **Window** menu enables you to:

- Navigate between diagram
- Rearrange diagram windows
- Switch to another diagram
- Show the Start Page
- Close Diagrams

Help Menu

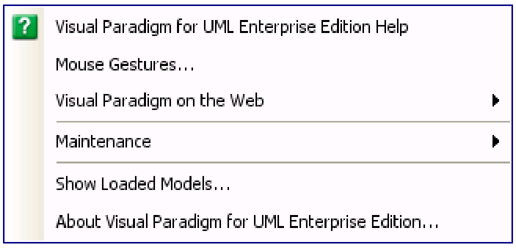

Visual Paradigm for UML Enterprise Edition Help
Mouse Gestures...
Visual Paradigm on the Web
Maintenance
Show Loaded Models...
About Visual Paradigm for UML Enterprise Edition...

Figure 3-11 *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
- Check the environment using the About dialog box

Toolbar

Show/Hide Toolbar(s)

A toolbar can be shown or hidden. To show a toolbar, right-click on any toolbar, and select the toolbar to show. Similary, you can uncheck a toolbar to hide it.

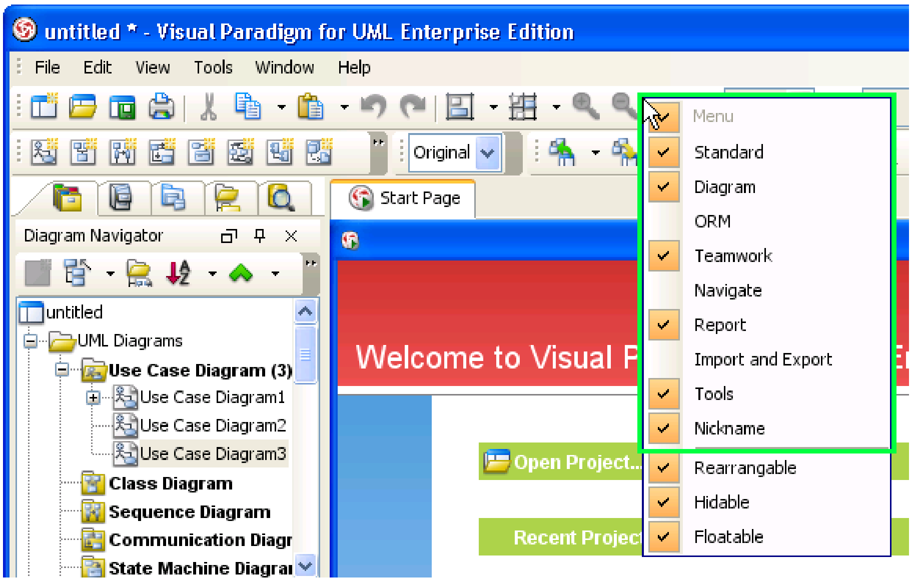


Figure 3-12 Show or hide toolbar(s) by unchecking toolbar(s) from toolbars' popup menu

Lock Toolbars

A toolbar can be repositioned by pressing on the left hand side of toolbar...

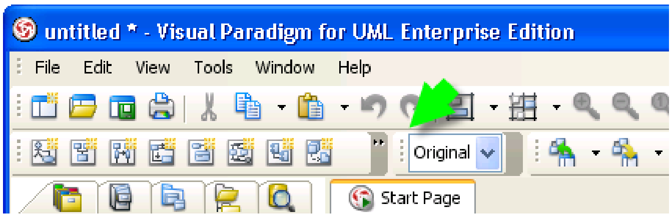


Figure 3-13 Drag toolbar by pressing and moving the left of toolbar

and dragging around.

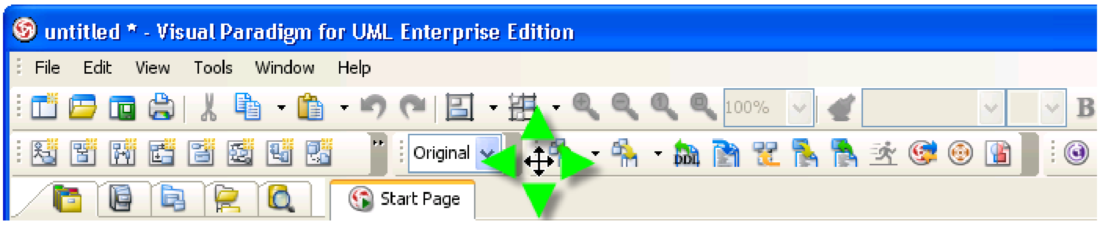


Figure 3-14 Freely movable toolbar

To lock the toolbars' position, right click on any toolbar and uncheck Rearrangable from the popup menu.

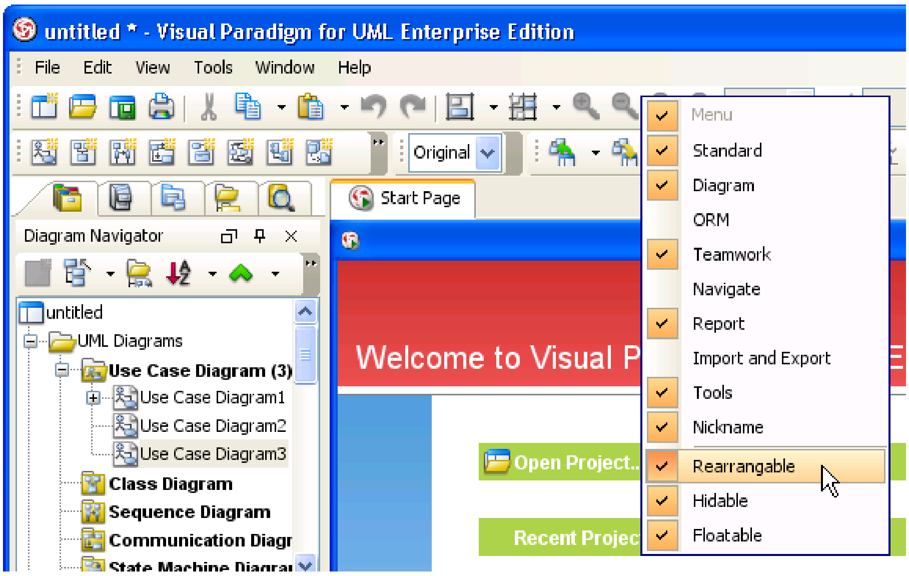


Figure 3-15 *Make toolbars not movable by deselecting the Rearrangable option in toolbars popup menu*

Standard Toolbar

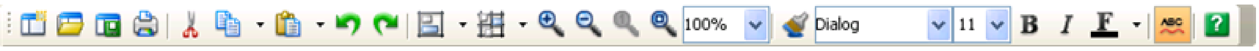


Figure 3-16 *The Standard toolbar*

Icon	Name	Description
	New Project	Create a project
	Open Project	Open a project
	Save Project	Save the changes made in the opening project
	Print	Print diagram(s)
	Cut	Cut selected diagram elements
	Copy within VP-UML	Copy selected diagram elements ready to be used within VP-UML
	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 elements
	Undo	Roll back undesired chagnes
	Redo	Rerun an undone task
	Align Shapes	Open the Align Shapes dialog box for aligning shapes
	Align Top	Align selected shapes to the top of the range of selected shapes
	Align Bottom	Align selected shapes to the bottom of the range of selected shapes

	Align Left	Align selected shapes to the left of the range of selected shapes
	Align Right	Align selected shapes to the right of the range of selected shapes
	Align Horizontal Center	Align selected shapes to the center of the range of selected shapes
	Align Vertical Center	Align selected shapes to the vertical center of the range of selected shapes
	Same Width	Make selected shapes in same width, following the last selected shape
	Same Height	Make selected shapes in same height, following the last selected shape
	Same Width and Height	Make selected shapes in same width and height, following the last selected shape
	Distribute Shapes	Open the Distribute Shapes dialog box for distributing shapes
	Distribute Shapes Horizontally	Distribute selected shapes horizontally
	Distribute Shapes Vertically	Distribute selected shapes vertically
	Distribute Shapes by Left Edges	Distribute selected shapes by their left edges
	Distribute Shapes by Horizontal Centers	Distribute selected shapes by their horizontal centers
	Distribute Shapes by Right Edges	Distribute selected shapes by their right edges
	Distribute Shapes by Top Edges	Distribute selected shapes by their top edges
	Distribute Shapes by Vertical Centers	Distribute selected shapes by their vertical centers
	Distribute Shapes by Bottom Edges	Distribute selected shapes by their bottom edges
	Zoom In	Magnify the opening diagram
	Zoom Out	Diminish the opening diagram
	Zoom to 100% (Actual Size)	Reset zoom ratio back to 100%
	Zoom to Region	Zoom by selecting the range to view on diagram
100% ▾	Zoom Ratio	Current zoom ratio. You can change it by selecting a ratio from the drop down menu or by typing a new one.
	Format Copier	Copy selected diagram element's format and apply on another shape
Dialog ▾	Font Name	Name of font being used by the selected diagram element
11 ▾	Font Size	Font size of the selected diagram element
B	Bold	Set the text of selected diagram element bold or not bold
I	Italic	Set the text of selected diagram element italic or not italic
F	Font Color	Select the font color of selected diagram element
ABC	Auto Spell Check	Enable or disable automatic spell checking
	Help	Launch the Help Contents

Diagram Toolbar

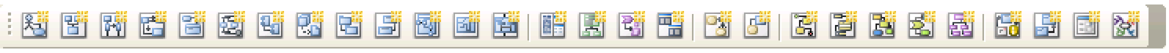


Figure 3-17 *The Diagram toolbar*

Icon	Name	Description
	New Use Case Diagram	Create a Use Case Diagram
	New Class Diagram	Create a Class Diagram
	New Sequence Diagram	Create a Sequence Diagram
	New Communication Diagram	Create a Communication Diagram
	New State Machine Diagram	Create a State Machine Diagram
	New Activity Diagram	Create an Activity Diagram
	New Component Diagram	Create a Component Diagram
	New Deployment Diagram	Create a Deployment Diagram
	New Package Diagram	Create a Package Diagram
	New Object Diagram	Create an Object Diagram
	New Composite Structure Diagram	Create a Composite Structure Diagram
	New Timing Diagram	Create a Timing Diagram
	New Interaction Overview Diagram	Create an Interaction Overview Diagram
	New Textual Analysis	Create a Textual Analysis
	New Requirement Diagram	Create a Requirement Diagram
	New Basic Diagram	Create a Basic Diagram
	New CRC Card Diagram	Create a CRC Card Diagram
	New Entity Relationship Diagram	Create an Entity Relationship Diagram
	New ORM Diagram	Create an ORM Diagram
	New Business Process Diagram	Create a Business Process Diagram
	New Data Flow Diagram	Create a Data Flow Diagram
	New EPC Diagram	Create a EPC Diagram
	New Process Map Diagram	Create a Process Map Diagram
	New Organization Chart	Create an Organization Chart
	New EJB Diagram	Create a EJB Diagram
	New Overview Diagram	Create an Overview Diagram
	New User Interface	Create a User Interface

	New Mind Mapping Diagram	Create a Mind Mapping Diagram

ORM Toolbar

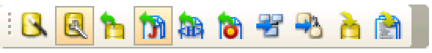


Figure 3-18 The ORM toolbar

Icon	Name	Description
	ORM Wizards	Open the ORM Wizards
	Database Configuration	Open the Database Configuration dialog box to configure database connections
	Reverse Database	Reverse engineering from database
	Reverse Java Classes	Reverse engineering from Java Class
	Reverse Hibernate	Reverse engineering from Hibernate mapping file
	Reverse Enterprise Object Framework	Reverse engineering from Enterprise Object Framework
	Synchronize to Class Diagram	Synchronize from Entity Relationship Diagram to Class Diagram
	Synchronize to Entity Relationship Diagram	Synchronize from Class Diagram to Entity Relationship Diagram
	Generate Database	Open the Database Code Generation dialog box to generate database
	Generate ORM Code	Open the Database Code Generation dialog box to generate code

Teamwork Toolbar

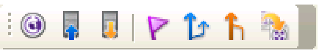


Figure 3-19 The Teamwork toolbar

Icon	Name	Description
	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

Navigate Toolbar

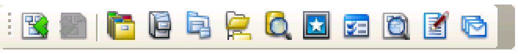


Figure 3-20 The Navigate toolbar

Icon	Name	Description
	Previous Diagram	Navigate to the previous diagram

	Next Diagram	Navigate to the next diagram
	Diagram Navigator	Open the Diagram Navigator pane
	Model Explorer	Open the Model Explorer pane
	Class Repository	Open the Class Repository pane
	Logical View	Open the Logical View pane
	ORM	Open the ORM pane
	Stencil	Open the Stencil pane
	Property	Open the Property pane
	Diagram Overview	Open the Diagram Overview pane
	Documentation	Open the Documentation pane
	Documentation	Open the Documentation pane

Report Toolbar

Figure 3-21 *The Report toolbar*

Icon	Name	Description
	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
	Project Publisher	Publish Project to Web pages through Project Publisher

Import and Export Toolbar

Figure 3-22 *The Import and Export toolbar*

Icon	Name	Description
	Import VP-UML Project	Import a VP-UML project into the opening project
	Import Rose	Import diagrams and model elements from a Rational Rose model
	Import XMI	Import diagrams and model elements from XMI
	Import XML	Import diagrams and model elements from XML
	Import MS Word to Use Case Model Element	Import Use Case Report (MS Word) back into the opening project
	Import Erwin Project (XML)	Import diagrams and model elements from Erwin Project

	Import Telelogic Rhapsody Project	Import diagram and model elements from Telelogic Rhapsody Project
	Import Telelogic System Architect	Import diagram and model elements from Telelogic System Architect
	Import Rational Model	Import diagram and model elements from Rational Model
	Import Rational DNX	Import diagram and model elements from Rational DNX
	Import Excel	Import Excel file back into the opening project
	Import Visio	Import Visio diagrams into VP-UML
	Import NetBeans UML Project	Import NetBeans UML diagrams into VP-UML
	Export VP-UML Project	Export the open project into a VP-UML project file
	Export XMI	Export XMI from the opening project
	Export XML	Export XML from the opening project
	Export Active Diagram as Image	Export active diagram as image file
	Export Diagram as Image	Export any diagram(s) as image file(s)
	Export Selection as Image	Export selection on active diagram as image file
	Export Use Case Model to MS Word	Export Use Case Report (MS Word) from Use Case diagrams and use case models
	Export Active Diagram to Excel	Export active diagram into Excel report
	Export Excel	Export any diagram(s) into Excel report(s)

Tools Toolbar

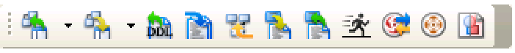


Figure 3-23 *The Tools toolbar*

Icon	Name	Description
	Instant Reverse	Reverse Engineering Class Diagram from many kinds of source code
	Instant Generator	Generate source code from the opening project
	Reverse DDL	Reverse data definition language file and form ERD
	Update to Code	Update source code from model
	Update to Model Element	Update model element from source code
	Generate State Machine Code	Generate state machine code from class diagram and state machine diagram
	Reverse State Machine Code	Reverse state machine code back into the opening project
	Animacian	Launch Animacian for animating the active diagram
	IDE Integration	Integrate with IDE(s) through IDE integration
	DB-VA SQL	Launch DB-VA SQL, a tool for querying data in database

	Visual Diff	Launch Visual Diff, a tool for comparing diagrams

Nickname Toolbar

Original

Figure 3-24 *The Nickname toolbar*

Manage and select the nickname to be applied on the opening project.

Dockable User Interface

VP-UML 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.

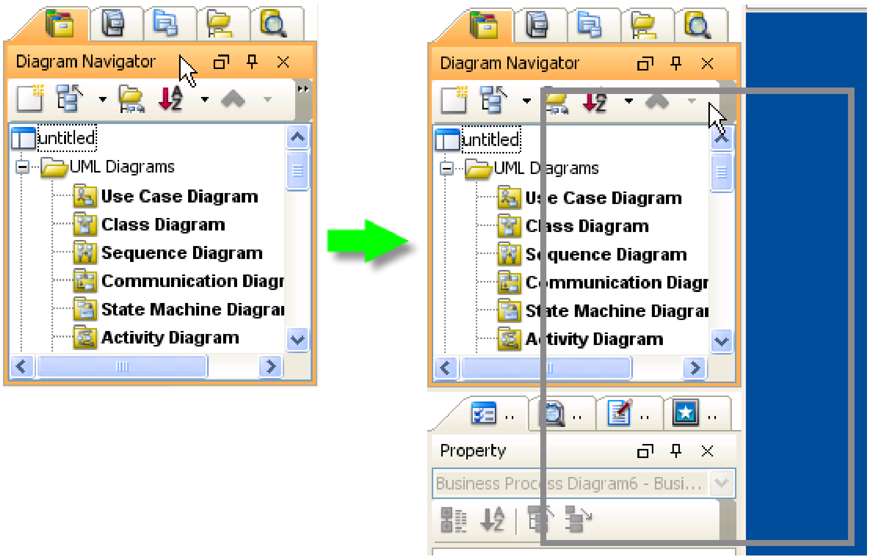


Figure 3-25 *Frame can be dragged out and dock to elsewhere*

You will notice a gray outline appears 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 move the cursor to certain 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.

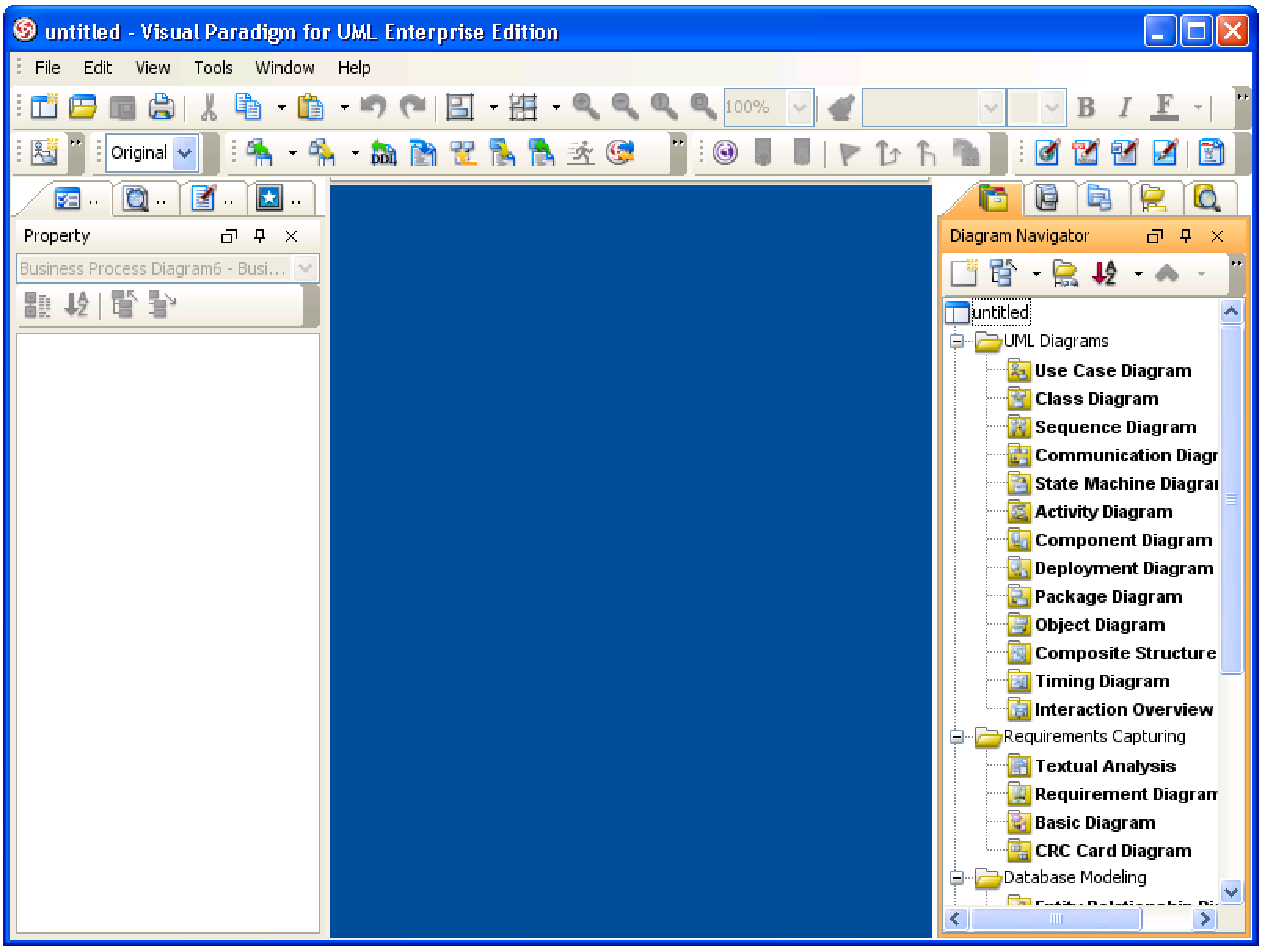


Figure 3-26 *A frame docked to the right of application screen*

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

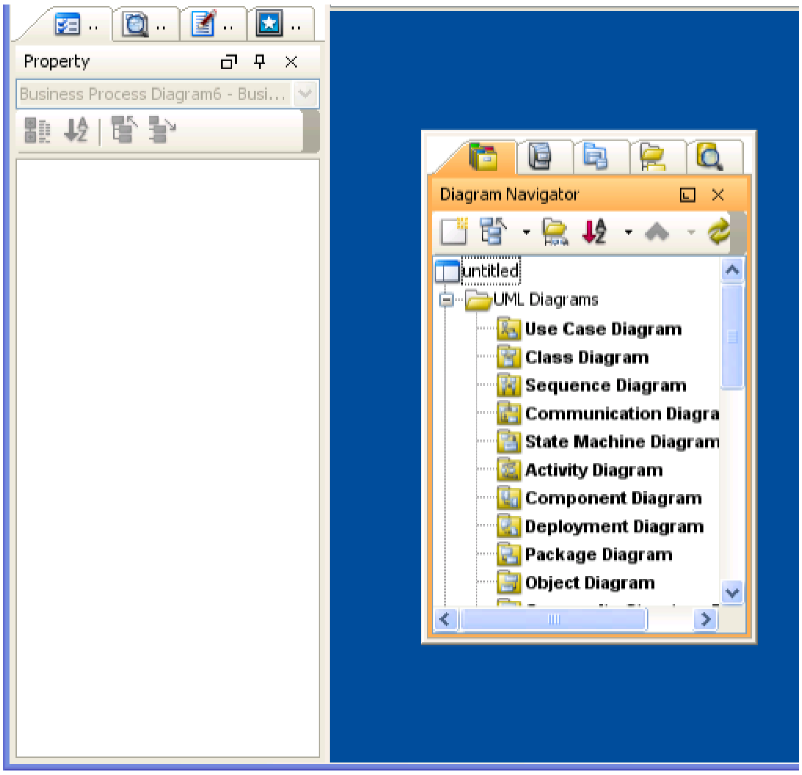


Figure 3-27 *A frame float on top of the application*

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

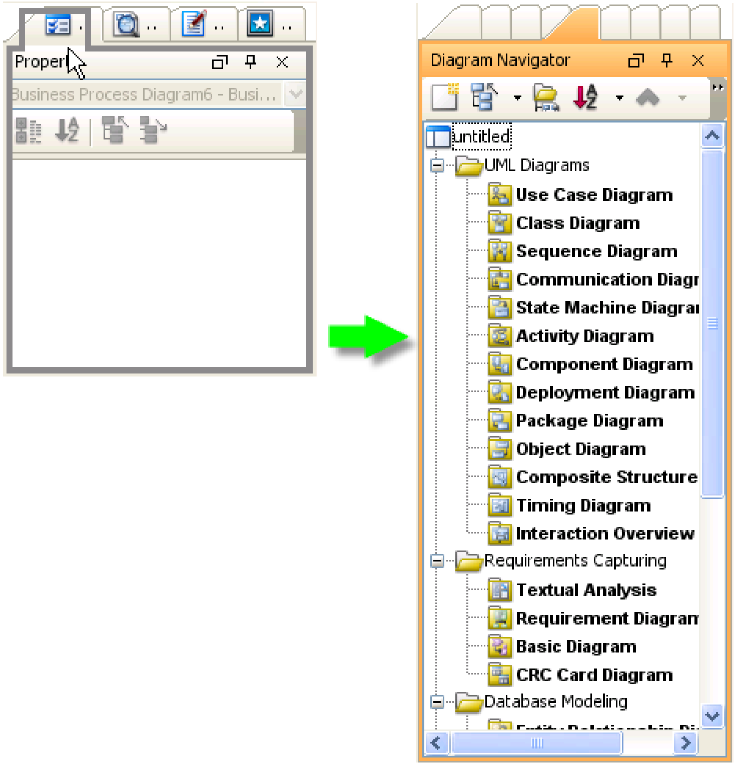


Figure 3-28 *A frame docked to another frame*

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 upper right corner of the frame (the button with a pin as the icon, see figure below)

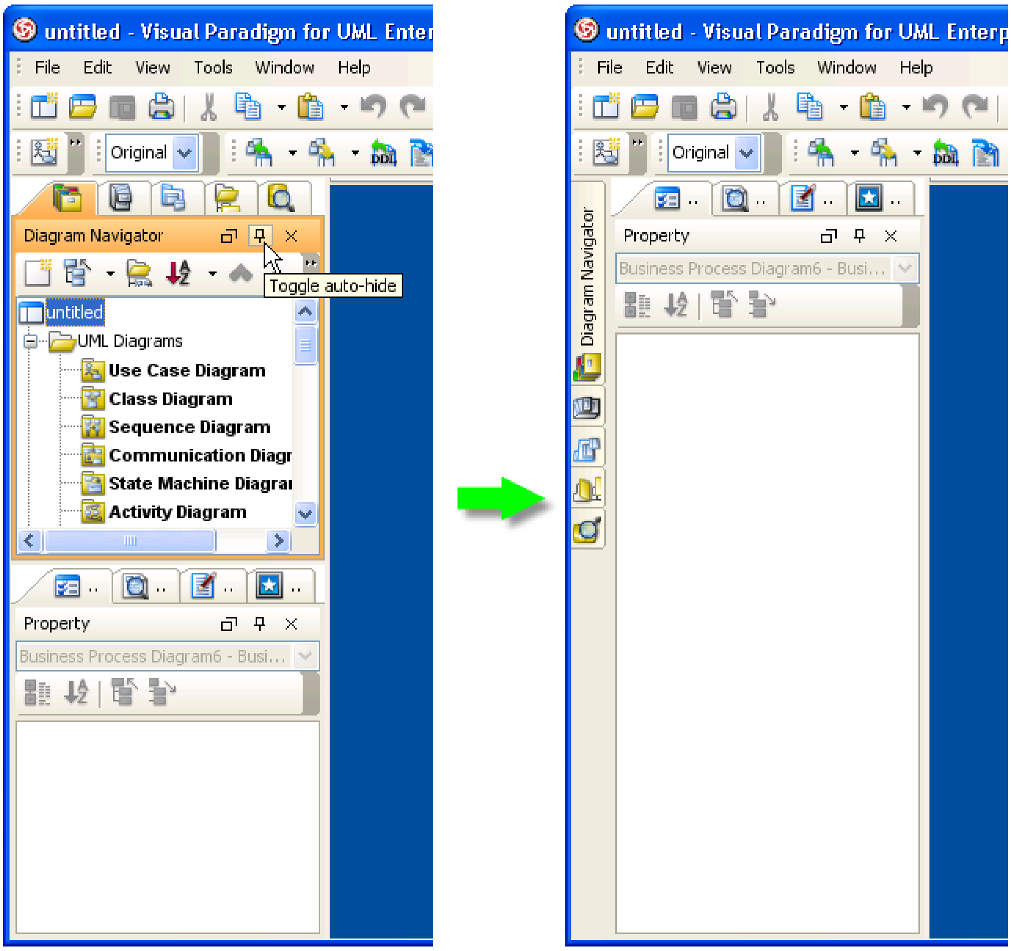

untitled - Visual Paradigm for UML Enter
File Edit View Tools Window Help
Original
Diagram Navigator
Toggle auto-hide
untitled
UML Diagrams
Use Case Diagram
Class Diagram
Sequence Diagram
Communication Diagr
State Machine Diagra
Activity Diagram
Property
Business Process Diagram6 - Busi...
untitled - Visual Paradigm for UML Enterp
File Edit View Tools Window Help
Original
Diagram Navigator
Property
Business Process Diagram6 - Busi...

Figure 3-29 *Automatically hiding Diagram Navigator*

Project Content Explorers

Diagram Navigator

The Diagram Navigator displays diagrams and their diagram elements in the project, categorized by diagram type.

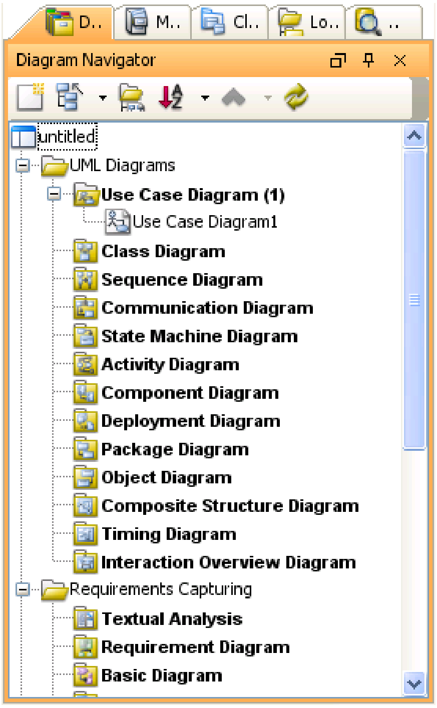


Figure 3-30 The Diagram Navigator

Icon	Name	Description
	New Diagram...	Create a diagram
	Collapse	Collapse all the nodes under the selected tree node
	Expand	Expand all the nodes under the selected tree 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
	Sort Diagram Element By Name	Sort all the nodes under the selected tree node by name
	Sort Diagram Element by Type	Sort all the nodes under the selected tree node by type
	Move Selected Diagram Up	Move the selected diagram node upward
	Move Selected Diagram Down	Move the selected diagram node downward
	Refresh	Refresh Diagram Navigator content

Popup menu of Diagram Navigator

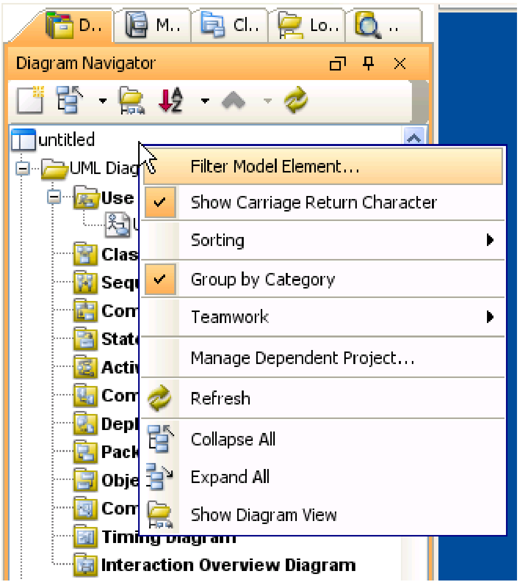


Figure 3-31 The Popup menu of Diagram Navigator

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 Explorer
Group by Category	Categorize diagrams base 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

Popup menu of Project

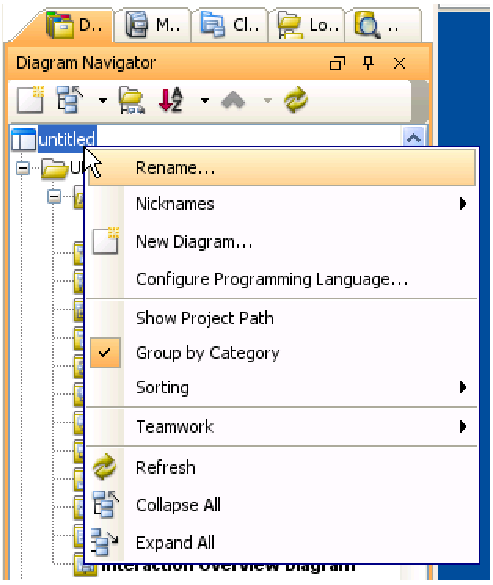


***Figure 3-32** The Popup menu of project node in Diagram Navigator*

Menu Title	Description
Rename...	Rename the project
Nicknames	Configure or switch to another nickname
New Diagram...	Create a diagram
Configure Programing 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

Popup menu of Diagram Category

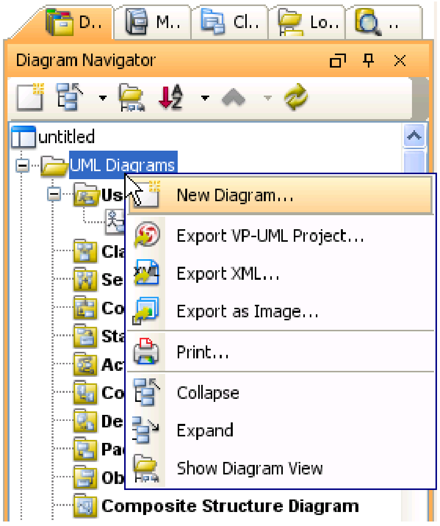


***Figure 3-33** The Popup menu of Diagram Category in Diagram Navigator*

Menu Title	Description
New Diagram...	Create a diagram
Export VP-UML Project...	Export all diagrams in the selected category as VP-UML 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

Popup menu of Diagram Type

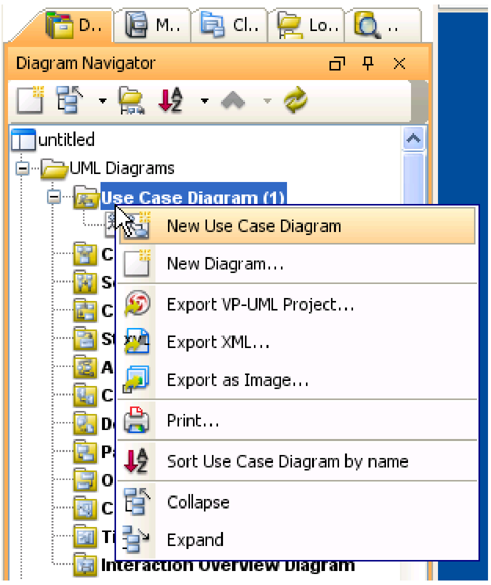


Figure 3-34 The Popup menu of Diagram Type in Diagram Navigator

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 VP-UML Project...	Export all diagrams in the selected type as VP-UML 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

Popup menu of Diagram

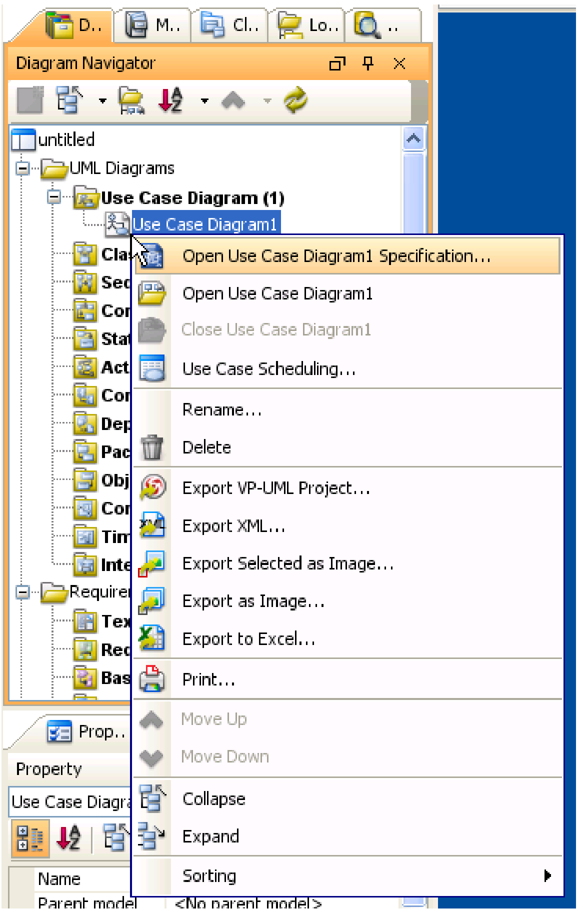


Figure 3-35 *The Popup menu of Diagram Type in Diagram Navigator*

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
Export VP-UML Project...	Export the selected diagram as VP-UML 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

Model Explorer

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

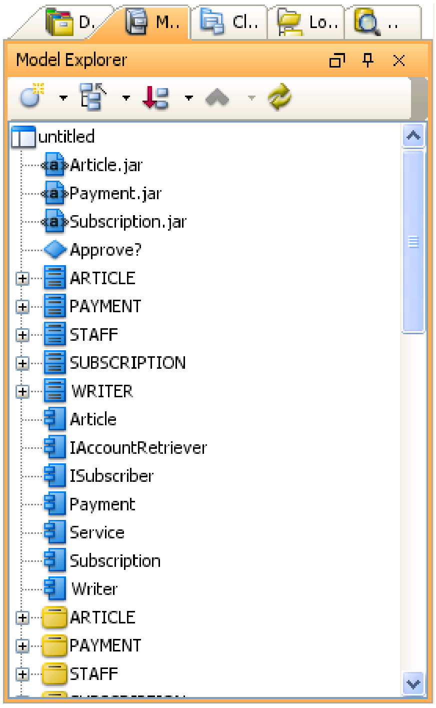


Figure 3-36 The Model Explorer

Icon	Name	Description
	New Model Element...	Create a new Model Element in Model Explorer without the need of creating through diagramming
	Collapse	Collapse all the nodes under the selected tree node
	Expand	Expand all the nodes under the selected tree node
	Sort Model Element By Name	Sort all the nodes under the selected tree node by name
	Sort Model Element by Type	Sort all the nodes under the selected tree node by type
▲	Move Selected Model Element Up	Move the selected model element node upward
▼	Move Selected Model Element Down	Move the selected model element node downward
	Refresh	Refresh Model Explorer content

Popup menu of Model Explorer

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

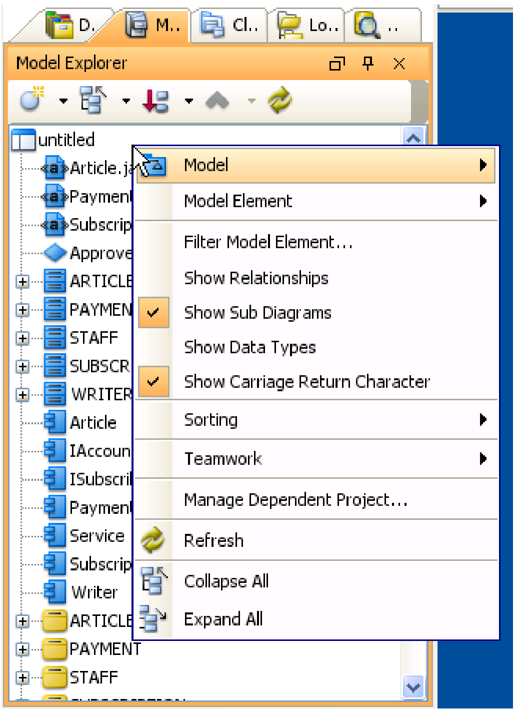


Figure 3-37 The Popup menu of Model Explorer

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

Popup menu of Project

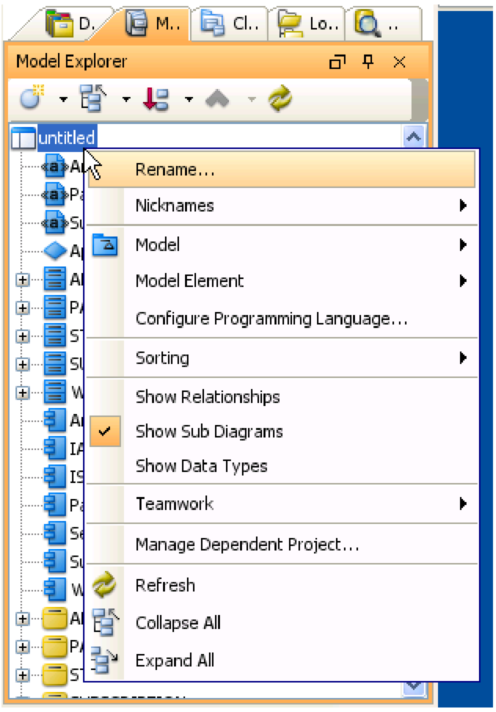


Figure 3-38 *The Popup menu of project node in Model Explorer*

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 Programing 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

Class Repository

The Class Repository lists the packages and class models in the project.

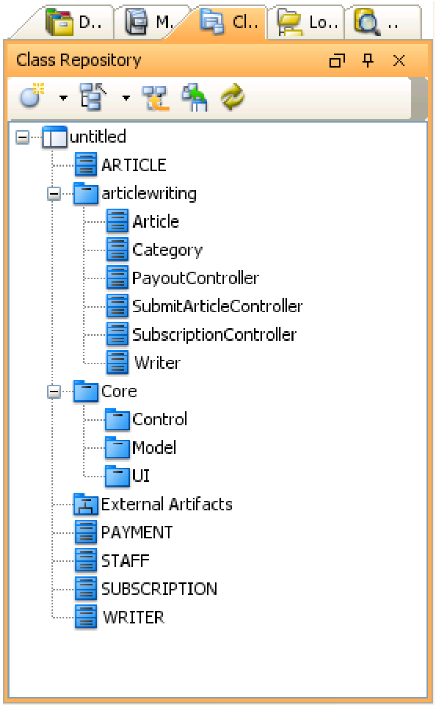


Figure 3-39 The Class Repository

Icon	Name	Description
	New Model Element...	Create a new Model Element in Model Explorer without the need of creating through diagramming
	Collapse	Collapse all the nodes under the selected tree node
	Expand	Expand all the nodes under the selected tree node
	Reverse Code...	Reverse code to model through Java Round-Trip
	Instant Reverse...	Reverse code to model through Instant Reverse
	Refresh	Refresh Class Repository content

Popup menu of Class Repository

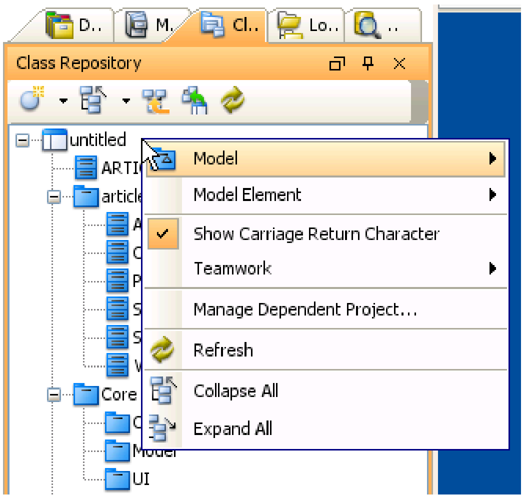


Figure 3-40 The Popup menu of Class Repository

Menu Title	Description
Model	Create a Model
Model Element...	Create a new Model Element in Class Repository without the need 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

Popup menu of Project

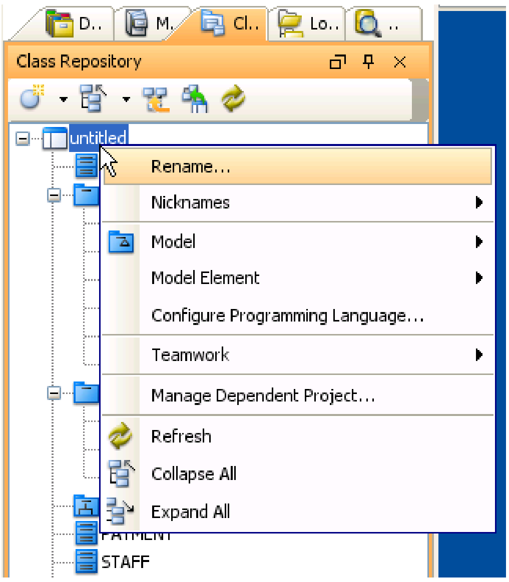


Figure 3-41 *The Popup menu of project node in Class Repository*

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 Class Repository without the need of creating through diagramming
Configure Programing 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

Logical View

The Logical View displays logical view of the project that can be customized to the desired structure.

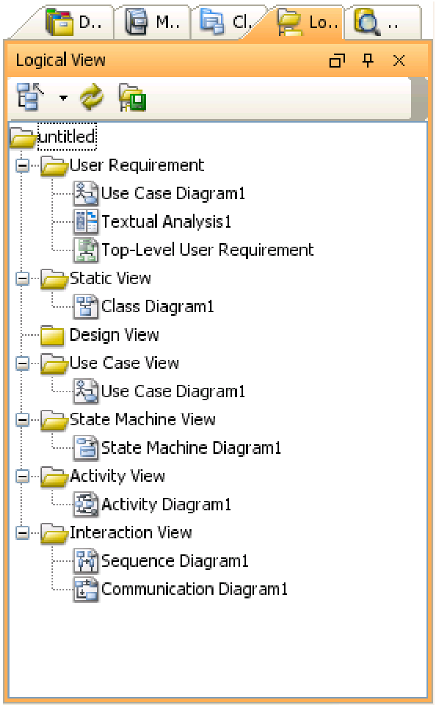


Figure 3-42 The Logical View

Icon	Name	Description
	Collapse	Collapse all the nodes under the selected tree node
	Expand	Expand all the nodes under the selected tree node
	Refresh	Refresh Class Repository content
	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

Popup menu of Logical View

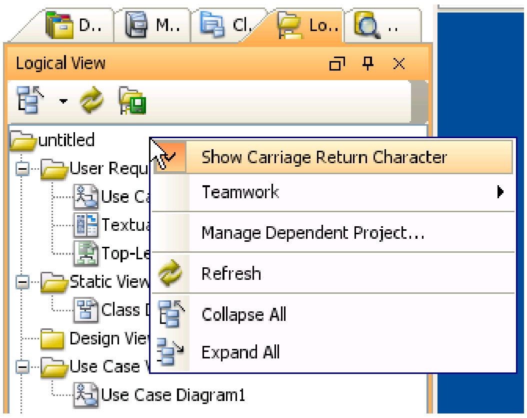


Figure 3-43 The Popup menu of Logical View

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

Popup menu of Project

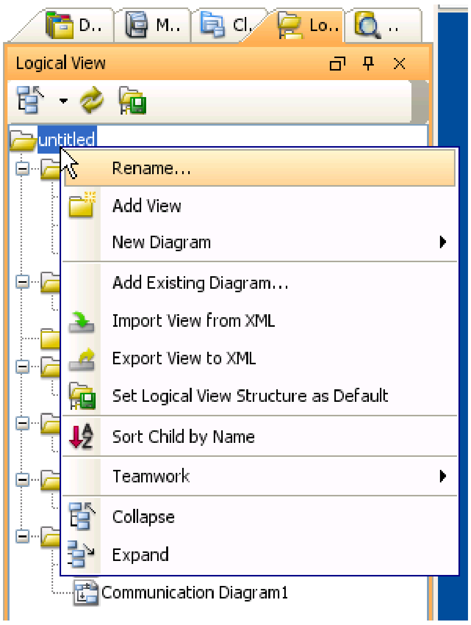


Figure 3-44 *The Popup menu of project node in Logical View*

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

Popup menu of View

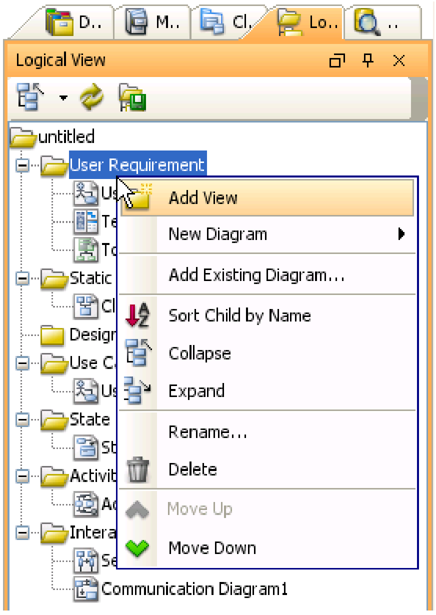
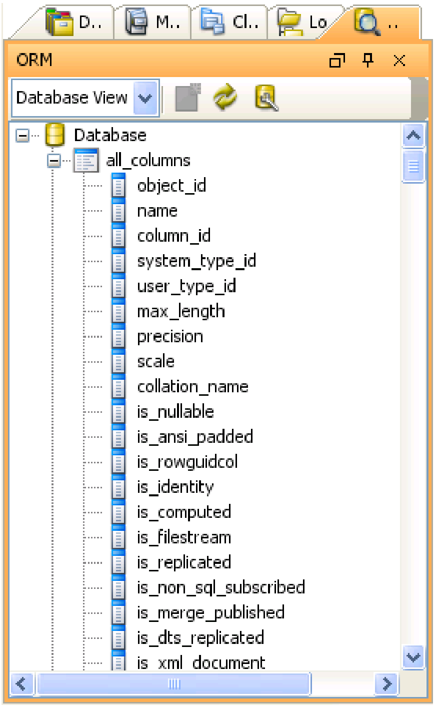


Figure 3-45 The Popup menu of view in Logical View

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
Move Up	Move the selected view upwards
Move Down	Move the selected view downwards

ORM

ORM Pane displays ORM-related views including classes and database configurations.



Database
all_columns
object_id
name
column_id
system_type_id
user_type_id
max_length
precision
scale
collation_name
is_nullable
is_ansi_padded
is_rowguidcol
is_identity
is_computed
is_filestream
is_replicated
is_non_sql_subscribed
is_merge_published
is_dts_replicated
is_xml_document

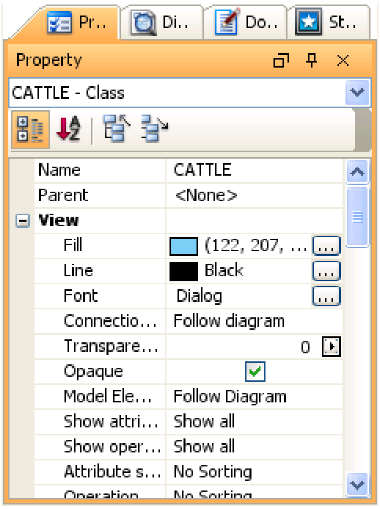
__Figure 3-46__ The ORM pane

Icon	Name	Description
Database View	Database/Class View	Switch between Database or Class view
	New Model Element...	Add or remove classpaths. Available only in Class View.
	Refresh	Refresh ORM pane
	Database Configuration	Configure the connection with database. Available only in Database View.

Editing Panes

Property Pane

Property Pane displays the properties of the selected element(s). You can also edit the properties here.
To open this pane:



CATTLE - Class

Name	CATTLE
Parent	<None>
View	
Fill	(122, 207, ...
Line	■ Black
Font	Dialog
Connectio...	Follow diagram
Transpare...	0
Opaque	☑
Model Ele...	Follow Diagram
Show attri...	Show all
Show oper...	Show all
Attribute s...	No Sorting
Operation	No Sorting

Figure 3-47 The Property pane

- Select **View > Panes > Property**
- Keyboard Shortcut: **Ctrl+Shift+P**

Diagram Overview Pane

This pane displays the overview of the active diagram which is scaled to fit the display area. The rectangle represents the visible area of diagram. You can navigate to different parts of the diagram by dragging the display area.

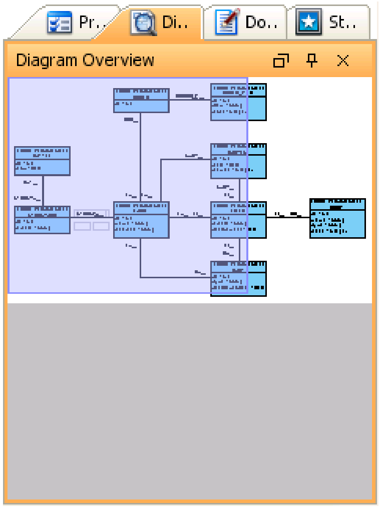


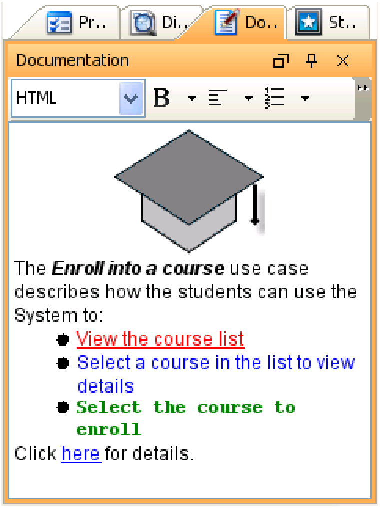
Figure 3-48 The Diagram Overview pane

To open this pane:

- Select **View > Panes > Diagram Overview**
- Keyboard Shortcut: **Ctrl+Shift+V**

Documentation Pane

Documentation Pane can display documentation of the selected element. VP-UML also provides rich text documentation. You can edit your documentation directly here.



The ***Enroll into a course*** use case describes how the students can use the System to:

- View the course list
- Select a course in the list to view details
- **Select the course to enroll**

Click here for details.

Figure 3-49 The Documentation pane

To open this pane:

- Select **View > Panes > Documentation**
- Keyboard Shortcut: **Ctrl+Shift+U**

Stencil Pane

Stencil Pane displays stencils that can be used to create custom shapes. You can select a stencil and drag it to diagram to create a shape. Stencils can be created in Shape Editor (Tools > Shape Editor).

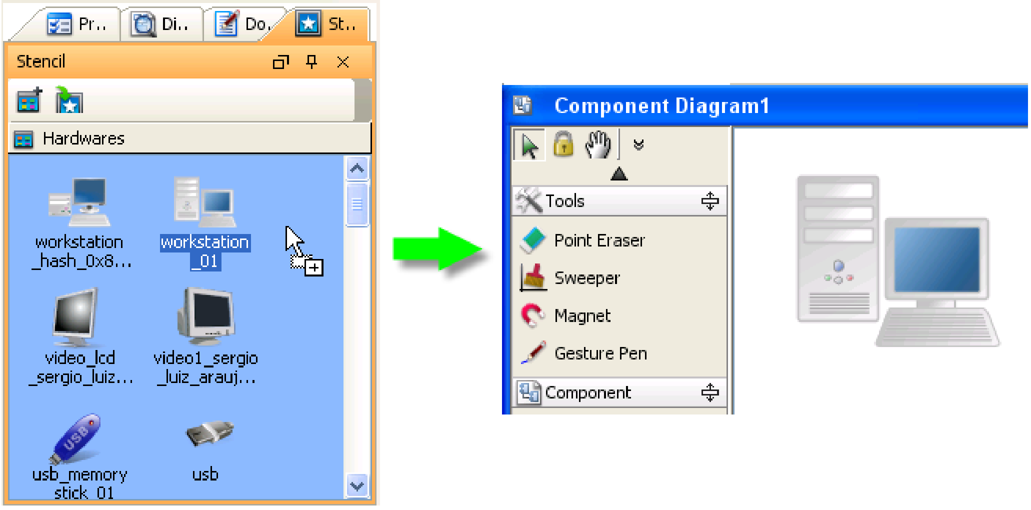


Figure 3-50 The Stencil pane

To open this pane:

- Select **View > Panes > Stencil**
- Keyboard Shortcut: **Ctrl+Shift+S**

To add a category, click on the Add Stencil button and select the category to add.

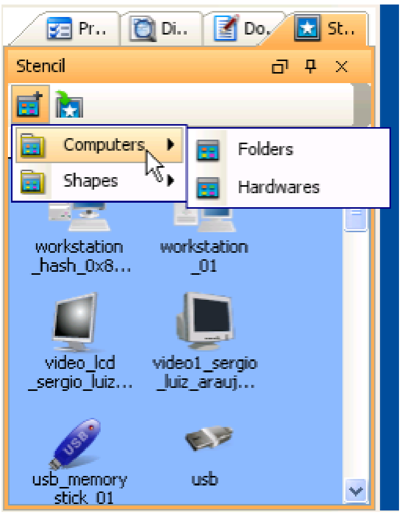


Figure 3-51 *To add a category in Stencil pane*

You can add multiple categories. A category can be opened by clicking on it.

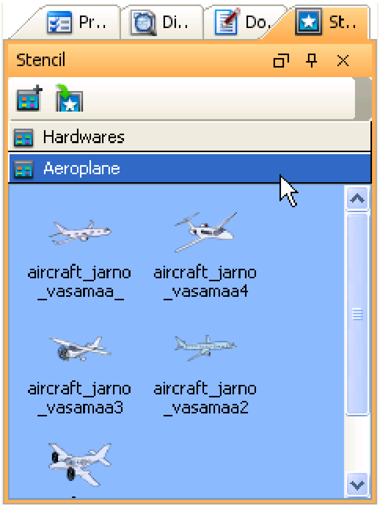


Figure 3-52 *Click a category to open it*

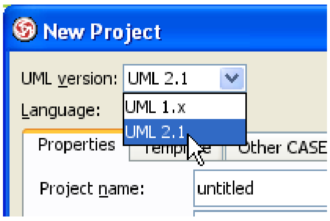
Creating Project

Visual Paradigm for UML stores information like model elements and diagrams in a project. Therefore, you need to create a project before performing modeling.

To create a project, select menu **File** > **New Project**. The **New Project** dialog box appears.

Selecting UML version

The **UML version** combo box lets you select the version of UML notation for the project. Normally you would select **UML 2.1** so you can create UML diagrams of the latest standard. However if you want to create diagrams in older notation you should select **UML 1.x**.



New Project

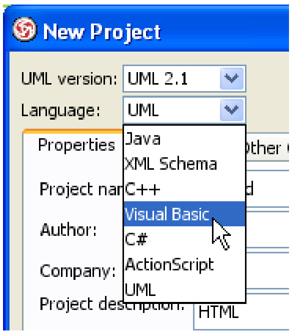
UML version: UML 2.1

Project name: untitled

__Figure 4-1__ Select UML version of project

Selecting Language

The **Language** combo box lets you select the programming/scripting language for the project.



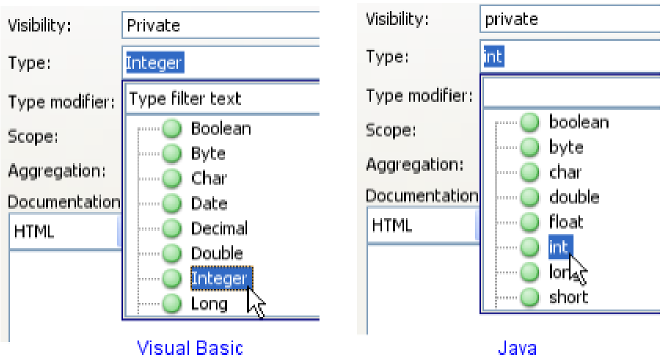
New Project

UML version: UML 2.1

Language: UML

__Figure 4-2__ Select language of project

The language you selected mainly affects the class modeling. For example, the selectable visibilities and primitive types vary among languages.



Visibility: Private

Type: Integer

Visibility: private

Type: int

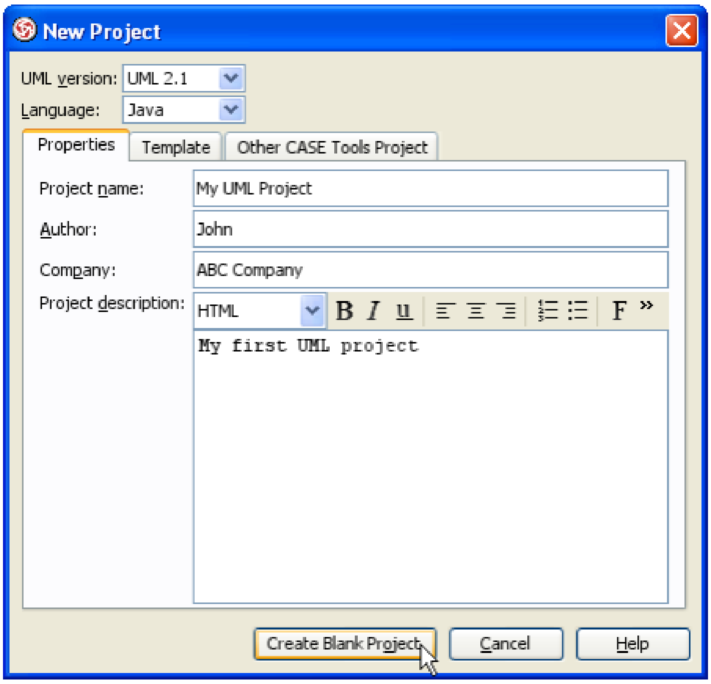
Visual Basic

Java

__Figure 4-3__ Different selectable primitive types for different languages

Creating Blank Project

To create a blank project, ensure the **Properties** page is being selected, and then click the **Create Blank Project** button.



Project name: My UML Project

Author: John

Company: ABC Company

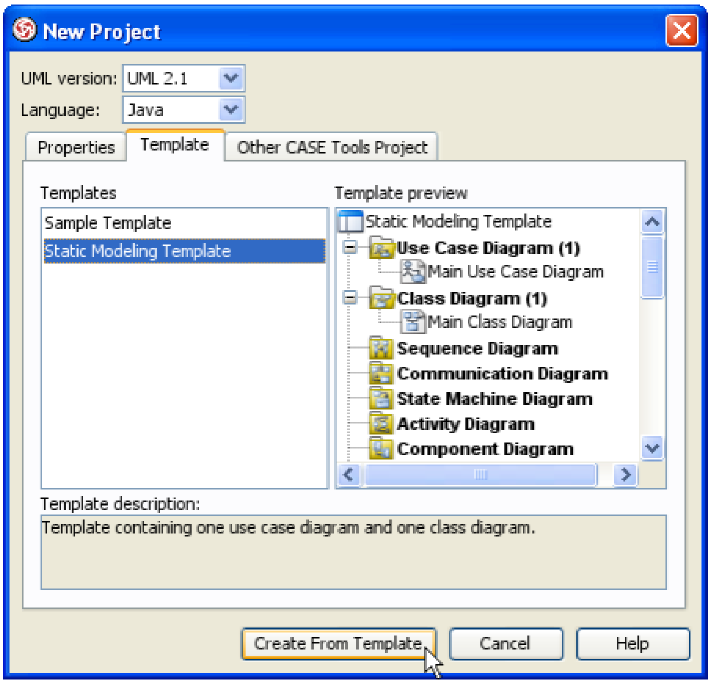
Project description: My first UML project

Figure 4-4 *Create a blank project*

Creating Project from Project Template

If you often create projects of similar structure, you can save the project as a template and use it for creating project later. For example, you can save a project with a use case diagram and a class diagram as template called "Static Modeling Template". When we create project from this template, the project will have the same structure as the template automatically.

To create project from template, select the **Template** page and select a template to use, then click the **Create From Template** button.



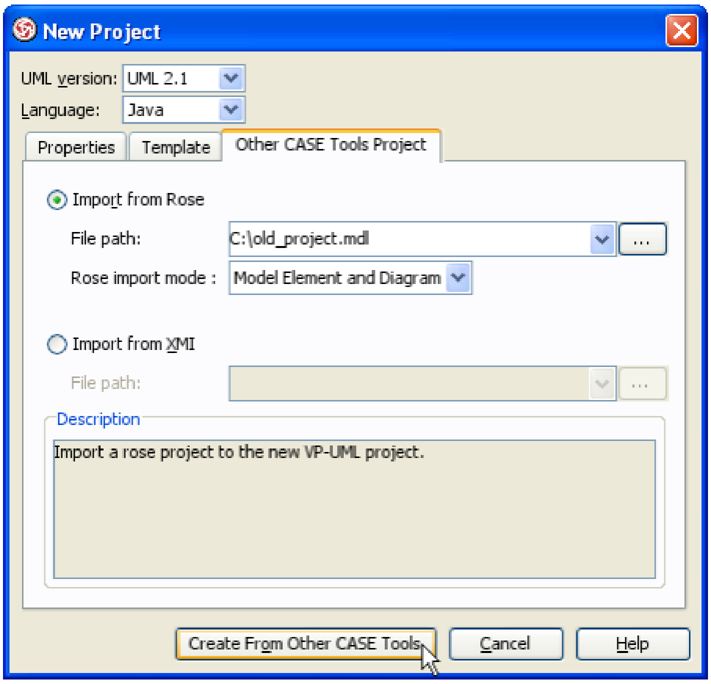
Template description: Template containing one use case diagram and one class diagram.

Figure 4-5 *Create a project from template*

Creating Project by importing other CASE Tool Project

If you have legacy projects created with other CASE tools, you can import them to VP-UML.

Select the **Other CASE Tools Project** page, and select either **Import from Rose** to import a Rational Rose project (.mdl), or **Import from XMI** to import XMI file exported from the CASE tool. Click **Create From Other CASE Tools** to import the project.



UML version: UML 2.1

Language: Java

● Import from Rose

File path: C:\old_project.mdl

Rose import mode : Model Element and Diagram

○ Import from XMI

File path:

Import a rose project to the new VP-UML project.

Figure 4-6 *Create a project by importing other CASE tool project*

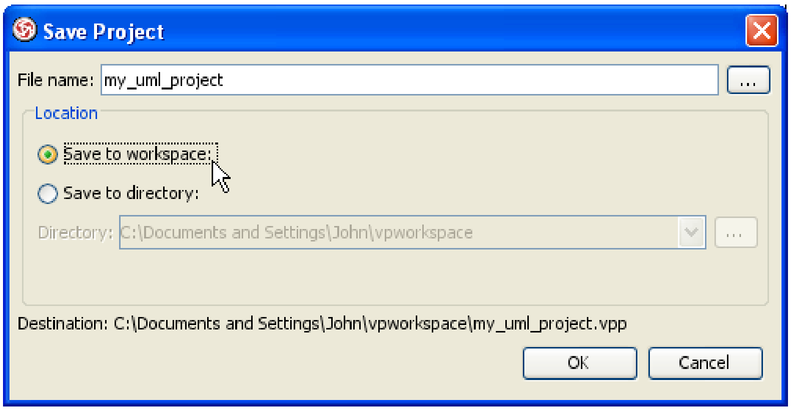
Saving Project

Visual Paradigm for UML saves all project content to a single file, with file extension .vpp.

To save project, select menu **File** > **Save Project**. The first time you save the project, the **Save Project** dialog box will appear.

Saving Project to Workspace

To save project to workspace, select **Save to workspace**. The project file will be saved in the workspace directory.



Save Project

File name: my_uml_project

Location

● Save to workspace:

○ Save to directory:

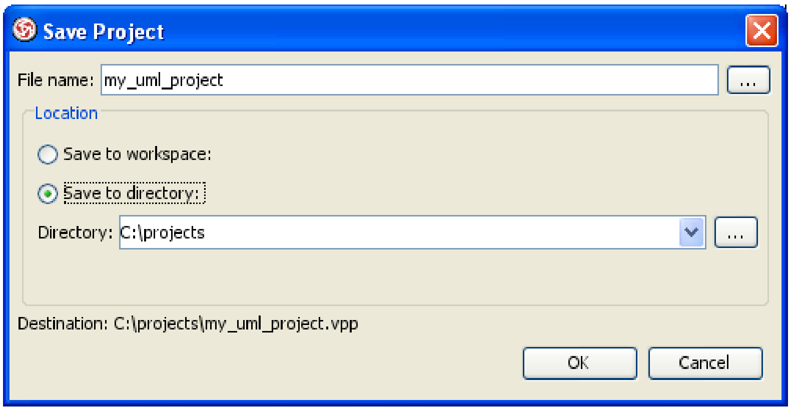
Directory: C:\Documents and Settings\John\vpworkspace

Destination: C:\Documents and Settings\John\vpworkspace\my_uml_project.vpp

Figure 4-7 *Save project to workspace*

Saving Project to Specified Directory

To save project to a specified directory, select **Save to directory**, and then specify the directory path in the **Directory** field.



Save Project

File name: my_uml_project

Location

○ Save to workspace:

● Save to directory:

Directory: C:\projects

Destination: C:\projects\my_uml_project.vpp

Figure 4-8 *Save project to a specified directory*

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.

VP-UML 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 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 popup 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.

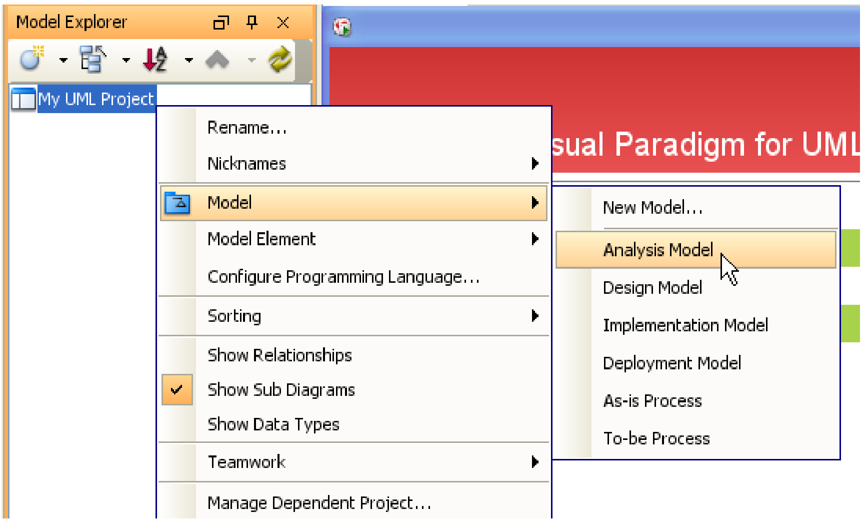


Figure 4-9 *Save project to a specified directory*

Creating Diagram in Model

To create diagram in Model, right-click on the Model and select **Diagram** from the popup menu, and then select to either create a new diagram or add existing diagrams to the Model.

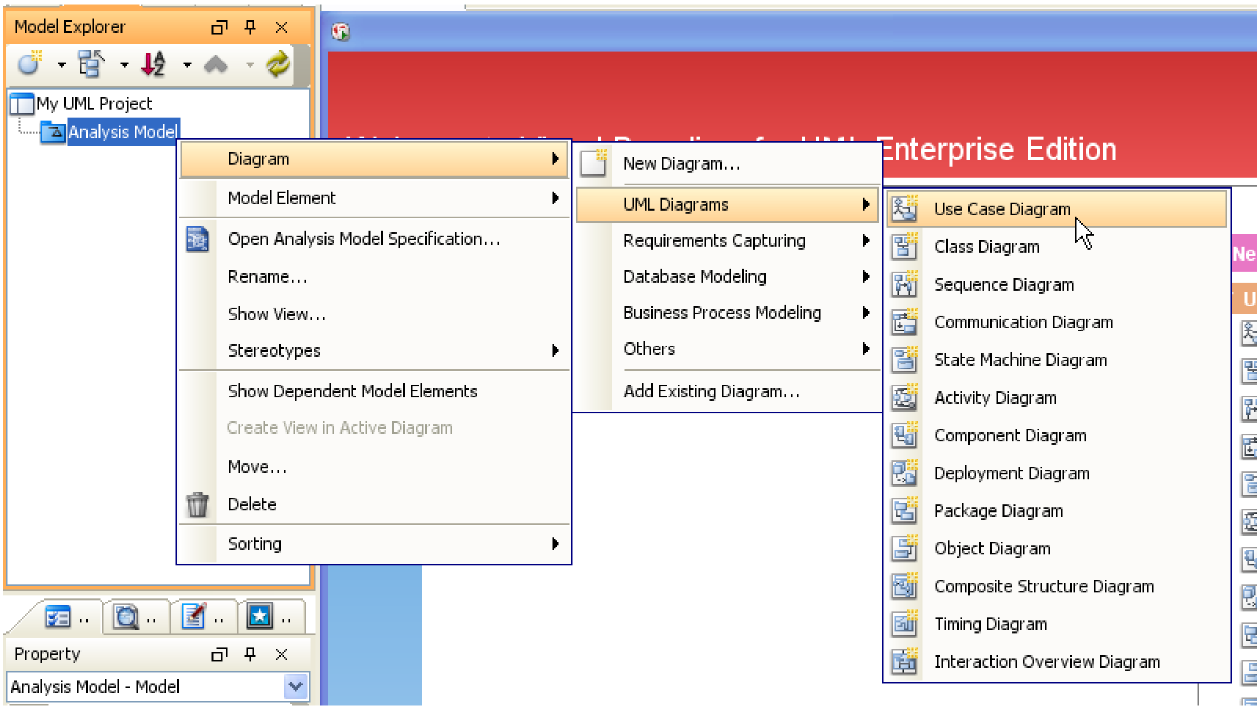


Figure 4-10 *Create a diagram in Model*

Moving Diagrams between Models

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 popup menu.

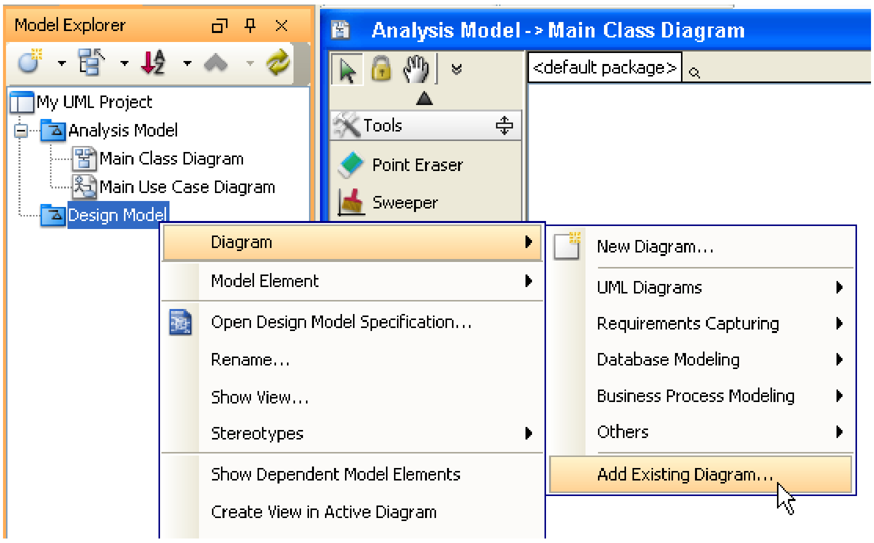
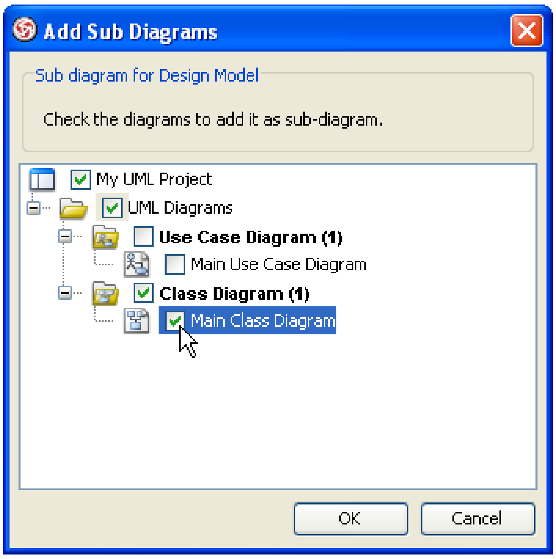


Figure 4-11 *Add existing diagram to Model*

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



Add Sub Diagrams

Sub diagram for Design Model

Check the diagrams to add it as sub-diagram.

- ☑ My UML Project
  - ☑ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Main Use Case Diagram
    - ☑ **Class Diagram (1)**
      - ☑ Main Class Diagram

OK Cancel

Figure 4-12 *Select diagrams to move*

The selected diagrams will be moved to the target Model.

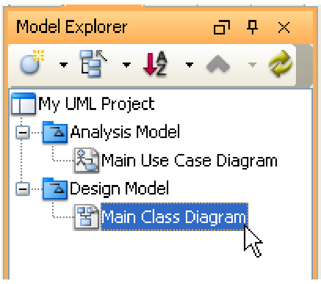


Figure 4-13 *Diagram moved*

Reference Model Elements of other Projects

Project referencing is a feature that enables us to reference to an external project, and uses its model elements in our own project. With this feature, we can analyze projects' data by visualizing different projects' model elements in a single diagram. Besides, we can organize our model elements in a more disciplined approach by having one Visual Paradigm project per library project. This also helps to slim up projects by separating part of it to isolated projects.

Managing Dependent Projects

To reference other project, right-click on the Model Explorer and select **Manage Dependent Project...** from the popup menu.

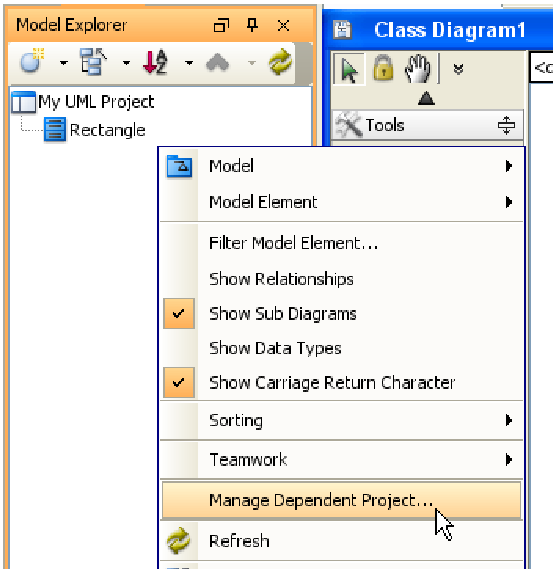
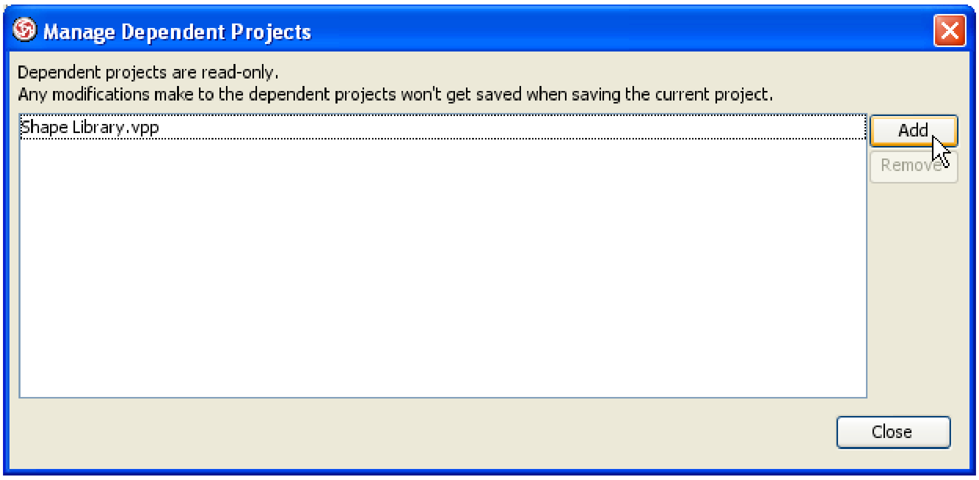


__Figure 4-14__ Manage dependent project

The **Manage Dependent Projects** dialog box appears. Click **Add** to browse for dependent project to add. Click **Close** when you have done.



Manage Dependent Projects

Dependent projects are read-only.
Any modifications make to the dependent projects won't get saved when saving the current project.

Shape Library.vpp

__Figure 4-15__ Add dependent project

Using Model Elements of Dependent Projects

Under the toolbar of Model Explorer, select the dependent project.

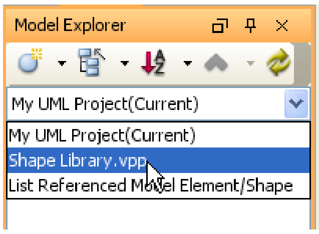


__Figure 4-16__ Select dependent project

Create a new diagram or open an existing diagram, and then drag model elements from the Model Explorer to the diagram. Alternatively, you can right-click on the model elements and select **Create View in Active Diagram** from the popup menu.

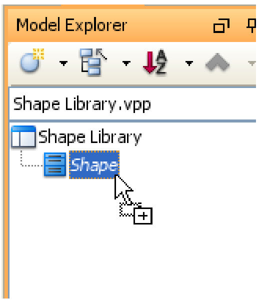


Figure 4-17 *Dragging model element of dependent project*

A view will be created for the model element. Then you can create relationships between it and the model elements of the current project.

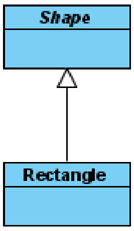


Figure 4-18 *View created for model element of dependent project*

Creating Diagrams

You can create diagrams in different ways:

- Using toolbar
- Using New Diagram dialog box
- Using popup menu of Diagram Navigator

To use toolbar to create:
Click on the icon on the toolbar.

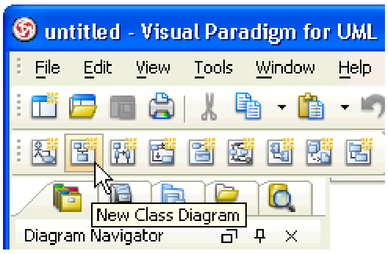


Figure 1-1 *New Class Diagram icon on the toolbar*

To use New Diagram dialog box to create:
1.Select **File > New Diagram > New Diagram...** from the main menu. The New Diagram dialog box is displayed.

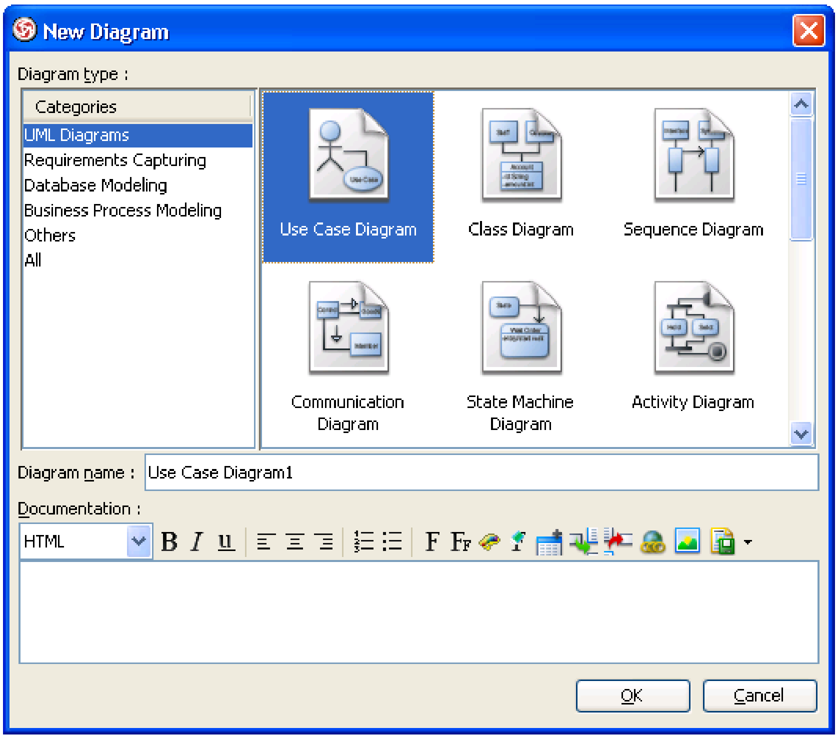


Diagram type : UML Diagrams — Use Case Diagram

Diagram name : Use Case Diagram1

Documentation : HTML

Figure 1-2 *New Diagram dialog box*

2.Then, select the category and select a diagram type in the category. You should also specify a diagram name. You may also specify the type of documentation.

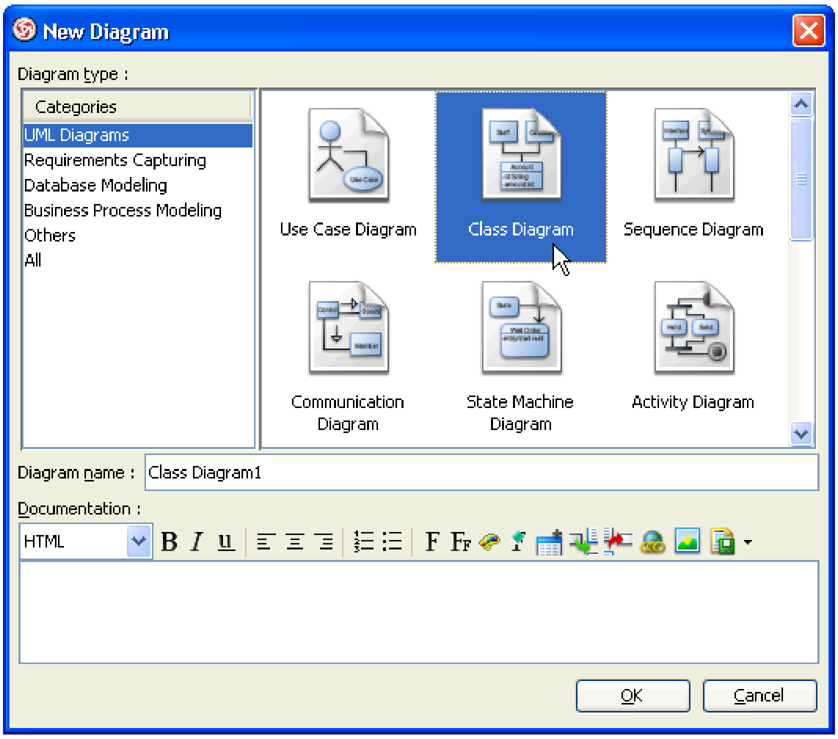


Diagram name : Class Diagram1

__Figure 1-3__ Select Class Diagram

To use the popup menu of Diagram Navigator to create:
Right click on the diagram type node in Diagram Navigator and select **New Class Diagram** in popup menu.

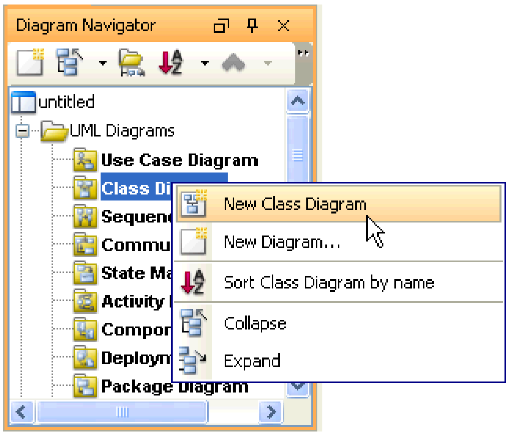


__Figure 1-4__ Select New Class Diagram from popup menu of Diagram Navigator

Drawing Shapes

After creating a new diagram, you can create diagram elements using the diagram toolbar. In this section, we will introduce the techniques of how to draw shapes:

- Creating Shapes
- Creating Connectors
- Creating Self-Connection

Creating Shapes

To create a shape, click on a diagram element button from the diagram toolbar and click on the diagram pane to create it. The element generated will have a default size.

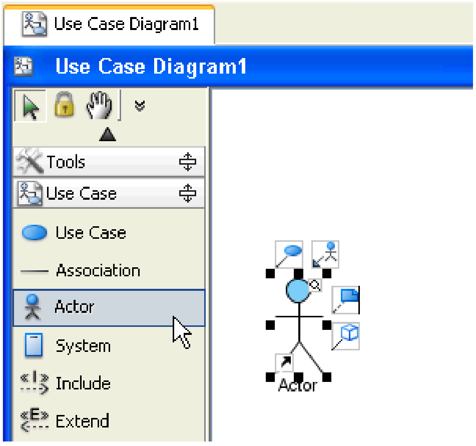


Figure 1-5 *Click to Create Shapes*

You can also drag a specific boundary before releasing the mouse to define a shape's initial size.

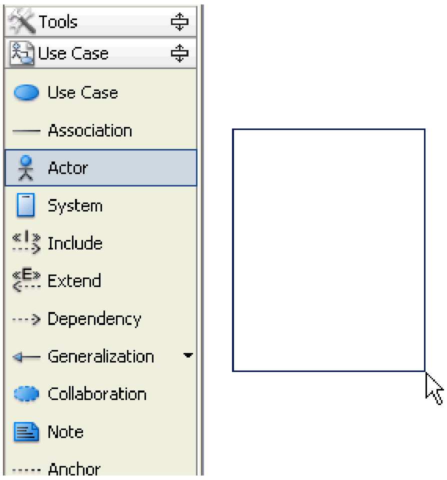


Figure 1-6 *Create Shapes with specific size*

Alternatively, you can also create a diagram element by dragging a diagram element button then dropping it on the diagram pane.

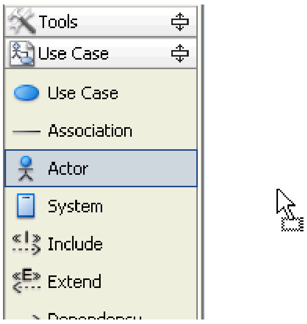


Figure 1-7 *Drag and drop to Create Shapes*

Apart from that, you can use the diagram popup menu to add a shape.

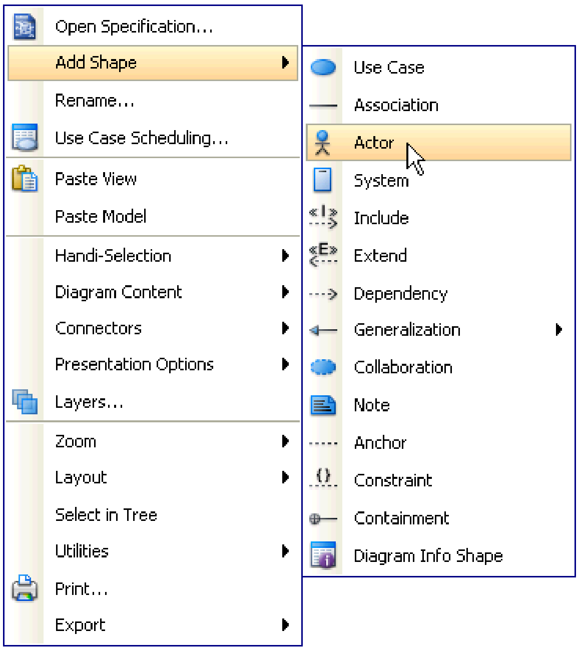


Figure 1-8 *Create shapes using diagram popup menu*

Creating Connectors
To create a connector, select the desired connector from the diagram toolbar and click on the source shape. Drag the connector to the destination shape.
VP-UML provides continuous UML syntax checking. You will see a stop sign when you try to create an invalid connection, e.g. you cannot create a generalization relationship between an actor and a use case.

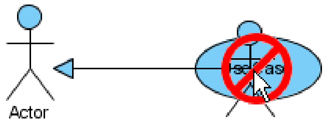


Figure 1-9 *Try to create an invalid connection*

If the connection is valid you will see a blue rounded rectangle surrounding the destination shape.

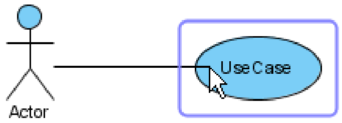


Figure 1-10 *Try to create a valid connection*

You may also use resource to create connectors.
Click on the Association resource of a shape and drag over the shape you want to connect to. If you release the mouse on an empty space, a shape will be created with the connector.

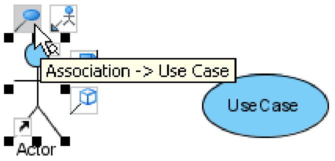


Figure 1-11 *Click on resource*

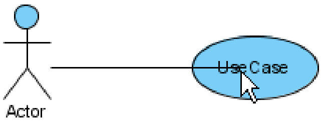


Figure 1-12 *Drag over the shape*

Creating Self-Connection

Some of the shapes can have a connection to itself, for example Self-Association of a Class or Self-Link of an Object in a Communication Diagram. To create a self-connection, click on the connector button on the diagram toolbar and click once on the target object.
Alternatively, you can click on the **Self Association** resource.

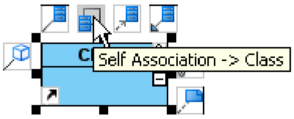


Figure 1-13 *Create Self-Connection*

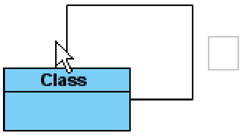


Figure 1-14 *Self-Connection*

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 no longer needs to go back and forth 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:

- Connection Resource
- Manipulation Resource
- Branching Resource

Connection Resource

It is designed for creating elements and making connections.

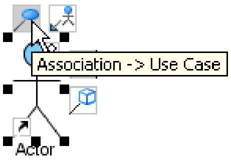


Figure 1-15 *Create association with connecting a new or existing use case*

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.

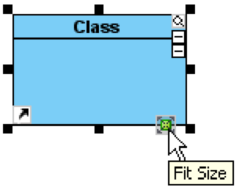


Figure 1-16 *Fit size by manipulation resource*

Branching Resource

Branching Resource helps you to create decision structure in diagram.

Drawing Freehand Shapes

Freehand Shape is general graphic shapes. There are pen shapes, pencil shapes, and some regular shapes like circle, rectangle and arrow. You can use freehand shape for annotating diagram.

For example, you can use freehand shapes to represent some classes which come from other libraries.

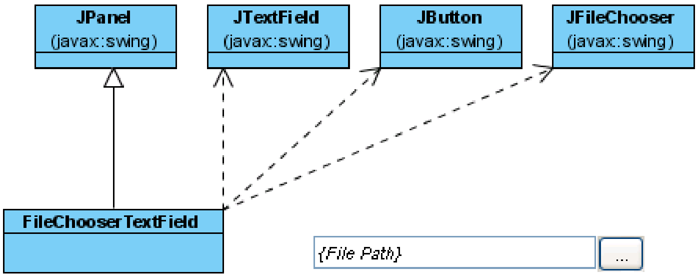


Figure 1-17 *Classes come from javax.swing*

Use Pencil to highlight the classes from javax.swing.

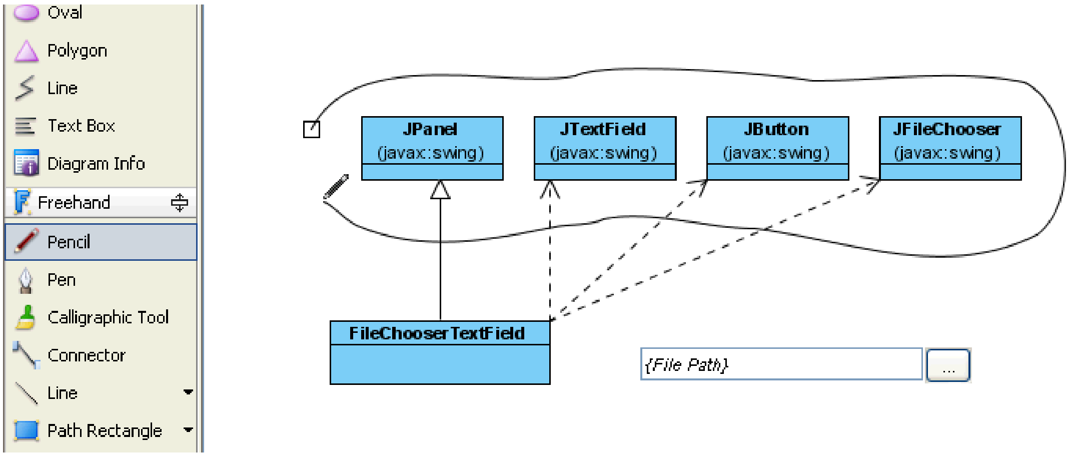


Figure 1-18 *Pencil*

Use Word Art to show text.

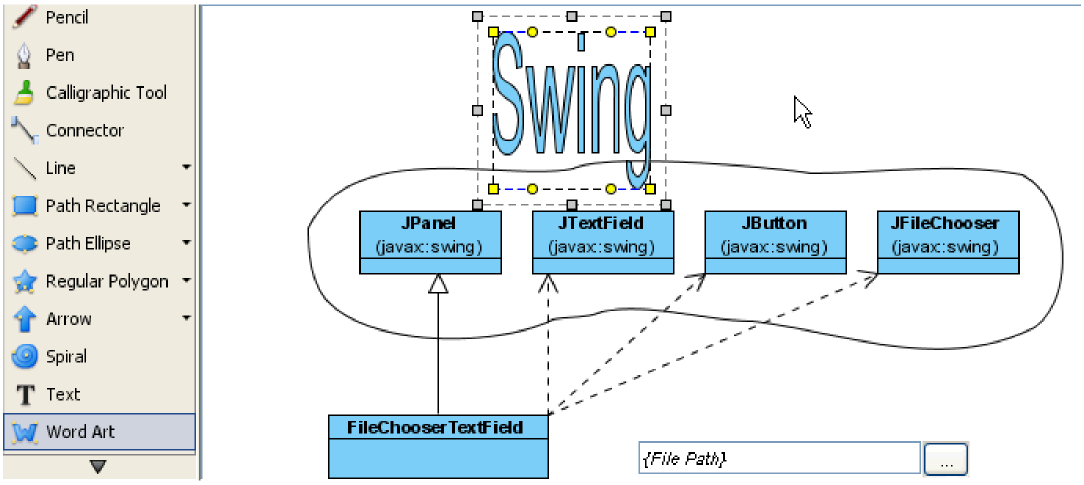


Figure 1-19 *Word Art*

Set freehand shapes style

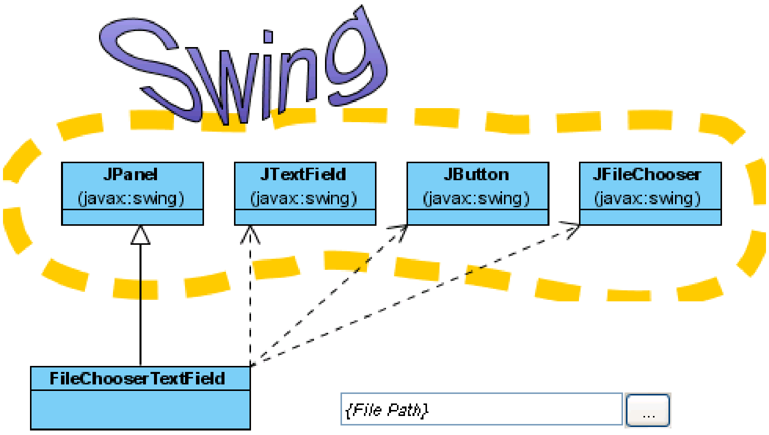


Figure 1-20 *Styled freehand shapes*

Changing Package Header

You can specify the parent package of any diagram through Package Header.

Package Header when diagram created

When diagram create, the package header will be focused and let you to specify the parent package of the diagram. (Specify the package by entering the fully qualifier name of the package)

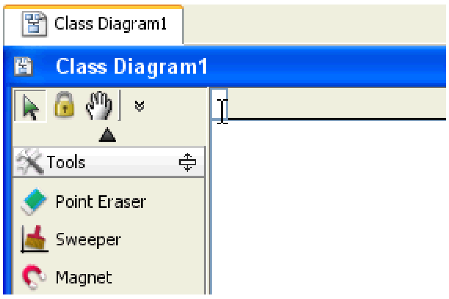
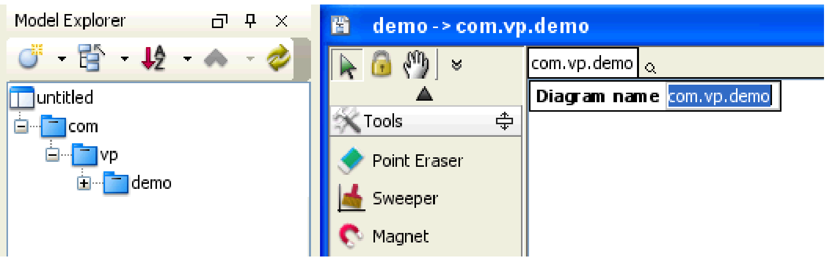


Figure 1-21 *Specify parent package in package header*

After you confirm the parent package. The diagram will be named same as the package.

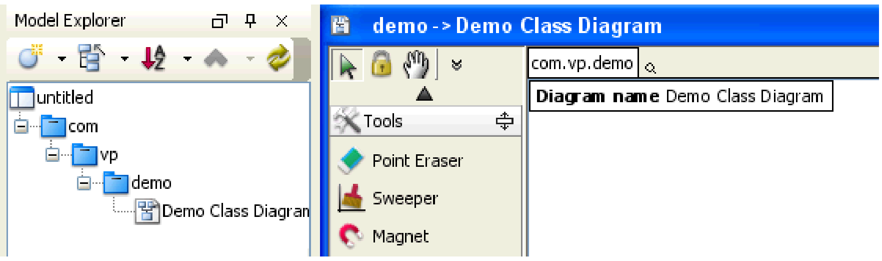


com.vp.demo

Diagram name com.vp.demo

Figure 1-22 *Diagram name will be same as fully qualify of parent package*

You can modify the diagram name. (The parent package won't be changed to follow diagram name)



com.vp.demo

Diagram name Demo Class Diagram

Figure 1-23 *Diagram name can be edited to be different with parent package*

You can open specification of parent package.

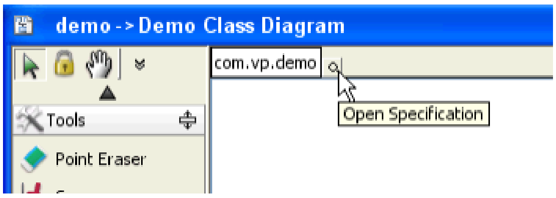


Figure 1-24 *Open Specification*

You also can change the parent package of the diagram by double click the package header.

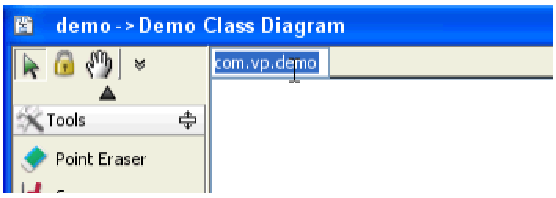


Figure 1-25 *Double-click to change parent package*

A new package will be created if the package name you entered does not exist in project. If the old parent package does not contain any elements, it will be deleted, so, the documentation (or other properties) or old parent package will be lost.

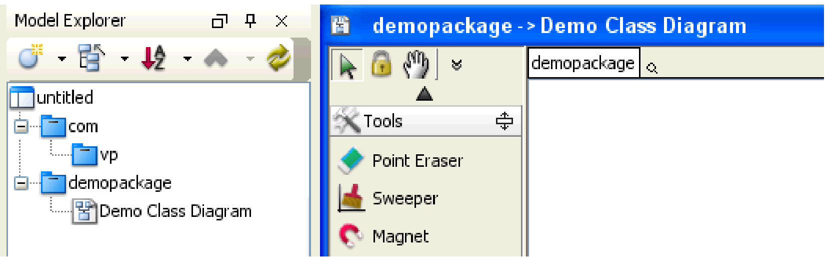


Figure 1-26 *Old parent package may be deleted if new parent package is specified*

You can show/hide the package header by diagram popup menu: **Presentation Options > Show Package Header**

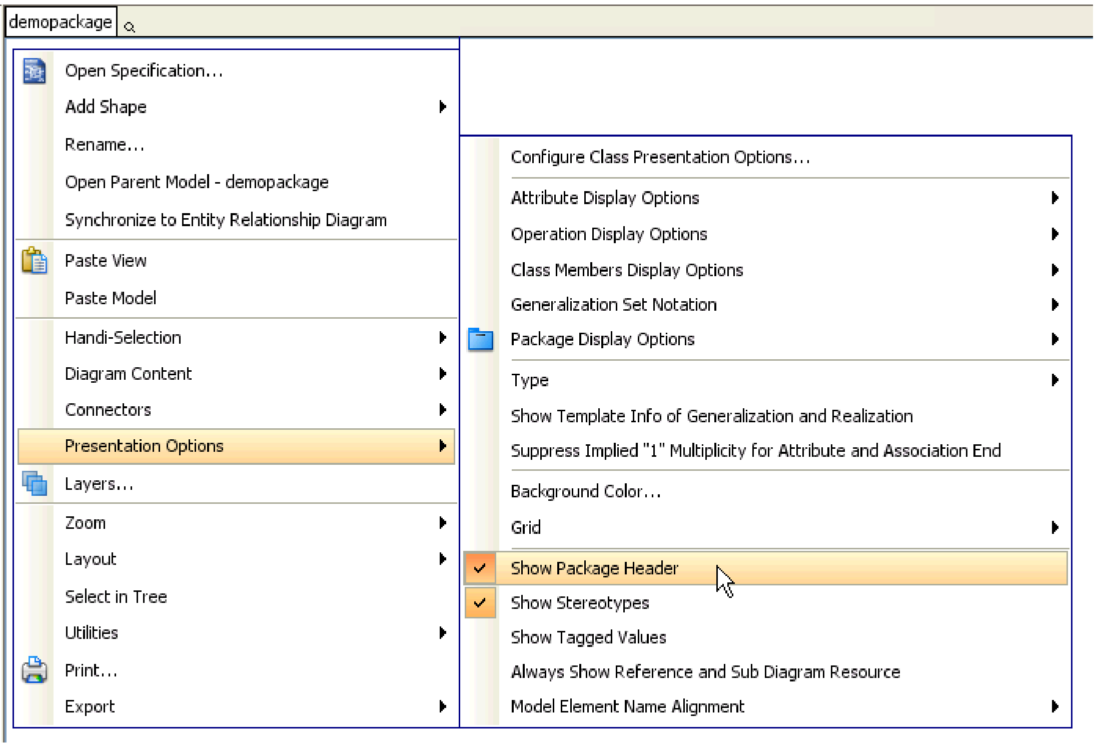

demopackage
Open Specification...
Add Shape
Rename...
Open Parent Model - demopackage
Synchronize to Entity Relationship Diagram
Paste View
Paste Model
Handi-Selection
Diagram Content
Connectors
Presentation Options
Layers...
Zoom
Layout
Select in Tree
Utilities
Print...
Export
Configure Class Presentation Options...
Attribute Display Options
Operation Display Options
Class Members Display Options
Generalization Set Notation
Package Display Options
Type
Show Template Info of Generalization and Realization
Suppress Implied "1" Multiplicity for Attribute and Association End
Background Color...
Grid
Show Package Header
Show Stereotypes
Show Tagged Values
Always Show Reference and Sub Diagram Resource
Model Element Name Alignment

Figure 1-27 *Show/hide package header*

Justify shape name

In Visual Paradigm, shape name is aligned center horizontally, and top or middle vertically, depending on the characteristic of shapes. It is possible to realign the shape name, which is useful for language that is written from right to left, like Modern Hebrew.

Adjusting Shape Name's Position

1.Right click on the diagram and select **Presentation Options > Model Element Name Alignment** from the popup menu. There you can choose the desired position to display the name.

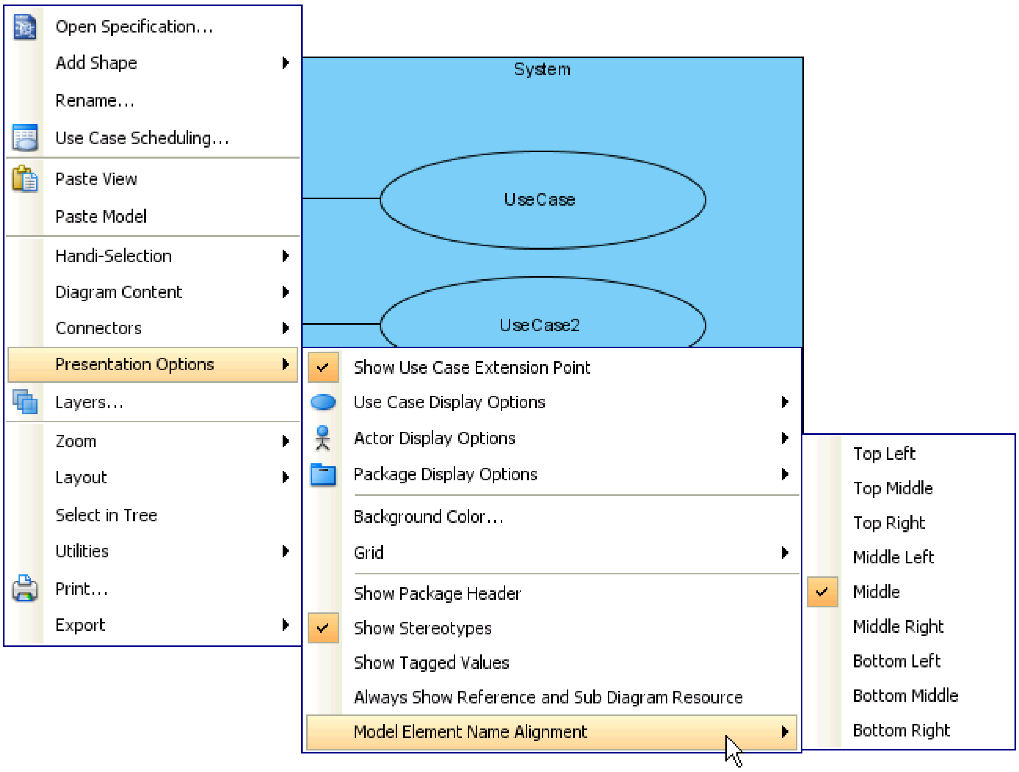


Figure 1-28 Select model element name alignment on diagram

2.This is the result of &ldquo;Middle Right&rdquo;. All shapes which Model Element Name Alignment is Follow Diagram are now aligned middle right.

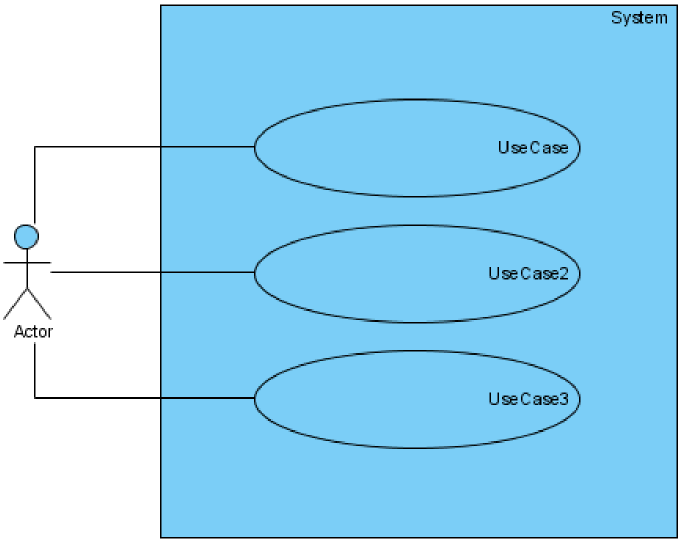


Figure 1-29 Align to Middle Right

Besides the diagram-wide setting, you can also set for specific shape. To do this:
1.Right click the shape and select **Presentation Options > Model Element Name Alignment** in the popup menu, and select the desired alignment option.

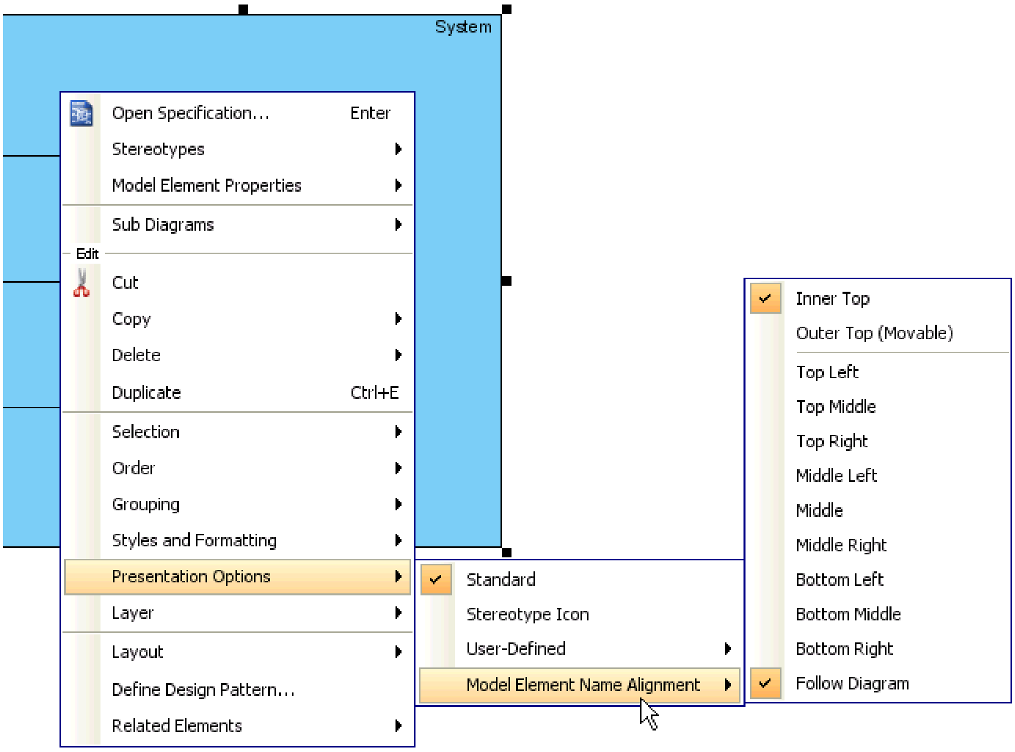


Figure 1-30 *Select model element name alignment on shape*

2.This is the result of setting System's name to align top middle.

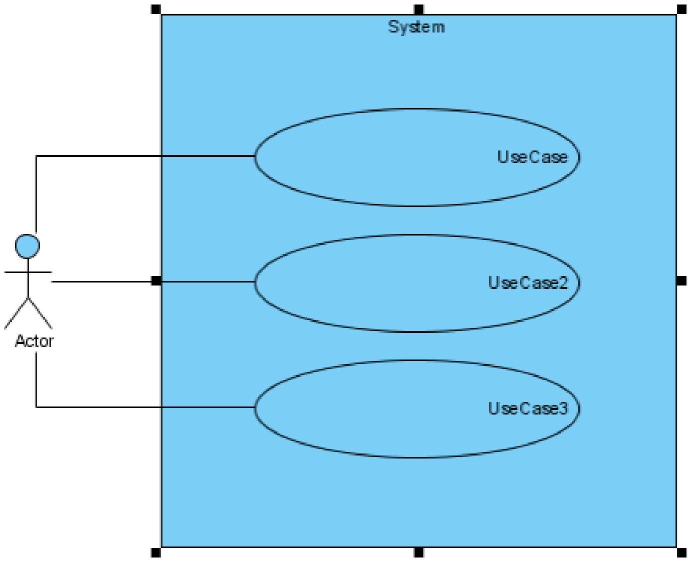
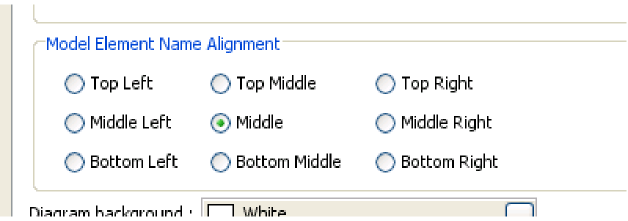


Figure 1-31 *Selected shape's name is aligned to Top Middle*

Besides setting of current diagram, you can also set for future diagrams.
1.Select **Tools > Options...** to open Options dialog box.
2.In the Options dialog box, open the **Diagramming** category, select the **Appearance** tab. You can select Model Element Name Alignment.


Model Element Name Alignment

○ Top Left ○ Top Middle ○ Top Right
○ Middle Left ● Middle ○ Middle Right
○ Bottom Left ○ Bottom Middle ○ Bottom Right

Diagram background : White

Figure 1-32 *Define default Model Element Name Alignment in Options dialog*

Exceptions
Some of the shapes are not able to justify name due to their characteristic. There are mainly two kinds of shapes. First, shapes with floating name label (freely movable) or with label not put inside the shape cannot be justified. Actor, Initial Pseudo Node and BP Start Events are examples of this kind of shape.

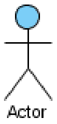


Figure 1-33 *Floating name label*

Container shapes have a limited scope of displaying names since their &ldquo;bodies&rdquo; are for containing other shapes, and are not available for positioning names. Package, State and System are example of this kind of shape.

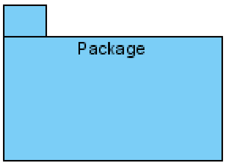


Figure 1-34 *Container shape*

Enable and disable minimum shape size

All shapes have their default minimum size. Users are not allowed to resize shapes to smaller than the minimum size, which help to ensure the shapes are compact yet clear enough to be seen on a diagram under normal conditions. The minimum size can be determined by fitting a shape's size.

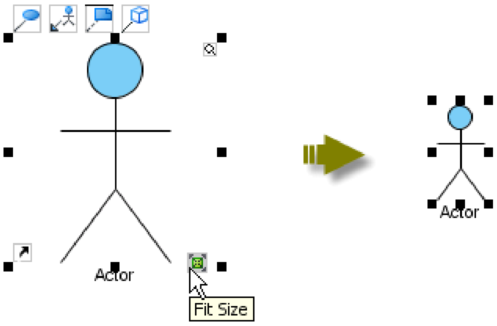


Figure 1-35 *Minimum size can be determined by fitting shape size*

Now, it is possible to disable such checking such that shapes can be resized to very small in size, ignoring the minimum size. For example:

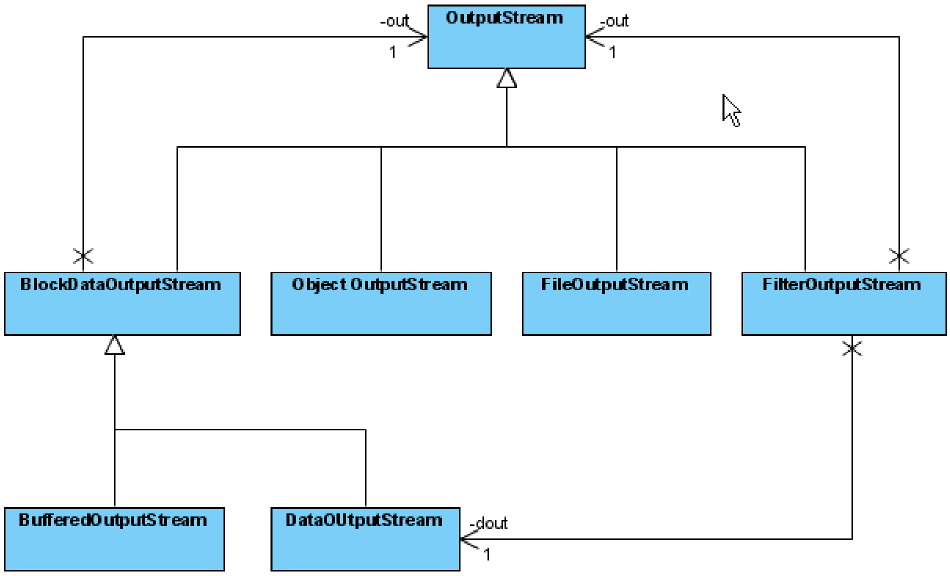


Figure 1-36 *Example, shapes are fitted size*

1.To disable the minimum size checking, select **Tools > Options...** to open **Options** dialog box.
2.In the Options dialog box, open the **Diagramming** category, select the **Appearance** tab and uncheck **Enable minimum size**. Click **OK** to confirm.

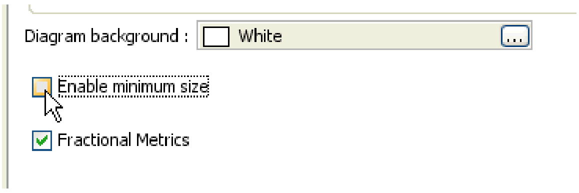


Diagram background : White

□ Enable minimum size

☑ Fractional Metrics

Figure 1-37 *Disable minimum size checking in Option dialog*

3.From Now on, you can resize the shapes to very small in size.

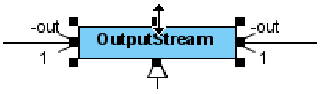


Figure 1-38 *Resize to very small*

4.And then select all the classes and select **Same Height**.

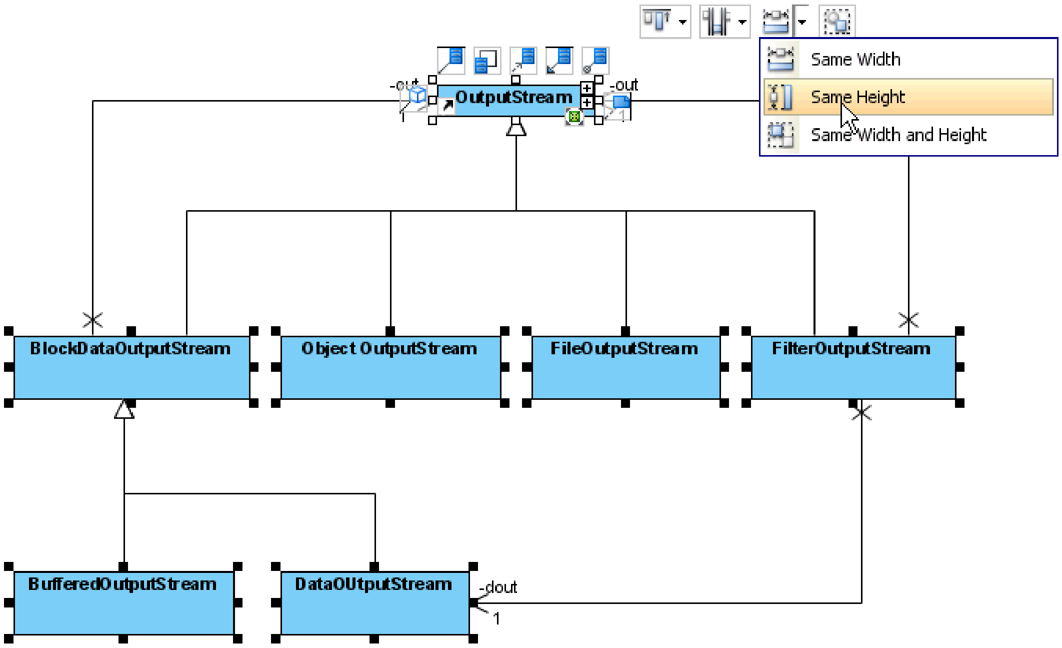


Figure 1-39 *Same height as smallest size shape*

5.This is the final result:

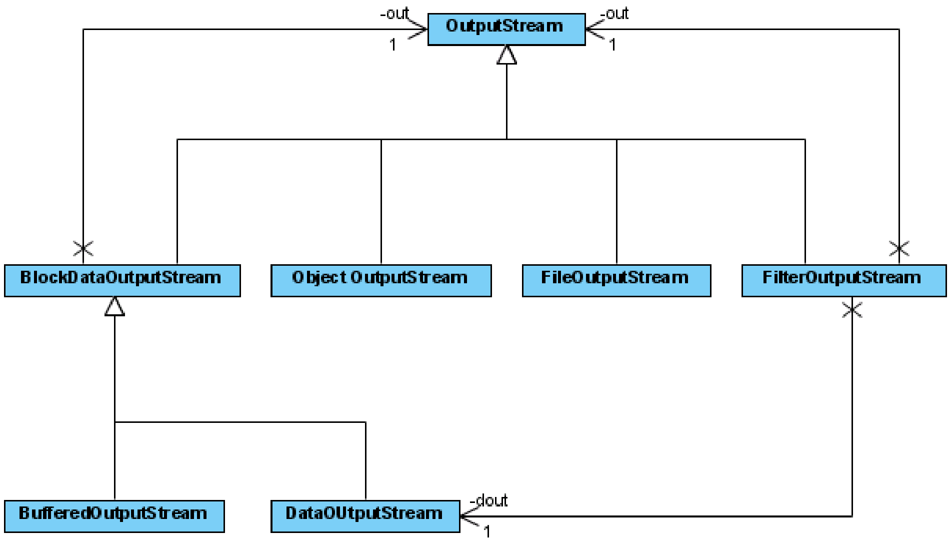


Figure 1-40 *All shapes are small size*

Undo and Redo

When you create and edit a diagram, you may make mistakes like accidentally deleting a diagram element. You can use the Undo function to cancel the previous action. On the other hand, you may re-perform the action using the Redo action. The undo/redo feature in VP-UML is diagram based.

Undo

You can roll back undesirable changes by performing Undo. To undo an action, perform one of the following actions:

- Select **Edit > Undo** from main menu.
- Click on the **Undo** button on toolbar.
- Press **Ctrl-Z**.

Redo

This feature is to re-perform actions that were just undone. To redo an action, perform one of the following actions:

- Select **Edit > Redo** from main menu.

- Click on the **Redo** button on toolbar.
- Press **Ctrl-Y**.

Showing Name for Undo and Redo Action

Since Undo and Redo are easy and fast ways to cancel or restore the actions we've done in the project, but it's somehow inconvenient to rely on our own memories of the actions done.
You can know the Undo and Redo name from main menu

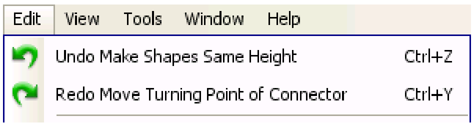

Edit
View
Tools
Window
Help
Undo Make Shapes Same Height
Ctrl+Z
Redo Move Turning Point of Connector
Ctrl+Y

Figure 1-41 *Menu shows Undo/Redo name*

or tooltip shown on toolbar button

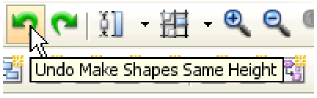


Figure 1-42 *Toolbar button's tooltip shows Undo/Redo name*

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 no longer needs to go back and forth 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.

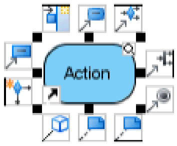


Figure 1-43 *Resources of Action*

Select **View > Resource Centric** in main menu, you can enable/disable **Resources, Group Resources**, **Extra Resources** and **Generic Resources Only**.

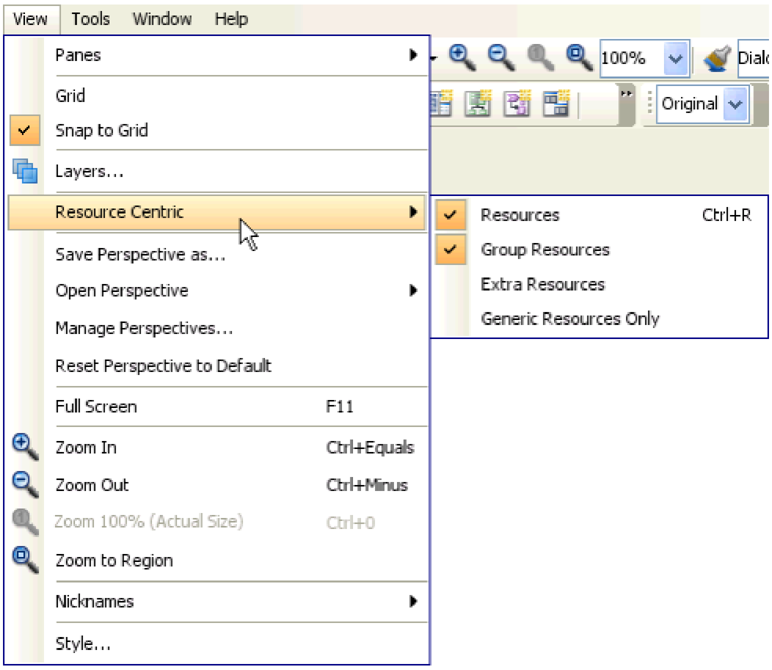
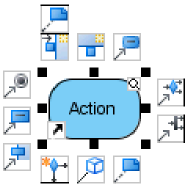


Figure 1-44 *Enable/disable resources on main menu*



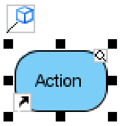


Figure 1-45 *Extra Resources of Action* ***Figure 1-46*** *Generic Resources Only of Action*

Creating shapes by Resource Centric Interface

You can create another shape by Resource Centric.
1.Select an actor, select the **Association -> Use Case** resource centric.

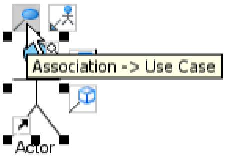


Figure 1-47 *Resources to create Use Case with Association*

2.Drag on the empty area on diagram.

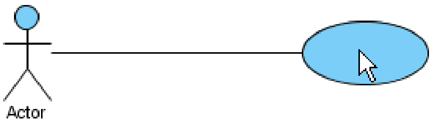


Figure 1-48 *Dragging on empty area*

3.After drop on the diagram, a Use Case will be created and connected by an association.

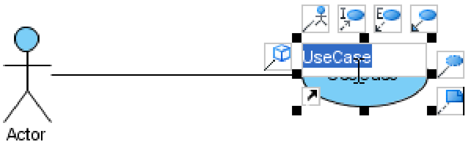


Figure 1-49 *Dropping to create Use Case with connecting Association*

Connecting shapes by Resource Centric Interface

You can connect to existing shape by Resource Centric
1.Select an actor, select the **Association -> Use Case** resource centric.

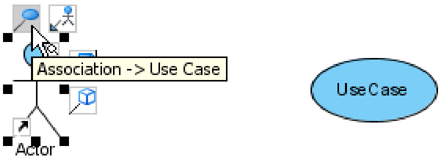


Figure 1-50 *Resources to create Use Case with Association*

2.Drag on an existing shape.

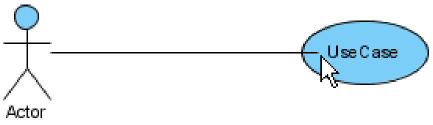


Figure 1-51 *Dragging on an existing Use Case*

3.After drop on the shape, an association is created and connects the actor and use case.

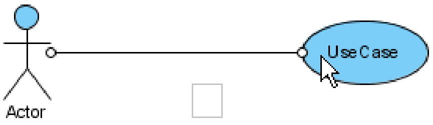


Figure 1-52 *Dropping to create Association connecting with the Use Case*

Using Quick connect

You can trigger Quick Connect by Resource Centric to connect existing shape.
1.Select an actor, click the **Association -> Use Case** resource centric.

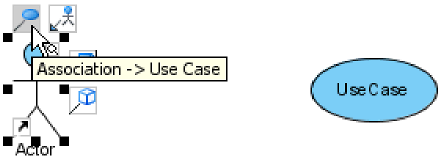
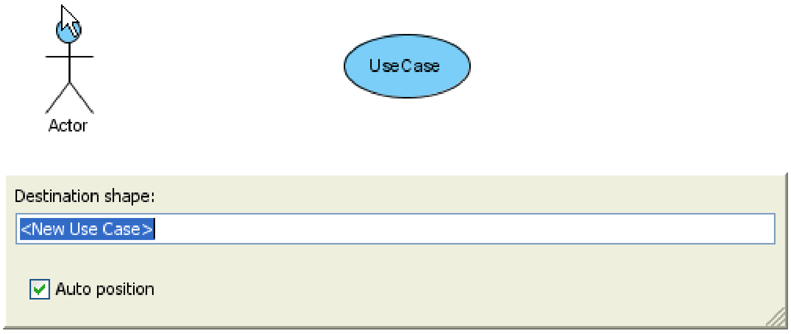


Figure 1-53 *Resources to create Use Case with Association*

2.The Quick Connect dialog shown to let you create a new Use Case or select an existing Use Case.



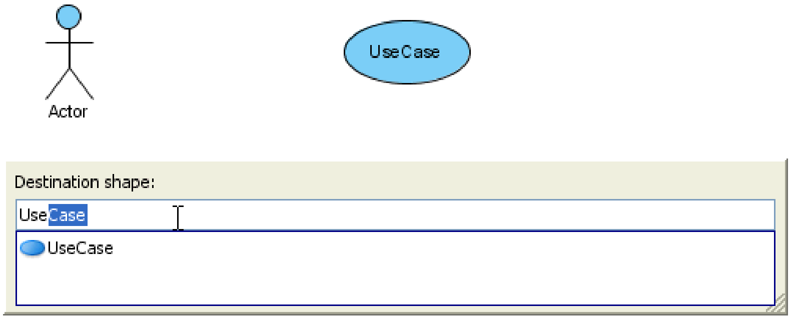
Destination shape:

<New Use Case>

☑ Auto position

__Figure 1-54__ Clicking on the resource to show Quick Connect dialog

3.Enter the use case name on the dialog. It shows a list of use cases which match with you typed.



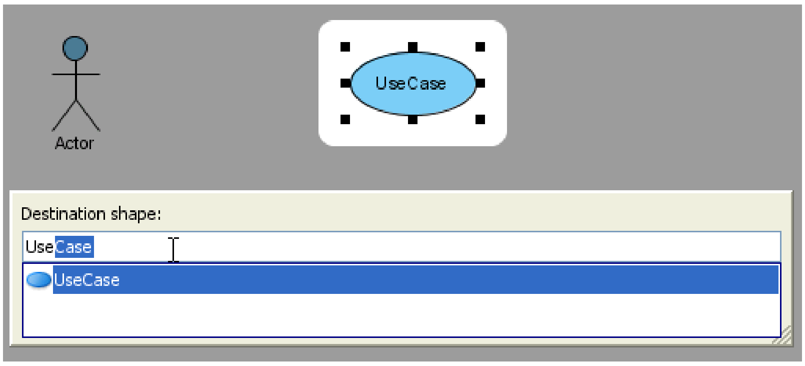
Destination shape:

UseCase

UseCase

__Figure 1-55__ Entering name of model element

4.If a use case is selected on list. After a moment, the use case will be focused on diagram.



Destination shape:

UseCase

UseCase

__Figure 1-56__ Use Case will be focused

5.Confirm selecting the use case by pressing ENTER, an association is created and connects the actor and use case.

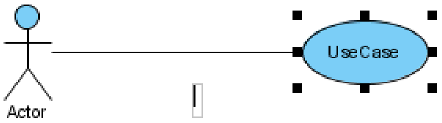


__Figure 1-57__ Association created

Split connection by shape

For some connectors (e.g. Control Flow in Activity Diagram), Resource Centric supports splitting the connector by adding another shape.
1.Select a control flow, select **Split with Decision Node**.

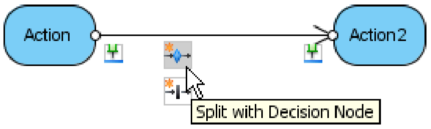


Figure 1-58 *Select resource of Control Flow*

2.A decision node is created and the control flow split.

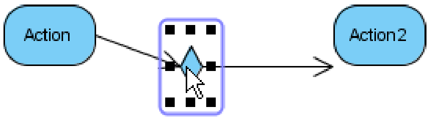


Figure 1-59 *Control Flow is splited by Decision Node*

Creating structure

For some model (e.g. Action in Activity Diagram), Resource Centric supports creating a structure of models.
1.Select an action, select **Create Branch with Decision Node**.

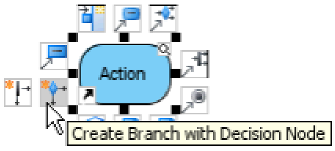


Figure 1-60 *Select Create Branch with Decision Node on Action resource centric*

2.Drag on the diagram.

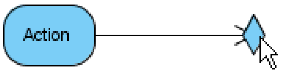


Figure 1-61 *Dragging on empty area*

3.After drop, 2 actions and a decision node are created and connected.

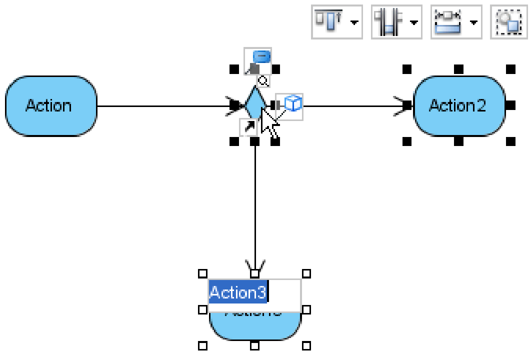


Figure 1-62 *Dropping to create Decision Node and Actions*

Group Selection Resource

Resource Centric supports Alignment and Grouping when selecting several shapes.
1.Select several shapes on diagram.

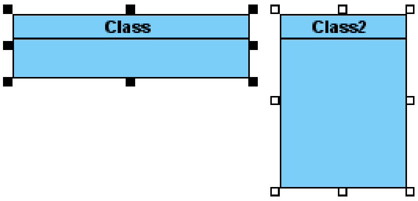


__Figure 1-63__ Select several shapes

2.Mouse over on any one of the selected shapes. Group Selection Resources are shown.

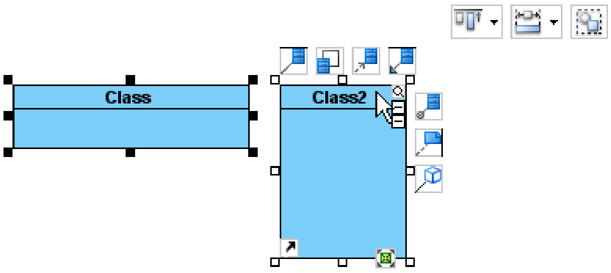


__Figure 1-64__ Group Selection Resources are shown

3.You may align the shapes by resources.

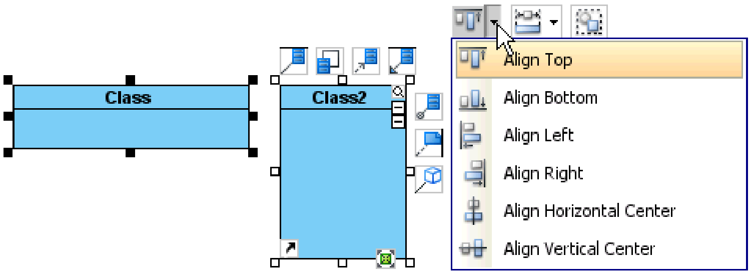
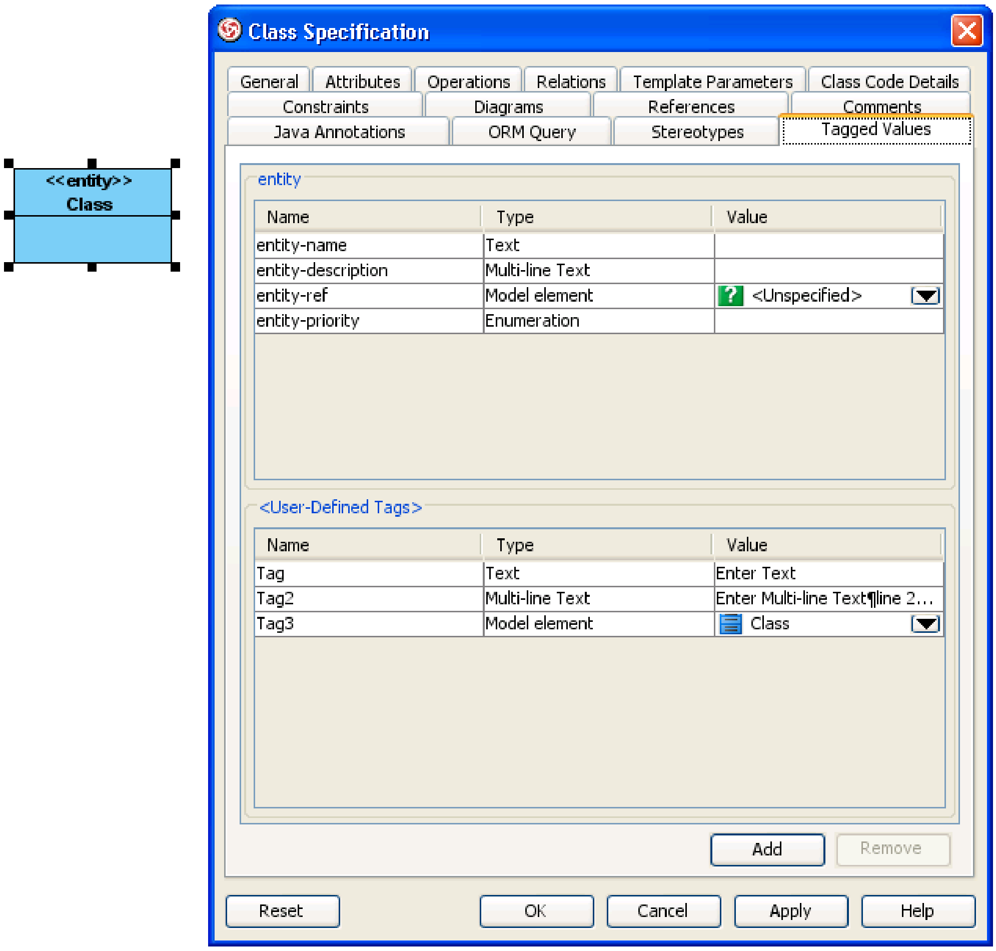


__Figure 1-65__ Aligns the selected shapes

Tagged Values

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.
VP-UML supports defined tagged values in stereotype, also support tagged values without stereotype.



entity

Name	Type	Value
entity-name	Text	
entity-description	Multi-line Text	
entity-ref	Model element	<Unspecified>
entity-priority	Enumeration	

<User-Defined Tags>

Name	Type	Value
Tag	Text	Enter Text
Tag2	Multi-line Text	Enter Multi-line Text¶line 2...
Tag3	Model element	Class

Figure 1-66 Stereotype-defined/user-defined tagged values are shown on Specification dialog of its owner model element

Adding User-Defined Tags

1.To define tagged values in model element. Open the specification of a model element.

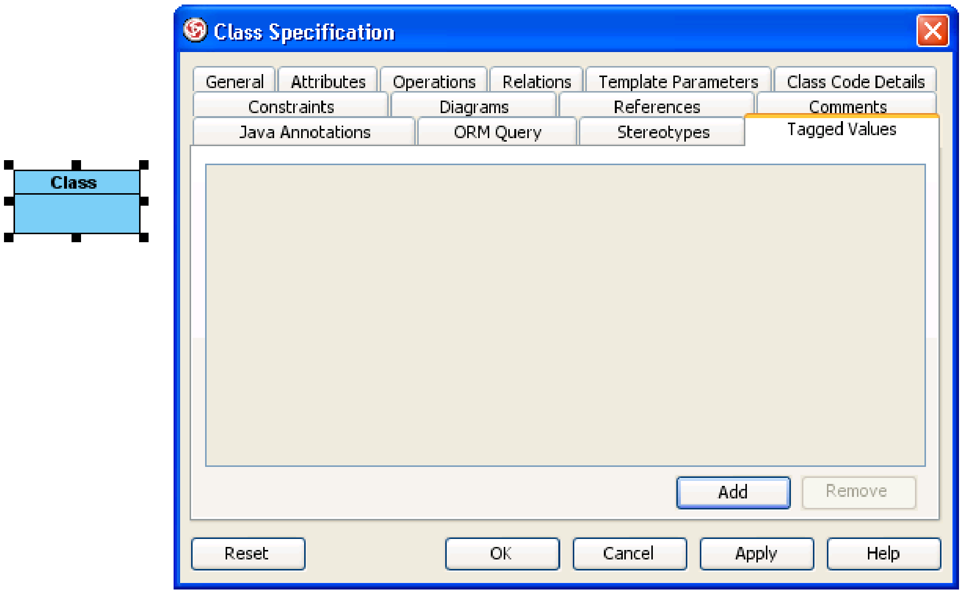


Figure 1-67 Open specification dialog of owner model element to create tagged value

2.Click on **Add** button, you can create tagged value with 3 types of value: **Text**, **Multi-line Text**, **Model Element**

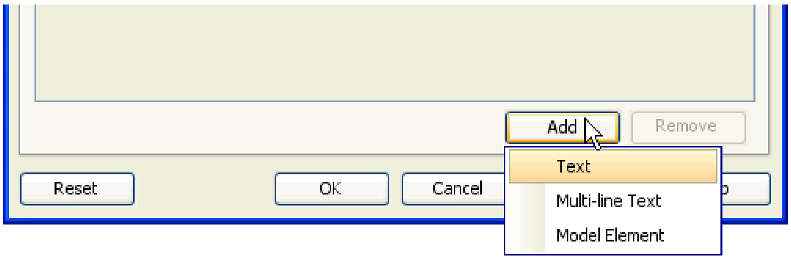
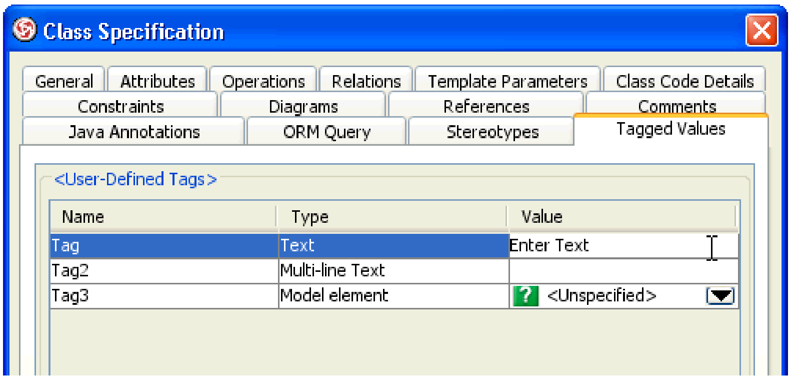


Figure 1-68 *3 kinds of tagged value can be created*

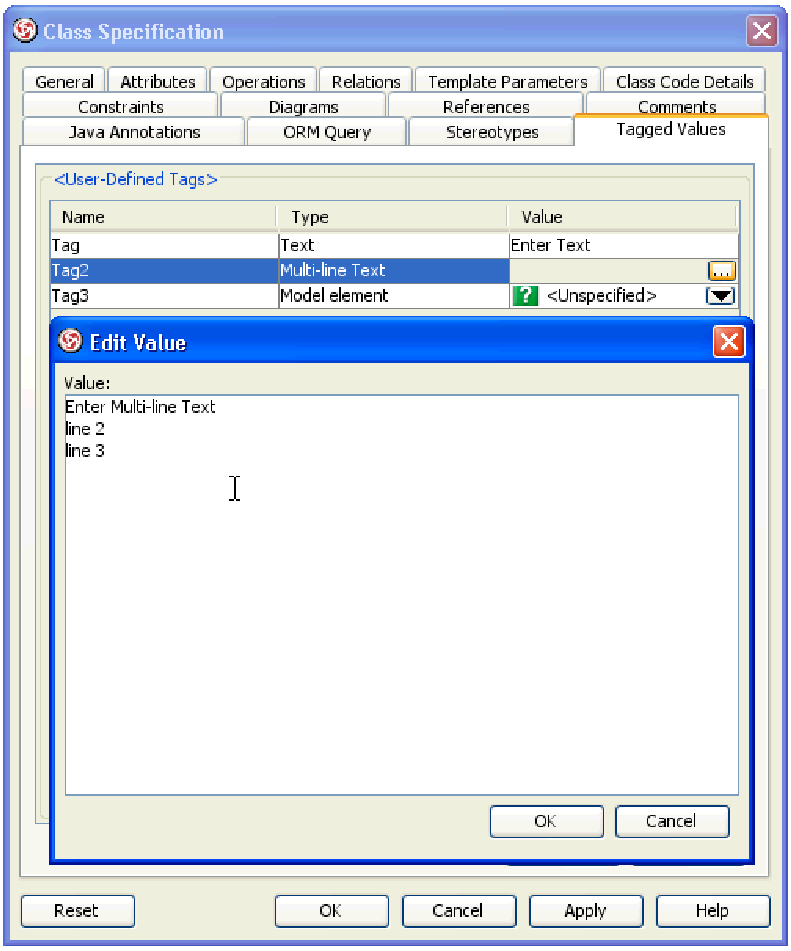
3. **Text** supports string-based value. You can enter the text on table directly.



Name	Type	Value
Tag	Text	Enter Text
Tag2	Multi-line Text	
Tag3	Model element	? <Unspecified>

Figure 1-69 *Text*

4. **Multi-line Text** supports multi-line string. You can enter multi-line text after clicking ... button.



Name	Type	Value
Tag	Text	Enter Text
Tag2	Multi-line Text	...
Tag3	Model element	? <Unspecified>

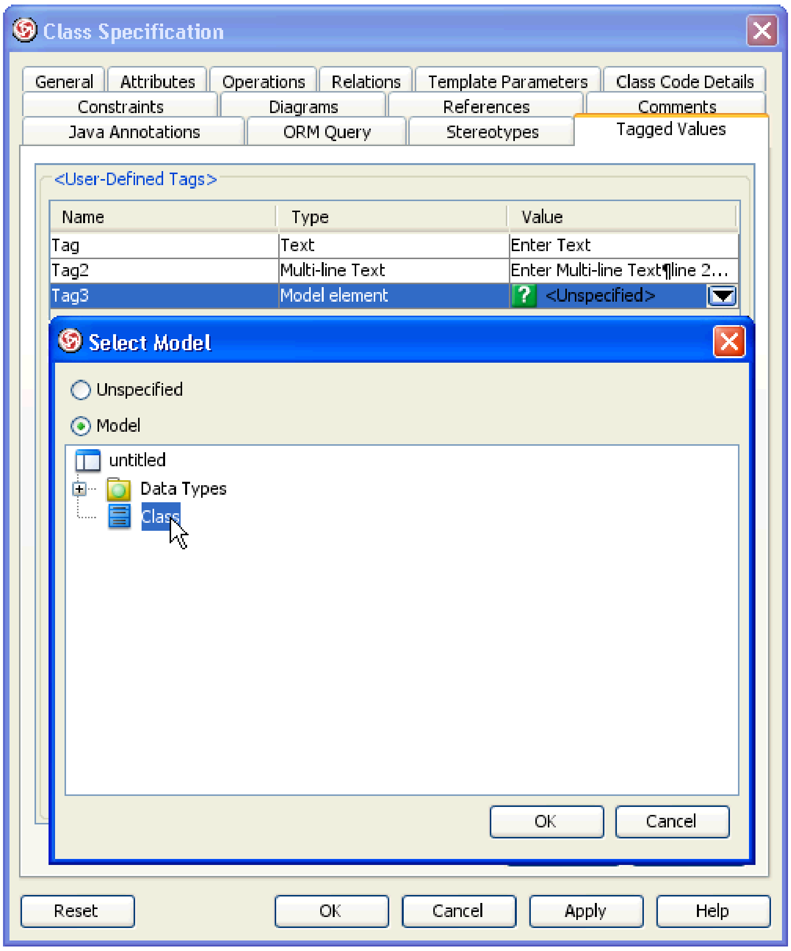
Edit Value

Value:

```
Enter Multi-line Text
line 2
line 3
```

Figure 1-70 *Multi-line Text*

5. **Model element** supports reference of model element. You can select the referenced model after clicking the ▼ button.



Name	Type	Value
Tag	Text	Enter Text
Tag2	Multi-line Text	Enter Multi-line Text¶line 2...
Tag3	Model element	<Unspecified>

○ Unspecified

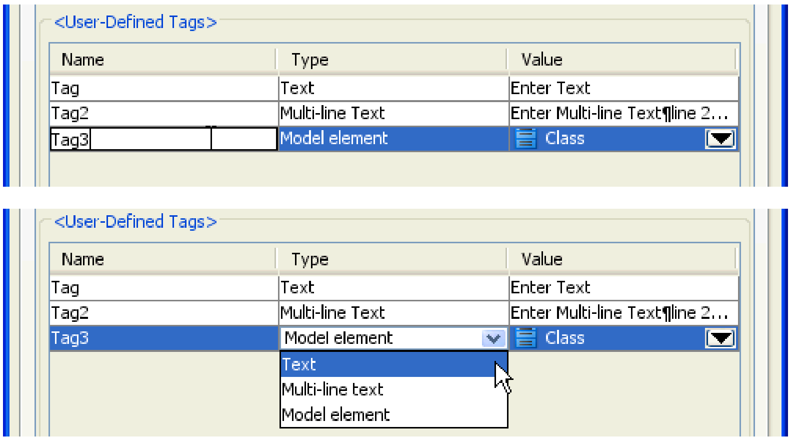
● Model

- untitled
  - Data Types
  - Class

Figure 1-71 Model element

Editing Tagged Values

You can edit the user-defined tagged value's name, type and value.

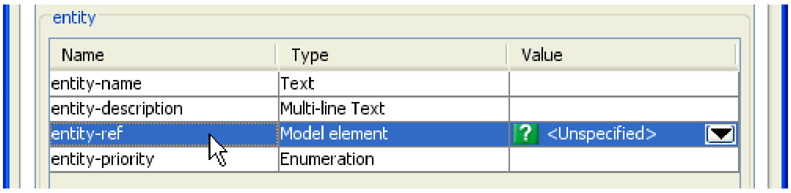


Name	Type	Value
Tag	Text	Enter Text
Tag2	Multi-line Text	Enter Multi-line Text¶line 2...
Tag3	Model element	Class

Name	Type	Value
Tag	Text	Enter Text
Tag2	Multi-line Text	Enter Multi-line Text¶line 2...
Tag3	Model element	Class

Figure 1-72 Can edit name, type and value on user-defined tagged value

But you can only edit the value of stereotype's tagged value. The name and type are not editable because they are defined in stereotype.



entity

Name	Type	Value
entity-name	Text	
entity-description	Multi-line Text	
entity-ref	Model element	? <Unspecified>
entity-priority	Enumeration	

Figure 1-73 *Only can edit value on stereotype-defined tagged value*

Spell Checking

The Spell Checking feature supports spell checking in all inline editing, as well as in Textual Analysis. It provides intelligent suggestions for words, and you can add your own words into your personal dictionary.

Correcting Spelling Mistakes

When you enter a word that is absent from the dictionary, the red wavy line appears to remind you there is a spelling mistake.

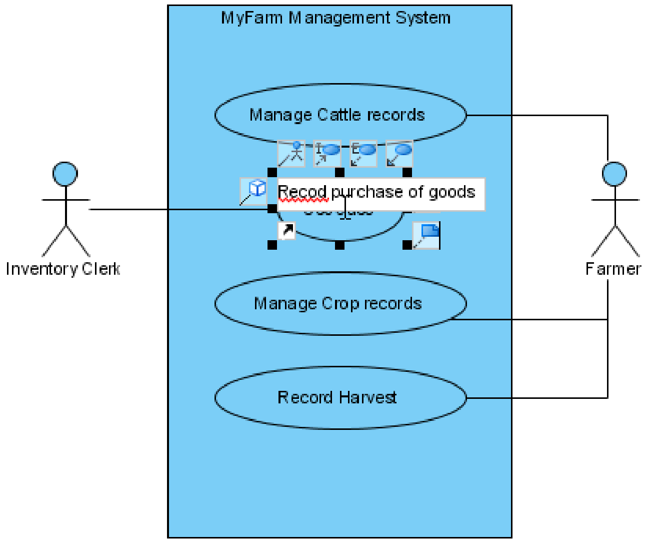


Figure 1-74 *Incorrect word is highlighted*

Right click on the incorrect word, a list of corrections will be shown on popup menu.

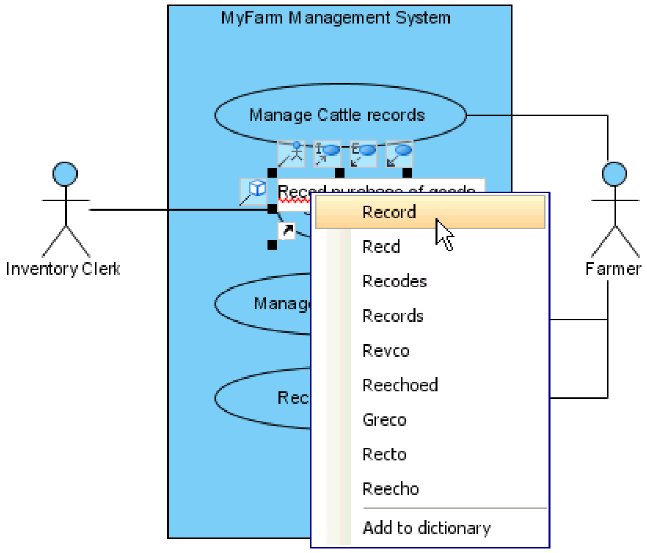


Figure 1-75 *A list of possible words are shown for correction*

After select the correction, the word is corrected and thus the red wavy line disappeared.

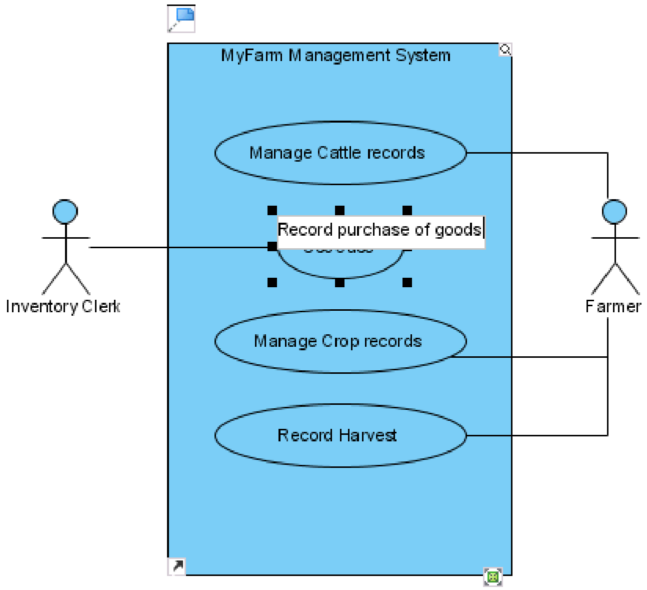


Figure 1-76 *Word is corrected, red wavy line is disappeared*

However, being underlined does not necessary mean that is an incorrect word. For example, the name of company may not appear in the dictionary. In such case, you may want to add the word to the dictionary.

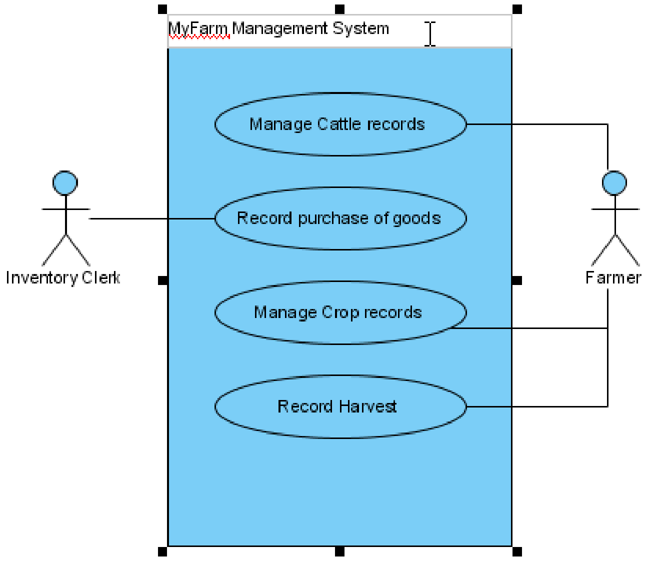


Figure 1-77 *You may want to define a word as correct word*

To add the word into dictionary. Right click on the word. Select **Add to dictionary** on popup menu.

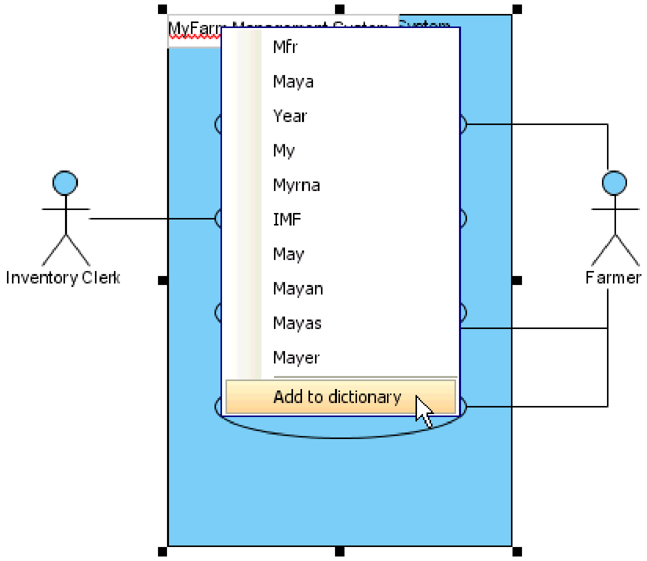


Figure 1-78 *Select Add to dictionary*

Since the word has been added to the dictionary, it will be treated as a correct word from now on.

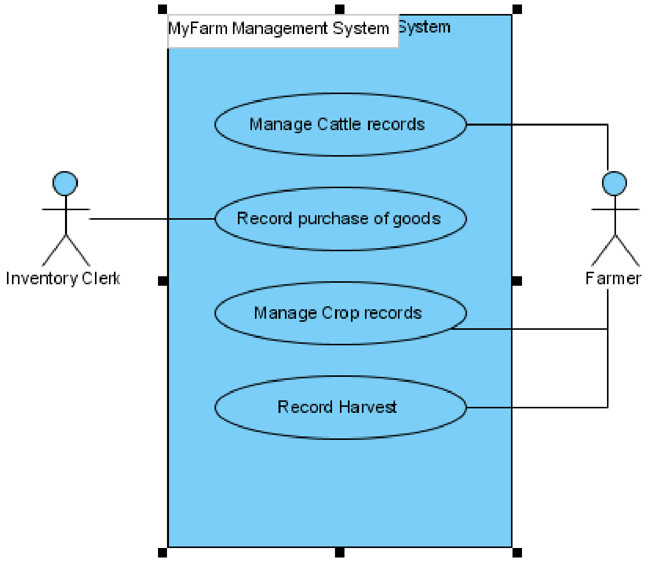
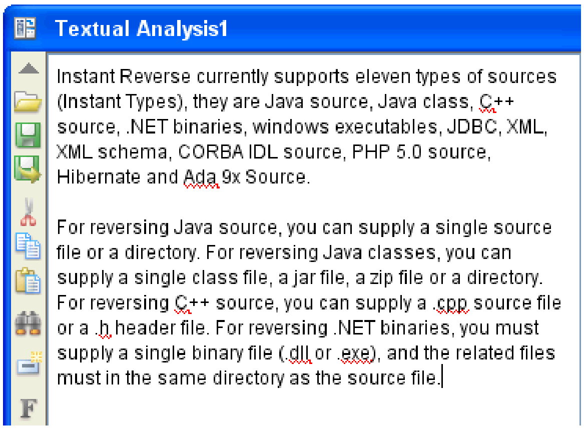


Figure 1-79 *Word is included in dictionary, red wavy line is disappeared*

Turn on and off of Spell Checking

On some case, you may want to turn off spell checking. For Example, contents in textual analysis contain a lot of names that are not included in dictionary. You may be annoyed with a lot of red wavy lines.

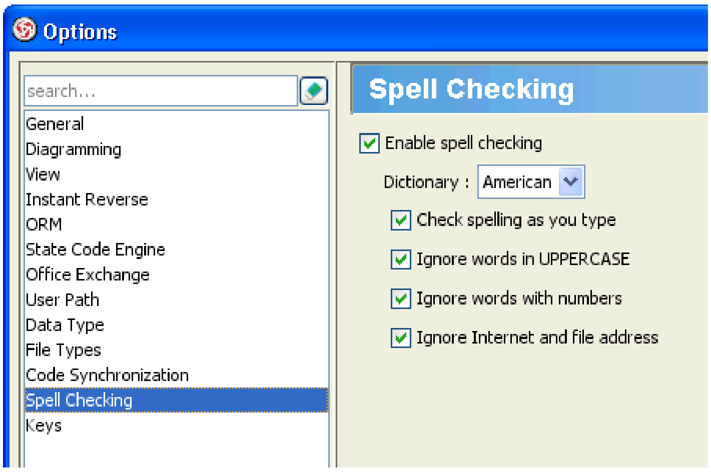


Instant Reverse currently supports eleven types of sources (Instant Types), they are Java source, Java class, C++ source, .NET binaries, windows executables, JDBC, XML, XML schema, CORBA IDL source, PHP 5.0 source, Hibernate and Ada 9x Source.

For reversing Java source, you can supply a single source file or a directory. For reversing Java classes, you can supply a single class file, a jar file, a zip file or a directory. For reversing C++ source, you can supply a .cpp source file or a .h header file. For reversing .NET binaries, you must supply a single binary file (.dll or .exe), and the related files must in the same directory as the source file.

Figure 1-80 Textual Analysis with a lot of red wavy lines

1.To turn off spell checking. Select **Tools > Options...** to open Options dialog.



☑ Enable spell checking

Dictionary : American

☑ Check spelling as you type

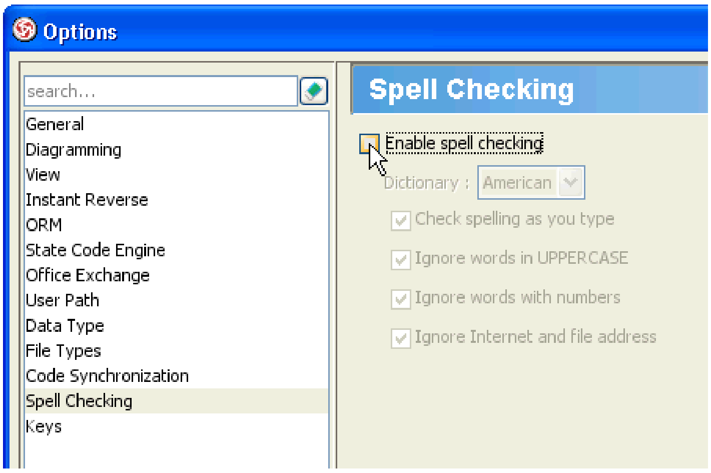
☑ Ignore words in UPPERCASE

☑ Ignore words with numbers

☑ Ignore Internet and file address

Figure 1-81 Options dialog

2.Select **Spell Checking**, uncheck **Enable spell checking** to turn off spell checking.



☐ Enable spell checking

Dictionary : American

☑ Check spelling as you type

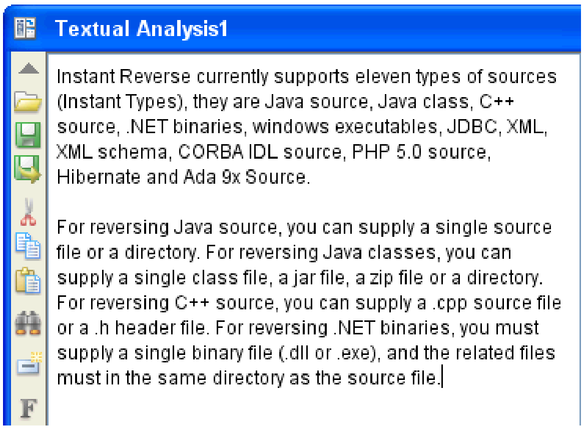
☑ Ignore words in UPPERCASE

☑ Ignore words with numbers

☑ Ignore Internet and file address

Figure 1-82 Disable spell checking

Spell Checking is turned off, no red wavy lines are shown.



Instant Reverse currently supports eleven types of sources (Instant Types), they are Java source, Java class, C++ source, .NET binaries, windows executables, JDBC, XML, XML schema, CORBA IDL source, PHP 5.0 source, Hibernate and Ada 9x Source.

For reversing Java source, you can supply a single source file or a directory. For reversing Java classes, you can supply a single class file, a jar file, a zip file or a directory. For reversing C++ source, you can supply a .cpp source file or a .h header file. For reversing .NET binaries, you must supply a single binary file (.dll or .exe), and the related files must in the same directory as the source file.

Figure 1-83 *Spell checking is disabled*

Automatic Diagram Layout

VP-UML provides a layout facility for arranging diagram elements in diagrams. It re-layouts the diagram elements so that they do not overlap, and the relationship links are arranged so that they will not cross over one another. Different layout styles and configurable options are provided, which allows for very flexible and sophisticated layouts to be adopted for diagrams.

Automatic Layout Diagram

There are different kinds of layout: **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**, **Smart Organic Layout**

Auto Layout

Auto Layout can arrange the shapes by selecting the most suitable layout automatically. It is best for arranging the shapes when user has no special preference in choosing a specific layout. To apply **Auto Layout** to the diagram, right-click on the diagram and select **Layout > Auto Layout** from the popup menu.

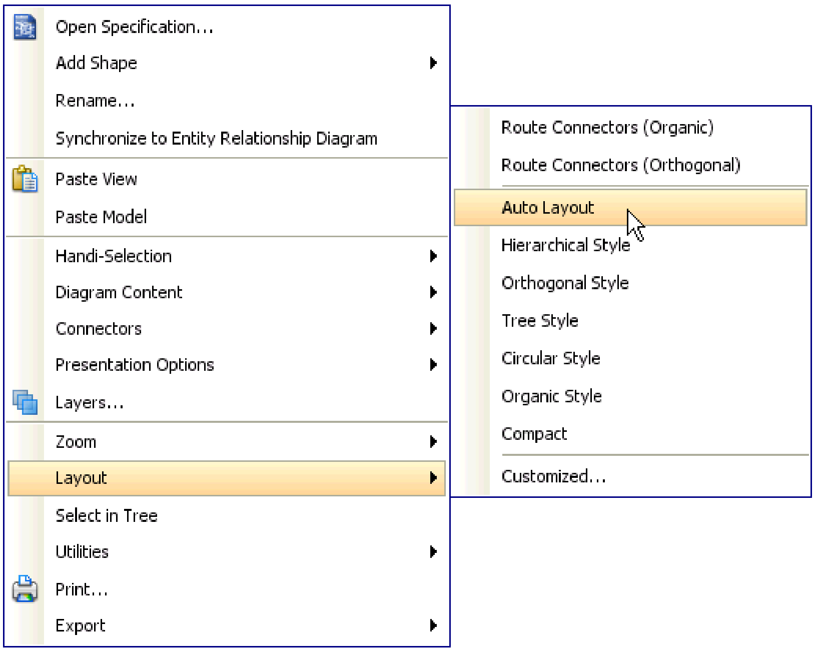


Figure 2-1 *Select Auto Layout*

Class Diagram (Hierarchy base / Factory class diagram)

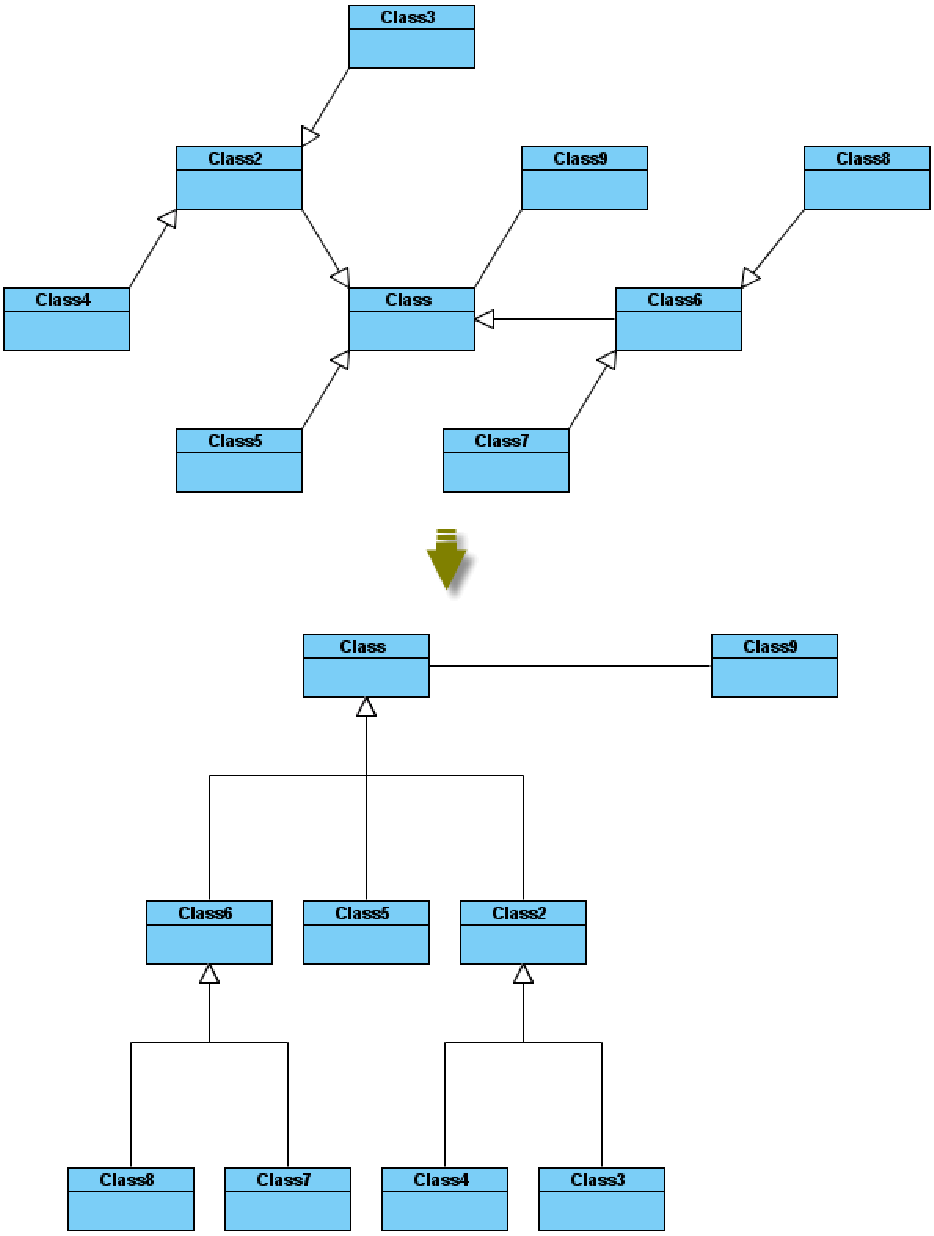


Figure 2-2 *Hierarchy base (Factory class diagram)*

Class Diagram (Navigation base / Mediator class diagram)

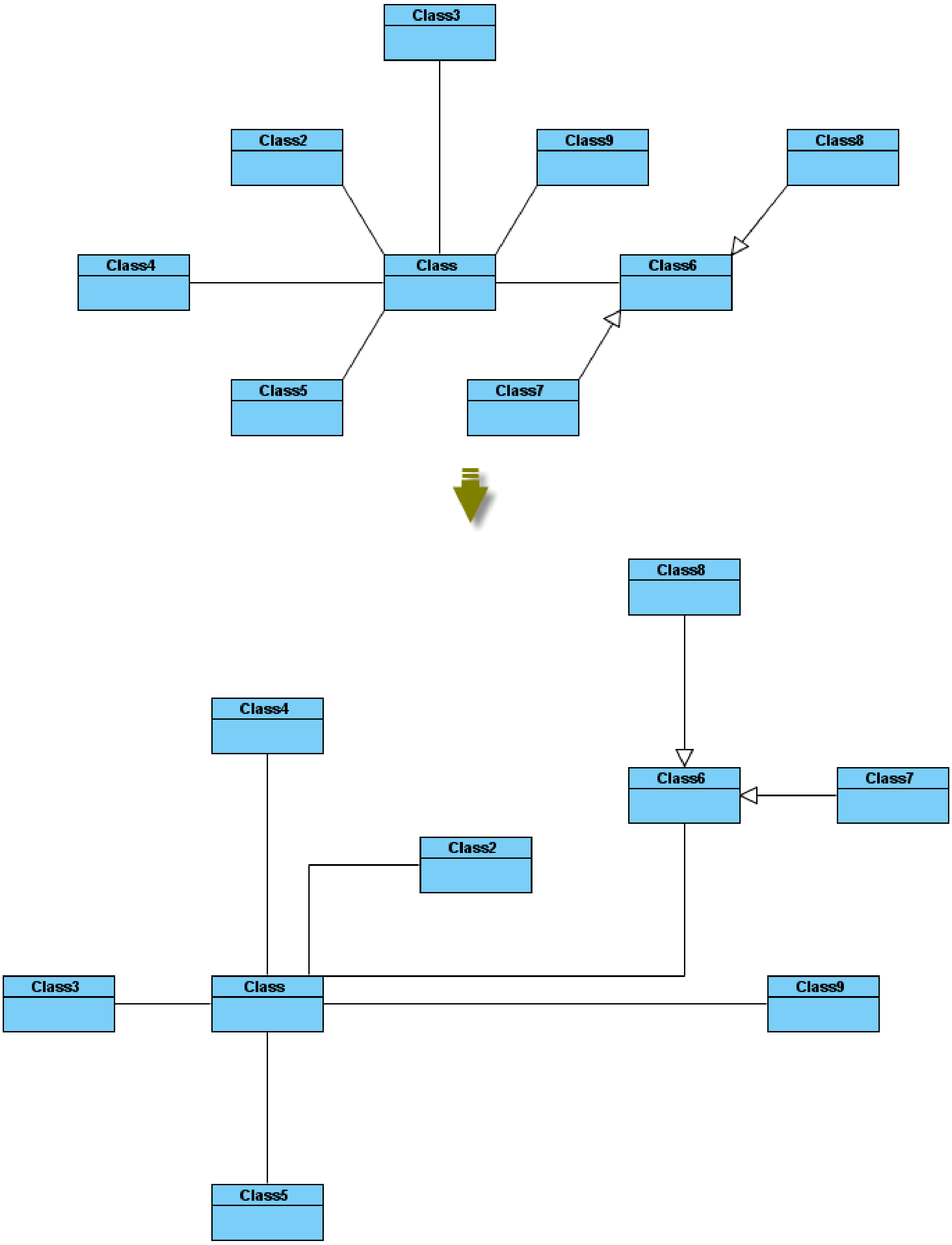


Figure 2-3 *Navigation base (Mediator class diagram)*

Activity Diagram

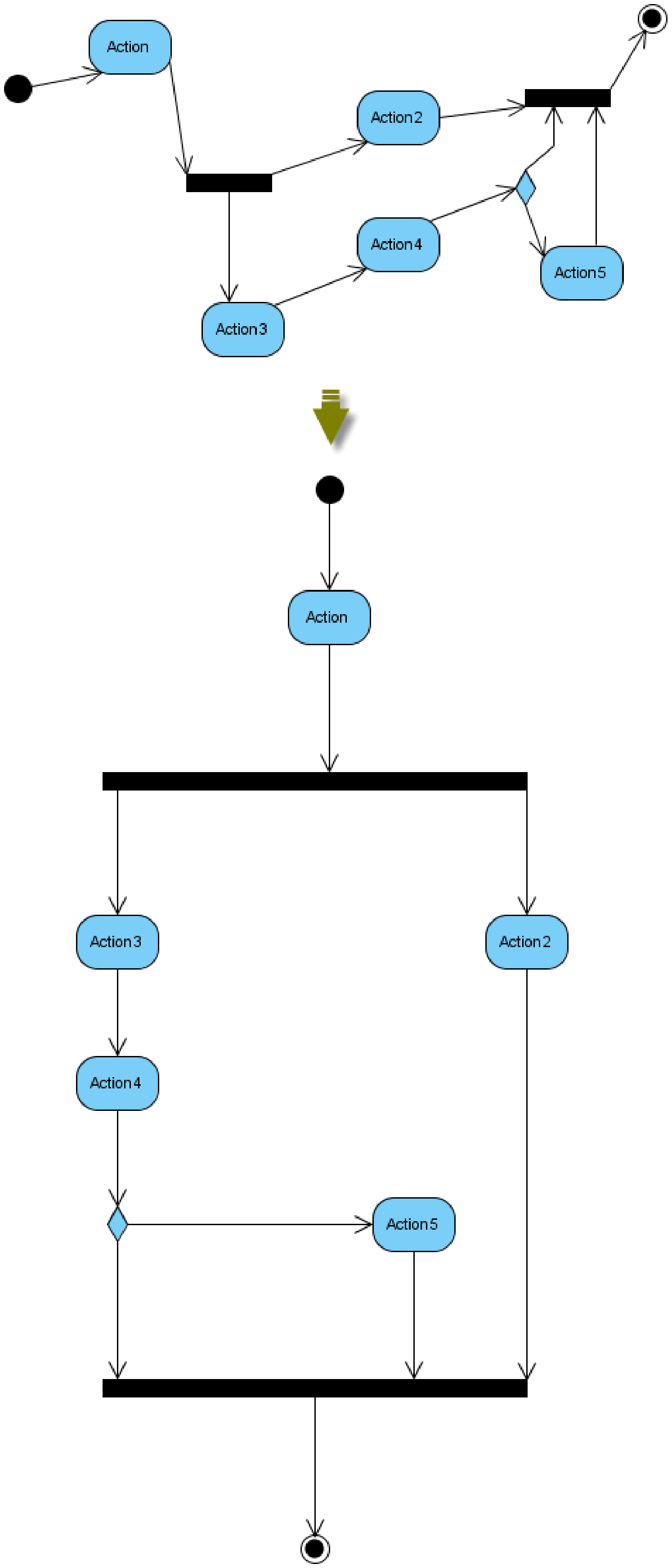


Figure 2-4 Auto Layout of activity diagram

State Machine Diagram

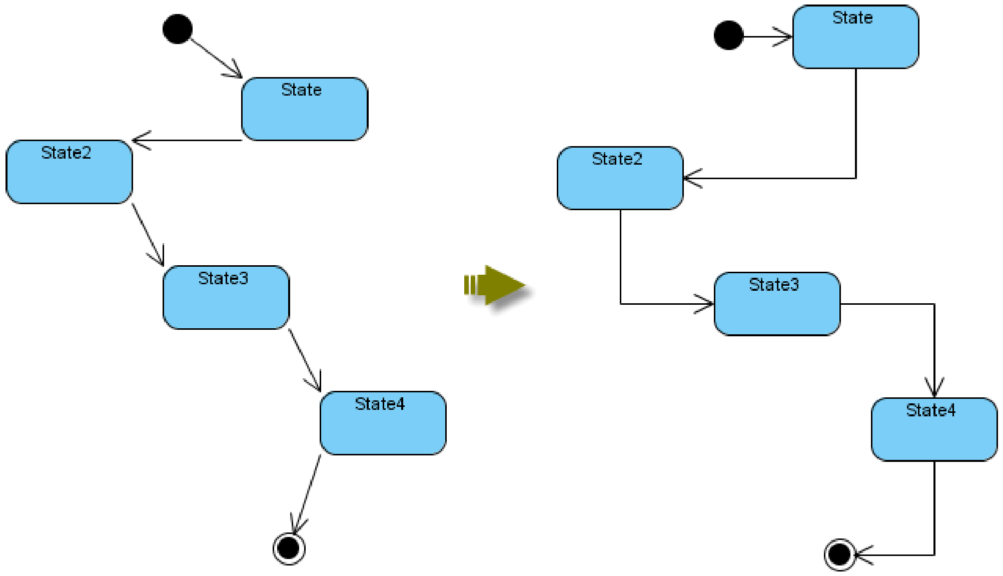


Figure 2-5 Auto layout of state machine diagram

Communication Diagram

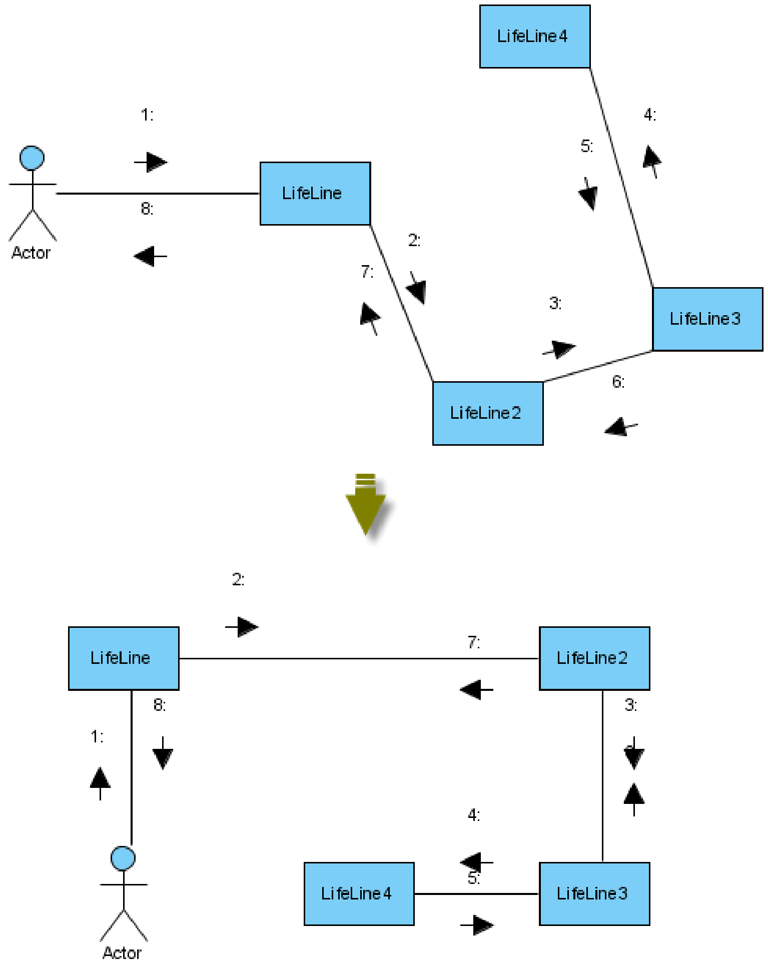


Figure 2-6 Auto layout of communication diagram

Other Diagrams

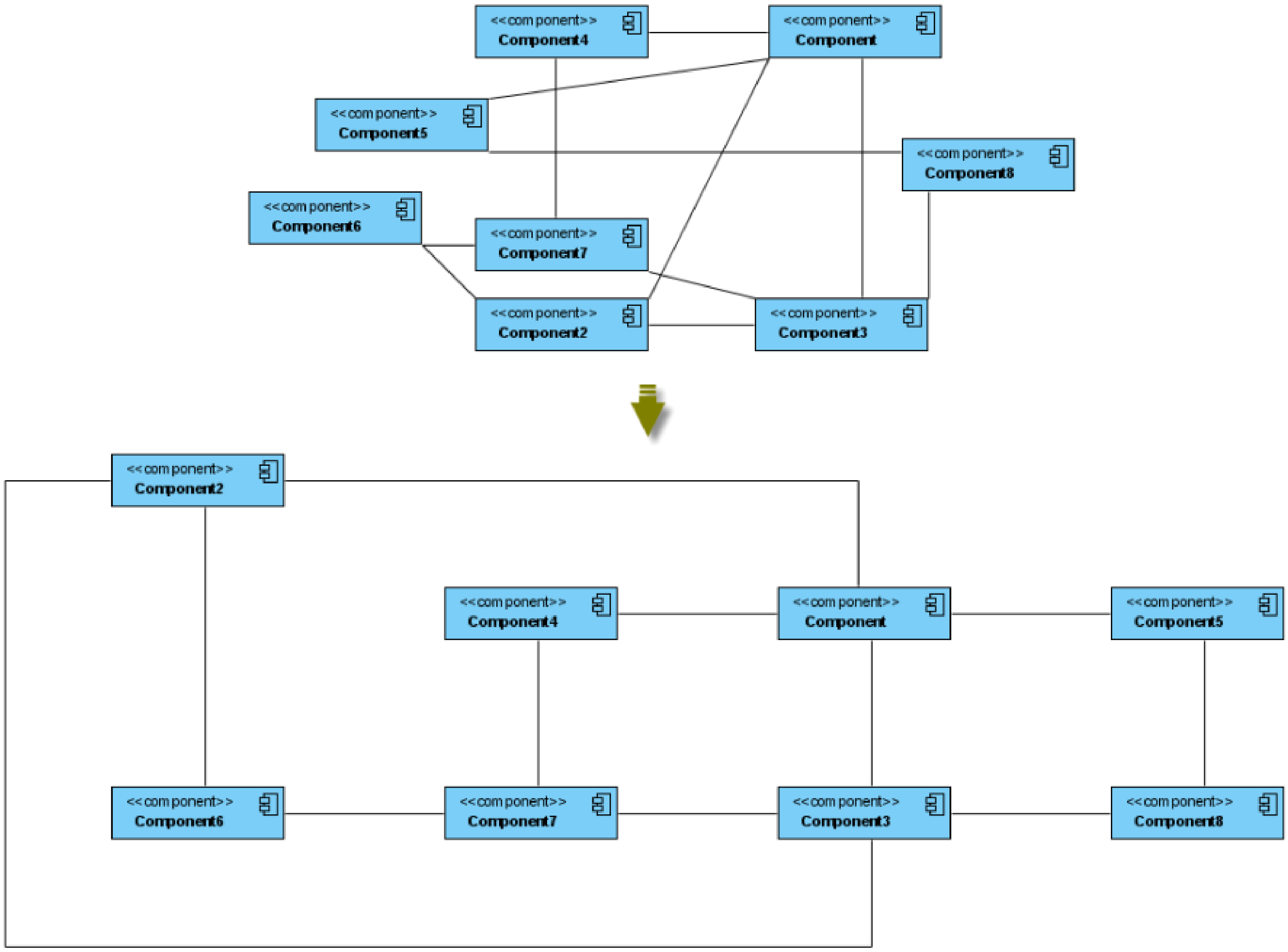
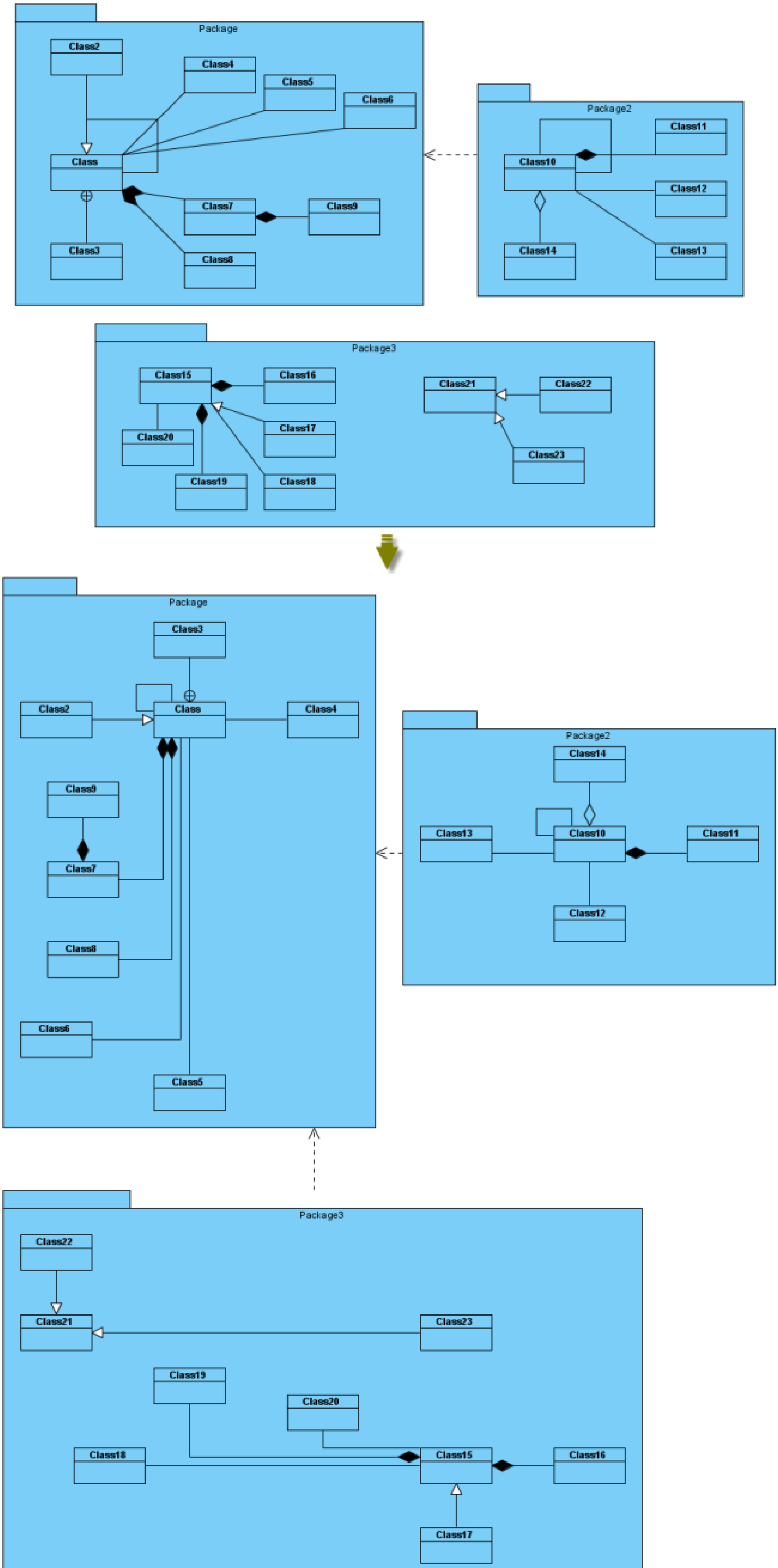


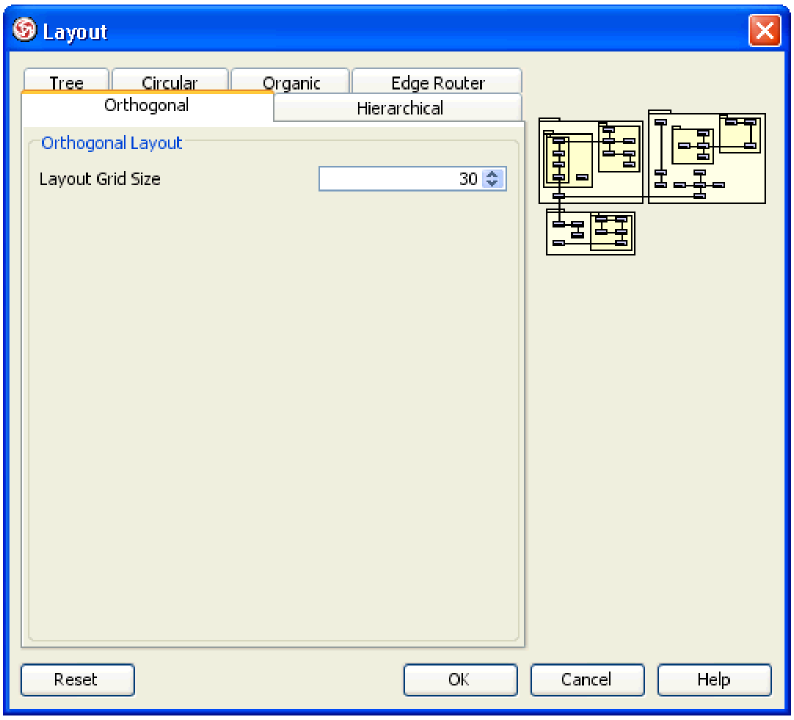
***Figure 2-7** Auto layout of other diagrams*

Orthogonal Layout

Orthogonal Layout arranges shapes based on the topology-shape-metrics approach. It is best of arranging shapes and connectors in Class Diagrams. It is default layout in VP-UML. 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.



Layout Grid Size: the virtual grid size for layout. Each shape will be placed in a way so that its center point lays on a virtual grid point.



Layout Grid Size: 30

Figure 2-9 *Orthogonal Layout setting*

Hierarchic Layout

Hierarchic Layout arranges shapes in a flow. It is best for arranging shapes that have hierarchical relationships such as generalization relationships and realization relationships.

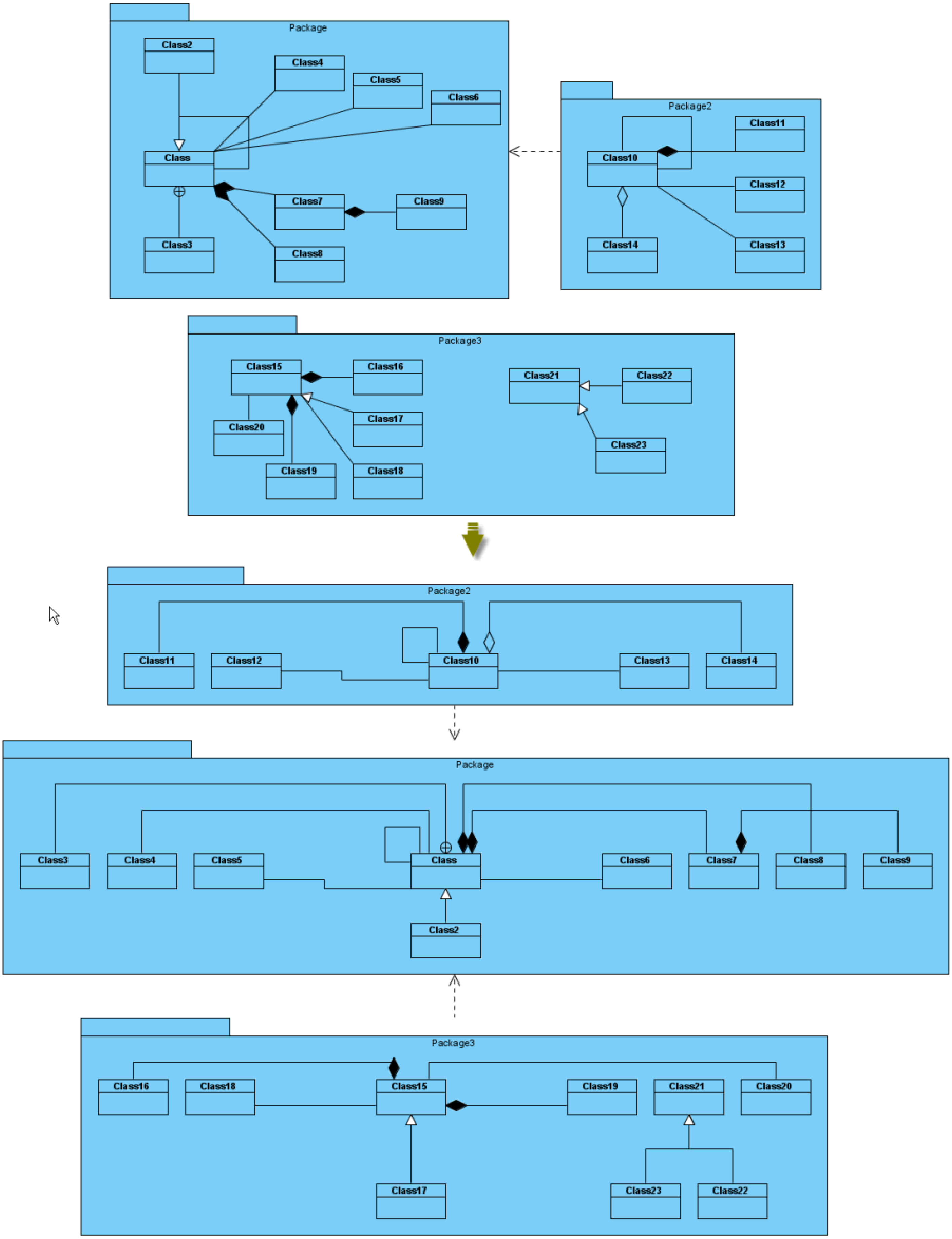
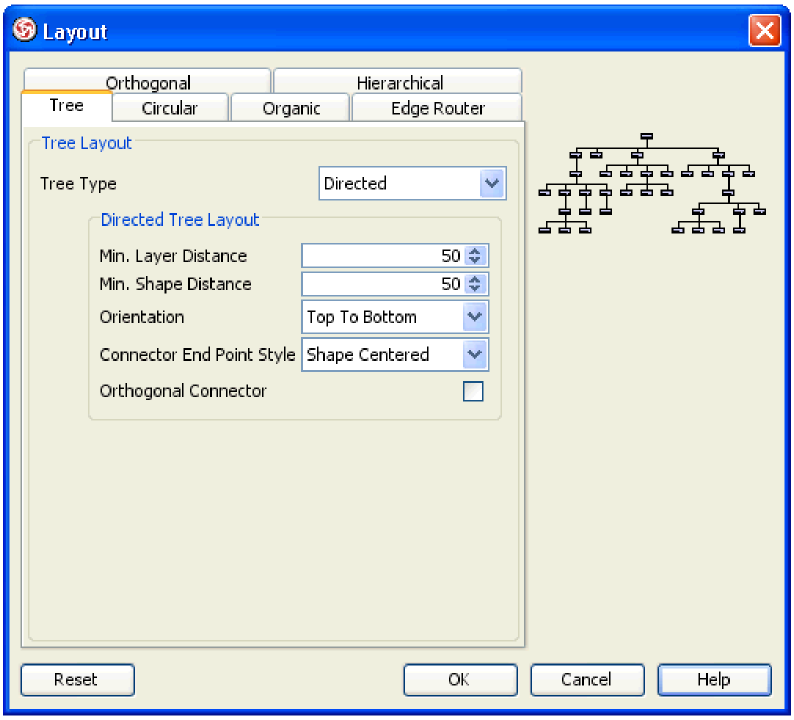


Figure 2-10 Hierarchic Layout

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 -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, simplex



Tree Layout

Tree Type: Directed

Directed Tree Layout

Min. Layer Distance: 50
Min. Shape Distance: 50
Orientation: Top To Bottom
Connector End Point Style: Shape Centered
Orthogonal Connector: □

***Figure 2-11** Hierarchic Layout setting*

Directed Tree Layout

Directed Tree Layout is one of the tree layouts in VP-UML. It can arrange shapes in a tree structure. It is best for arranging shapes except those which have hierarchical relationships such as generalization relationships and realization relationships.

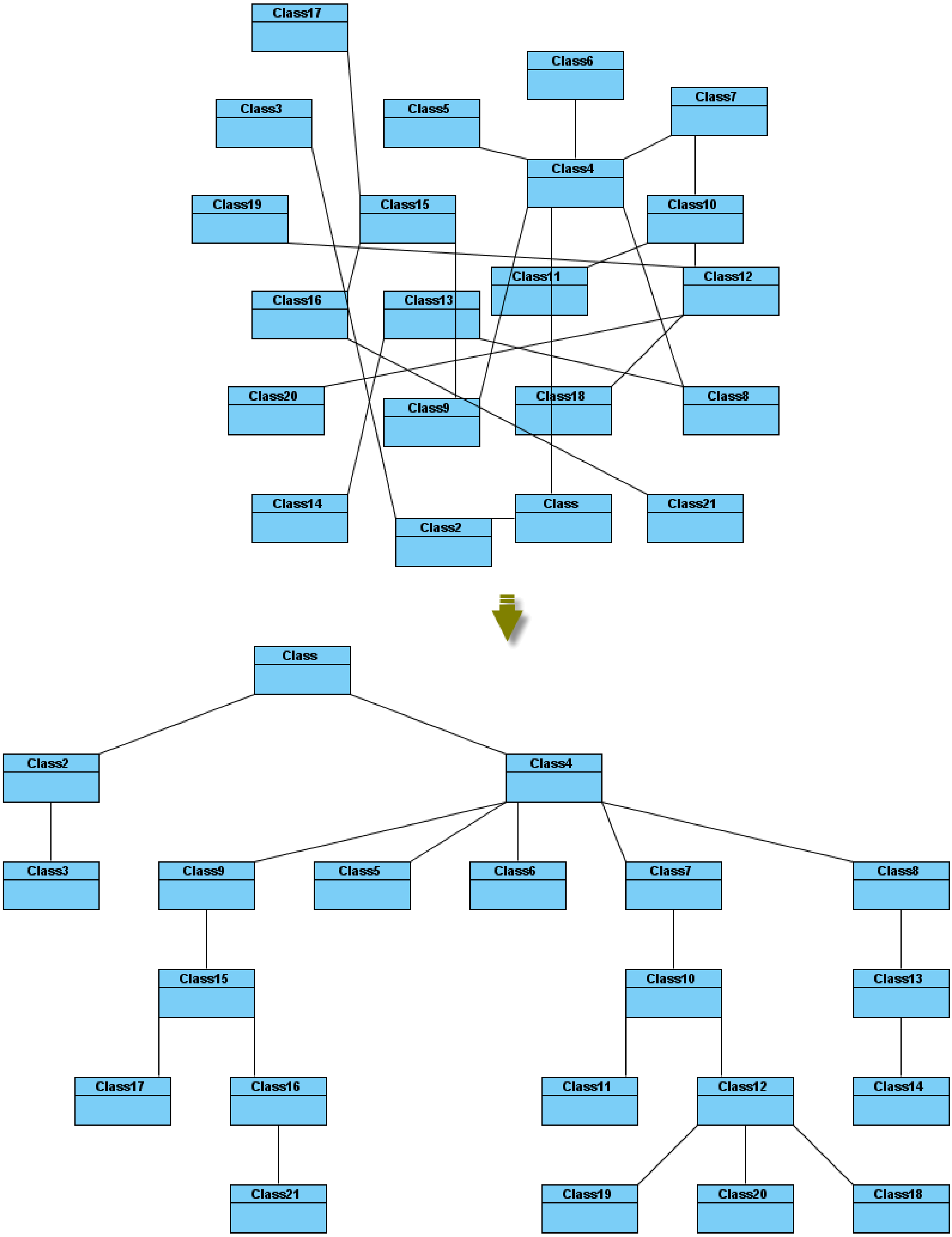
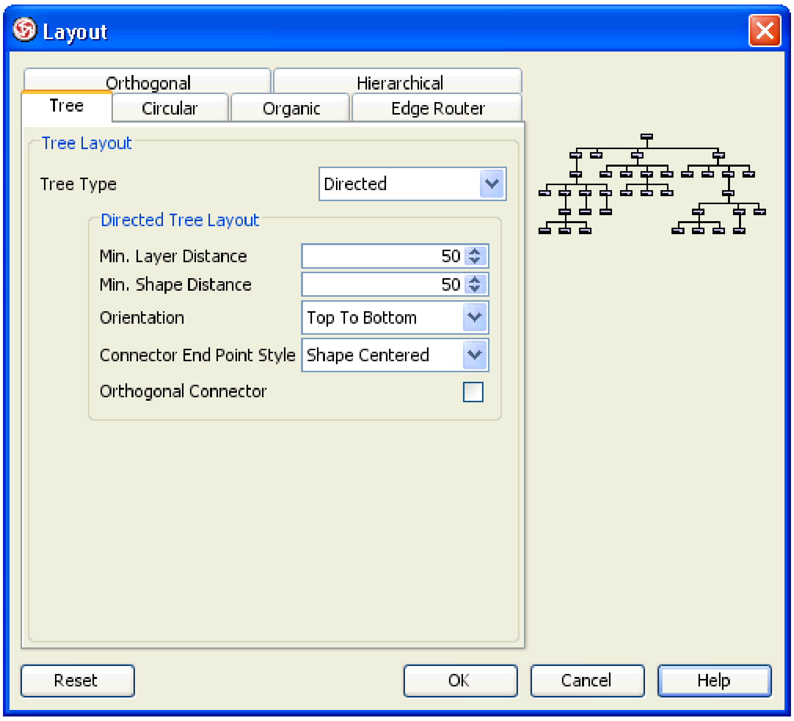


Figure 2-12 *Directed Tree Layout*

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 &ndash; 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 &ndash; shape centered, border centered, border distributed
Orthogonal Connector: whether the connectors will be arranged in orthogonal



Tree Type: Directed

Min. Layer Distance: 50

Min. Shape Distance: 50

Orientation: Top To Bottom

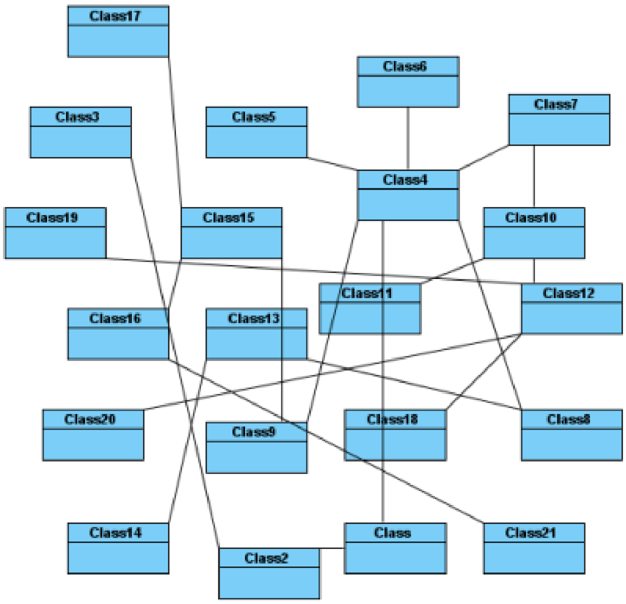
Connector End Point Style: Shape Centered

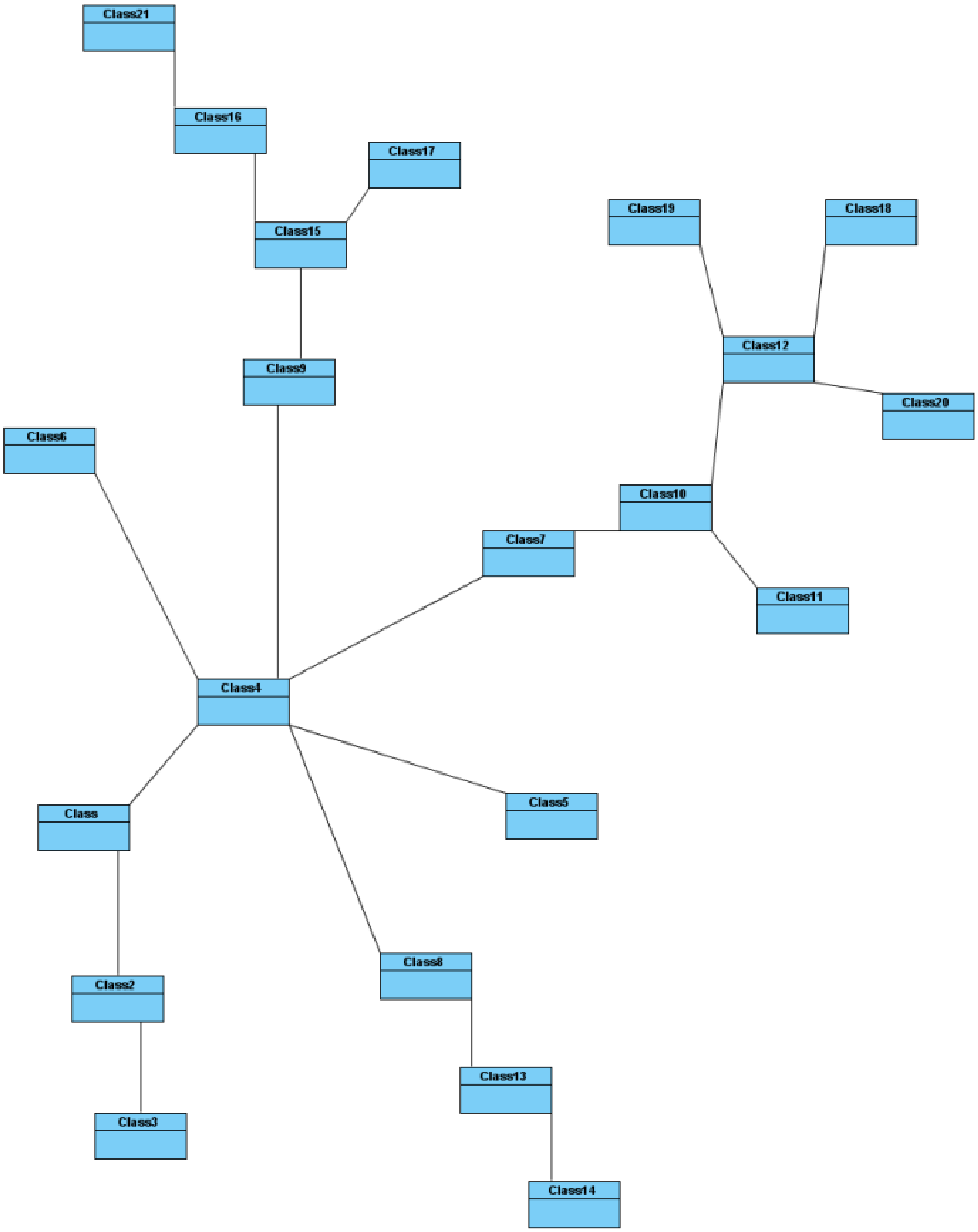
Orthogonal Connector: □

Figure 2-13 *Directed Tree Layout setting*

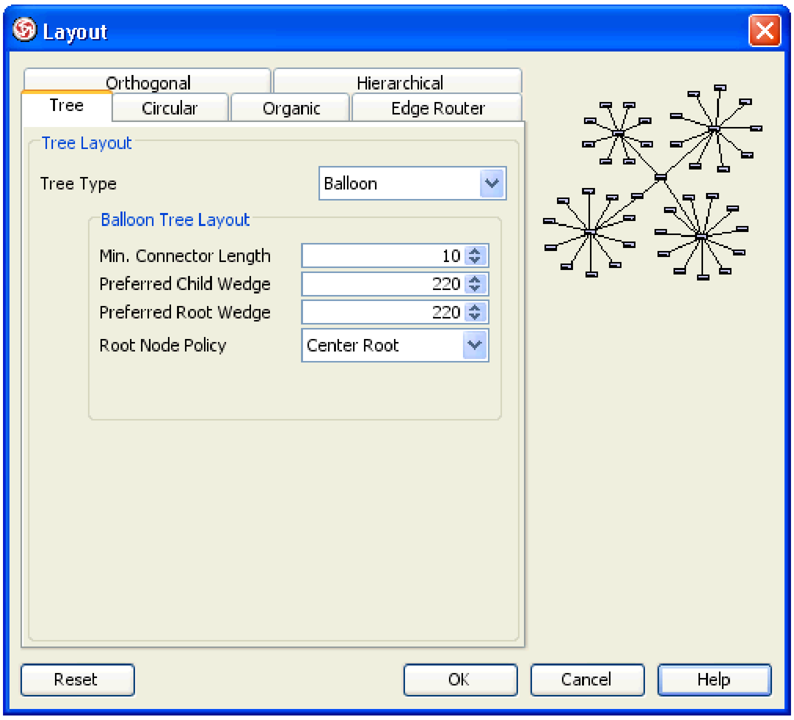
Balloon Tree Layout

Balloon Tree Layout is one of the tree layouts in VP-UML. It can arrange shapes in a tree structure in a radial fashion. It is best for arranging large trees.





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 &ndash; directed root, center root, and weighted center root



Tree Layout

Tree Type: Balloon

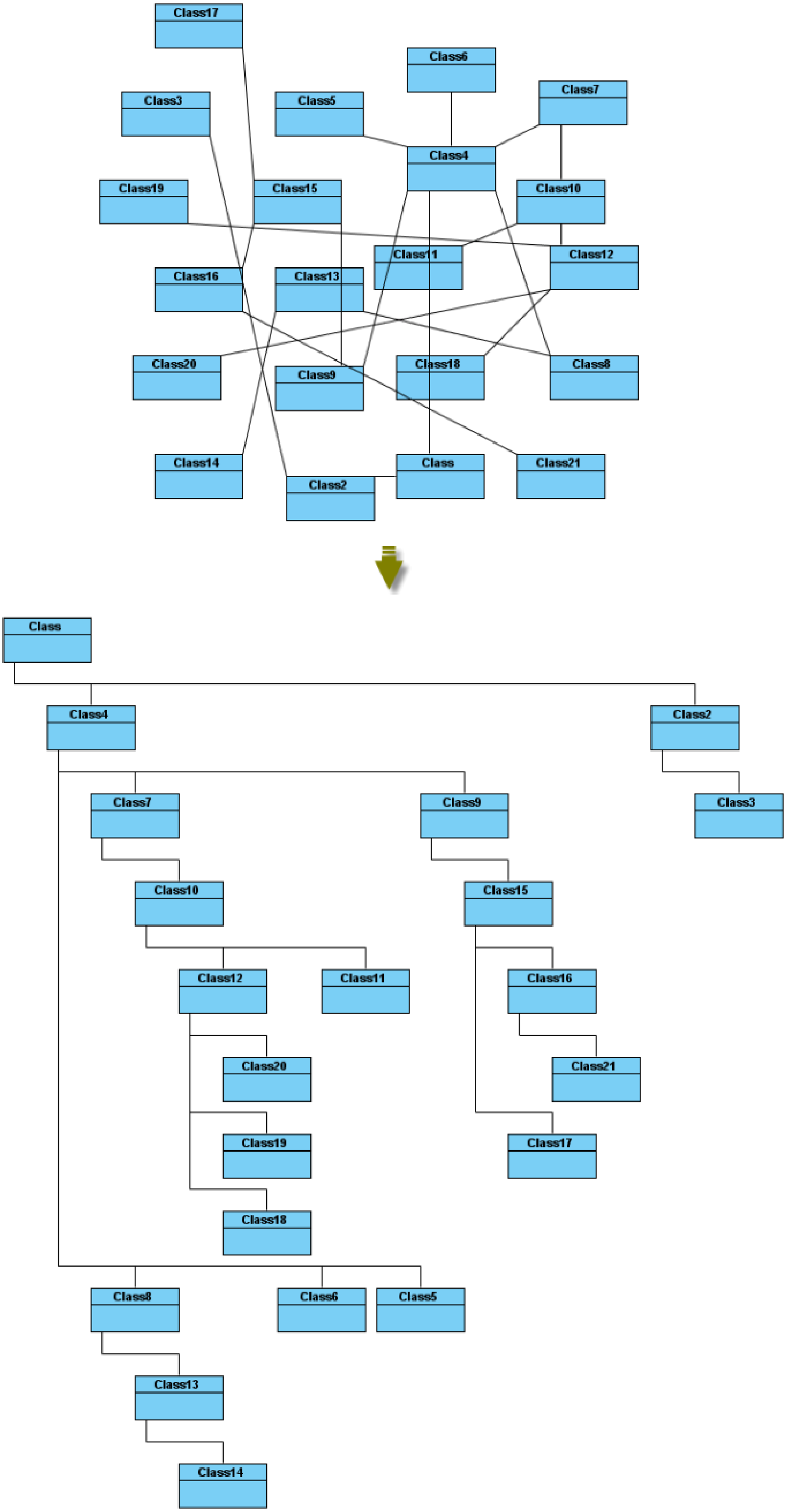
Balloon Tree Layout

Min. Connector Length: 10
Preferred Child Wedge: 220
Preferred Root Wedge: 220
Root Node Policy: Center Root

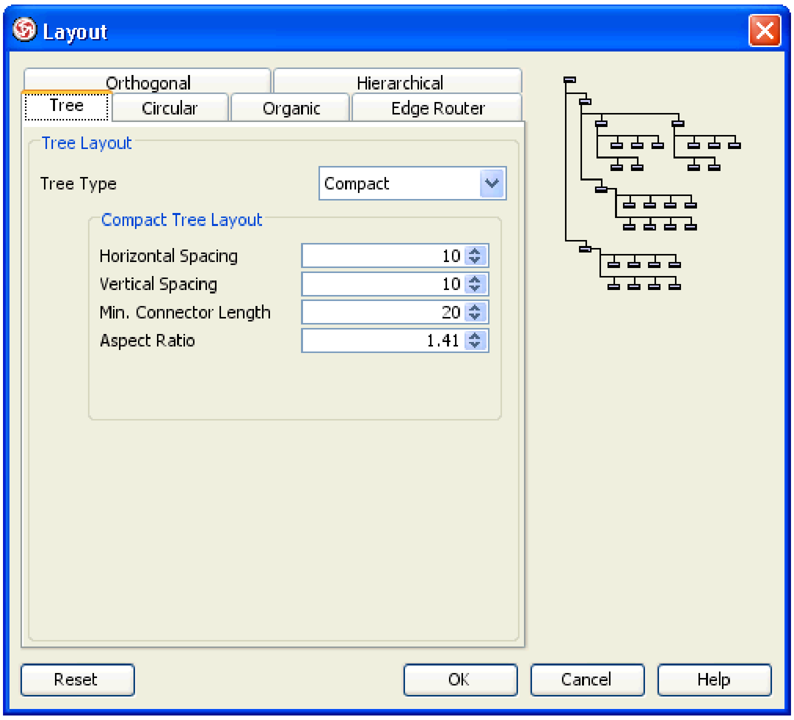
Figure 2-15 *Balloon Tree Layout setting*

Compact Tree Layout

Compact Tree Layout is one of the tree layouts in VP-UML. It can arrange shapes in a tree structure. You can set the aspect ratio (relation of tree width to tree height) of the resultant tree.



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 he tree height



Tree Layout

Tree Type: Compact

Compact Tree Layout

Horizontal Spacing	10
Vertical Spacing	10
Min. Connector Length	20
Aspect Ratio	1.41

Figure 2-17 *Compact Tree Layout setting*

Horizontal-Vertical Tree Layout

Horizontal-Vertical Tree Layout is one of the tree layouts in VP-UML. It can arrange shapes in a tree structure horizontally and vertically.

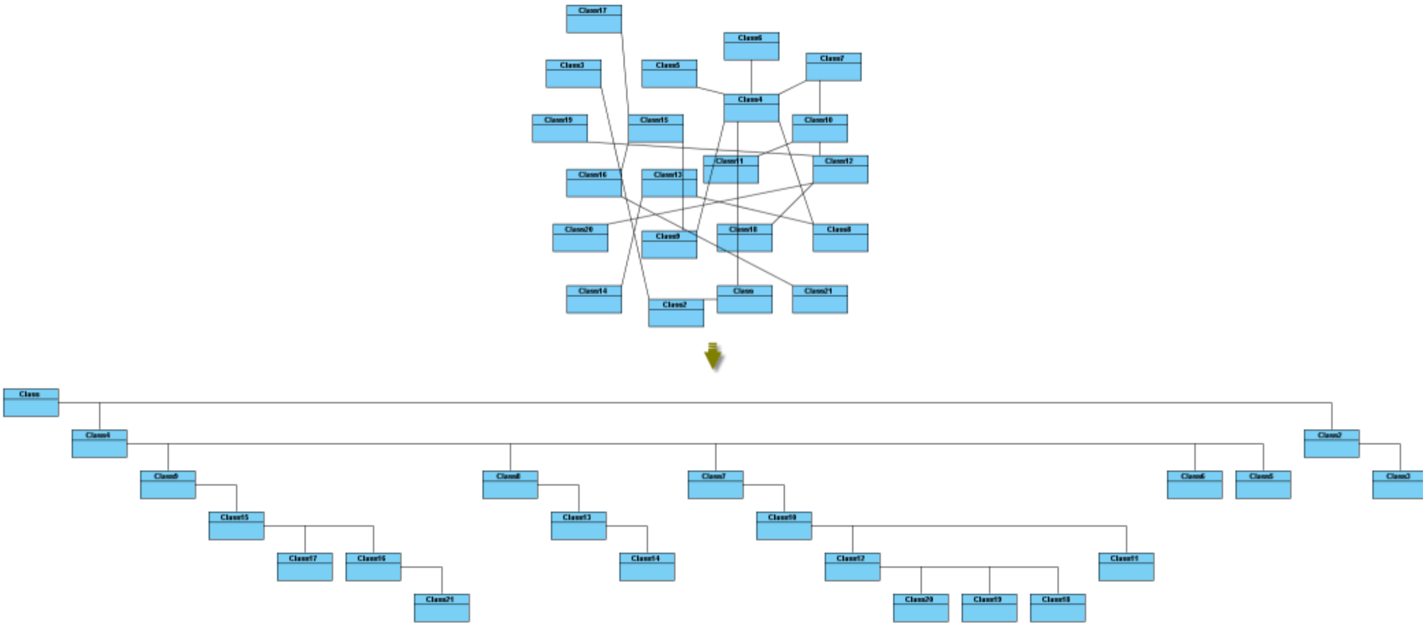
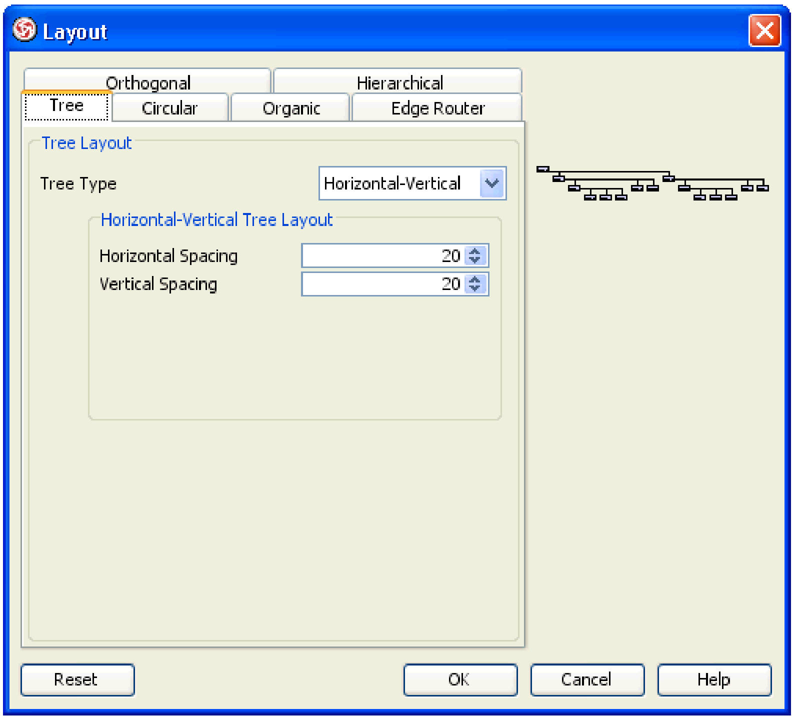


Figure 2-18 *Horizontal-Vertical Tree Layout*

Horizontal Spacing: the horizontal spacing between the shapes
Vertical Spacing: the vertical spacing between the shapes


Layout

Tree Layout

Tree Type: Horizontal-Vertical

Horizontal-Vertical Tree Layout

Horizontal Spacing: 20

Vertical Spacing: 20

Figure 2-19 Horizontal-Vertical Tree Layout setting

BBC Compact Circular Layout

BBC Compact Circular Layout is one of the circular layouts in VP-UML. It can arrange shapes in a radial tree structure. The detected group is laid out on the separate circles. It is best for arranging shapes that belongs to more than one group with a ring structure.

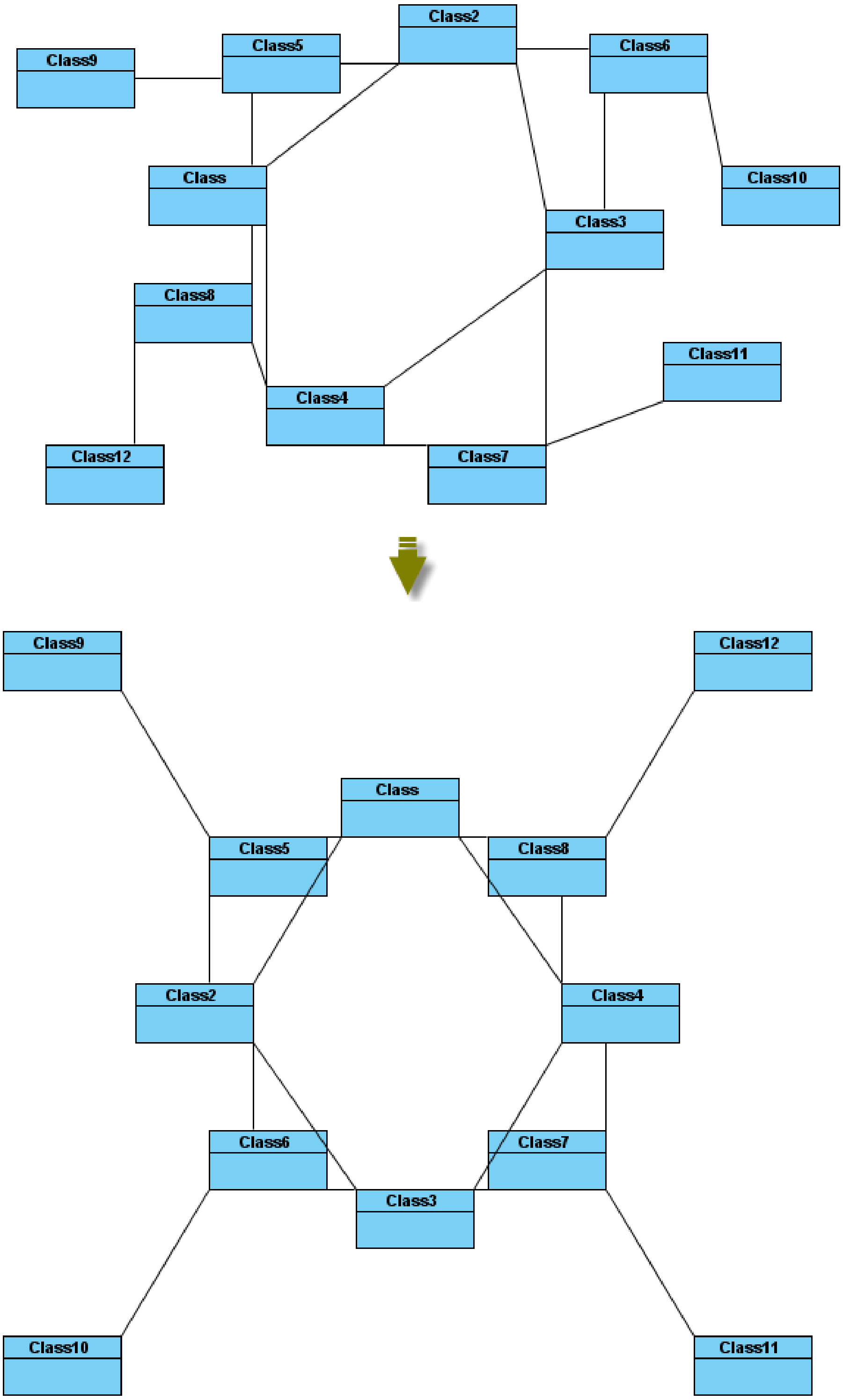
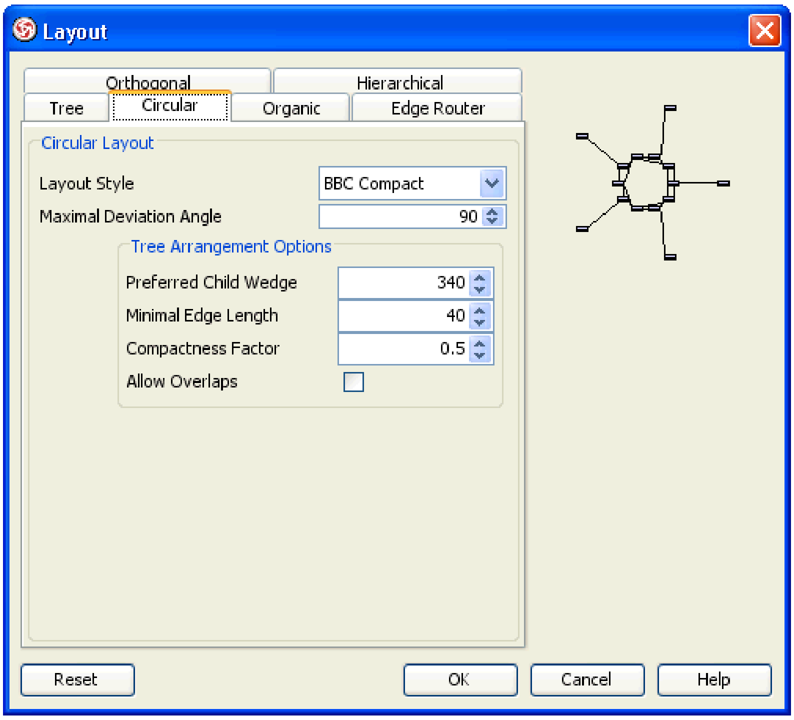


***Figure 2-20** BBC Compact Circular Layout*

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


Layout

Orthogonal | Hierarchical | Tree | Circular | Organic | Edge Router

Circular Layout

Layout Style: BBC Compact
Maximal Deviation Angle: 90

Tree Arrangement Options

Preferred Child Wedge: 340
Minimal Edge Length: 40
Compactness Factor: 0.5
Allow Overlaps: □

Figure 2-21 *BBC Compact Circular Layout setting*

BBC Isolated Circular Layout

BBC Isolated Circular Layout is one of the circular layouts in VP-UML. It can arrange shapes into many isolated ring structures. It is best for arranging shapes that belong to one group with ring structure.

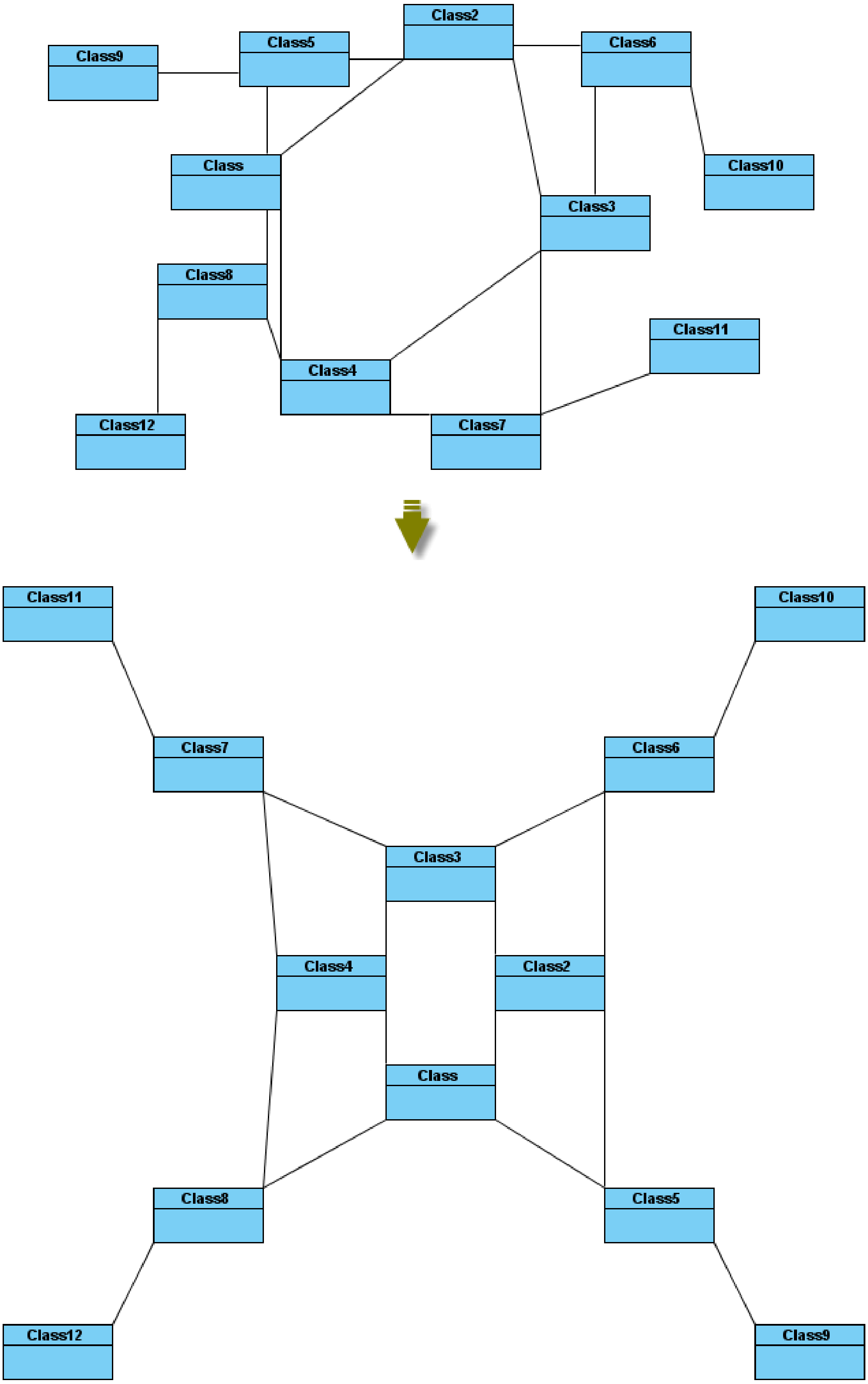
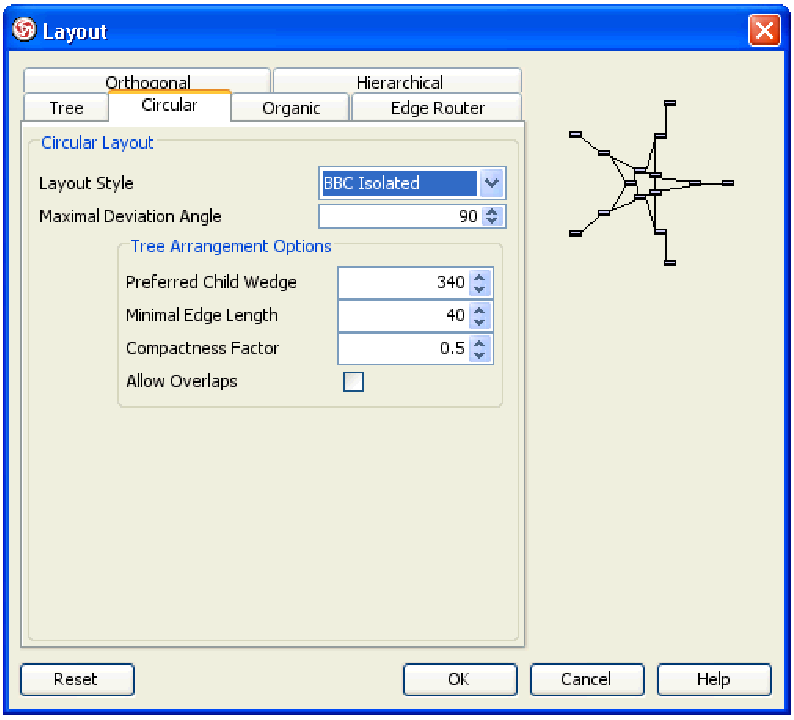


Figure 2-22 *BBC Isolated Circular Layout*

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



Circular Layout

Layout Style: BBC Isolated
Maximal Deviation Angle: 90

Tree Arrangement Options

Preferred Child Wedge: 340
Minimal Edge Length: 40
Compactness Factor: 0.5
Allow Overlaps: □

Figure 2-23 *BBC Isolated Circular Layout setting*

Single Cycle Circular Layout

Single Cycle Layout is one of the circular layouts in VP-UML. It can arrange shapes in circular strcture in single circle.

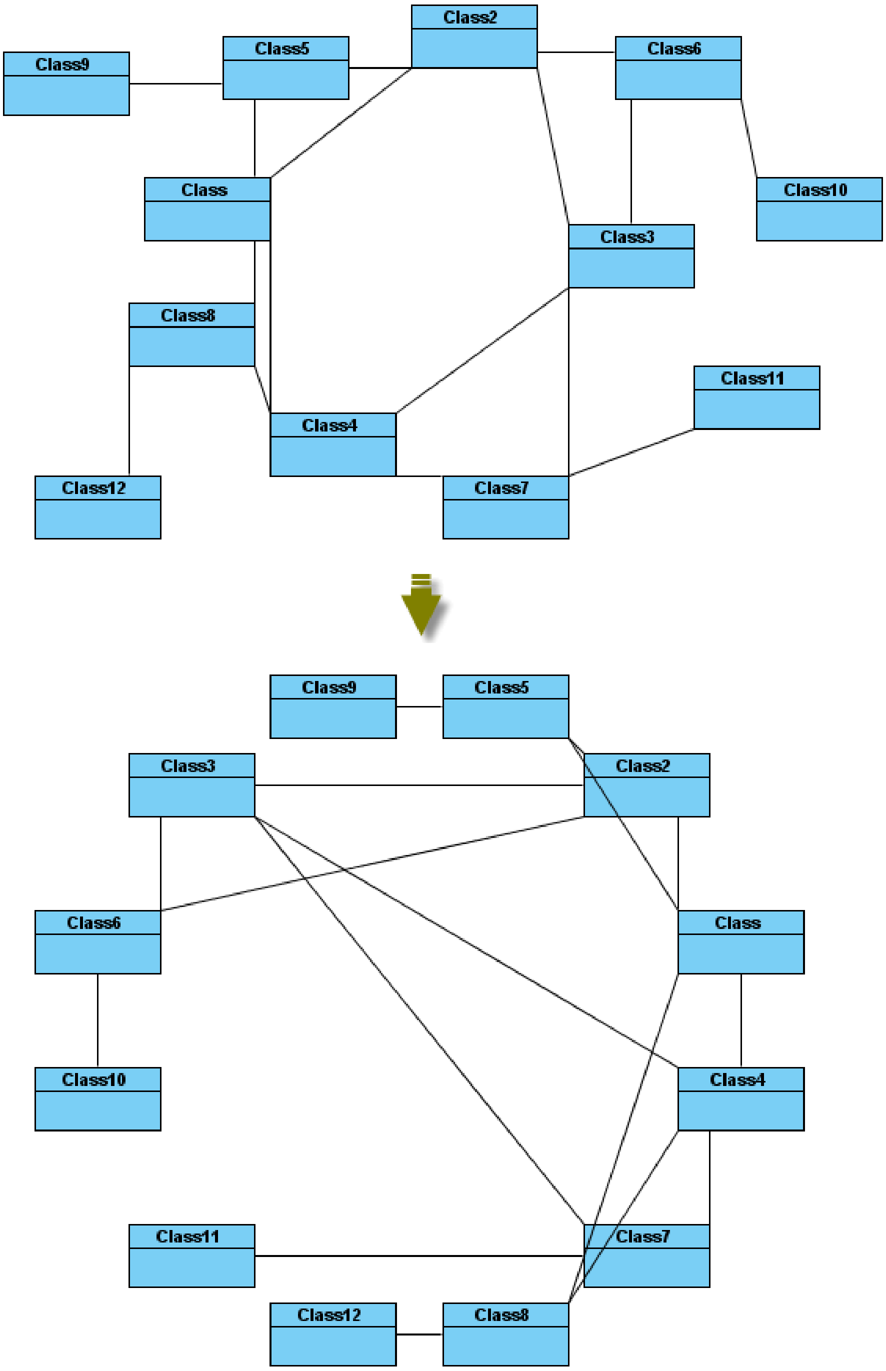
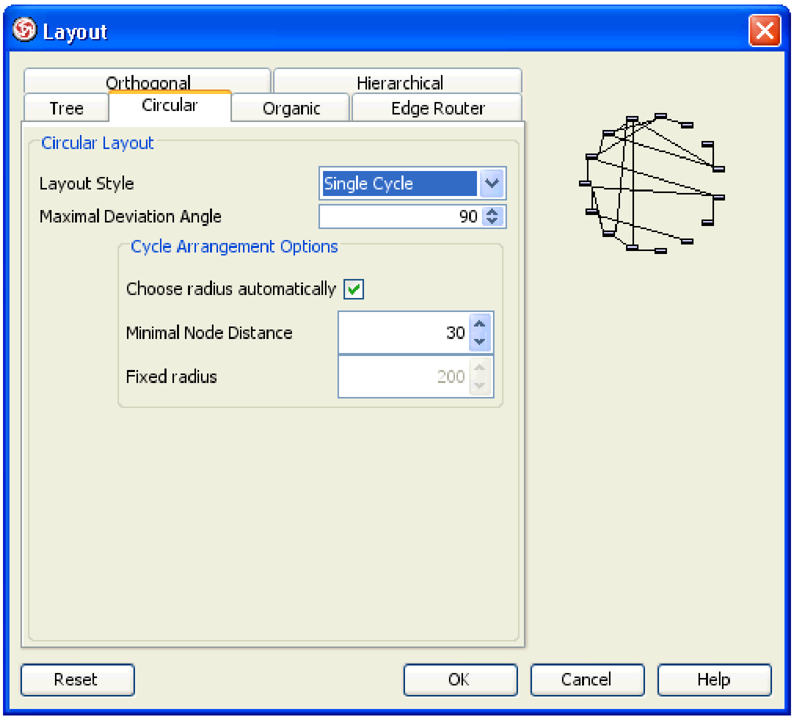


Figure 2-24 *Single Cycle Circular Layout*

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



Layout Style: Single Cycle

Maximal Deviation Angle: 90

Choose radius automatically ☑

Minimal Node Distance: 30

Fixed radius: 200

Figure 2-25 *Single Cycle Circular Layout setting*

Organic Layout

Organic Layout is one of the organic layouts in VP-UML. It can arrange shapes in a star or ring structure. It is best for arranging the shapes that have highly connectivity relationship.

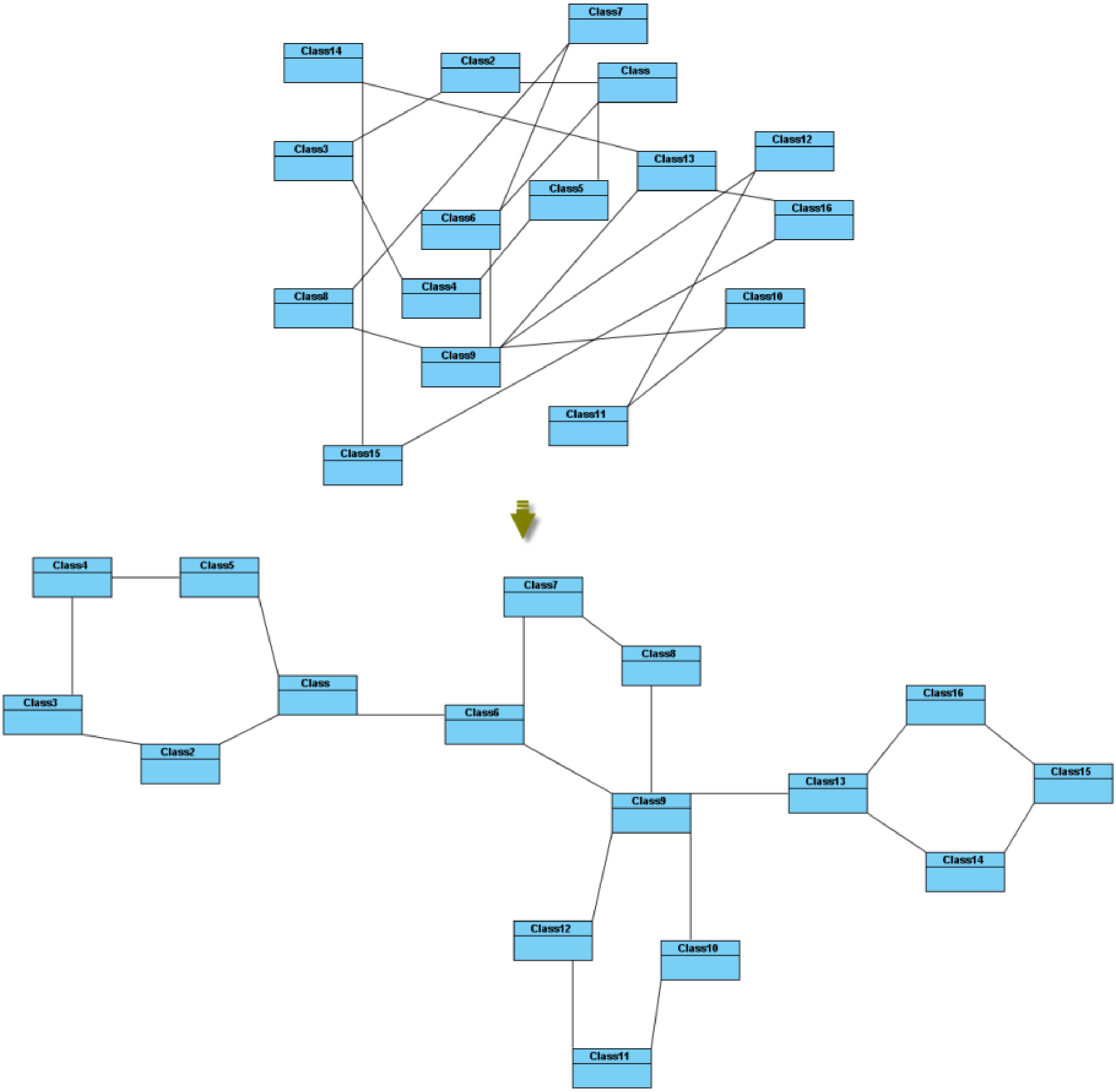
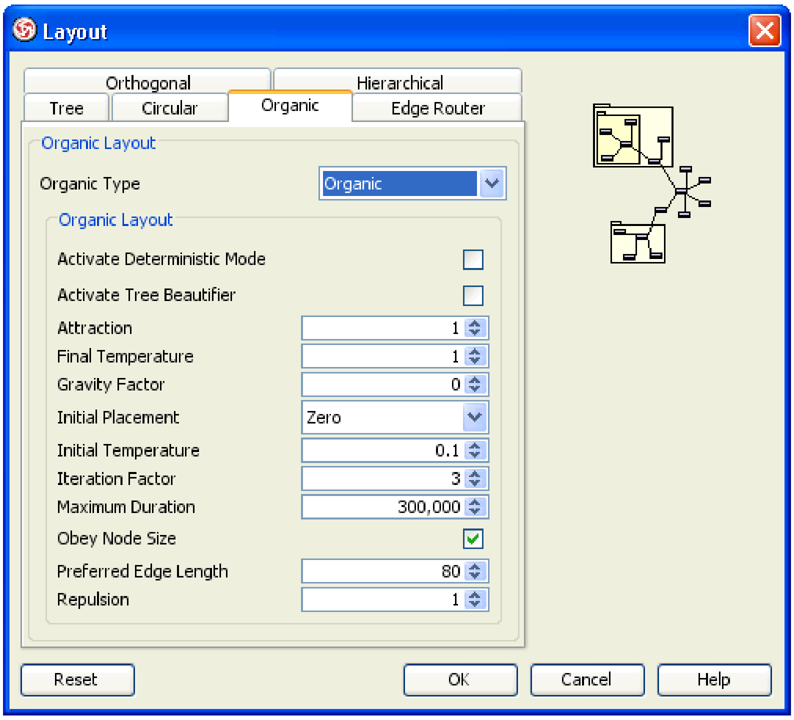


Figure 2-26 *Organic Layout*

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

Organic Type: Organic

Organic Layout

Activate Deterministic Mode ☐
Activate Tree Beautifier ☐
Attraction: 1
Final Temperature: 1
Gravity Factor: 0
Initial Placement: Zero
Initial Temperature: 0.1
Iteration Factor: 3
Maximum Duration: 300,000
Obey Node Size ☑
Preferred Edge Length: 80
Repulsion: 1

Figure 2-27 *Organic Layout setting*

Smart Organic Layout

Smart Organic Layout is one of the organic layouts in VP-UML. It 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.

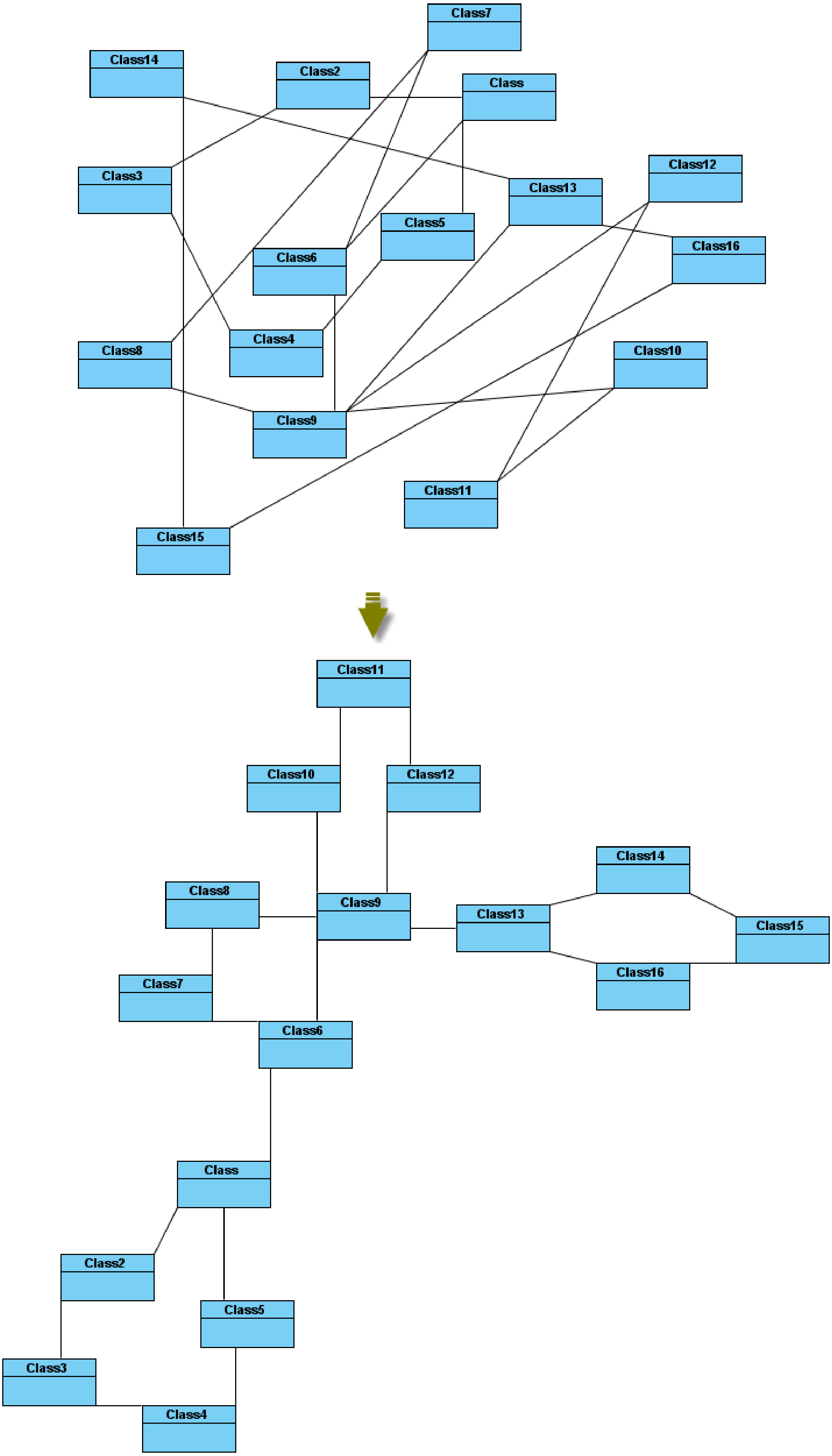
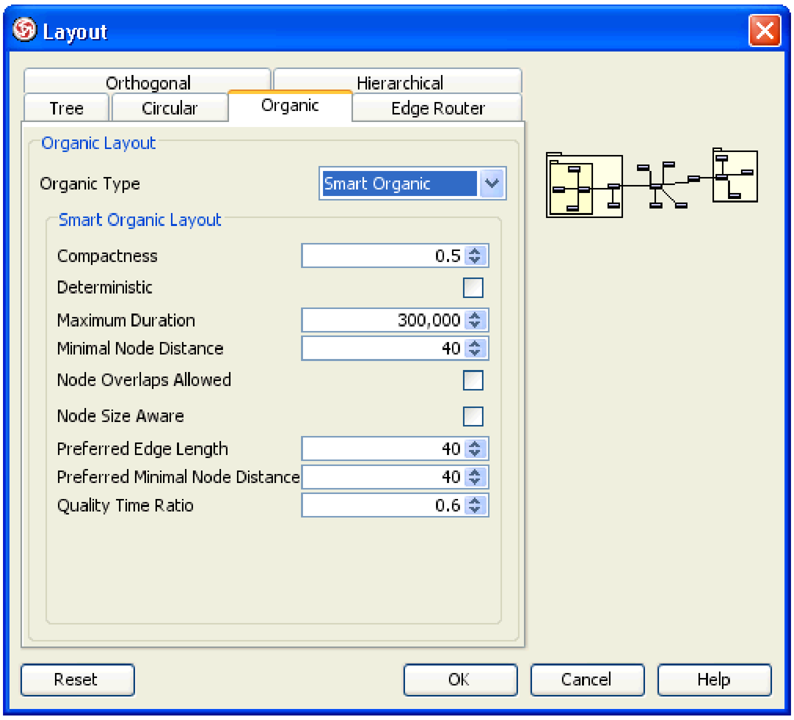


Figure 2-28 Smart Organic Layout

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

Organic Type: Smart Organic

Smart Organic Layout

Compactness: 0.5
Deterministic: □
Maximum Duration: 300,000
Minimal Node Distance: 40
Node Overlaps Allowed: □
Node Size Aware: □
Preferred Edge Length: 40
Preferred Minimal Node Distance: 40
Quality Time Ratio: 0.6

***Figure 2-29** Organic Layout setting*

Automatic Layout Selected Shapes

To layout all the shapes in the diagram, right-click on the diagram and select Layout from the popup menu.

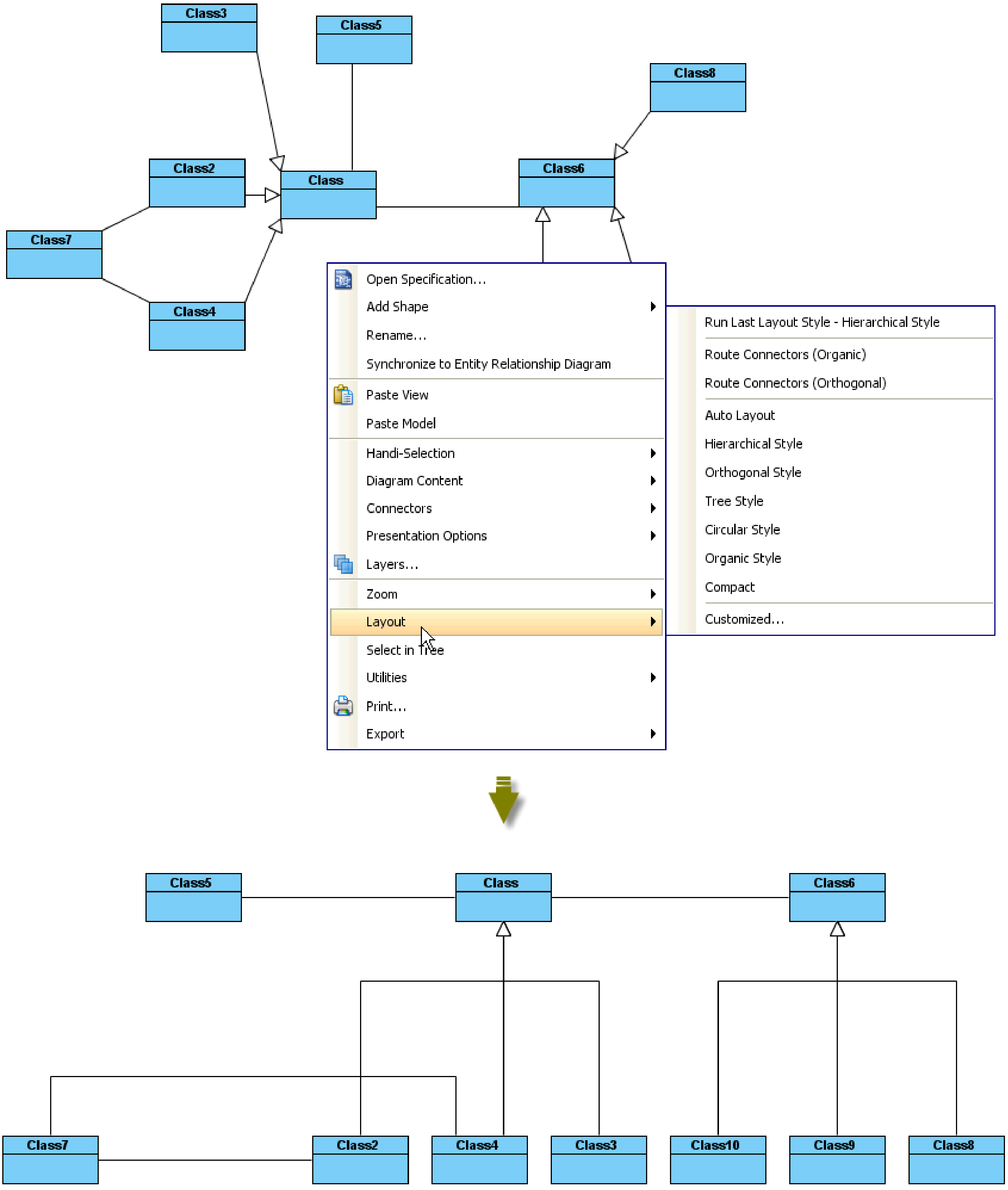


Figure 2-30 *Perform layout with all shapes of diagram*

To layout the selected shapes right-click on the selection and select Layout from the popup menu (make sure there are more than one diagram elements selected).

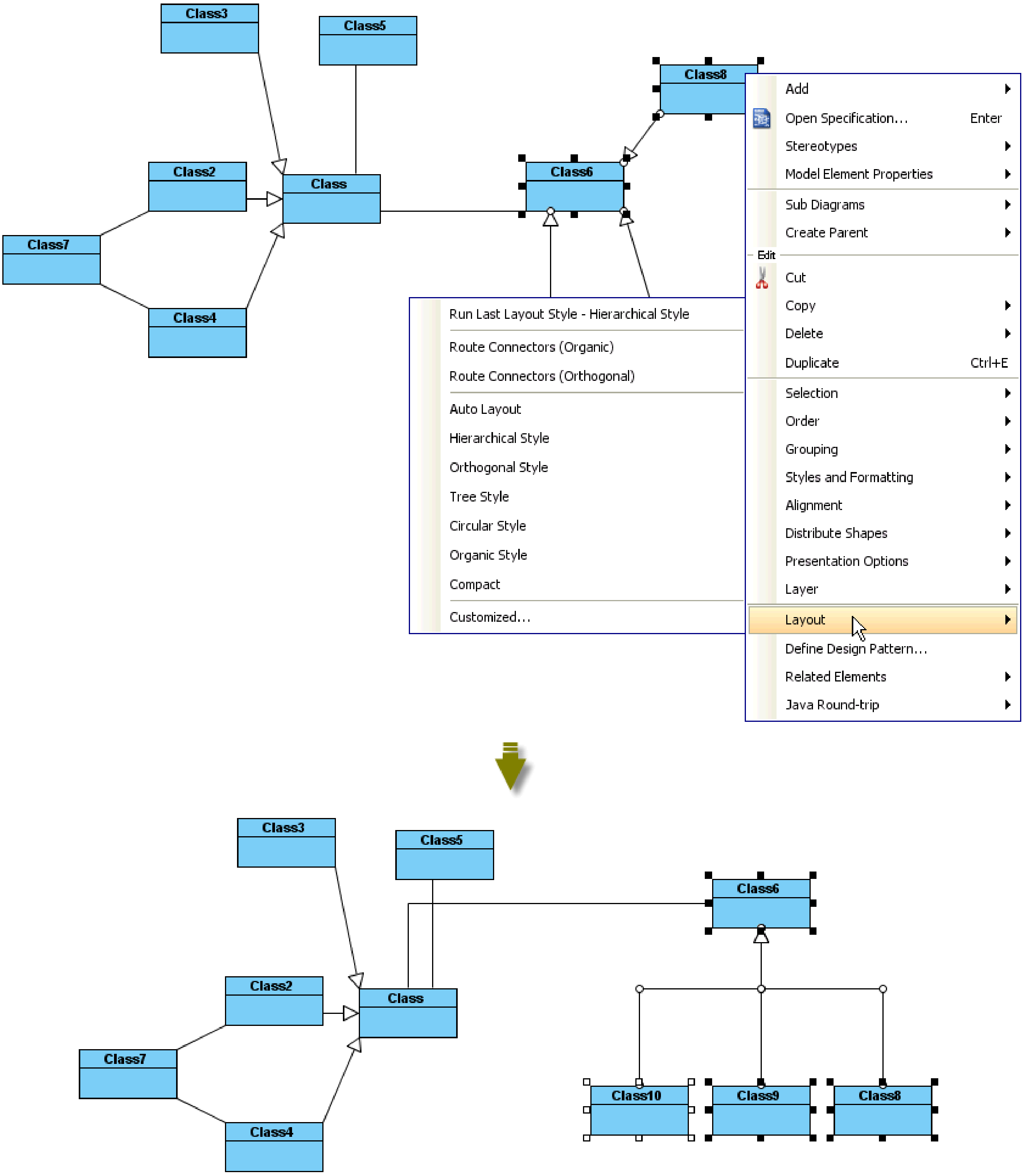


Figure 2-31 Perform layout with selected shapes

Automatic Route Connectors

There are 2 kinds of layout which do not change the location of shapes, only change the connectors: **Organic Edge Route Layout** and **Orthgonal Edge Route Layout**

Organic Edge Route Layout

Organic Edge Route Layout is one of the edge route layouts in VP-UML. It can arrange the connectors without affecting the location of shapes. It can ensure that the shapes will not overlap and keep a specific minimal distance.

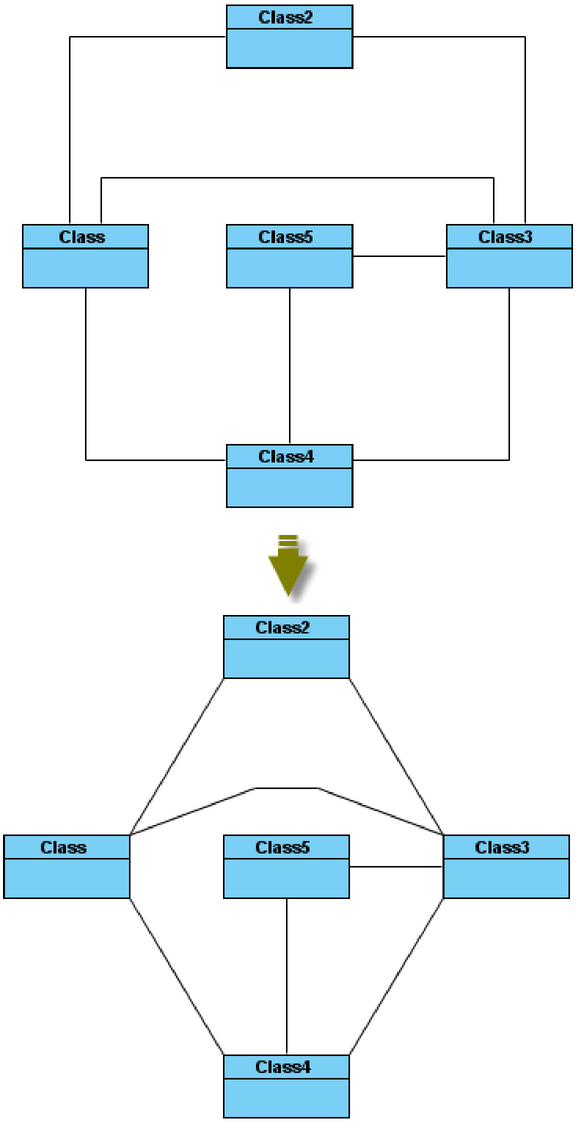
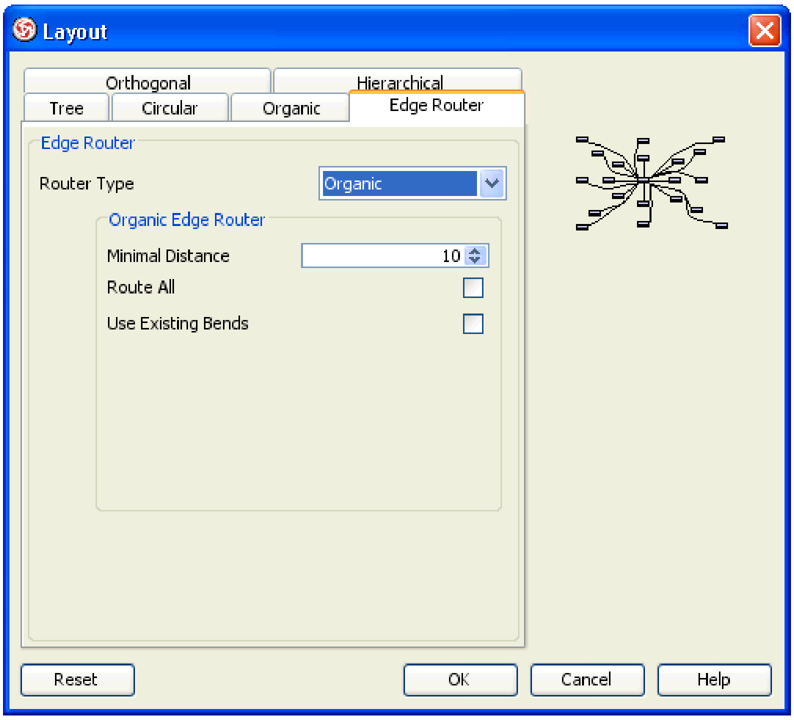


Figure 2-32 *Organic Edge Route Layout*

Minimal Distance: the minimal distance of the connectors
Route All: whether all the connectors will be routed
Use Existing Beans: whether using existing bends



Edge Router

Router Type: Organic

Organic Edge Router

Minimal Distance: 10

Route All ☐

Use Existing Bends ☐

Figure 2-33 *Organic Edge Route Layout setting*

Orthogonal Edge Route Layout

Route Connectors can arrange the connectors using vertical and horizontal line segments only. It is best for arranging the connectors that have complicated route.

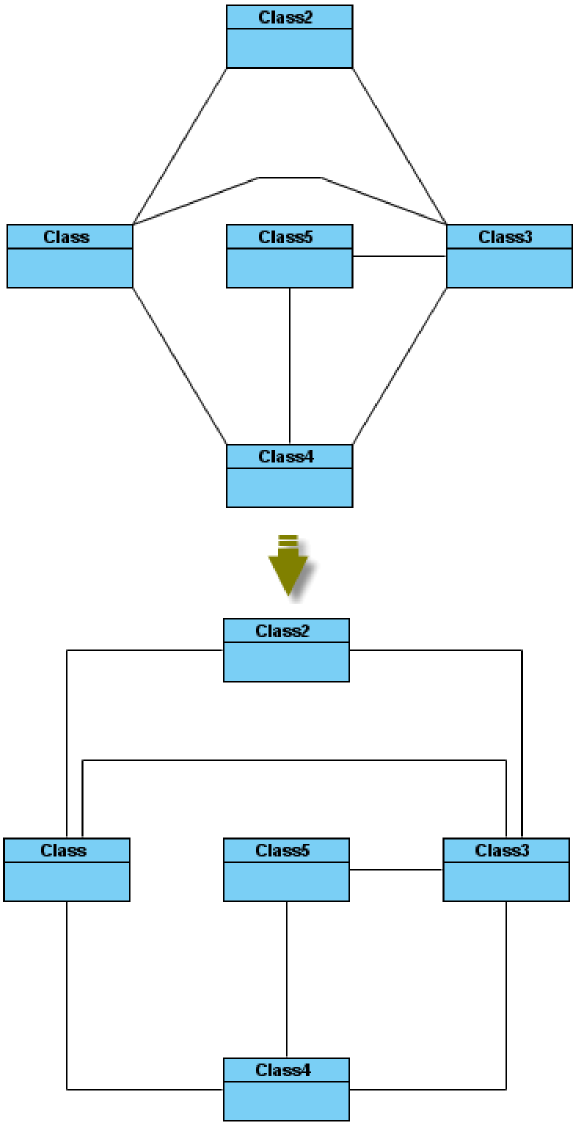
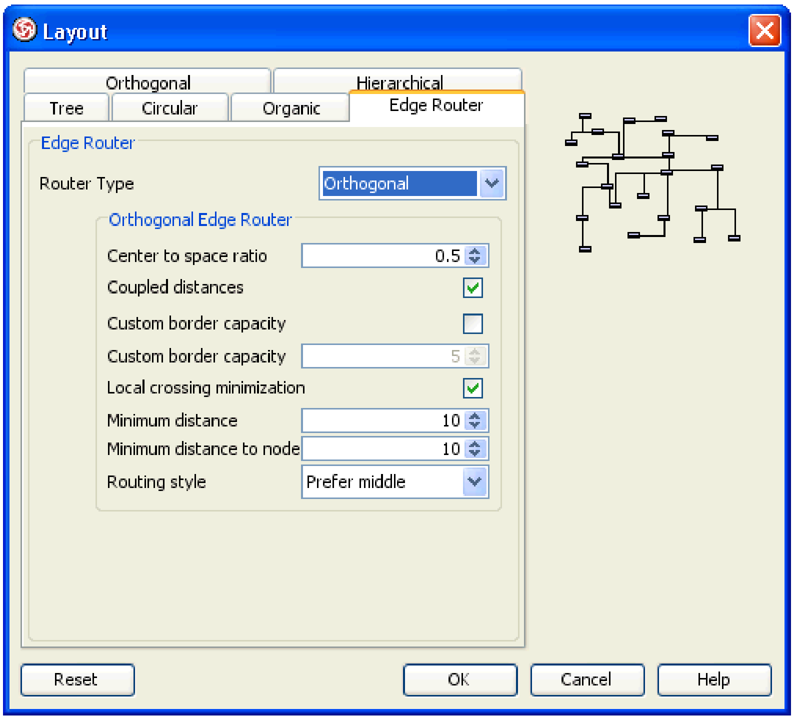


Figure 2-34 *Orthogonal Edge Route Layout*

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



Edge Router

Router Type: Orthogonal

Orthogonal Edge Router

Center to space ratio: 0.5
Coupled distances: ☑
Custom border capacity: □
Custom border capacity: 5
Local crossing minimization: ☑
Minimum distance: 10
Minimum distance to node: 10
Routing style: Prefer middle

***Figure 2-35** Orthogonal Edge Route Layout setting*

Fit Shape Size

Sometimes, a shape may be too large to show its content. So that you may want to resize it smaller to look it better. At this moment, Fit Size can help you.

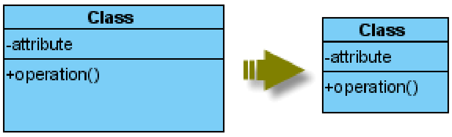


Figure 2-36 Fit size

Fit selected shapes size

To fit a shape size, selected the shape, **Fit Size** resource will be shown. Click the resource to fit size.

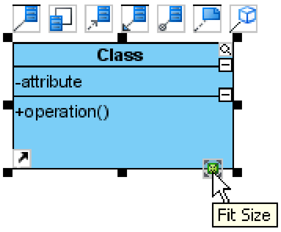


Figure 2-37 Fit size on 1 shape by resource centric

To fit several shapes size, selected the shapes, right-click on a selected shape, select **Selection > Fit Size** in popup menu.

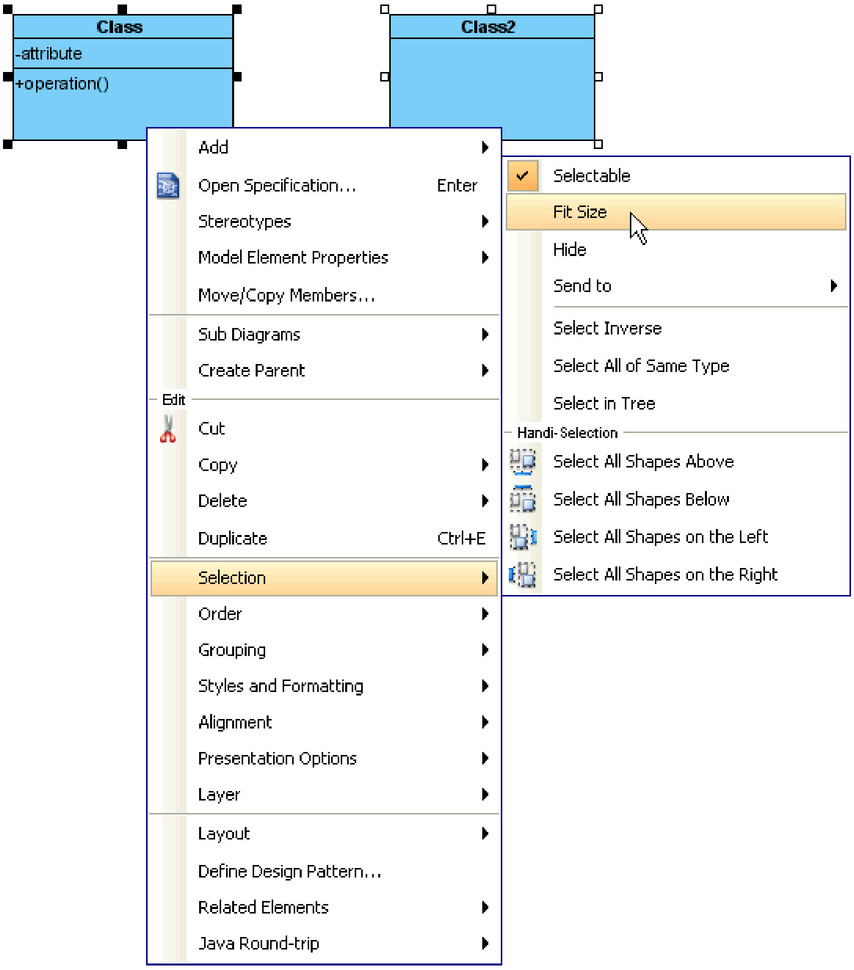


Figure 2-38 Fit size on several shapes by popup menu

The shape will be fitted to it size, won't make all selected shapes to be same size.

Class
-attribute
+operation()
Class2

Figure 2-39 *Shapes are fitted size*

Turn on and off automatic Fit Shape size Mode

You can turn on/off the Auto Fit Shapes Size on diagram to make all the shapes on the diagram to be fit size automatically. To do so, right-click on diagram, select **Diagram Content > Auto Fit Shapes Size** in popup menu.

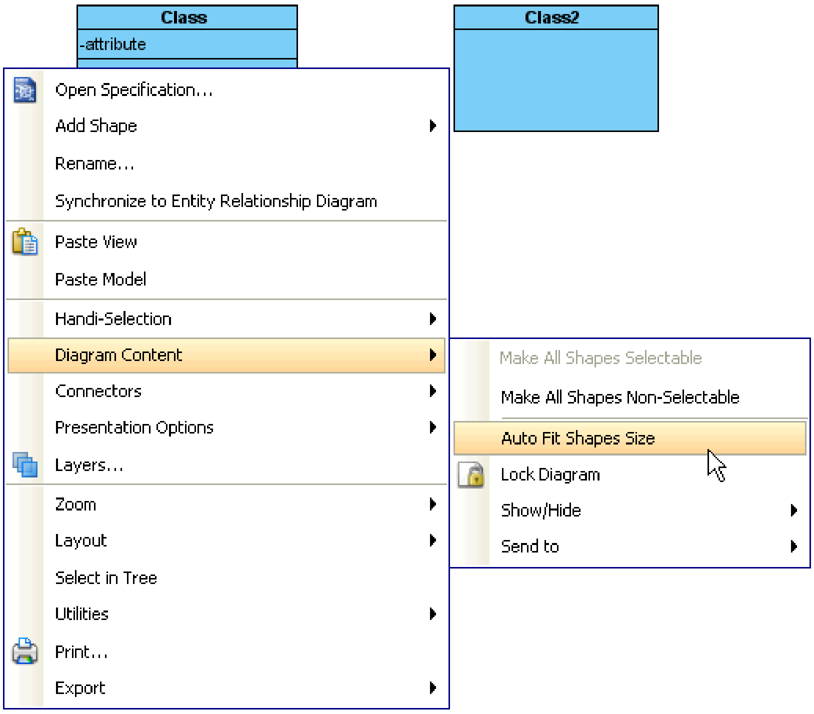


Figure 2-40 *Turn on Auto Fit Shapes Size on diagram*

All the shapes are fitted size, and they will become non-sizable.

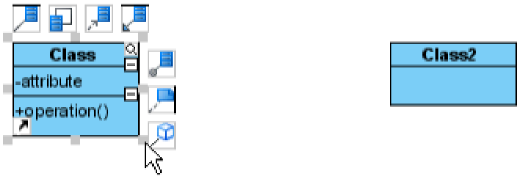


Figure 2-41 *Cannot resize the shapes if Auto Fit Shapes Size mode is turned on*

If the shape contents is changed, the shape will be resized automatically

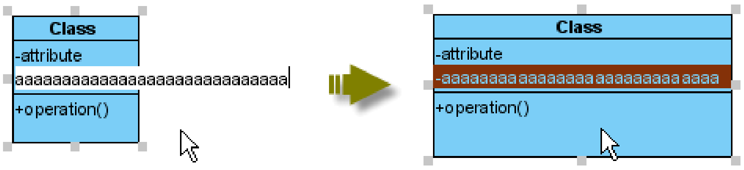
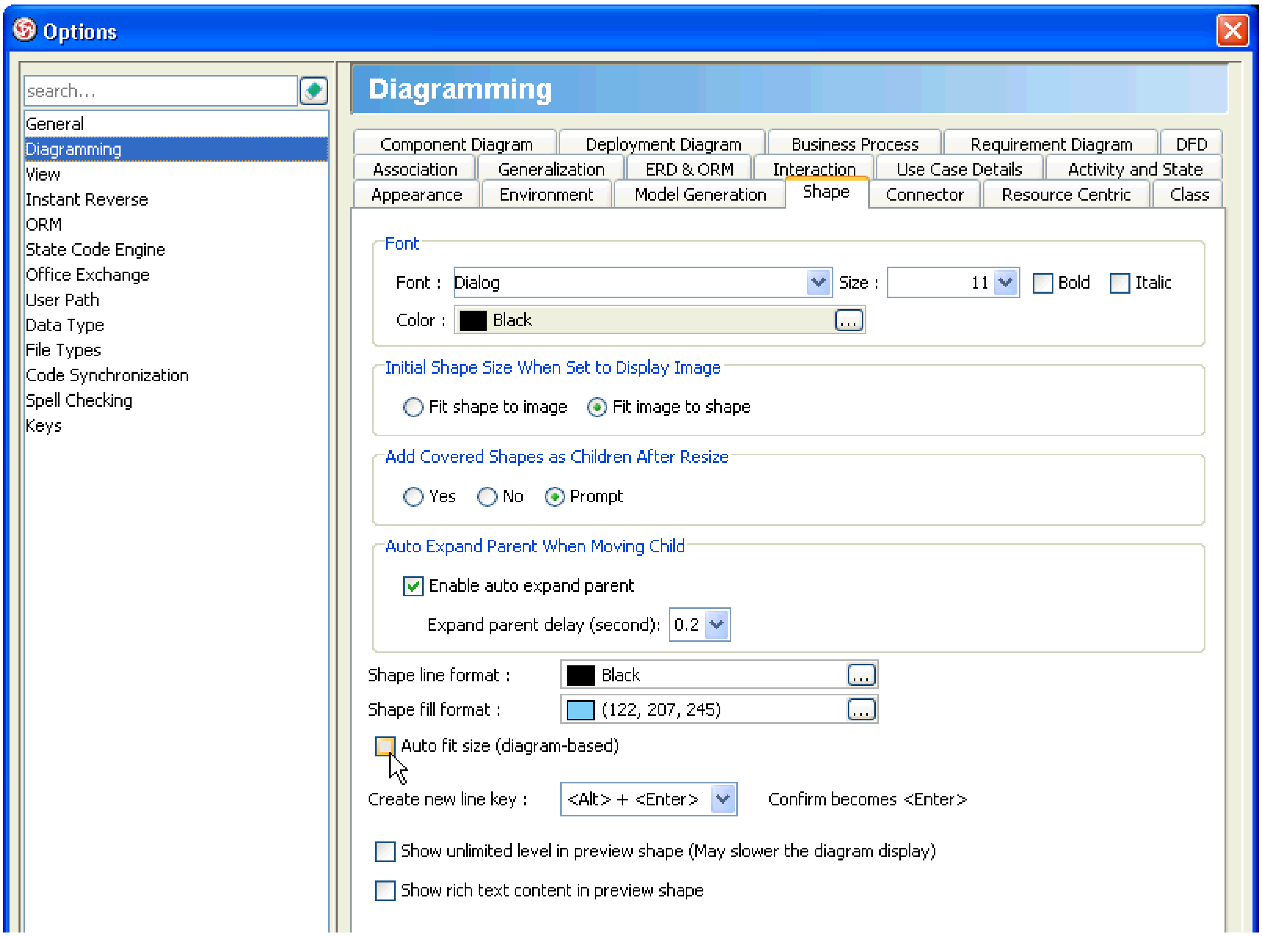


Figure 2-42 *New attribute added, class shape will be resized automatically*

You also can turn on/off Auto Fit Shapes Size for the future diagrams.
1.Select **Tools > Options...** to open Options dialog box.
2.In the Options dialog box, open the Diagramming category, select the Shape tab. You can select Auto fit size (diagram-based).



Font

Font : Dialog Size : 11 □ Bold □ Italic

Color : ■ Black

Initial Shape Size When Set to Display Image

○ Fit shape to image ● Fit image to shape

Add Covered Shapes as Children After Resize

○ Yes ○ No ● Prompt

Auto Expand Parent When Moving Child

☑ Enable auto expand parent

Expand parent delay (second): 0.2

Shape line format : ■ Black

Shape fill format : ■ (122, 207, 245)

□ Auto fit size (diagram-based)

Create new line key : <Alt> + <Enter> Confirm becomes <Enter>

□ Show unlimited level in preview shape (May slower the diagram display)

□ Show rich text content in preview shape

Figure 2-43 *Select Auto fit size (diagram-based)*

Diagram Element Selection

Multi-Selection

You can select several shapes by drag-and-drop on diagram or clicking shapes with pressing Ctrl or Shift key.

Drag and Drop

1.Dragging on diagram, a rectangle will be shown as selection.

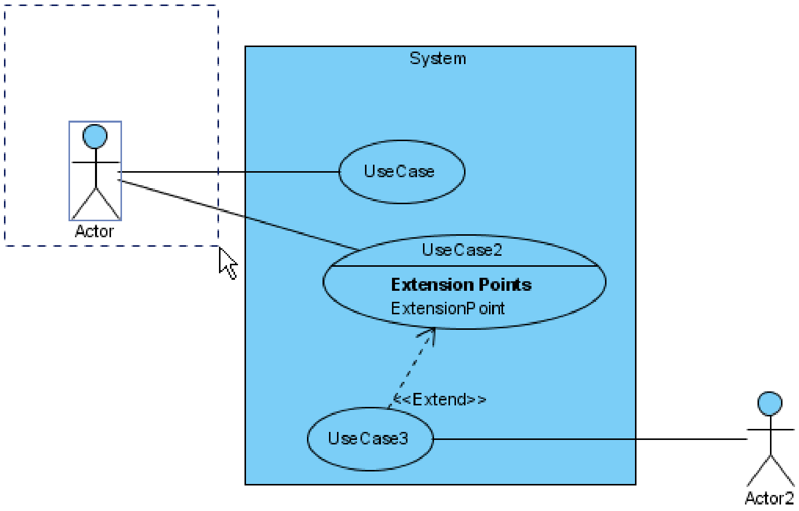


Figure 2-44 *Rectangle shown as selection*

2.The shapes inside the rectangle will be highlighted.

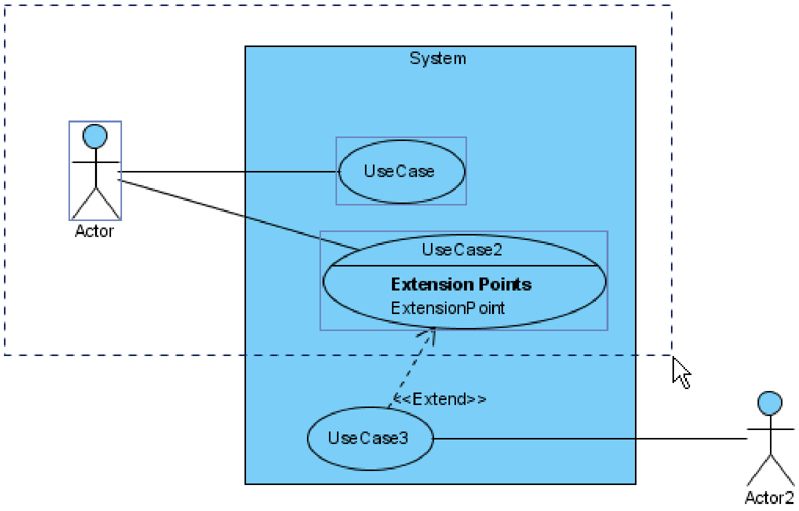


Figure 2-45 *Shapes are highlighted*

3.After drop the mouse, the shapes are selected.

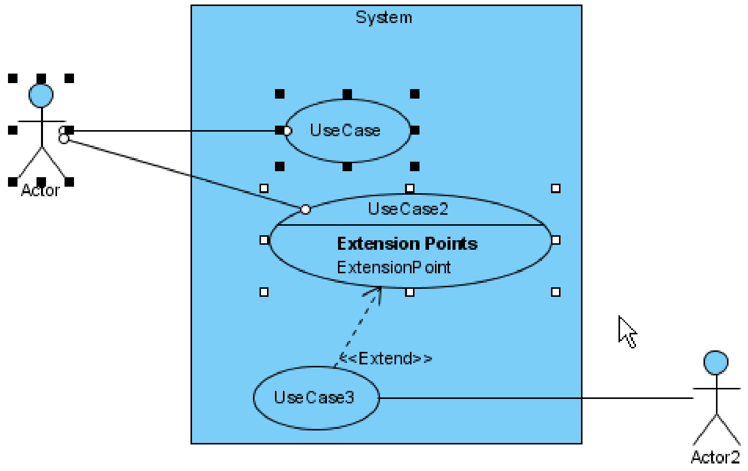


Figure 2-46 *Select shapes by drag and drop*

Clicking with Ctrl/Shift key

You also can select several shapes by clicking on the shape with pressing Ctrl or Shift key.

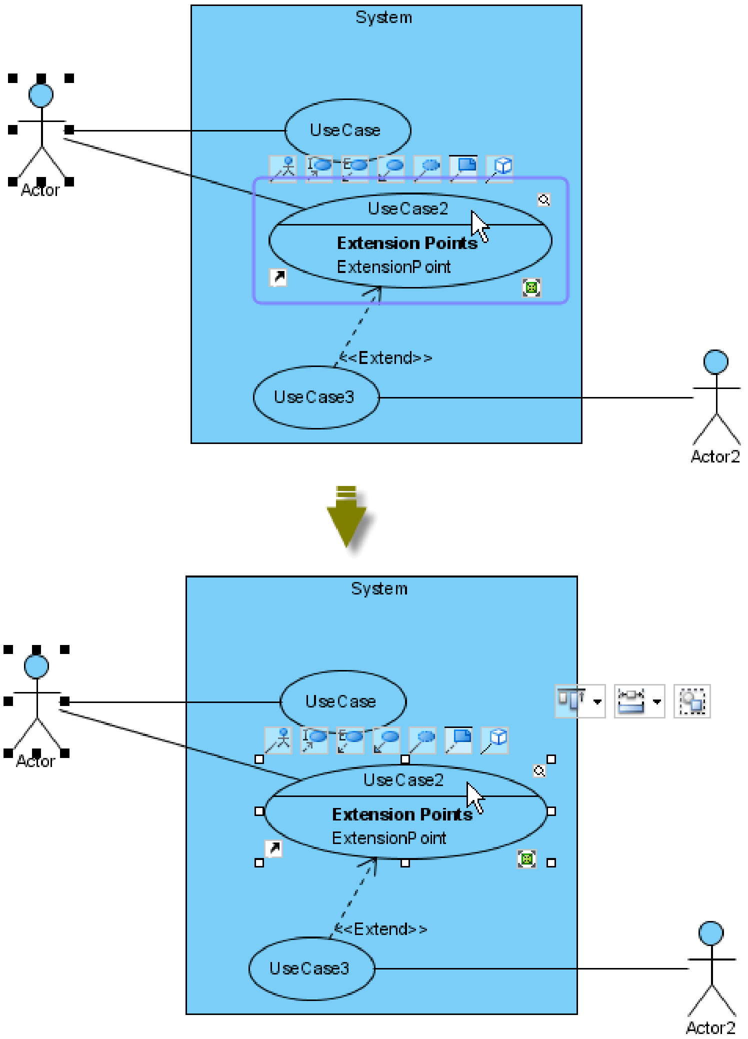


Figure 2-47 *Shape will be selected if clicking on it with pressing Ctrl or Shift key*

Handi-Selection

Sometimes, you may want to select a set of shapes based on the shapes' location, e.g. on a large diagram, it is difficult to drag and drop on the large diagram, or select one by one with clicking on many shapes. Then you may use **Handi-Selection** to select the shapes.

1.Assume you wanna select all shapes which are on right of the Actor, then you can right-click on right of the actor.

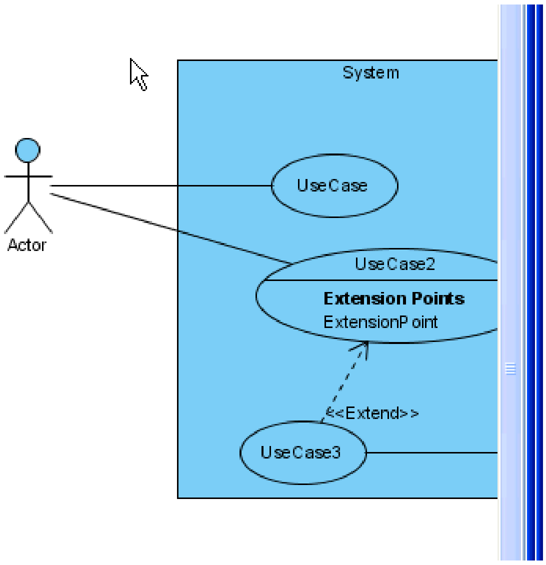


Figure 2-48 *Right-click on the right of the actor*

2.Select **Handi-Selection > Select All Shapes on the Right** from popup menu.

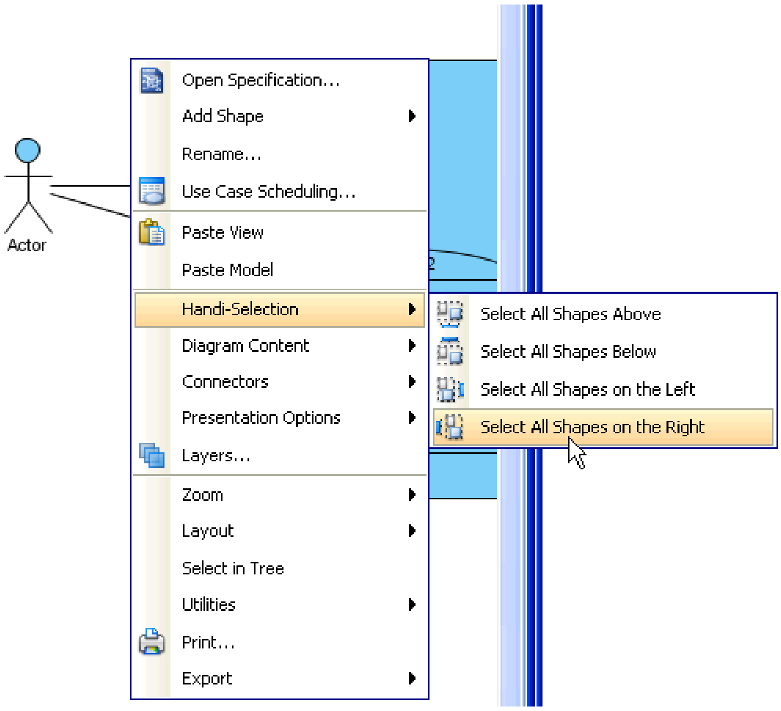


Figure 2-49 *Select All Shapes on the right*

3.All the shapes, that on the right of mouse cursor, are selected

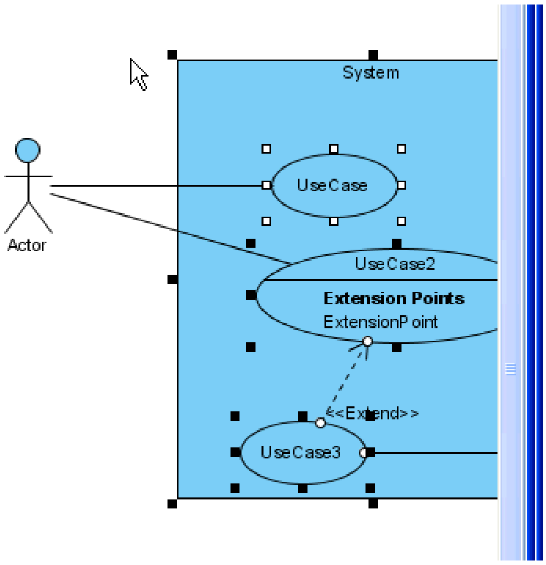


Figure 2-50 Shapes on right are selected

Selecting Same Type of Shapes

You also can select same type shapes. To do so, right-click on the shape, select **Selection > Select All of Same Type** from popup menu.

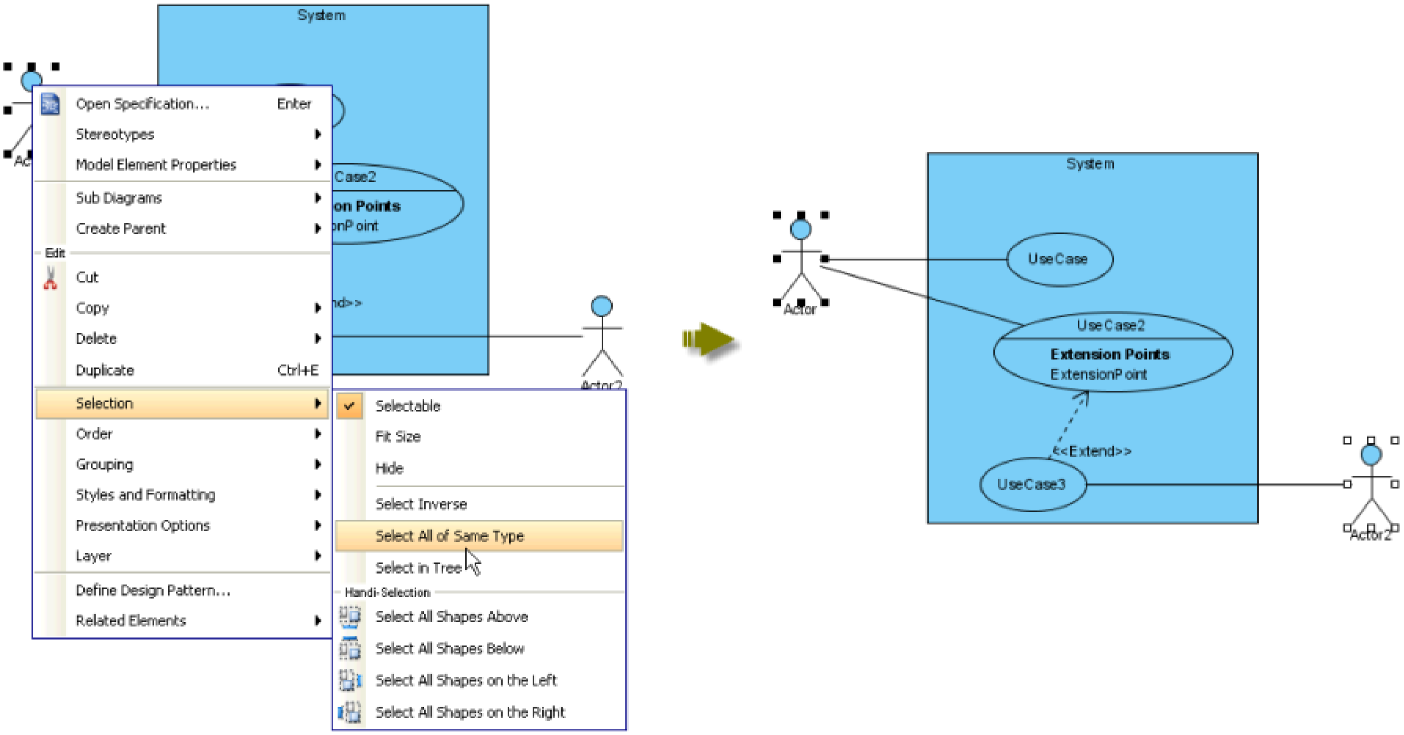


Figure 2-51 All shapes in same type are selected

Inverse Selection

You also can inverse the selection, right-click on the selected shape, select **Selection > Select Inverse** from popup menu.

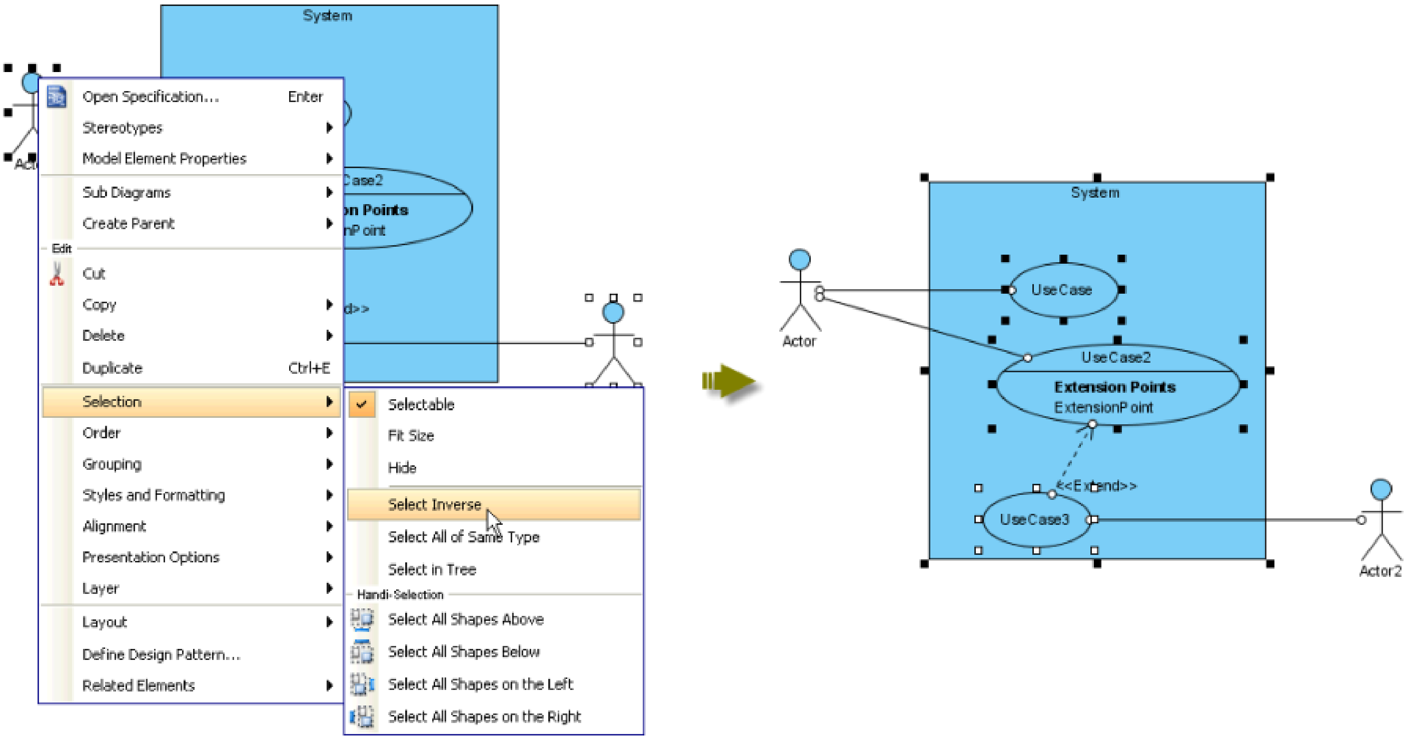


Figure 2-52 Selection is inversed

Copy and Paste

Cut and Paste

You can cut and paste the shapes between diagrams.
1.Right-click on selected shapes, select **Cut** from popup menu.

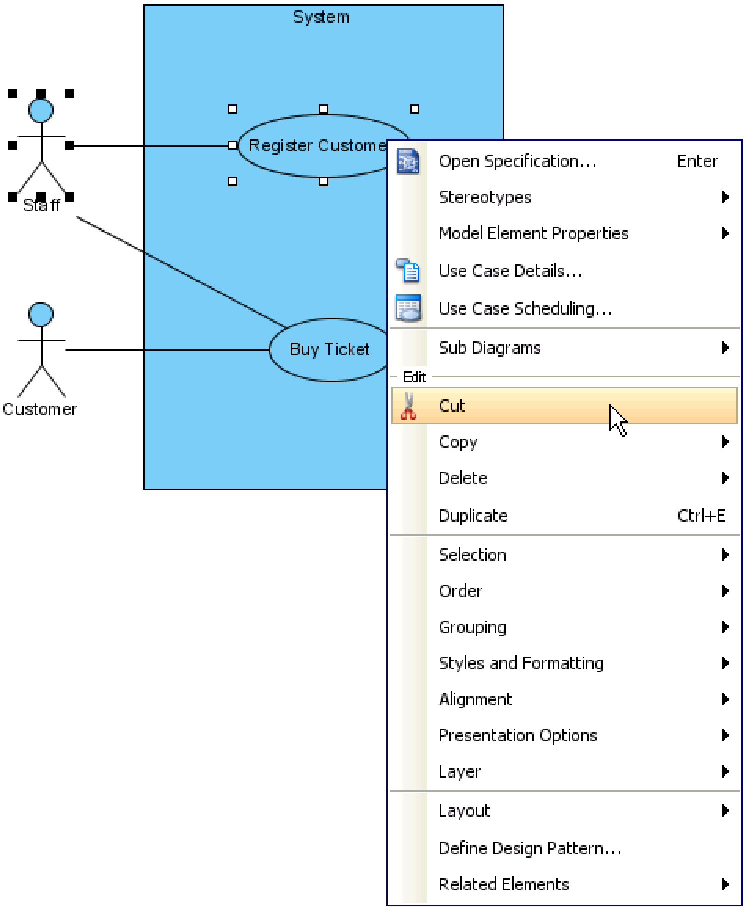


***Figure 2-53** Cut selected shapes*

2.On another diagram, right-click on empty area, select **Paste View** from popup menu

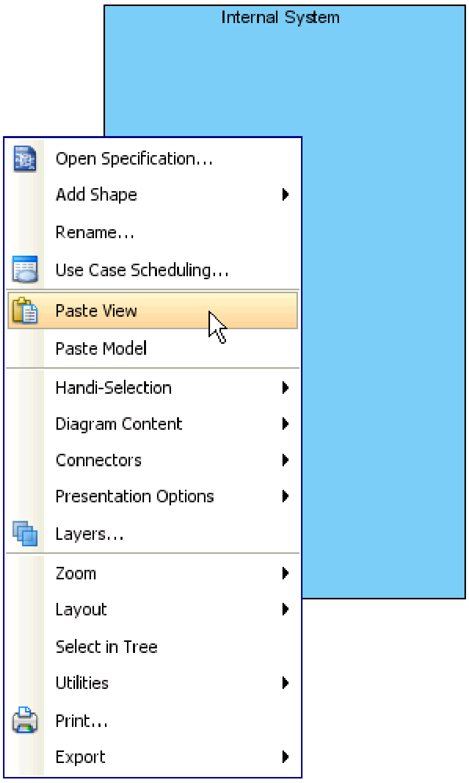


Figure 2-54 *Paste the selected shapes on another diagram*

3.Selected shapes are pasted.

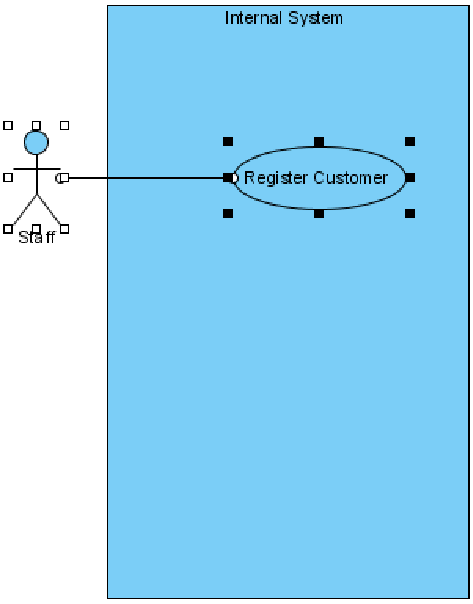
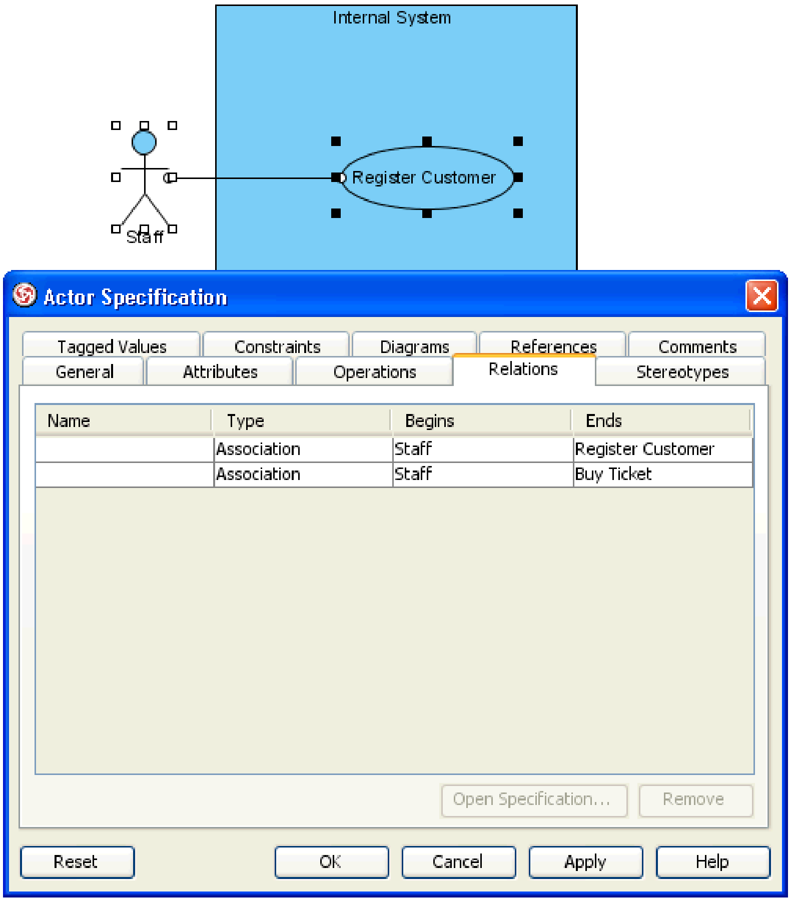


Figure 2-55 *Shapes are pasted*

4.You may found that the non-selected association is not shown on both diagrams, but the association is still exists, you can find that in open spec.



Name	Type	Begins	Ends
	Association	Staff	Register Customer
	Association	Staff	Buy Ticket

__Figure 2-56__ The non-selected association is still exists

Copy within VP-UML

Besides Cut and Paste, you can also Copy and Paste.

1.Right-click on the selected shapes, select **Copy > Copy within VP-UML EE**.

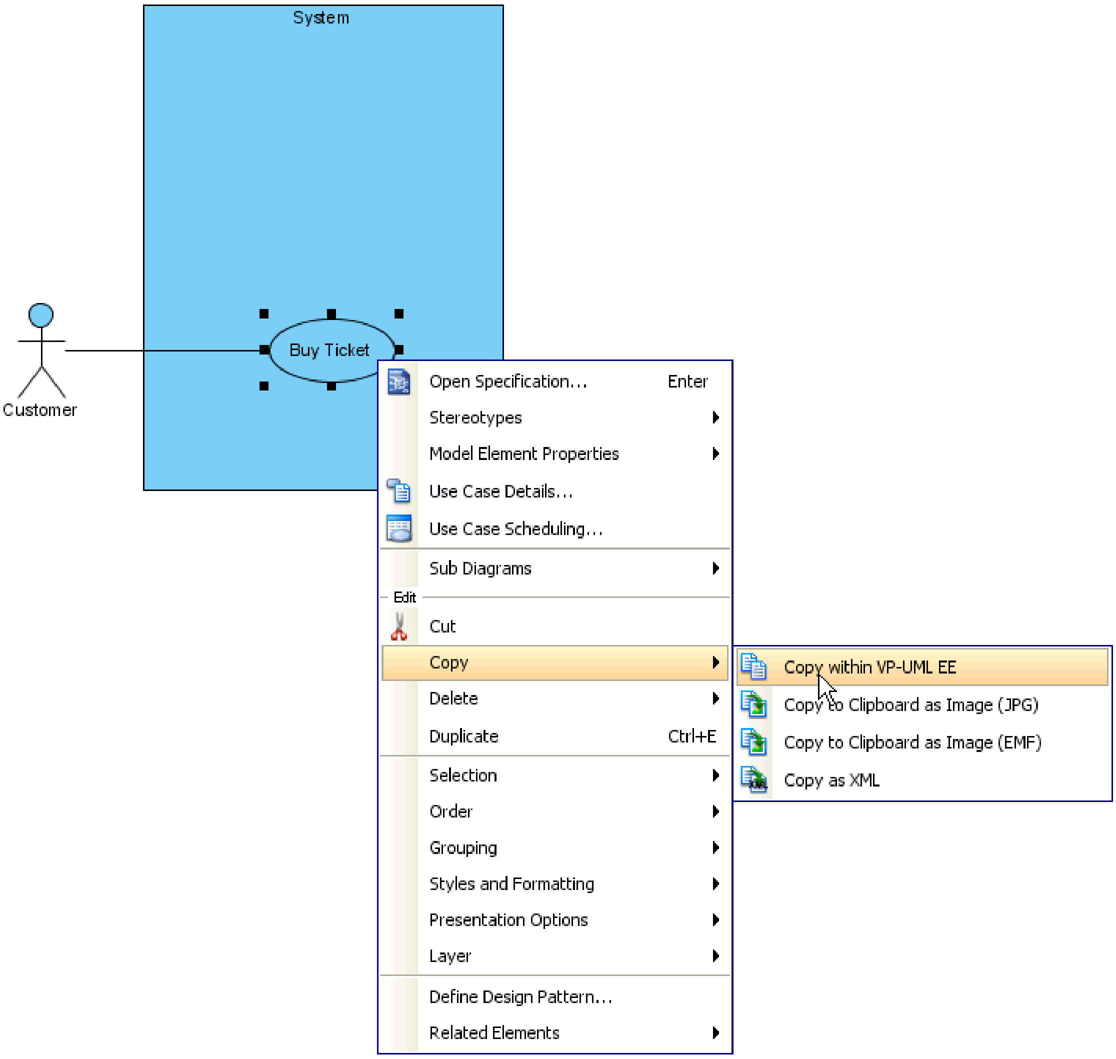


Figure 2-57 Copy selected shapes

2.On another diagram, right-click on empty area, select **Paste View** from popup menu.

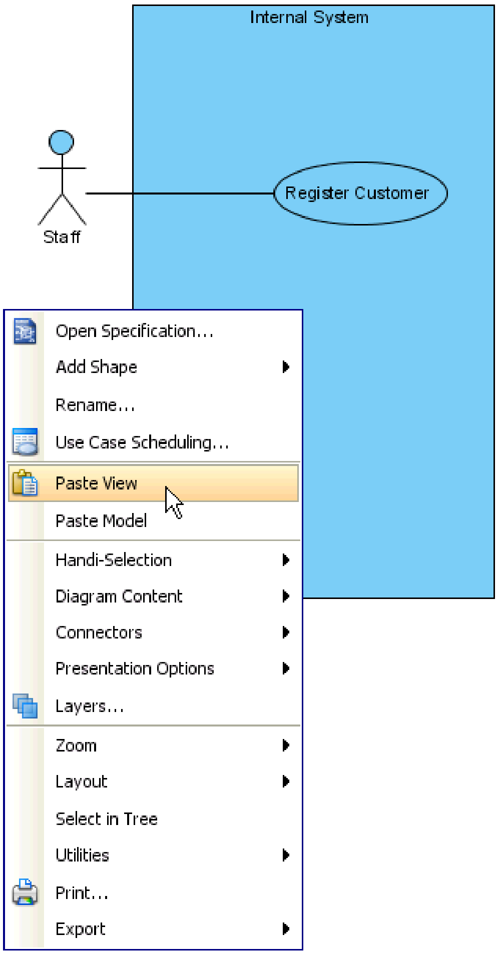


Figure 2-58 *Paste the selected shapes*

3.The shapes are pasted on diagram.

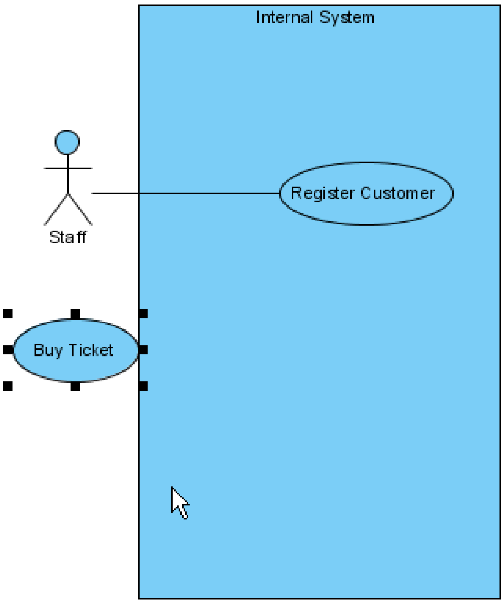


Figure 2-59 *Shapes are pasted*

4.You may found that, there is no Association shown between the *Staff* and *Buy Ticket*. To show the association, right-click on the shape. Select **Related Elements > Visualize Related Model Element...**.

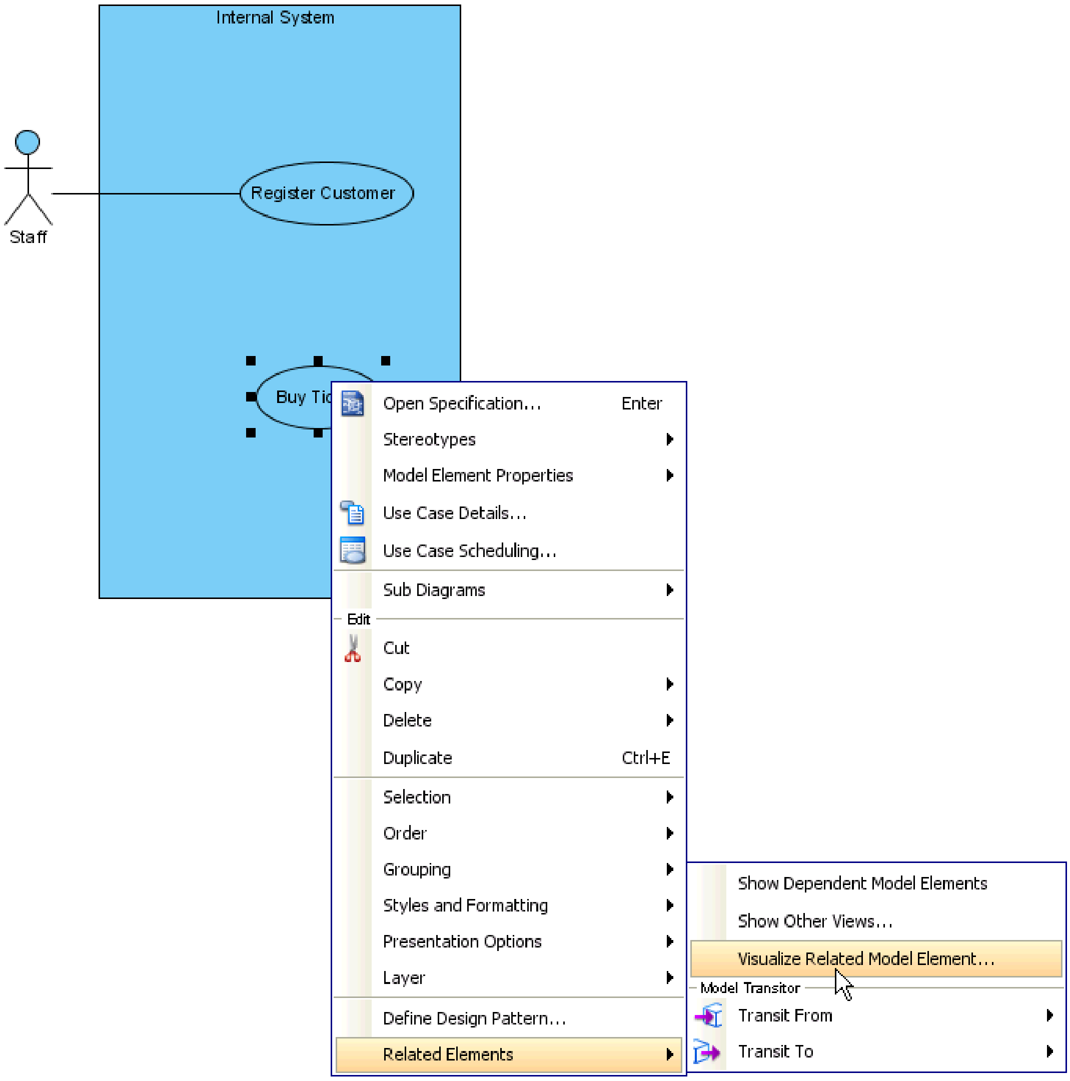
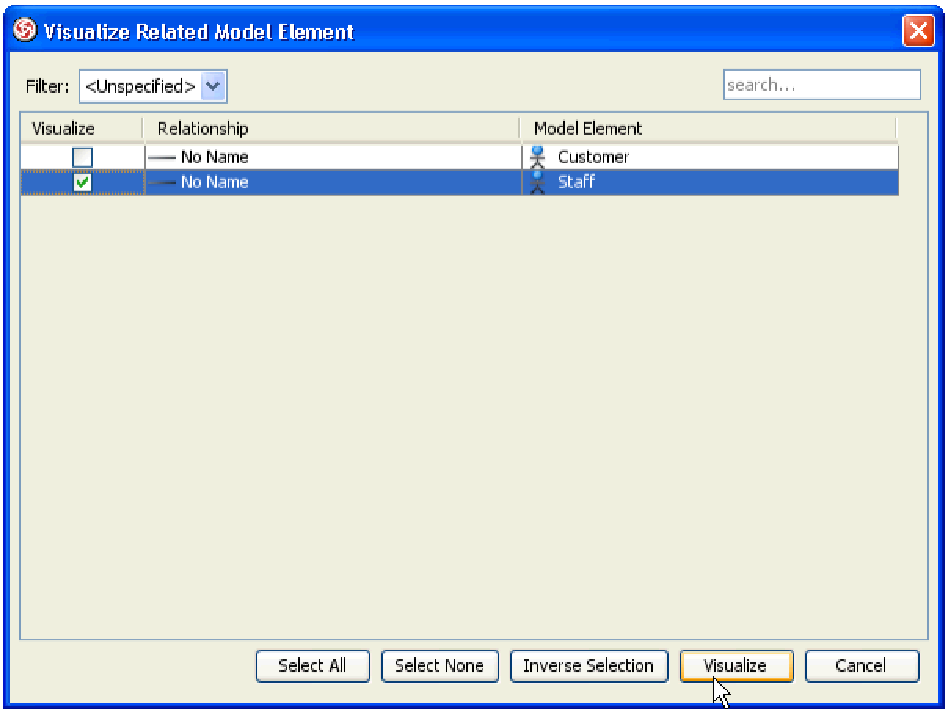


Figure 2-60 *Right-click on the shape*

5.Select the association with *Staff*, and click **Visualize**.



Filter: <Unspecified>

Visualize	Relationship	Model Element
□	No Name	Customer
☑	No Name	Staff

Figure 2-61 *Visualize the association*

6.The association is shown on the diagram.

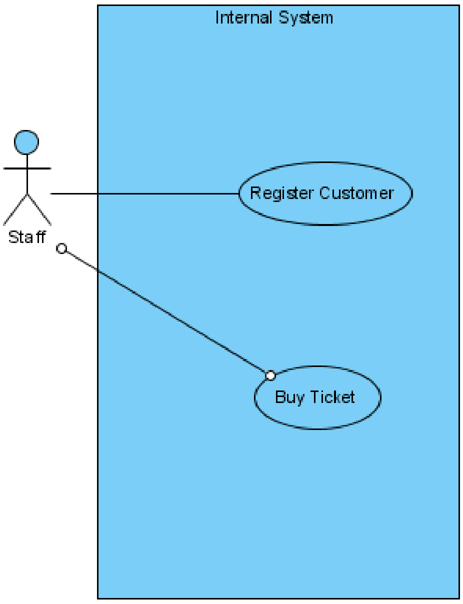


Figure 2-62 *Association is shown*

Copy to Clipboard as Image (JPG)

You can also copy the selected shapes as JPG image. Right-click on the selected shapes, select **Copy > Copy to Clipboard as Image (JPG)**.

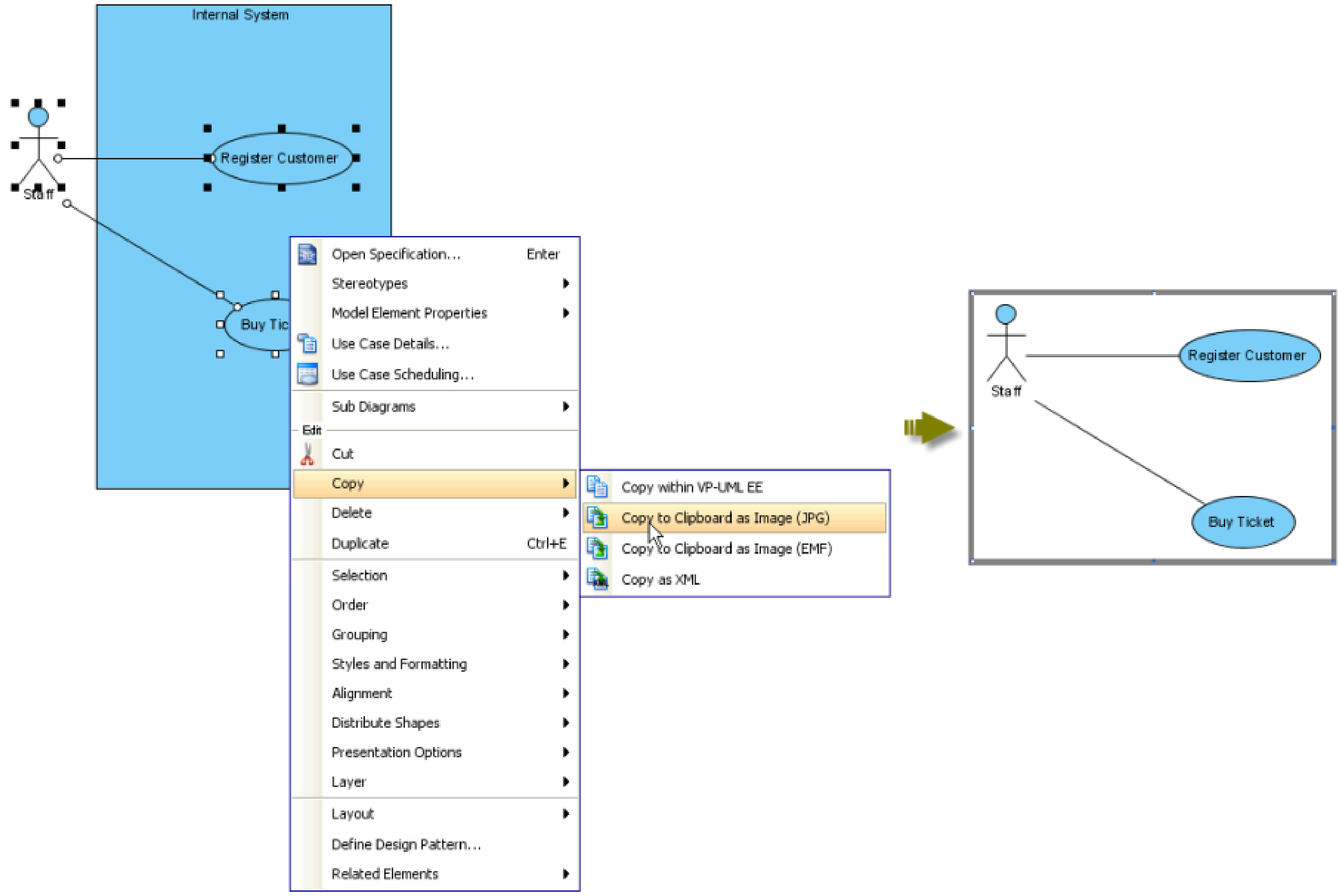


Figure 2-63 Copy to JPG image

Copy to Clipboard as Image (EMF)

You can also copy the selected shapes as EMF image. Right-click on the selected shapes, select **Copy > Copy to Clipboard as Image (EMF)**.

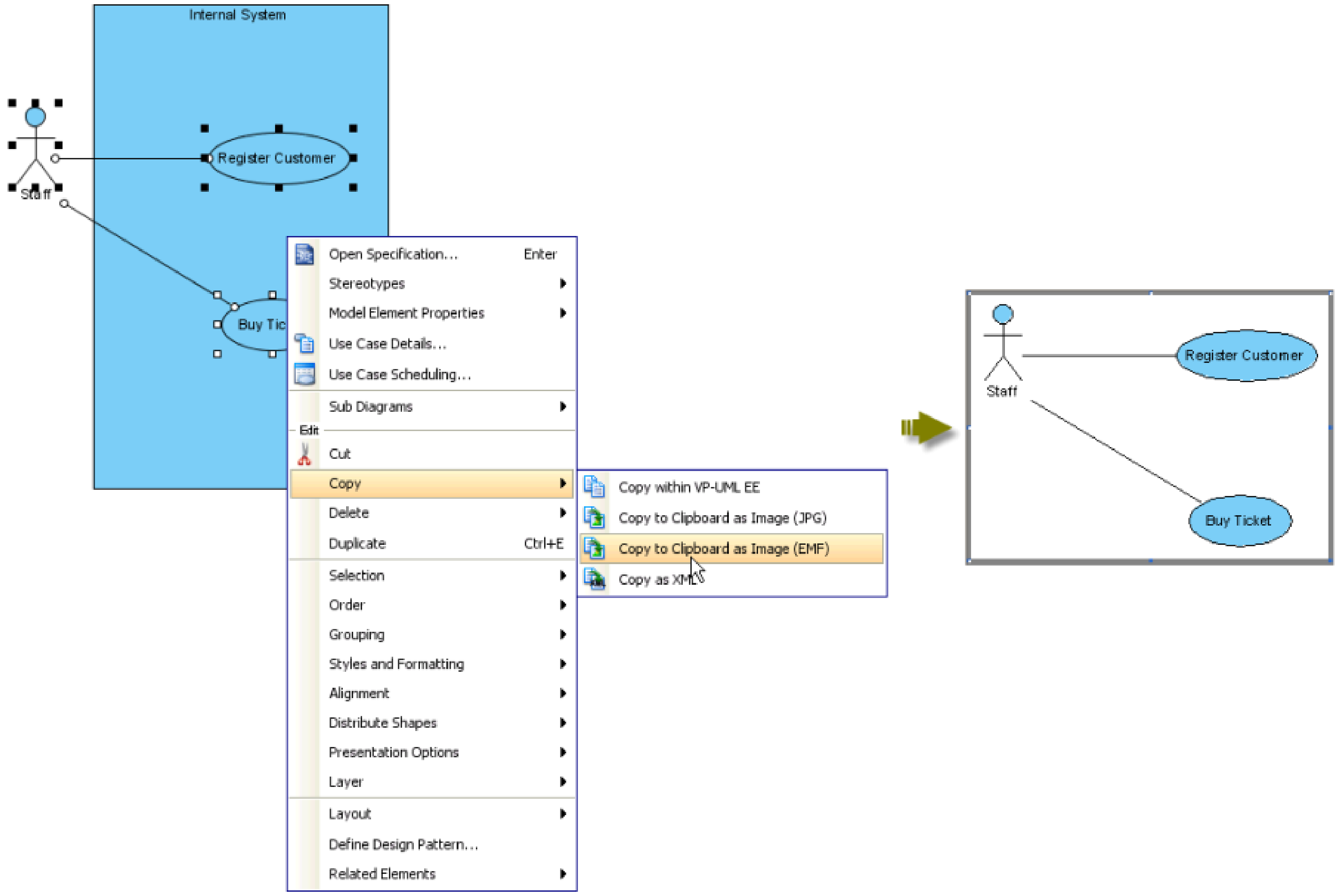
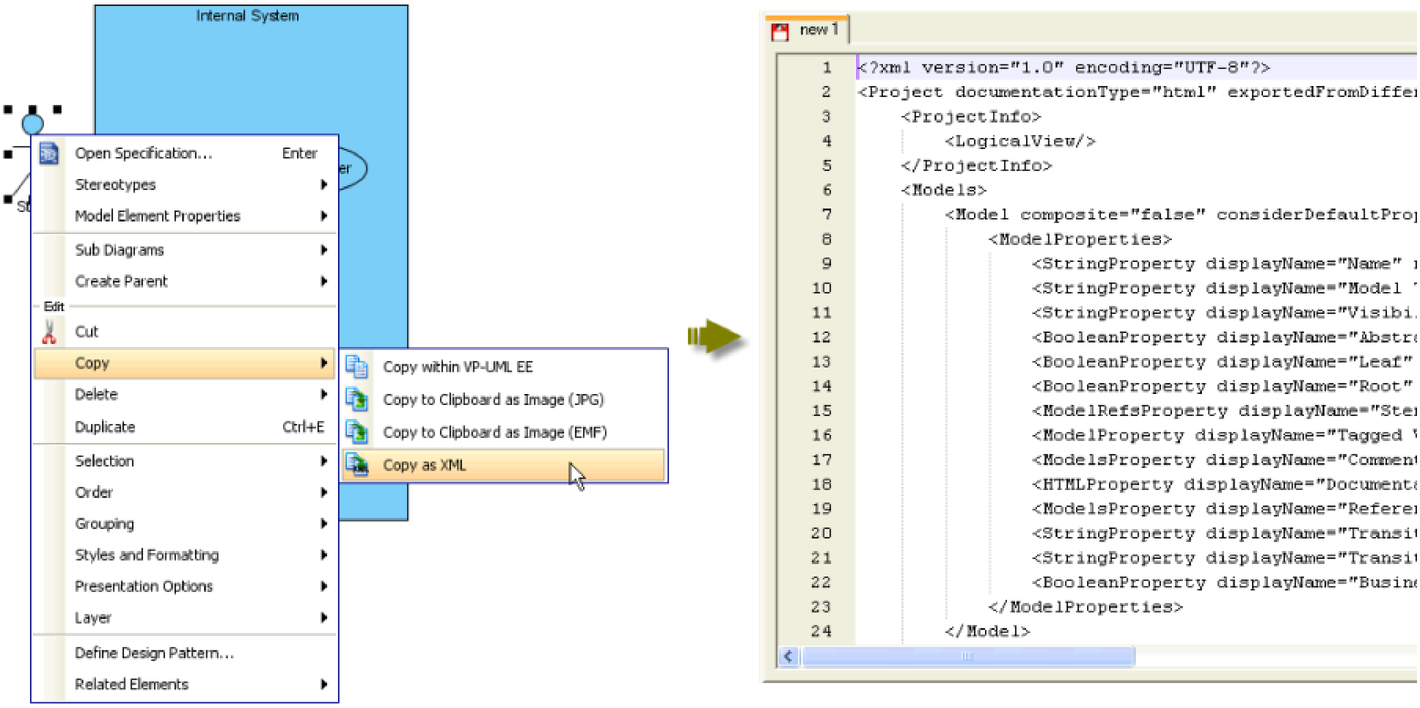


__Figure 2-64__ Copy to EMF image

Copy as XML

You can also copy the selected shapes as XML. Right-click on the selected shapes, select **Copy > Copy as XML**.



```
<?xml version="1.0" encoding="UTF-8"?>
<Project documentationType="html" exportedFromDiffe
    <ProjectInfo>
        <LogicalView/>
    </ProjectInfo>
    <Models>
        <Model composite="false" considerDefaultPro
            <ModelProperties>
                <StringProperty displayName="Name"
                <StringProperty displayName="Model
                <StringProperty displayName="Visibi
                <BooleanProperty displayName="Abstr
                <BooleanProperty displayName="Leaf"
                <BooleanProperty displayName="Root"
                <ModelRefsProperty displayName="Ste
                <ModelProperty displayName="Tagged
                <ModelsProperty displayName="Commen
                <HTMLProperty displayName="Document
                <ModelsProperty displayName="Refere
                <StringProperty displayName="Transi
                <StringProperty displayName="Transi
                <BooleanProperty displayName="Busin
            </ModelProperties>
        </Model>
```

Figure 2-65 *Copy to XML*

Paste View

After you copy the model, you can paste in VP-UML.

1.Right-click on empty area on a diagram. Select **Paste View** from popup menu.

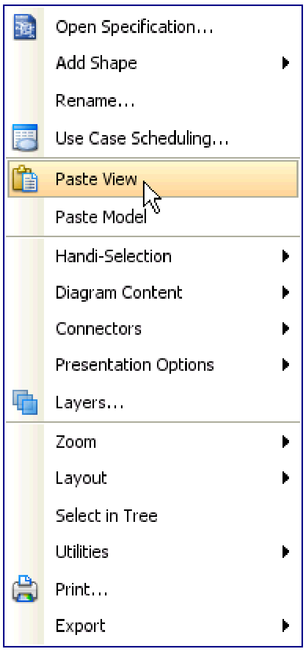


Figure 2-66 *Paste View on diagram*

2.The shape is pasted, no model will be created.

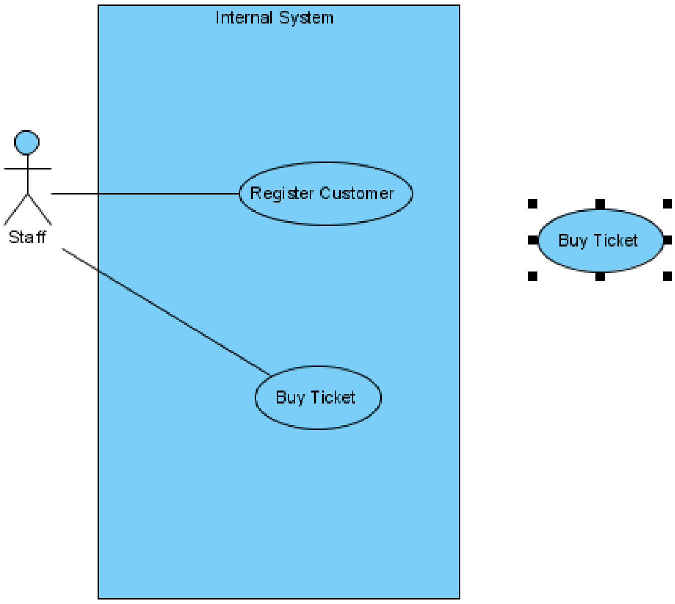


Figure 2-67 *Shape is pasted*

Paste Model Element

You can also paste with copying the model.
1.Right-click on empty area on a diagram, select **Paste Model** from popup menu.

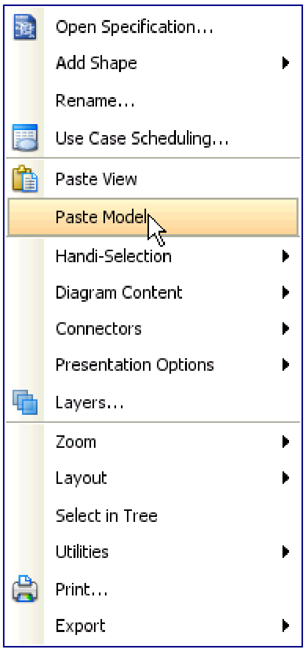


Figure 2-68 *Paste Model on diagram*

2.The model will be copied and pasted on the diagram. The new model will be named with appending a sequential number.

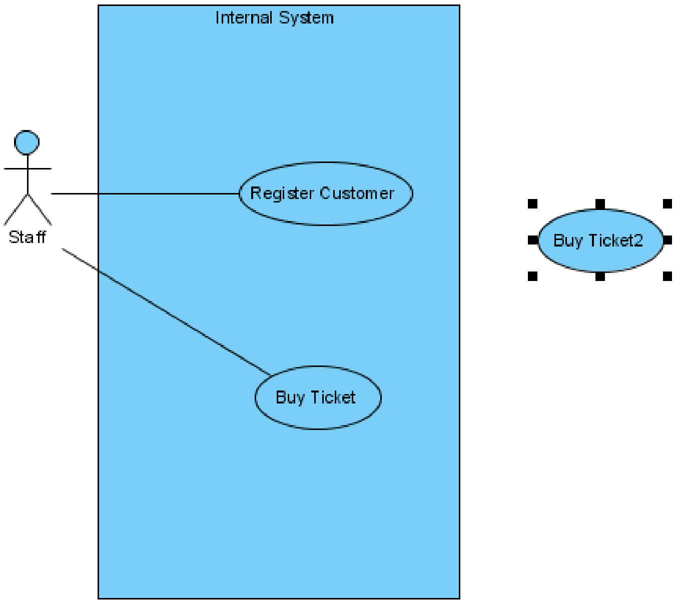


Figure 2-69 Model is pasted

Duplicate

You can duplicate a model on the diagram.

1.Right-click on the selected shapes, select Duplicate from popup menu.

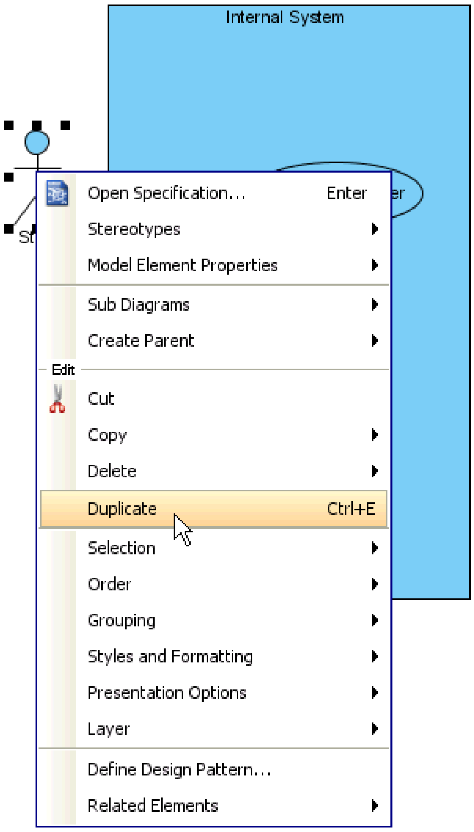


Figure 2-70 Duplicate the selected shapes

2.The model is copied and pasted on the diagram

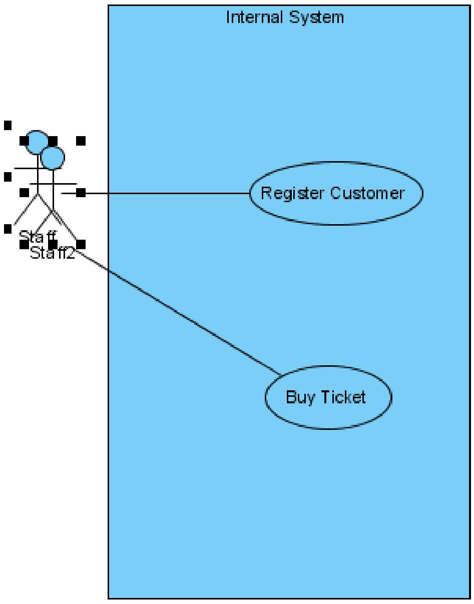


Figure 2-71 *Model is duplicated*

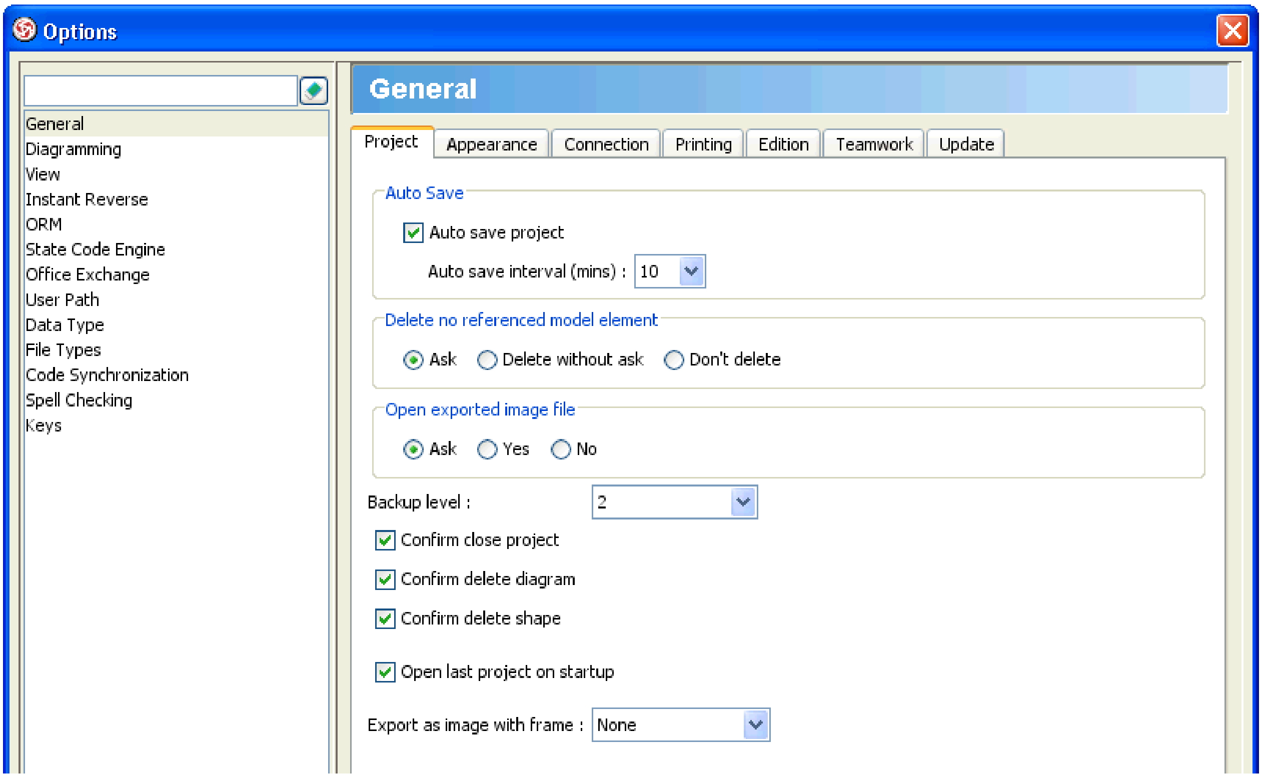
Alignment Guide

Alignment Guide may be shown on diagram when you dragging the shapes on diagram.

Change Alignment Guide Options

You can change the alignment guide on option dialog box.

1.Select **Tool > Options...** to show option dialog box.



Auto Save

☑ Auto save project

Auto save interval (mins) : 10

Delete no referenced model element

● Ask ○ Delete without ask ○ Don't delete

Open exported image file

● Ask ○ Yes ○ No

Backup level : 2

☑ Confirm close project

☑ Confirm delete diagram

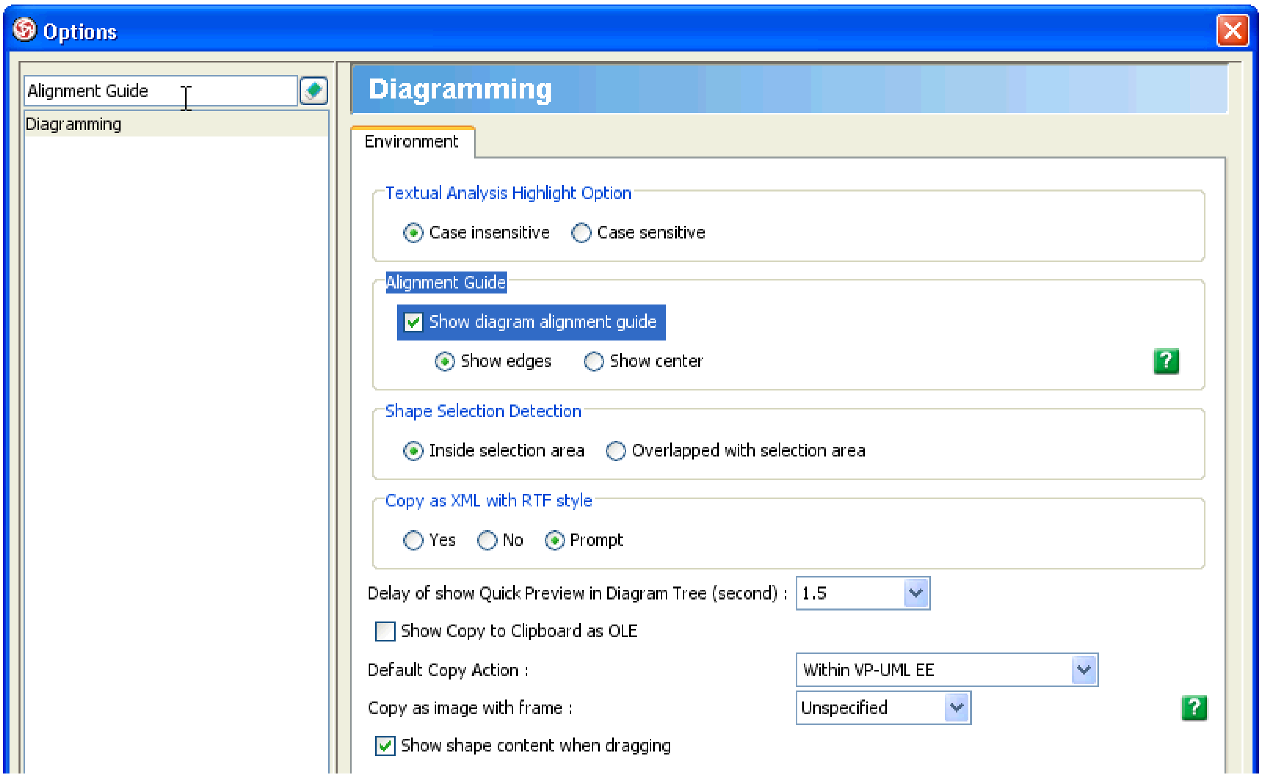
☑ Confirm delete shape

☑ Open last project on startup

Export as image with frame : None

Figure 2-72 *Option dialog box*

2.Typing *Alignment guide* on top-left text field, you will find that the options for alignment guide is shown on **Diagramming** category, **Environment** tab. You can enable/disable the alignment guide, and decide the alignment guide showing on edges or center of the shape.



Textual Analysis Highlight Option
● Case insensitive ○ Case sensitive

Alignment Guide
☑ Show diagram alignment guide
● Show edges ○ Show center

Shape Selection Detection
● Inside selection area ○ Overlapped with selection area

Copy as XML with RTF style
○ Yes ○ No ● Prompt

Delay of show Quick Preview in Diagram Tree (second) : 1.5

□ Show Copy to Clipboard as OLE

Default Copy Action : Within VP-UML EE

Copy as image with frame : Unspecified

☑ Show shape content when dragging

Figure 2-73 *Change Alignment Guide Options*

Show Edges Alignment Guide

1.Dragging up/down

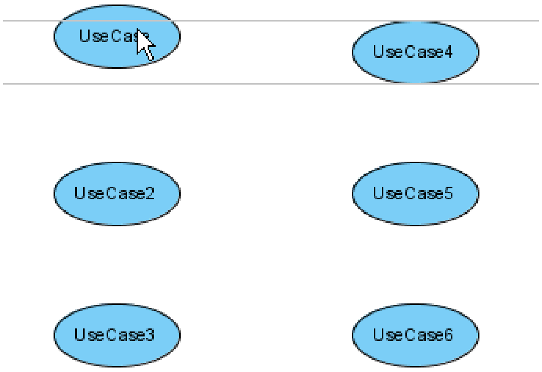


Figure 2-74 *Dragging up/down*

2.Dragging left/right

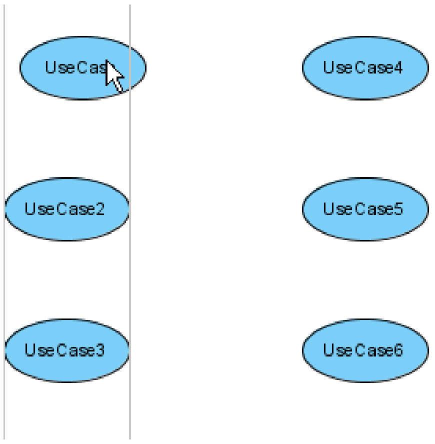


Figure 2-75 *Dragging left/right*

Show Center Alignment Guide

1.Dragging up/down

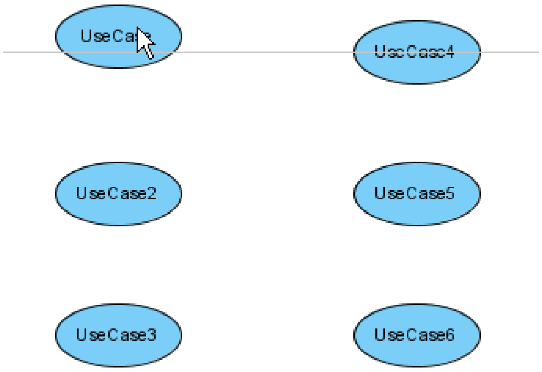


Figure 2-76 *Dragging up/down*

2.Dragging left-right

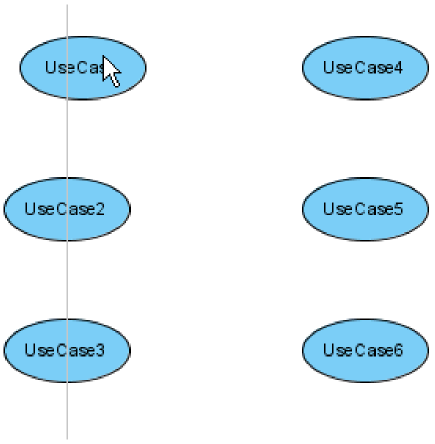


Figure 2-77 *Dragging left-right*

Reverse Connector Direction

VP-UML supports reversing the connector.

Reverse Connector Direction

1.For example, the connector comes from Action to Action2.

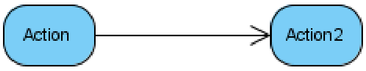


Figure 2-78 Try to reverse the connector

2.Right-click on the connector and select Reverse Connector from popup menu.

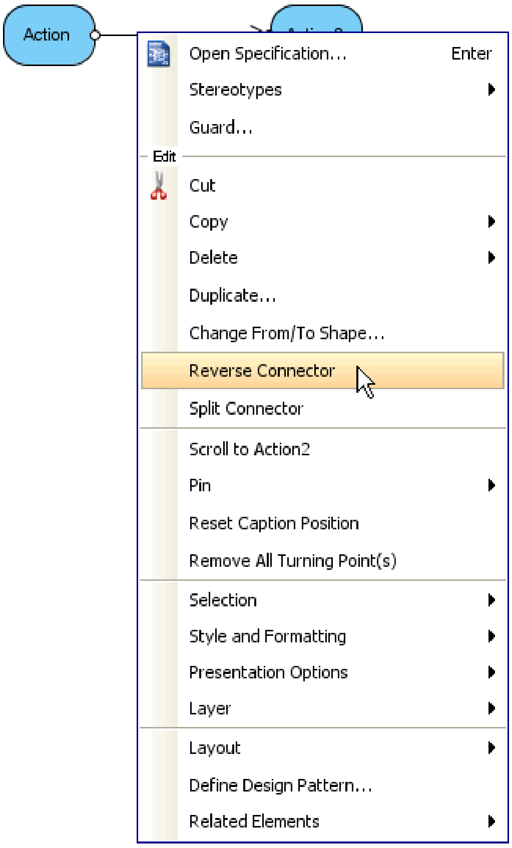


Figure 2-79 Reverse Connector

3.The connector is reversed, becomes from Action2 to Action.

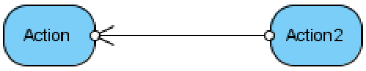


Figure 2-80 Connector reversed

Reverse Association Direction

Some connector contains more information, e.g. Association contains Association End that shows name, multiplicity, etc... Those information will be also reversed.

1.Try to reverse the Association with Association Ends

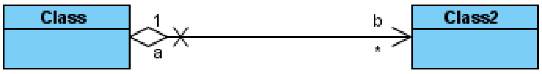


Figure 2-81 Association with Association Ends

2.Also right-click on the association and select Reverse Connector from popup menu.

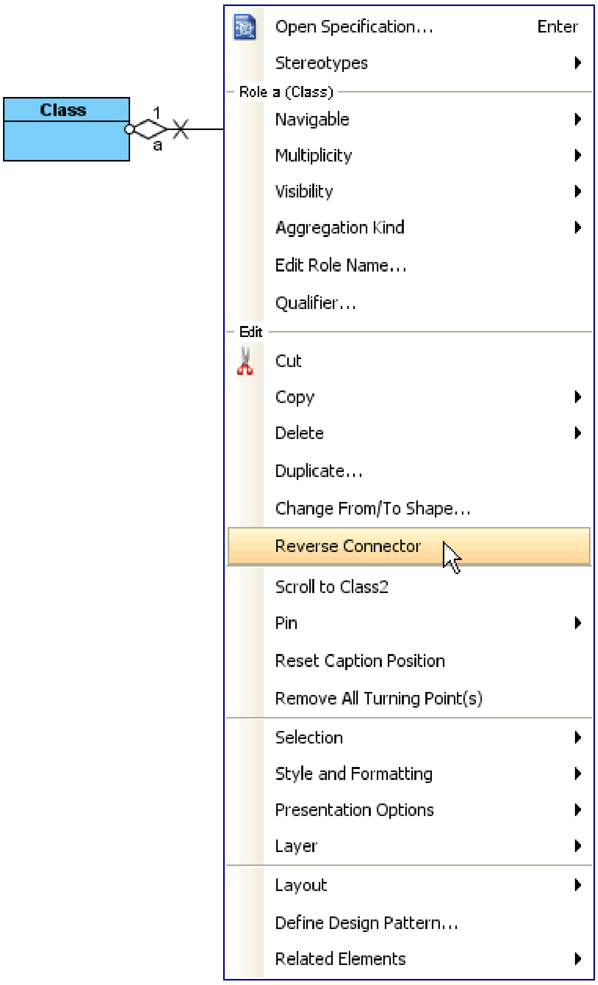


Figure 2-82 *Reverse Connector*

3.The association with its association ends are reversed.

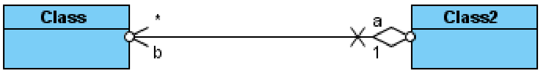


Figure 2-83 *Association reversed*

Reverse Sequence Message Direction

1.You can also reverse message on sequence diagram.

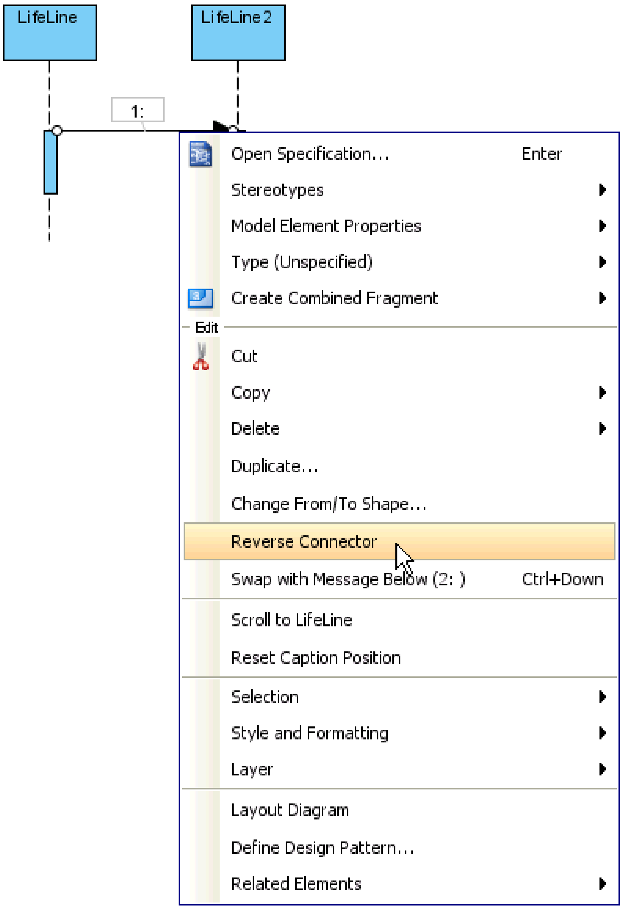


Figure 2-84 *Reverse Connector is supported on message*

2.But some messages do not support reverse connector, such as call message with referencing the lifeline's operation, and create message.

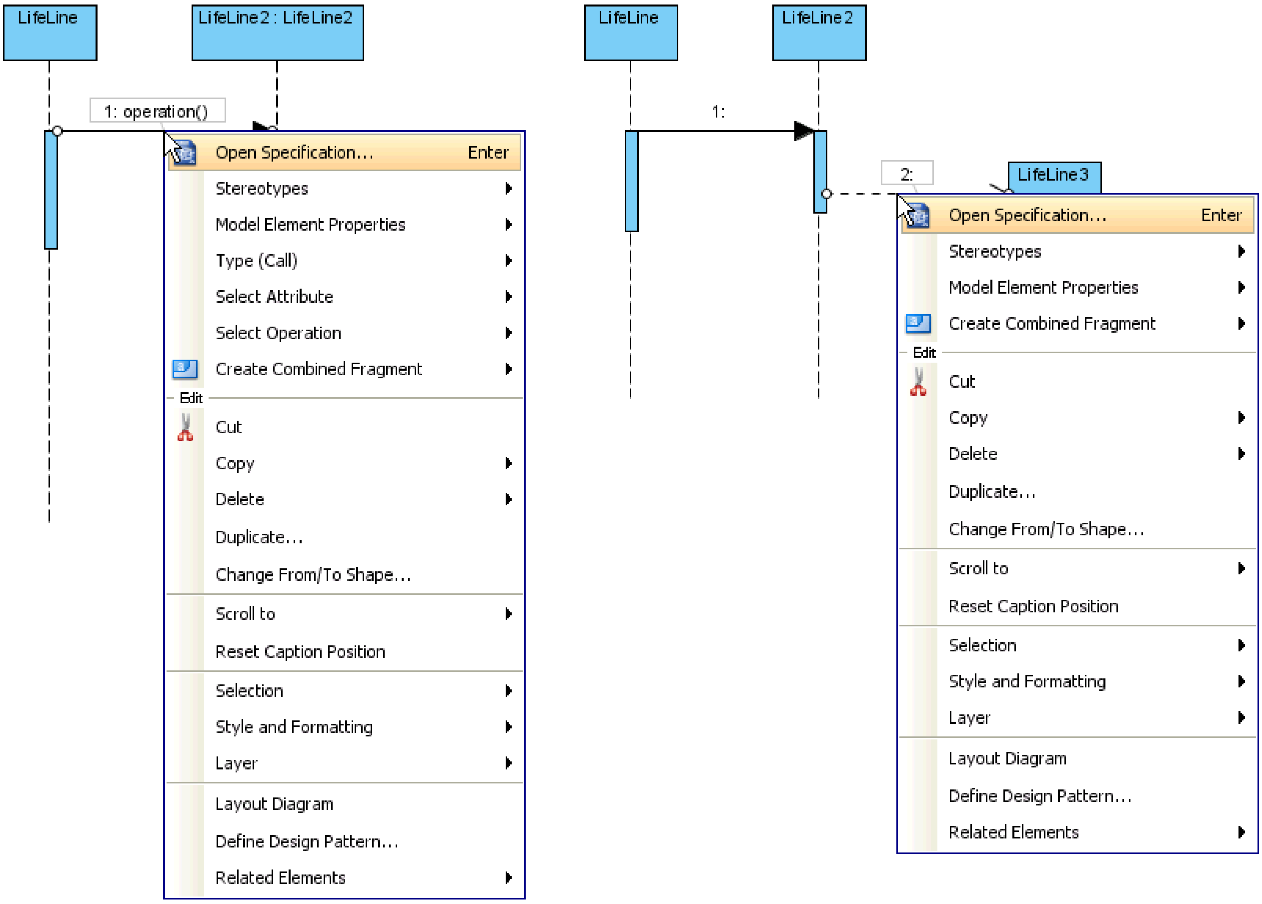


Figure 2-85 *Reverse Connector is not supported on some messages*

Visualize Model Element to Diagram

VP-UML supports models and diagrams. A model may be shown on more than one diagram. A diagram may also show a model more than one time.

Drag Model Element from Tree to Diagram

To show a model on a diagram, you may drag the model from **Model Explorer** (or **Diagram Navigator**) into a diagram.

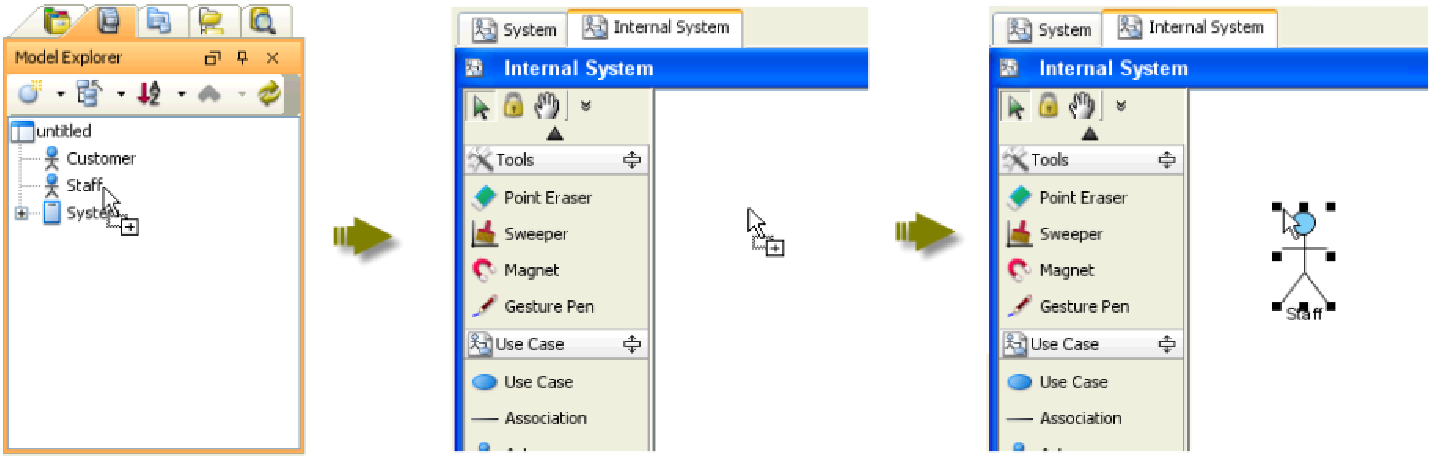


Figure 2-86 *Drag and drop from tree to diagram to show a model on diagram*

Visualize related model element

You may also want to show another models which related with the models on the diagram.
1.To do so, select the model on diagram, right-click and select **Related Elements > Visualize Related Model Element...** from popup menu.

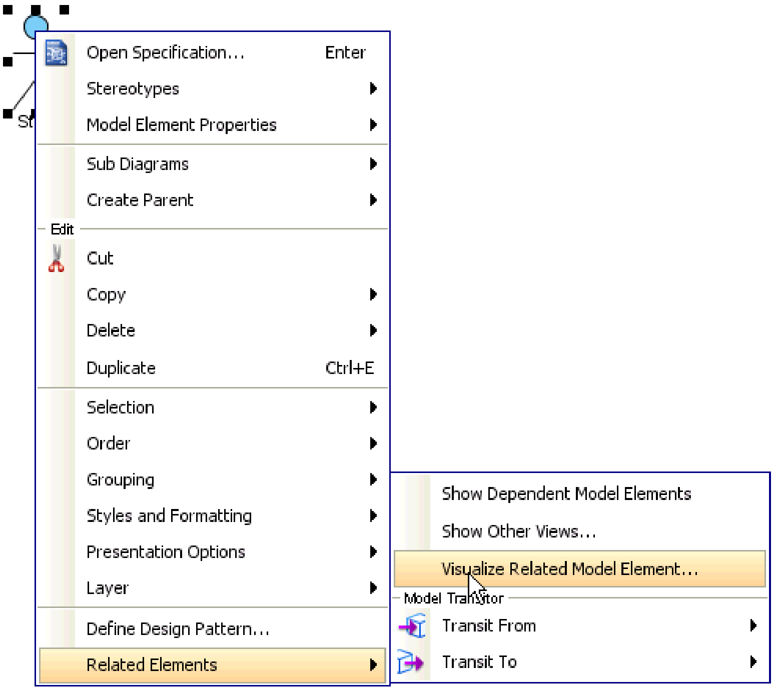
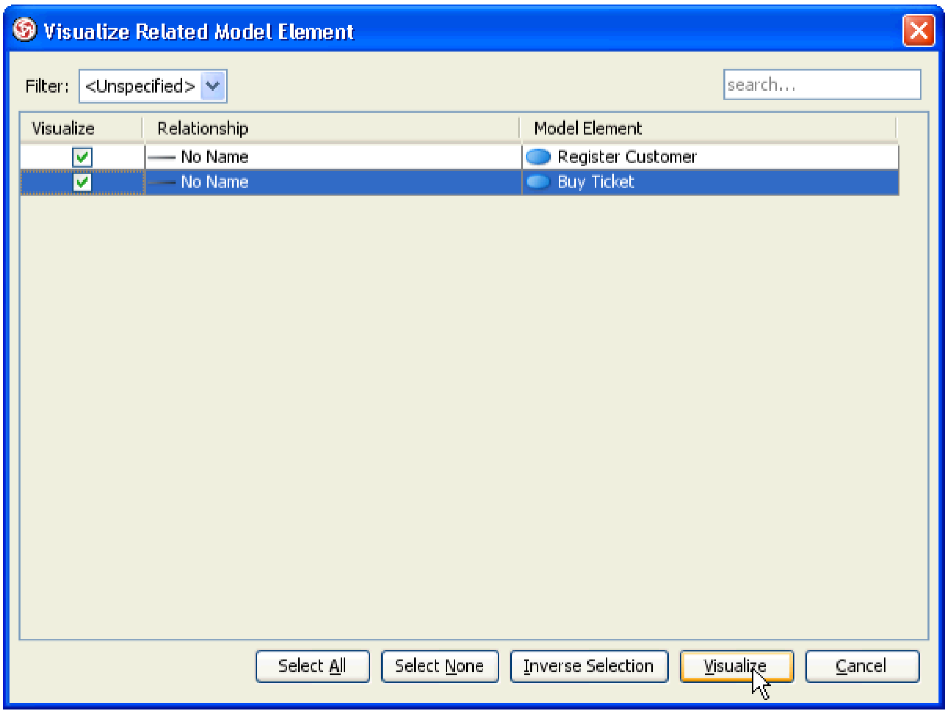


Figure 2-87 *Visual Related Model Element*

2.Select the related model elements you want to be shown on the diagram and select **Visualize.**



Visualize	Relationship	Model Element
☑	No Name	Register Customer
☑	No Name	Buy Ticket

Figure 2-88 *Visualize Related Model Element dialog box*

3.The related models are shown on the diagram.

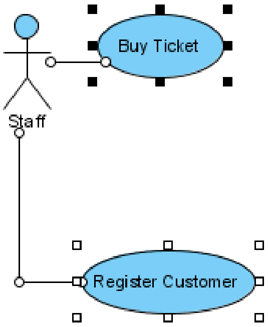


Figure 2-89 *Related model elements are visualized*

Model commenting

VP-UML supports comments on models.

Adding Comment to Model Element

1.To add comment into model, right-click on model and select **Open Specification...** from popup menu.

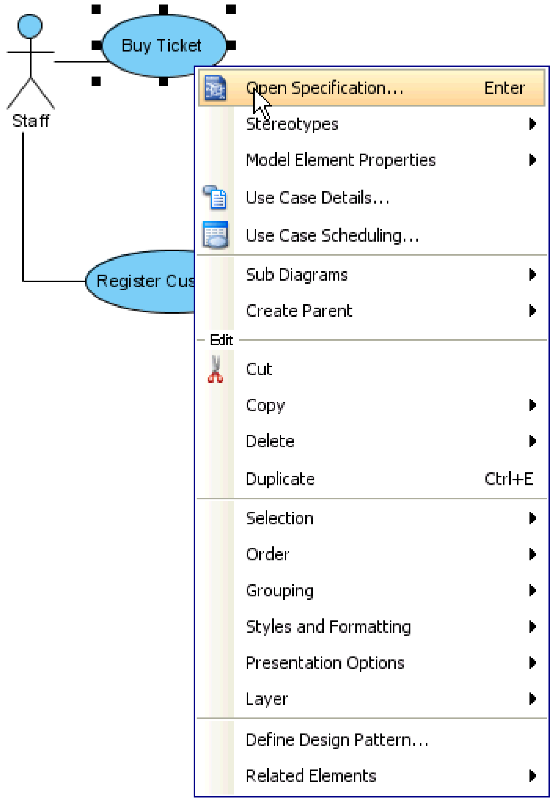
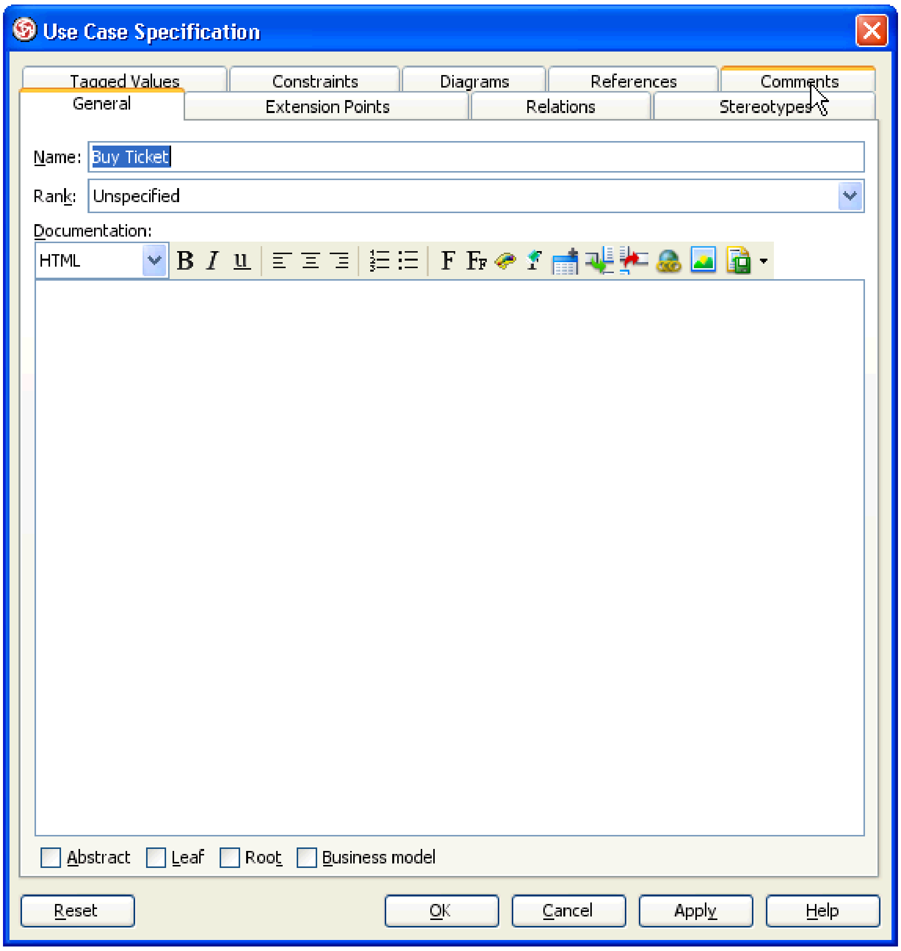


Figure 2-90 *Open Specification*

2.Select **Comments** tab.



Name: Buy Ticket

Rank: Unspecified

□ Abstract □ Leaf □ Root □ Business model

Figure 2-91 Comments tab

3.Select **Add** button to create a comment.

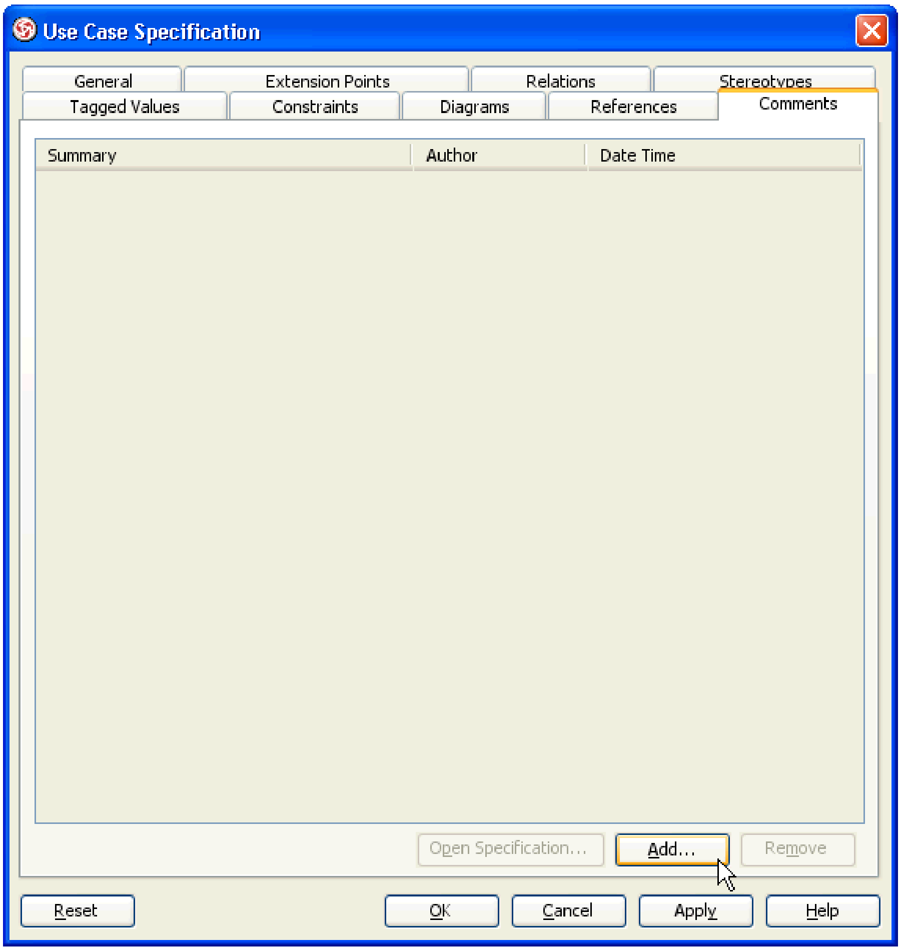
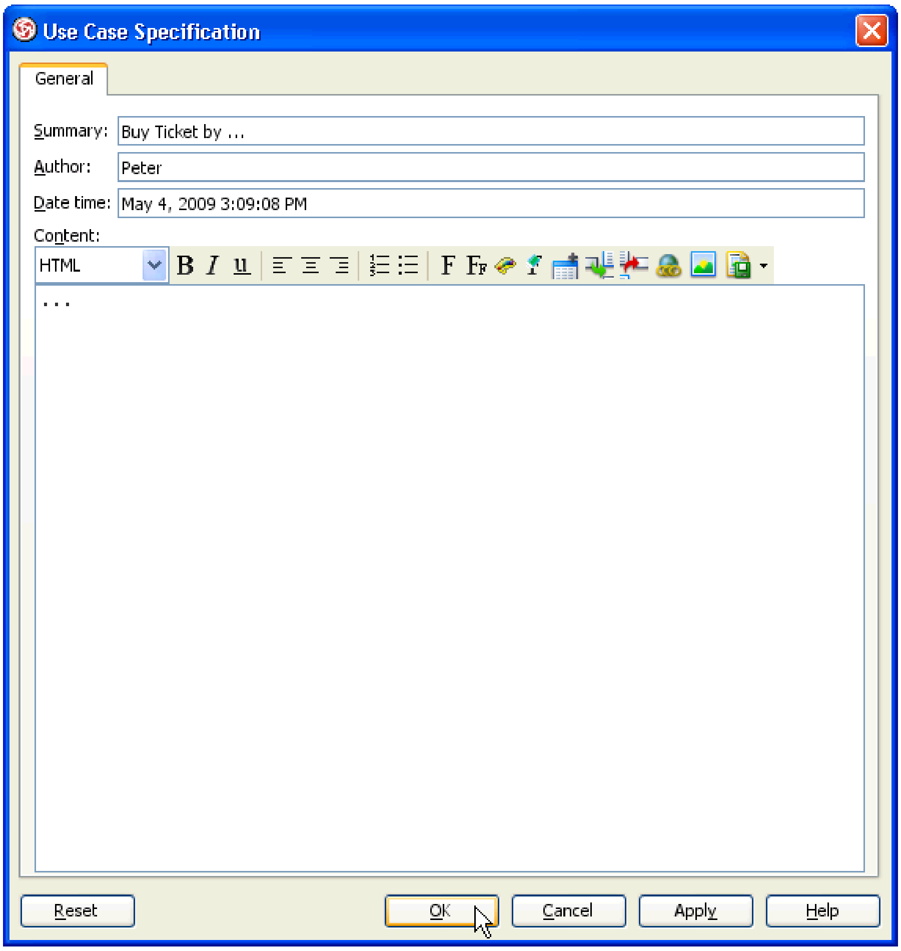


Figure 2-92 Add comment

4.Enter the name, author, etc... of the comment. And then select **OK** button to confirm adding the comment.



Summary: Buy Ticket by ...

Author: Peter

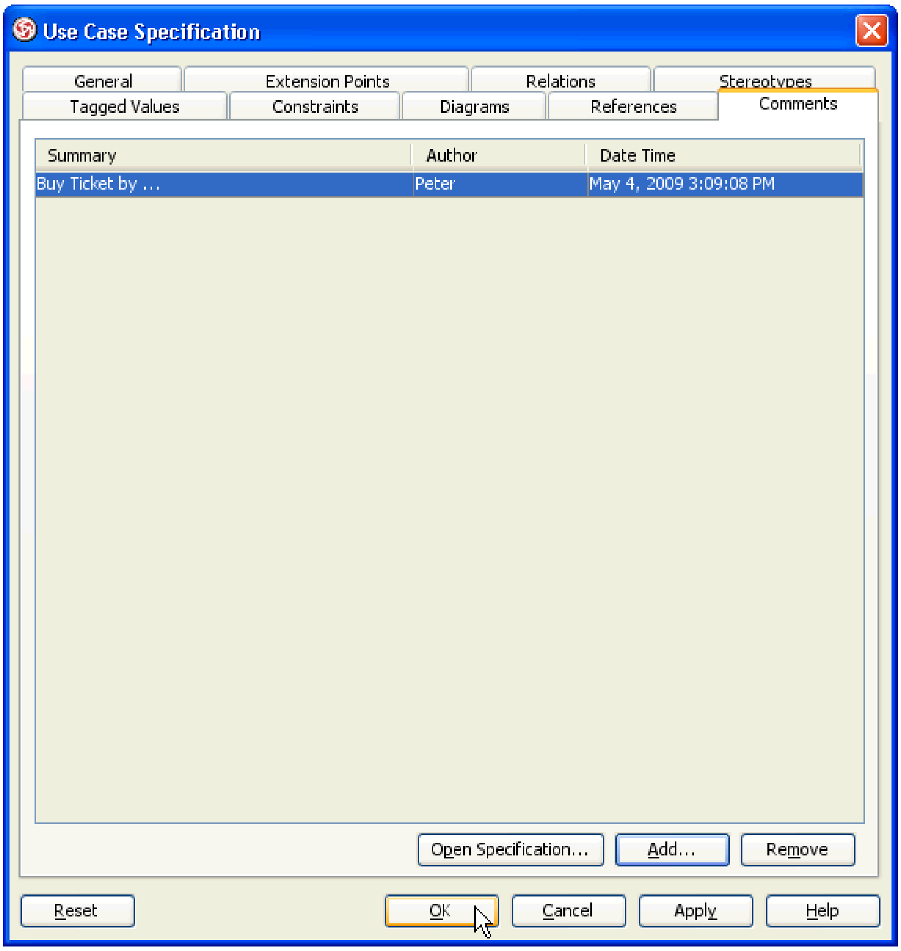
Date time: May 4, 2009 3:09:08 PM

Content: HTML

...

Figure 2-93 *Contents of comment*

5.Comment(s) is shown on **Comments** tab.

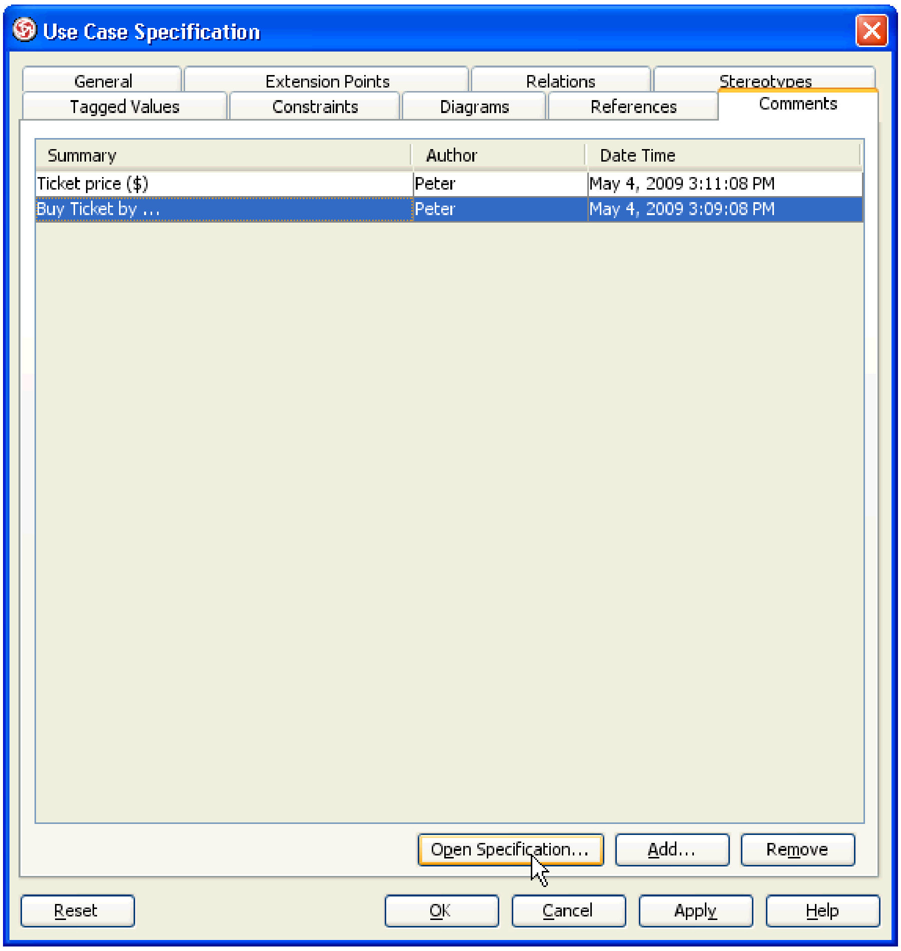


Summary	Author	Date Time
Buy Ticket by ...	Peter	May 4, 2009 3:09:08 PM

Figure 2-94 *Comments shown*

Managing Comment of Model Element

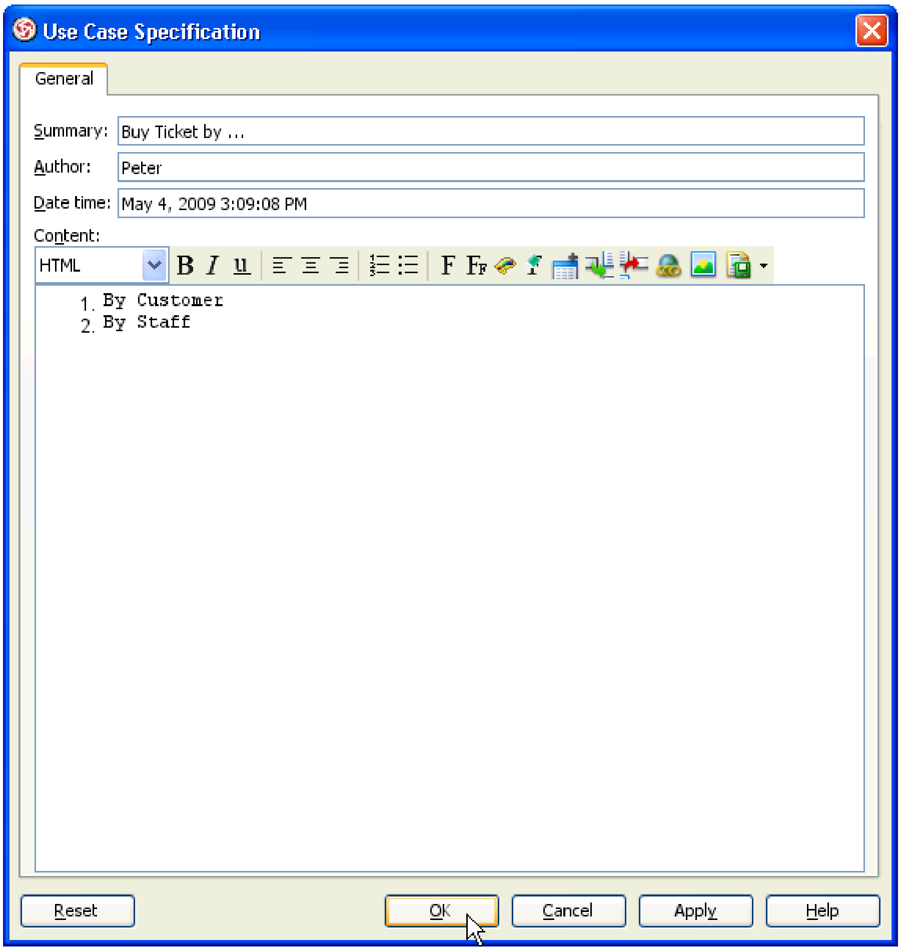
1.To modify the comment, also on **Specification** dialog box, **Comments** tab, select the comment and select **Open Specification...** button.



Summary	Author	Date Time
Ticket price ($)	Peter	May 4, 2009 3:11:08 PM
Buy Ticket by ...	Peter	May 4, 2009 3:09:08 PM

Figure 2-95 *Open Specification of the comment*

2.Modify the name, author, etc... of the comment and select **OK** to confirm changing.



Summary: Buy Ticket by ...

Author: Peter

Date time: May 4, 2009 3:09:08 PM

1. By Customer
2. By Staff

Figure 2-96 *Modify contents of comment*

Pinnable Connector end

Connector can be pinned with its shape.

Temporary adjust connector end

1.You may wants the connectors connect with shape at same point on the diagram. To do so, drag the connector's start/end point.

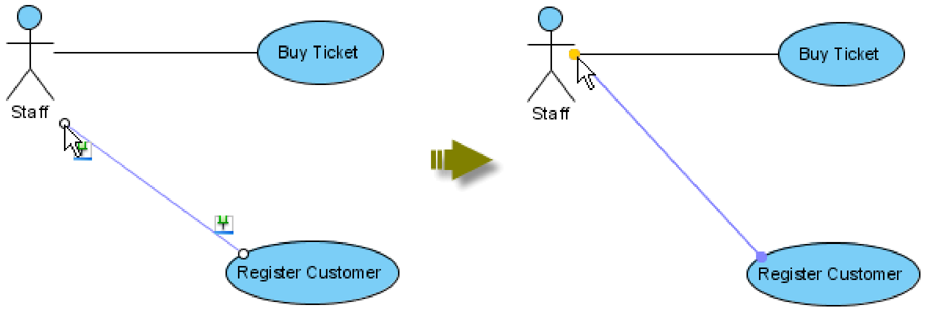
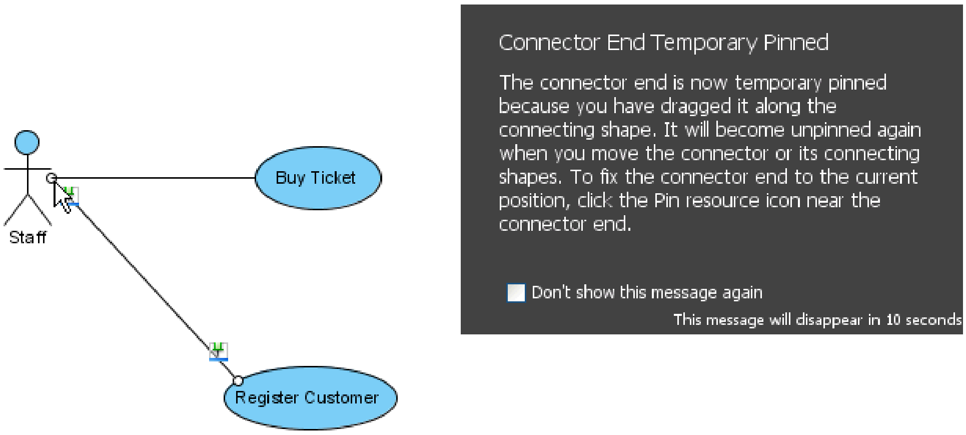


Figure 2-97 *Drag connector point*

2.The connector is temporary pinned. A description will be shown on diagram to describe how to pin the connector.



Connector End Temporary Pinned

The connector end is now temporary pinned because you have dragged it along the connecting shape. It will become unpinned again when you move the connector or its connecting shapes. To fix the connector end to the current position, click the Pin resource icon near the connector end.

□ Don't show this message again

This message will disappear in 10 seconds

Figure 2-98 *Connector is temporary pinned*

3.But moving the from/to shapes of the connector, the connector will be unpinned.

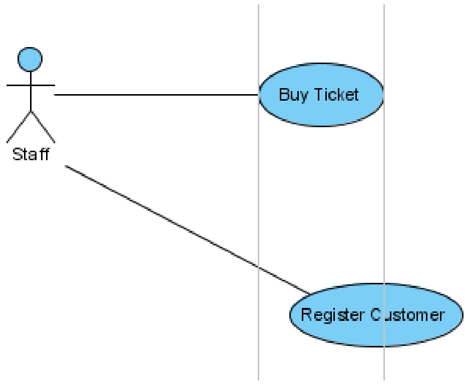


Figure 2-99 *Connector is unpinned automatically*

Pin Connector End

1.To pin the connector, select the connector and select the **Pin** resource.

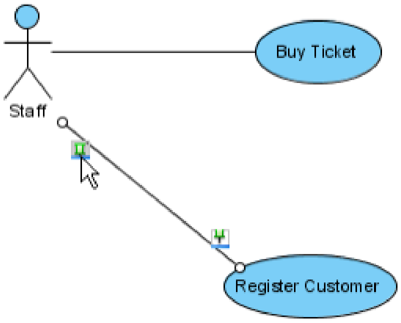


Figure 2-100 *Pin the connector by selecting Pin resource*

2.Drag the connector point again, no description of temporary pin will be shown this time.

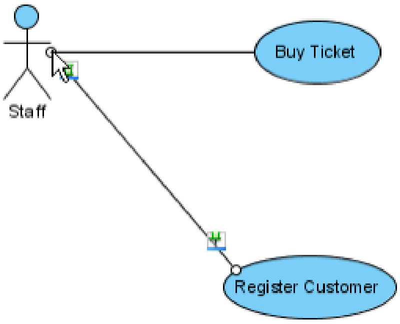


Figure 2-101 *Drag connector point*

3.Moving the shape, the connector won't be unpinned.

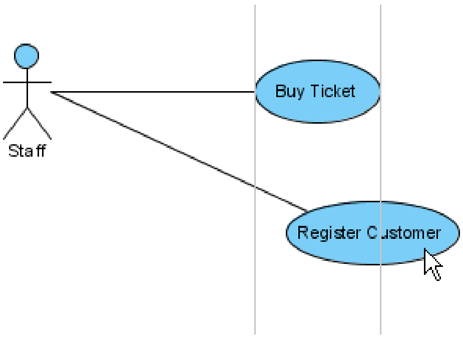


Figure 2-102 *Connector is still pinned*

Align and distribute Diagram Elements

This feature provides a facility to align selected diagram elements. You can align using toolbar, popup menu or group resource.

Aligning Diagram Elements

To align diagram elements, select the diagram elements on diagram.

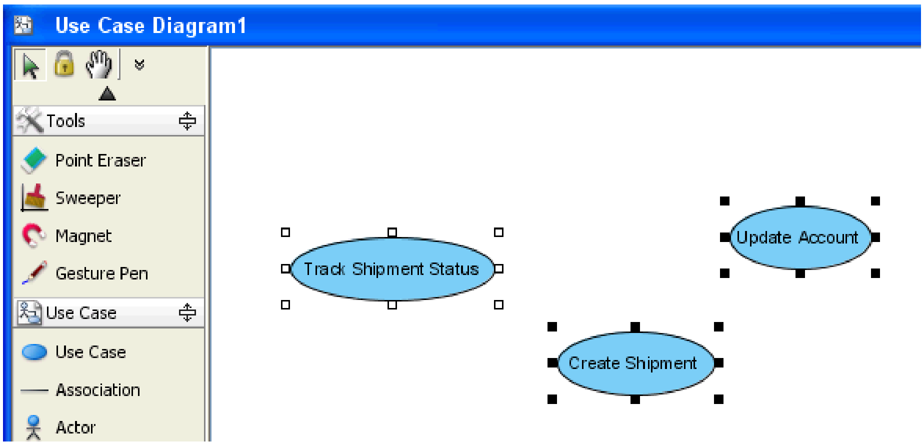


Figure 2-103 Select shapes

Using Toolbar

1.Select alignment on toolbar.

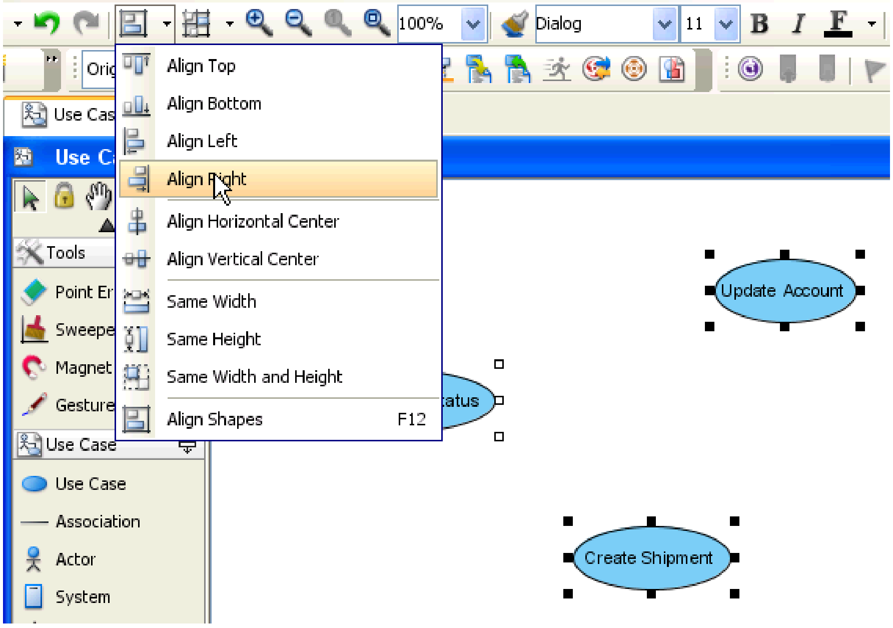


Figure 2-104 Align Right

2.The shapes are aligned to right. They are based on the shape with no-filled selector □.

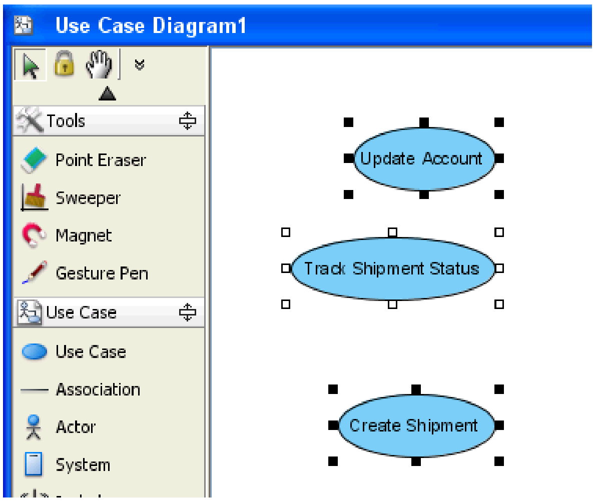


Figure 2-105 *Shapes are aligned following the no-filled selector shape*

3.To change the no-filled selector shape, you may press Ctrl key and click the shape 2 times to deselect and select the shape.

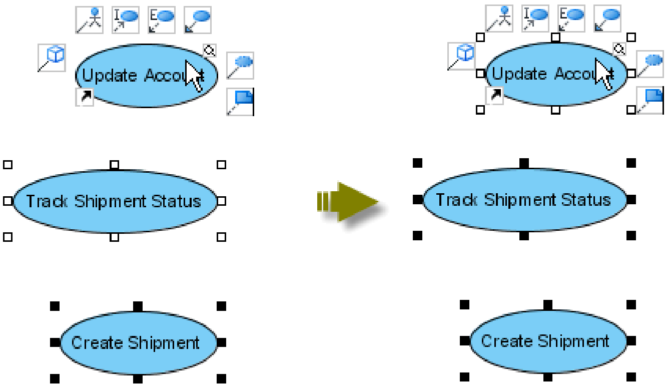


Figure 2-106 *Change shape to be no-filled selector shape*

Using Grouping Resources

1.Mouse over on any selected shape, Grouping Resources will be shown.

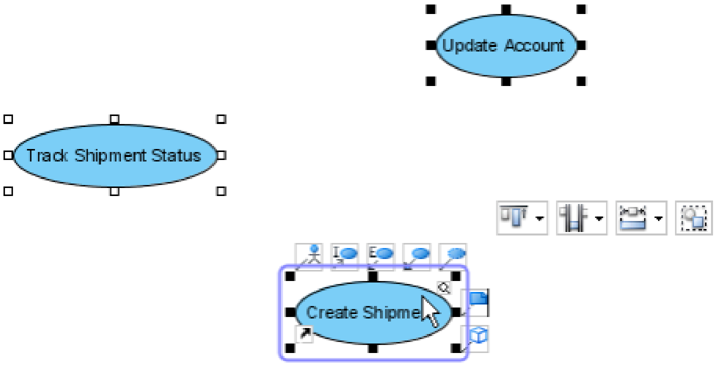


Figure 2-107 *Grouping Resources shown on selected shape*

2.Select alignment on the Grouping Resources.

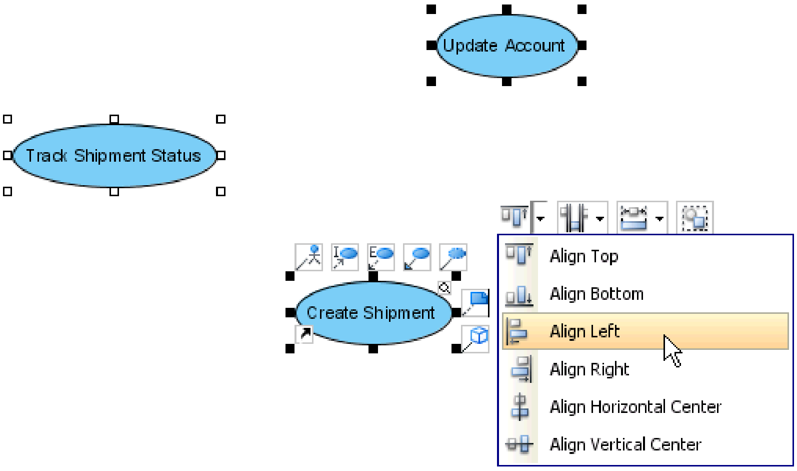


Figure 2-108 *Select Align Left on resources*

3.The selected shape will become no-filled selector shape and the alignment will follow the selected shape.

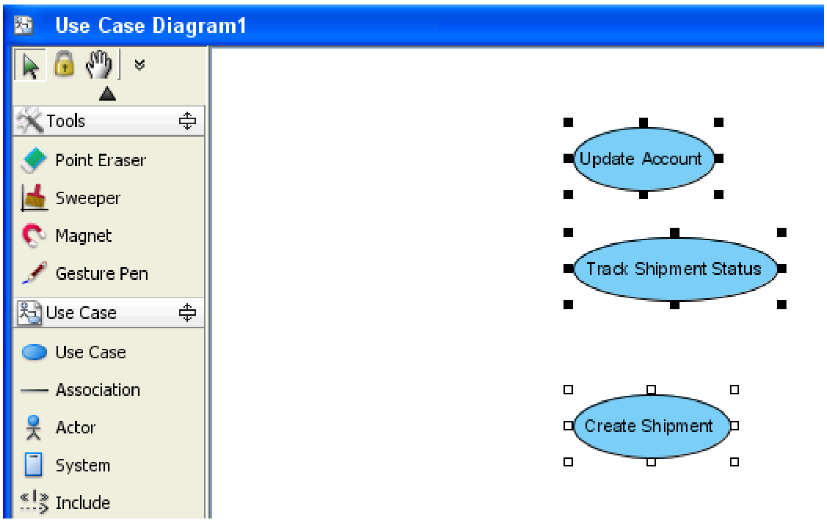


Figure 2-109 *Shapes are aligned*

Making Diagram Elements Same Width and Height

Besides align the shapes, also can resize the shapes.

Using Toolbar

Select **Same Width and Height** on toolbar.

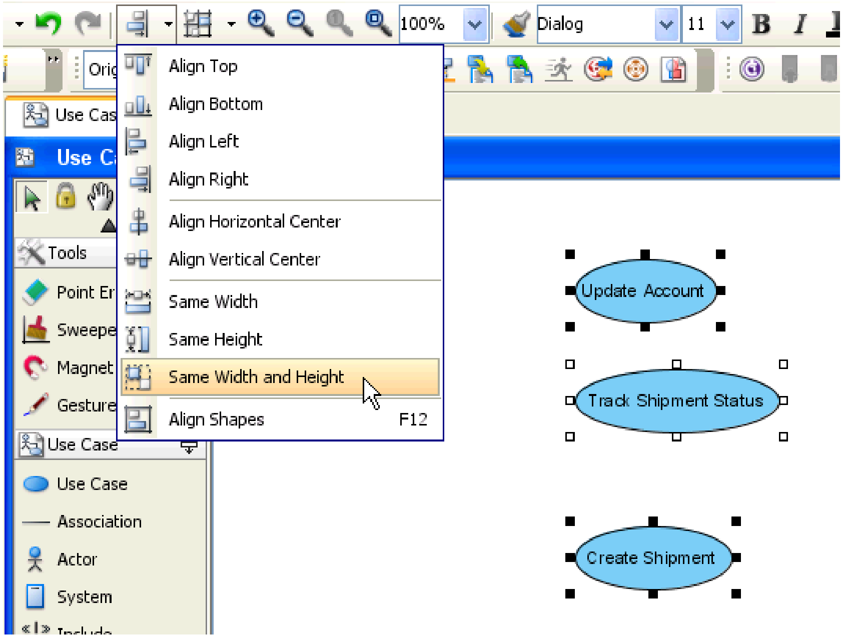


Figure 2-110 Resize by toolbar button

The shapes are resized.

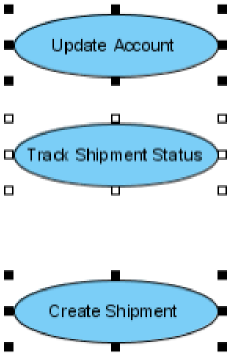


Figure 2-111 Shapes are resized

Using Grouping Resources

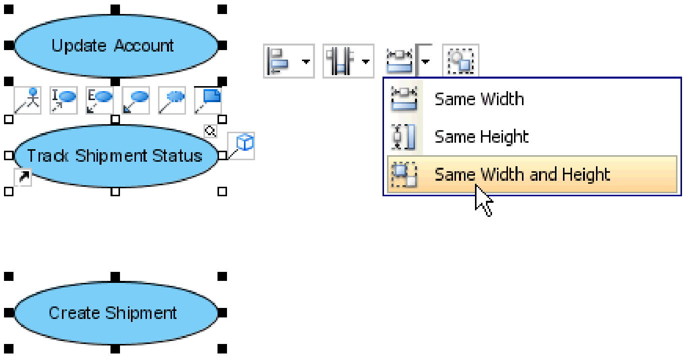


Figure 2-112 Resize by grouping resources

Using Align Shapes Dialog box

To show **Align Shapes Dialog**, you may select **Align Shapes** on toolbar or select **Edit > Align Shapes > Align Shapes...** in menu.

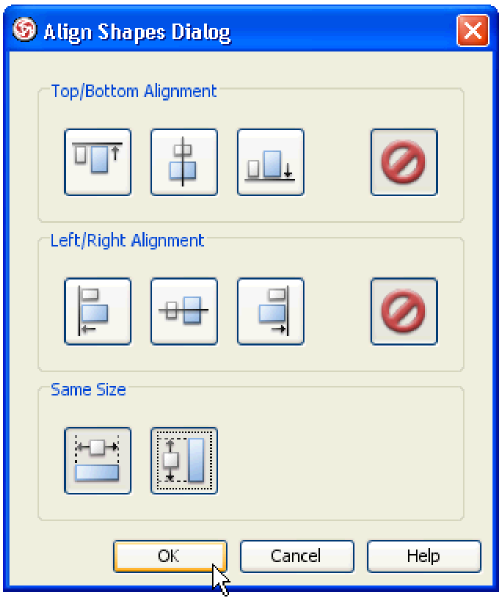


Figure 2-113 *Select Same Width and Same Height on Align Shapes Dialog*

Distribute Diagram Elements

Besides alignment and resize, you can distribute the diagram elements.

Using Toolbar

Select **Distribute Shapes Vertically**.

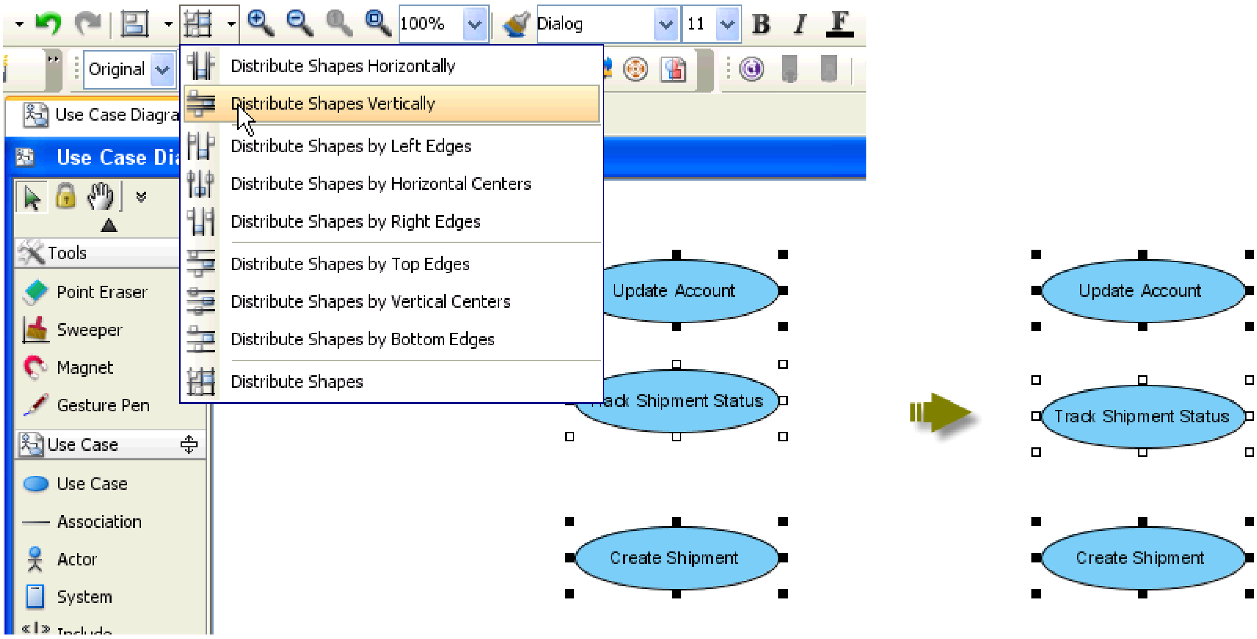

100%
Dialog
11
Original
Use Case Diagra
Use Case Di
Distribute Shapes Horizontally
Distribute Shapes Vertically
Distribute Shapes by Left Edges
Distribute Shapes by Horizontal Centers
Distribute Shapes by Right Edges
Distribute Shapes by Top Edges
Distribute Shapes by Vertical Centers
Distribute Shapes by Bottom Edges
Distribute Shapes
Tools
Point Eraser
Sweeper
Magnet
Gesture Pen
Use Case
Use Case
Association
Actor
System
Include
Update Account
rack Shipment Status
Create Shipment
Update Account
Track Shipment Status
Create Shipment

Figure 2-114 *Shapes are located with same distance*

Using Grouping Resources

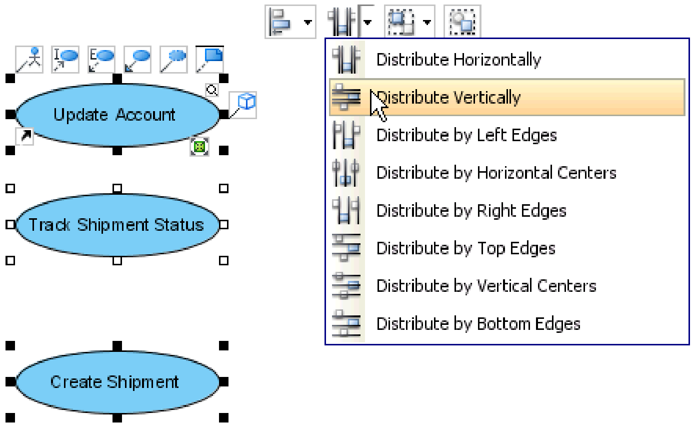


Figure 2-115 *Distribute shapes in Grouping Resource*

Annotate Diagram Elements with UML Note Shape

Creating UML Note with Resource Centric

1.Move mouse over Anchor - > Note resource.

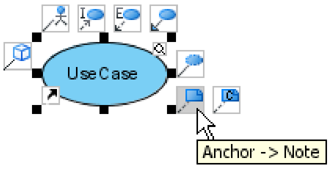


Figure 3-1 Mouse over recource

2.Drag resource to empty space on diagram pane.

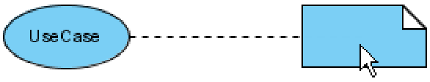


Figure 3-2 Dragging resource

3.Release the mouse, new connector and note are created.

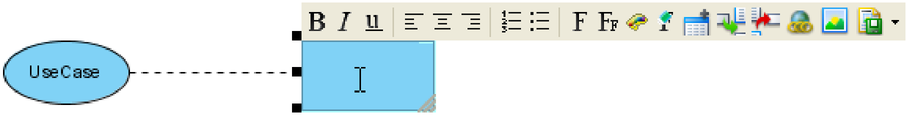


Figure 3-3 Note created

Connecting UML Note Shape with Anchor

1.Click Anchor button from diagram toolbar

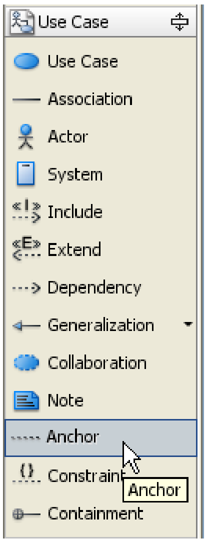


Figure 3-4 Diagram toolbar

2.Move mouse over shape which connect from.

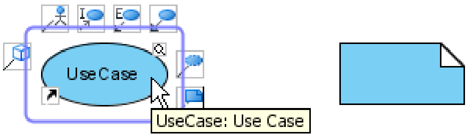


Figure 3-5 Connecting from shape

3.Drag to note which connect to.

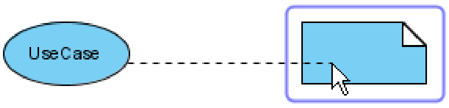


Figure 3-6 Conencting to note

4.Release the mouse, new anchor is created.

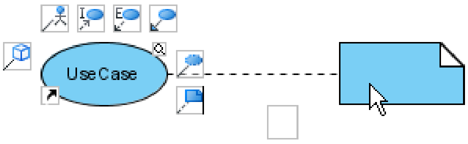


Figure 3-7 Anchor created

Editing UML Note content

1.Double click note to start editing content.

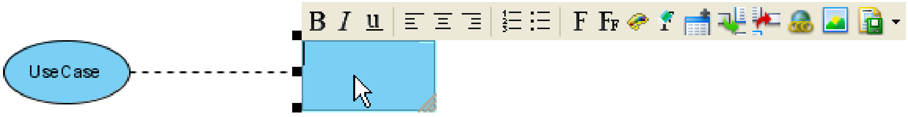


Figure 3-8 Start editing

2. Drag bottom right to resize note.

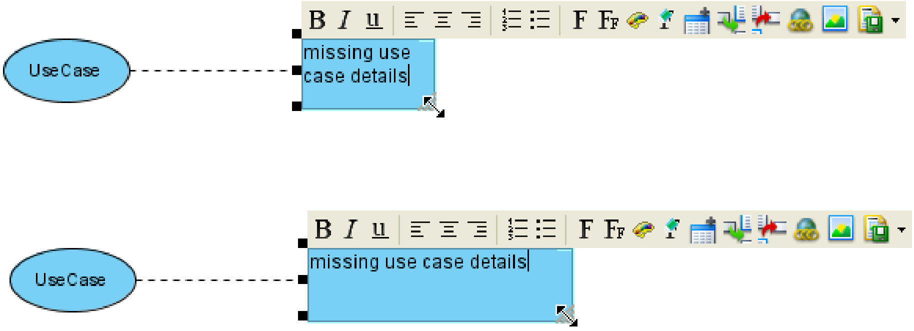


Figure 3-9 Resizing

3.Click on diagram to confirm editing.

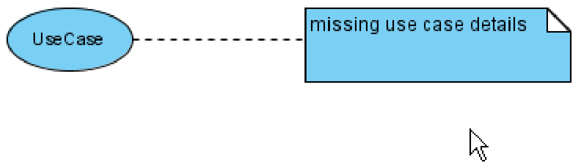


Figure 3-10 *Confirm editing*

Annotate Diagram Element with Callout Shape

Creating Callout Shape

1.Click Callout button from diagram toolbar.

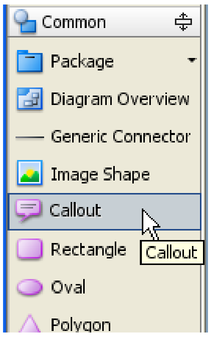


Figure 3-11 *Diagram toolbar*

2.Click empty space on diagram pane.

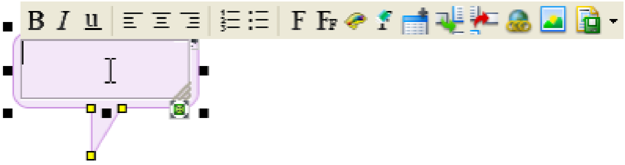


Figure 3-12 *Callout created*

Adjusting direction of Callout Shape Pointer

To adjust direction of callout shape pointer, simply drag the end point.

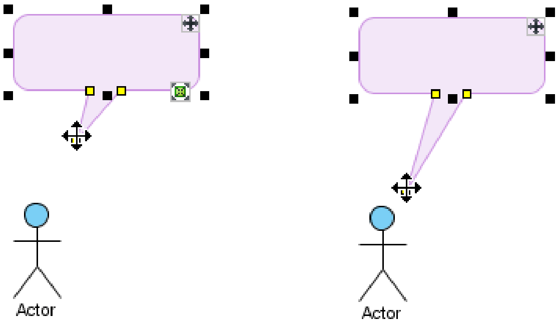


Figure 3-13 *Dragging end point*

To move callout shape without adjust callout shape pointer, drag top right move icon insist of shape.

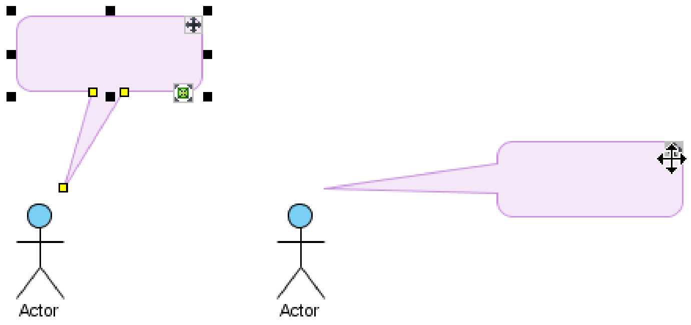


Figure 3-14 Dragging move icon

Editing Callout Shape content

1.Double click callout shape to start editing.

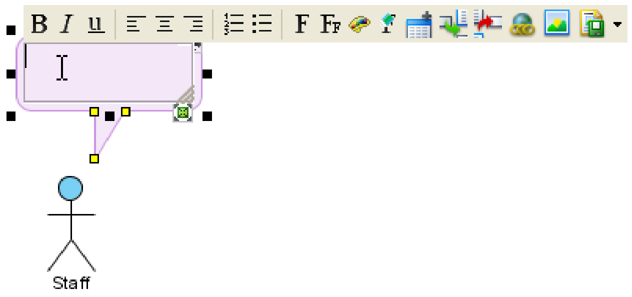
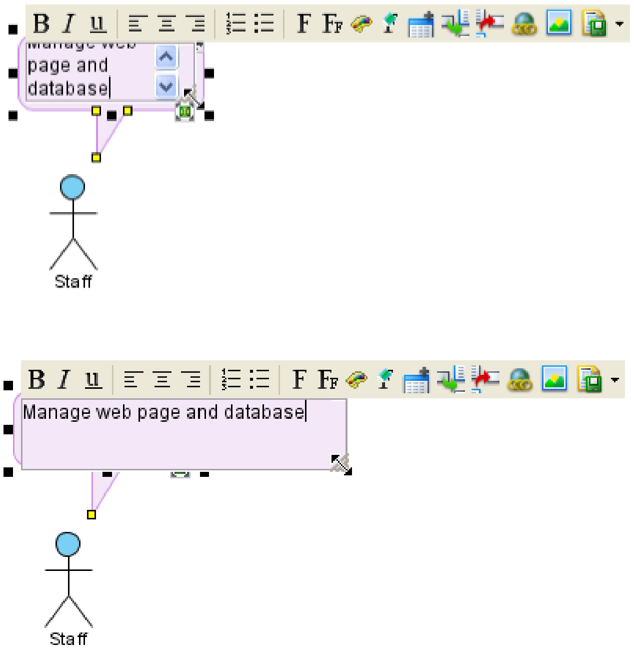


Figure 3-15 Start editing callout

2.Drag bottom right to resize callout shape.



page and
database

Manage web page and database

Figure 3-16 Resizing callout

3.Click on diagram to confirm editing.

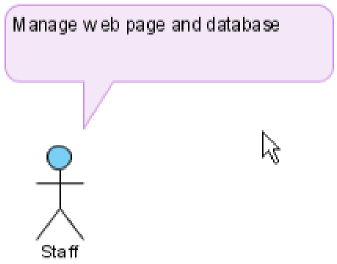


__Figure 3-17__ Confirm editing callout

Drawing Freehand Shapes

Showing Freehand Category

1.Click on diagram toolbar popup button, select **Category > Freehand.**

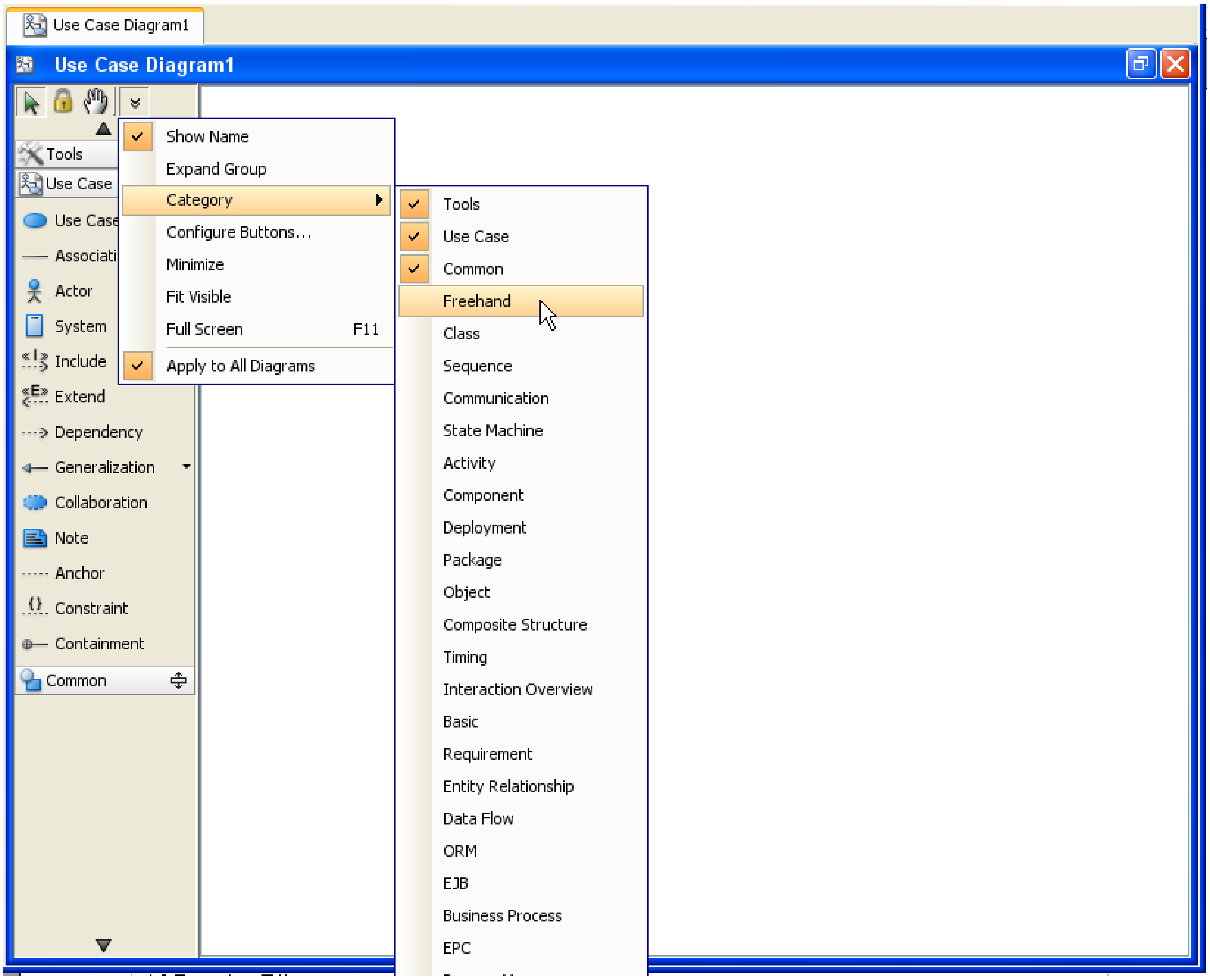


Figure 3-18 *Show freehand category by poup menu*

2.Freehand category shown.

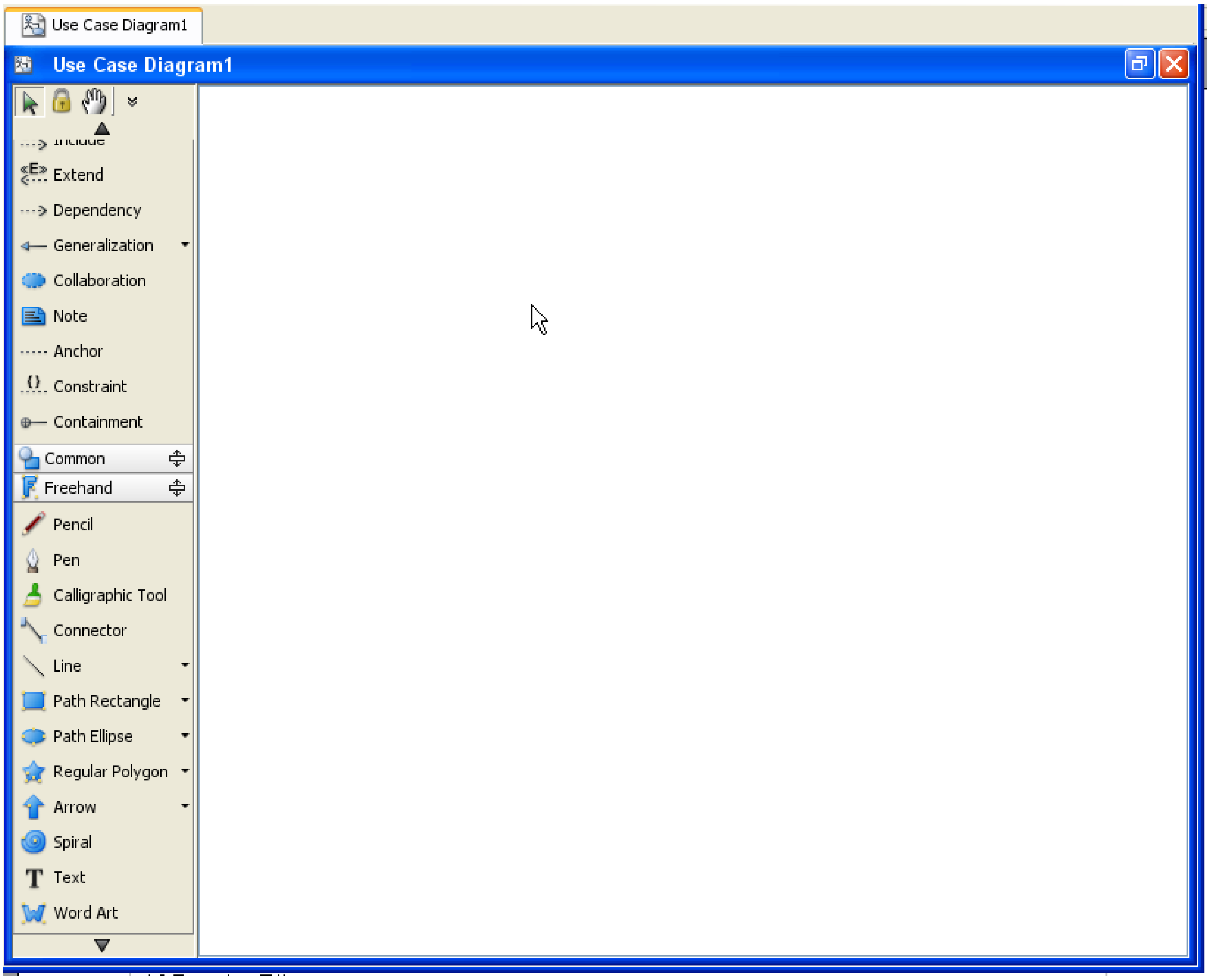
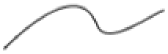
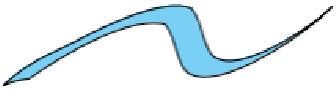
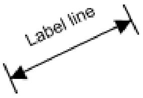
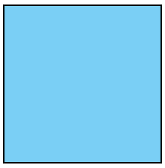
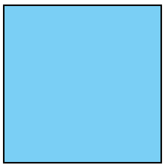
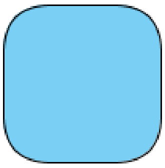
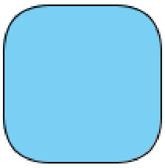
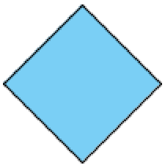
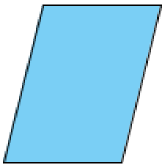
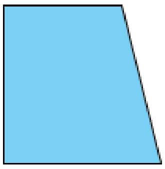
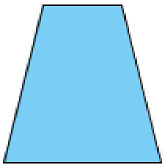
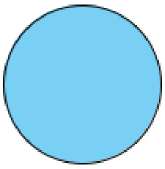
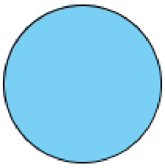
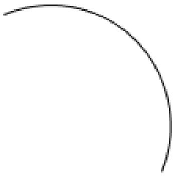
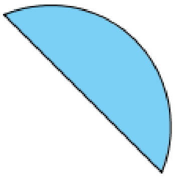
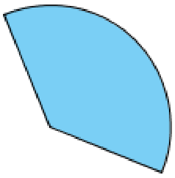
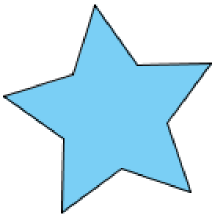


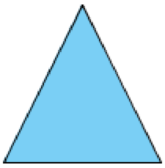
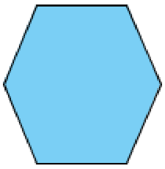
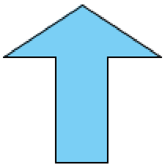
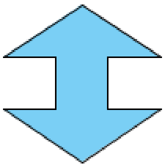
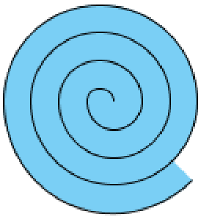
Figure 3-19 *Freehand category*

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	Text
Word Art	

Drawing Free Style Path with Pencil

1. Click Pencil button from diagram toolbar

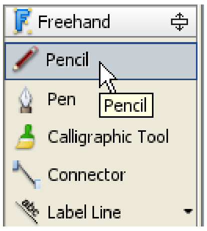


Figure 3-20 *Pencil button on diagram toolbar*

2. Press empty space on diagram pane and drag the outline

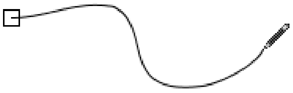


Figure 3-21 *Drawing with pencil*

3. Release the mouse, new freehand shape is created

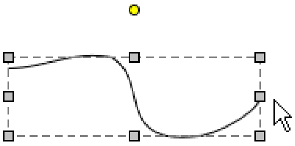


Figure 3-22 *Freehand shape created*

Activate the fine editing selector

Fine editing selector shows after freehand shape selected for 1 second
if you want to show it immediately, press keyboard 'N' key

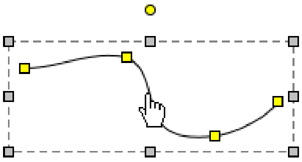


Figure 3-23 *Fine editing selector*

Press on the yellow box to select it, selected selector show as orange, more fine editing selector will show for curve adjustment.

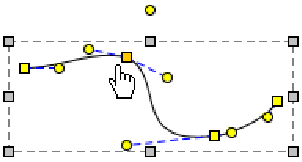


Figure 3-24 *Selected fine editing selector*

Drawing curve with Pen

1. Click Pen button from diagram toolbar.

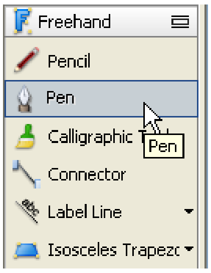


Figure 3-25 *Pen button on diagram toolbar*

2. Click empty space on diagram pane to create first point.

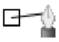


Figure 3-26 *Creating first poitnt*

3. Click other empty space on diagram pane to create second point, straight line created.

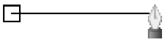


Figure 3-27 *Straight line created*

4. To creating curve, press and drag mouse, a indicate line shows.

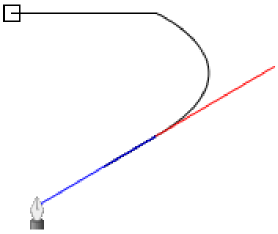


Figure 3-28
Creating curve

5. Release mouse to confirm.

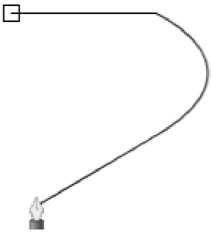


Figure 3-29
Curve created

6. Right click on the diagram can cancel the last point.

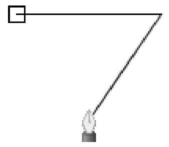


Figure 3-30 *Cancel*

7. Draw curve again.

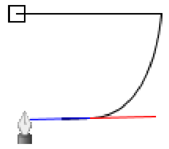


Figure 3-31 *Draw curve again*

8. Double click on diagram to confirm.

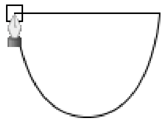


Figure 3-32 *Confirming creation*

9. New freehand shape is created.

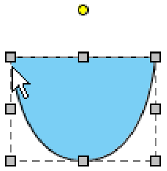


Figure 3-33 *Freehand shape created*

If finish creating on the start point square box, the shape becomes close path.

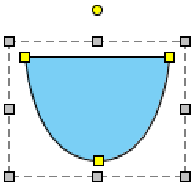


Figure 3-34 *Showing close path of freehand shape*

Drawing Calligraphic Path with Calligraph Tool

1. Click Calligraphic Tool button from diagram toolbar.

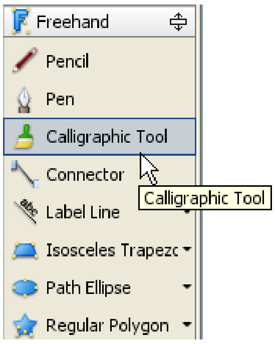


Figure 3-35 *Calligraphic Tool button on diagram toolbar*

2. Press empty space on diagram pane and drag the outline.

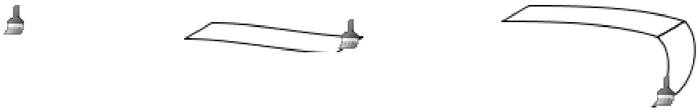


Figure 3-36 *Drawing Calligraphy*

3. Release the mouse, new freehand shape is created.

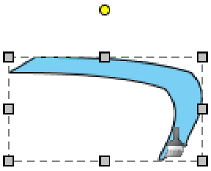


Figure 3-37 *Freehand shape created*

Try to combine more shapes.

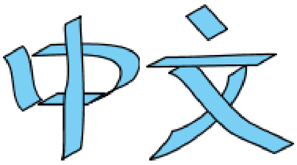


Figure 3-38 *Calligraphy example*

Draw Straight and Curve line with Connector

1. Click Connector button from diagram toolbar.

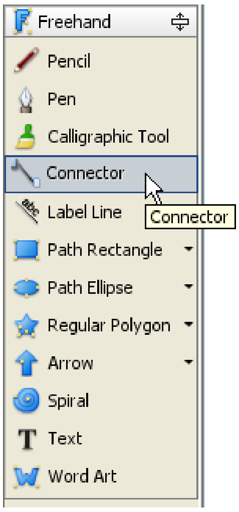


Figure 3-39 *Connector button on diagram toolbar*

2. Press on source shape and drag to destination shape.

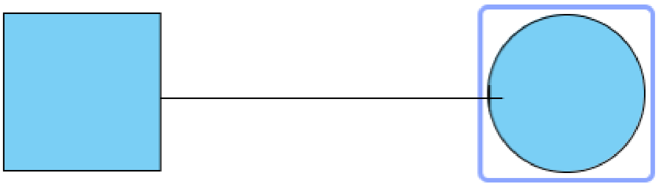


Figure 3-40
Connecting shapes

3. Release the mouse, new connector is created.

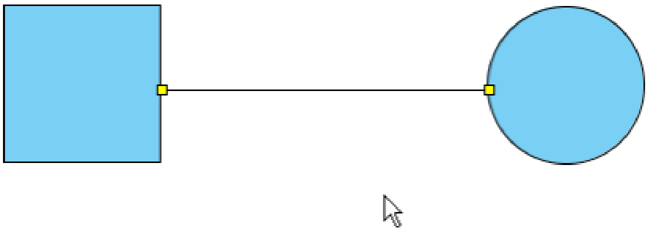


Figure 3-41
Line created

Edit straight line as curve line

1. Press on straight line.

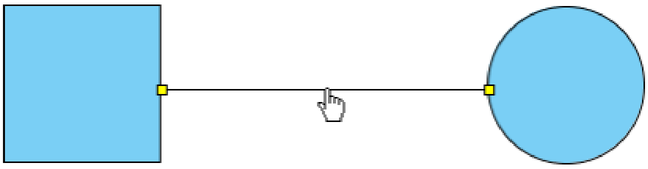


Figure 3-42 *Clicking on straight line*

3. Drag as curve.

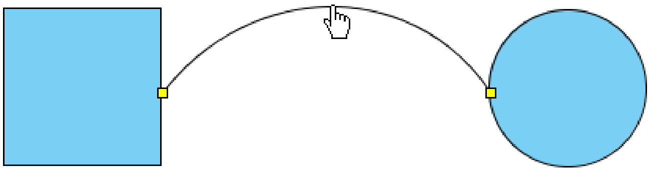


Figure 3-43 *Draging line as curve*

Split straight line

1. Ctrl click on split location, a mid point created.

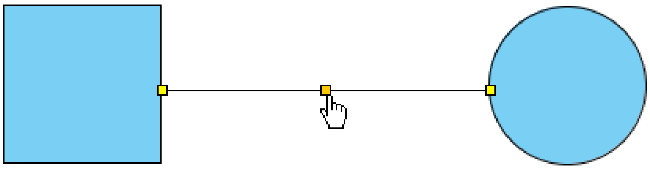


Figure 3-44
Splitting line

2. Drag mid point to split line.

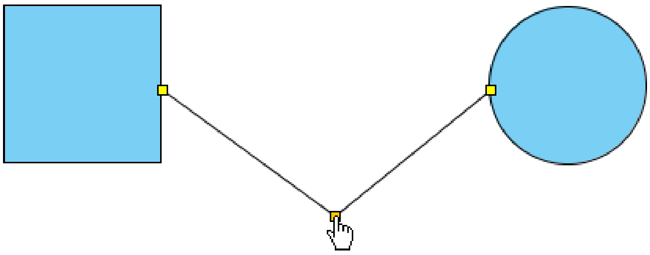


Figure 3-45
Moving mid point

Drawing Straight and Curved Line

1. Click Line button from diagram toolbar.

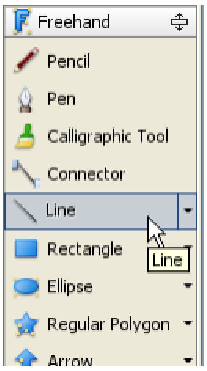


Figure 3-46 *Line button on diagram toolbar*

2. Press empty space on diagram pane and drag the outline.

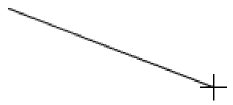


Figure 3-47
Dragging line

3. Release the mouse, new straight line is created.

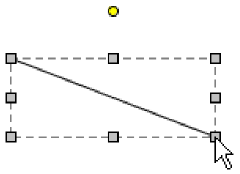


Figure 3-48
Line created

Edit straight line as curve line
1. Select shape for 1 second to show fine editing selector.

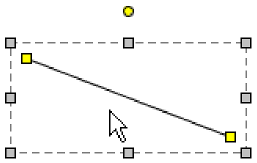


Figure 3-49 *Showing fine editing selector*

2. Press on straight line.

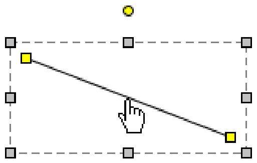


Figure 3-50 *Clicking on straight line*

3. Drag as curve.

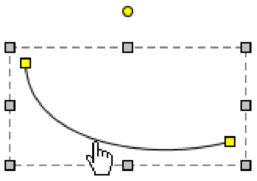


Figure 3-51 *Draging line as curve*

Split straight line

1. Ctrl click on split location, a mid point created.

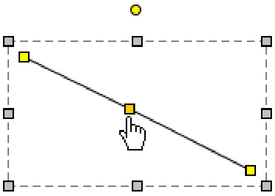


Figure 3-52
Splitting line

2. Fine editing selector activated, drag mid point to split line.

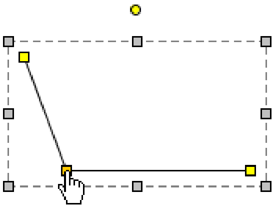


Figure 3-53
Moving mid point

Drawing Labelled Line

1. Click Label Line button from diagram toolbar.

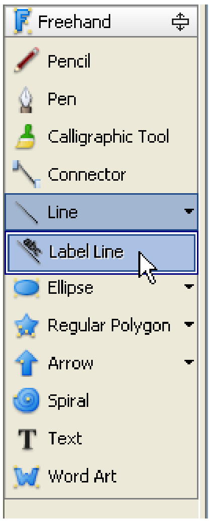


Figure 3-54 *Label line button on diagram toolbar*

2. Press empty space on diagram pane and drag the outline.

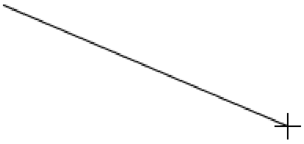


Figure 3-55 *Dragging*

3. Release the mouse, new label line is created.

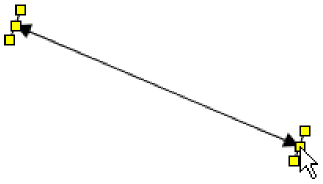


Figure 3-56 *Freehand shape created*

4.Double click to edit the label.

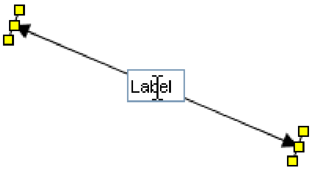


Figure 3-57 *Editing label*

5. Press enter to confirm.

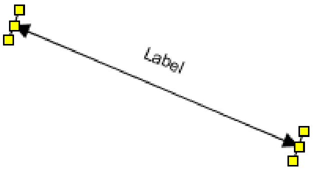


Figure 3-58 *Label Line*

6. You can drag yellow box to modify outline.

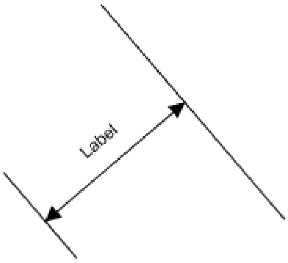


Figure 3-59 *Label Line modified*

Drawing Rectangle

1. Click Rectangle button from diagram toolbar

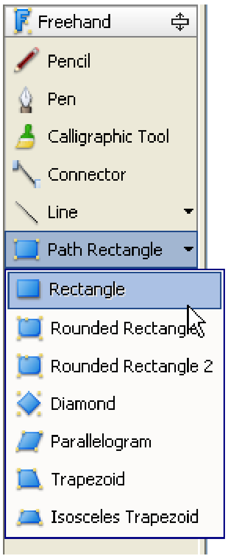


Figure 3-60 *Rectangle button on diagram toolbar*

2. Click empty space on diagram pane , new freehand shape is created

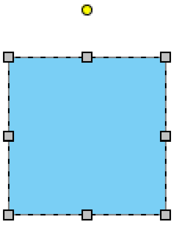


Figure 3-61 *Freehand shape created*

Drawing Path Rectangle

1. Click Path Rectangle button from diagram toolbar.

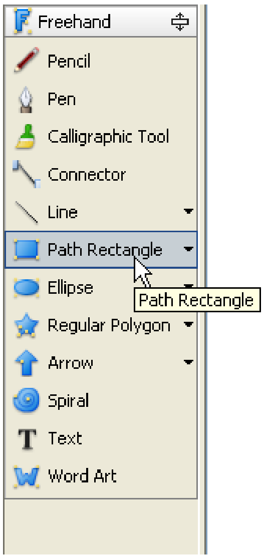


Figure 3-62 *Path Rectangle button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

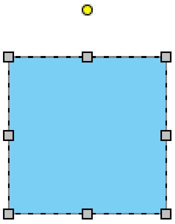


Figure 3-63 Freehand shape created

What's different between Rectangle and Path Rectangle?
Path Rectangle build by path, you can freely modify after created.
Rectangle always keep shape as rectangle.

Drawing Rounded Rectangle

1. Click Rounded Rectangle button from diagram toolbar.

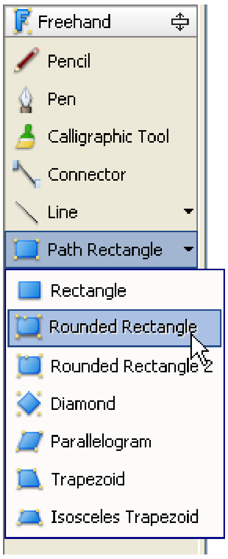


Figure 3-64 Rounded Rectangle button on diagram toolbar

2. Click empty space on diagram pane, new freehand shape is created.

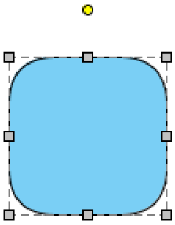


Figure 3-65 Freehand shape created

Drawing Rounded Rectangle 2

1. Click Rounded Rectangle 2 button from diagram toolbar.

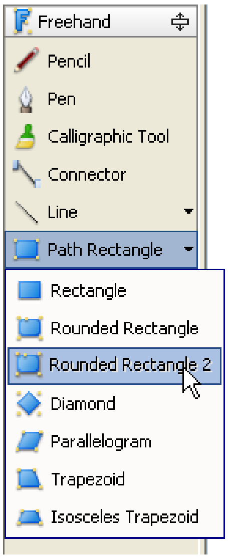


Figure 3-66 *Rounded Rectangle 2 button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

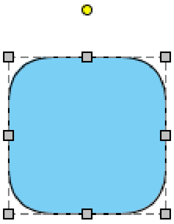


Figure 3-67 *Freehand shape created*

What's different between Rounded Rectangle and Rounded Rectangle 2?
Rounded Rectangle use one point to control the deep of round corner, it must be regular.
Rounded Rectangle 2 use two points to control the deep of round corner, it can be not regular.

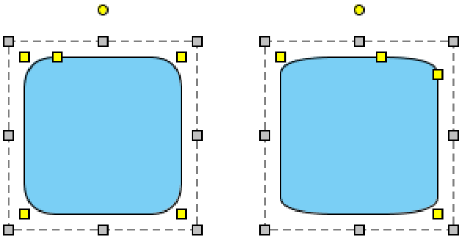


Figure 3-68 *Difference between two rounded rectangle*

Drawing Diamond

1. Click Diamond button from diagram toolbar

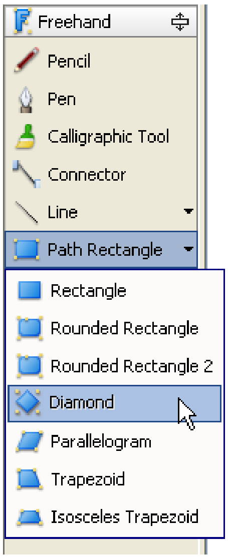


***Figure 3-69** Diamond button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

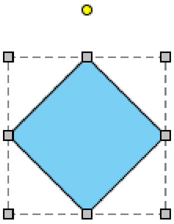


***Figure 3-70** Freehand shape created*

Drawing Parallelogram

1. Click Parallelogram button from diagram toolbar.

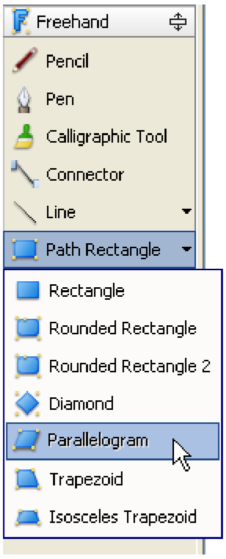


***Figure 3-71** Parallelogram button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

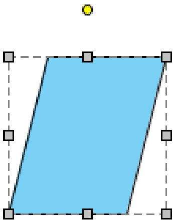


Figure 3-72 Freehand shape created

Drawing Trapezoid

1. Click Trapezoid button from diagram toolbar.

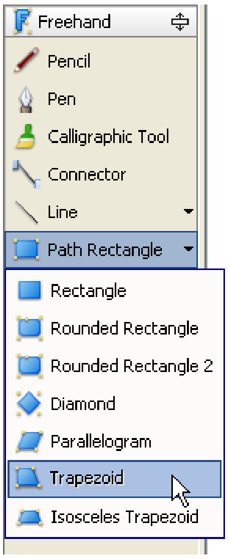


Figure 3-73 Trapezoid button on diagram toolbar

2. Click empty space on diagram pane, new freehand shape is created.

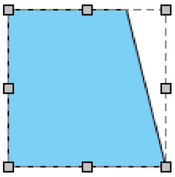


Figure 3-74 Freehand shape created

Activate fine editing selector can modify the slope.

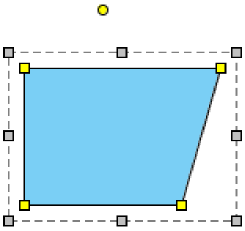


Figure 3-75 Other trapezoid outline

Drawing Isosceles Trapezoid

1. Click Isosceles Trapezoid button from diagram toolbar.

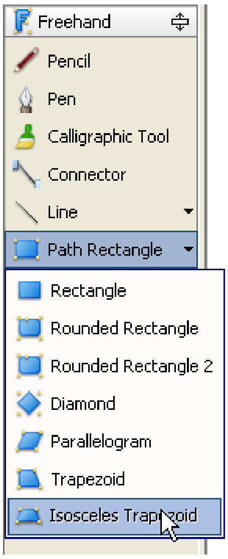


Figure 3-76 *Isosceles Trapezoid button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

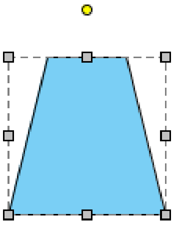


Figure 3-77 *Freehand shape created*

Activate fine editing selector can modify the slope.

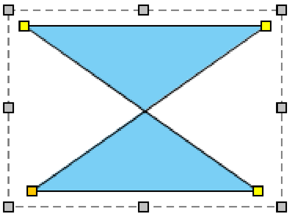


Figure 3-78 *Other isosceles trapezoid outline*

Drawing Elipse

1. Click Elipse button from diagram toolbar.

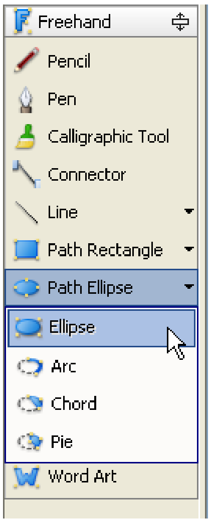


Figure 3-79 *Elipse button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

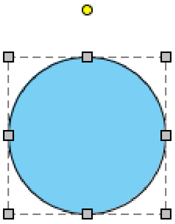


Figure 3-80 *Freehand shape created*

Drawing Path Elipse

1. Click Path Elipse button from diagram toolbar.

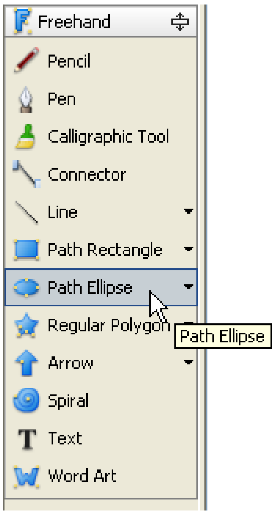


Figure 3-81 *Path Elipse button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

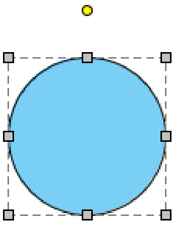


***Figure 3-82** Freehand shape created*

What's different between Elipse and Path Elipse?
Path Elipse build by path, you can freely modify after created.
Elipse always keep shape as oval.

Drawing Arc

1. Click Arc button from diagram toolbar.

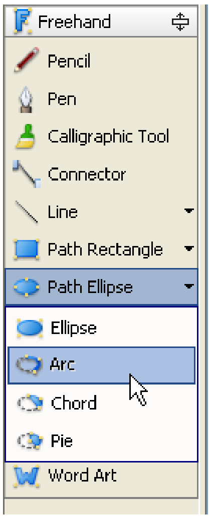


***Figure 3-83** Arc button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

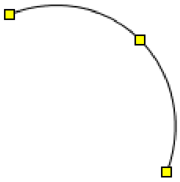


***Figure 3-84** Freehand shape created*

Activate fine editing selector can modify the length.

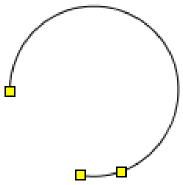


***Figure 3-85** Other arc outline*

Drawing Chord

1. Click Chord button from diagram toolbar.

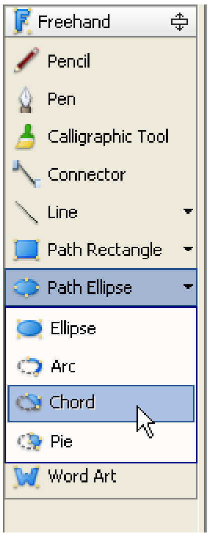


Figure 3-86 *Chord button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

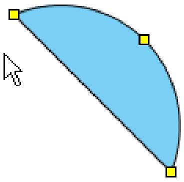


Figure 3-87 *Freehand shape created*

Activate fine editing selector can modify the length.

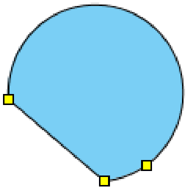


Figure 3-88 *Other Chord outline*

Drawing Pie

1. Click Pie button from diagram toolbar

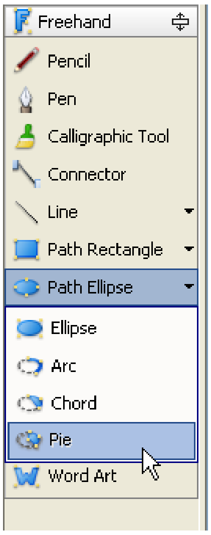


Figure 3-89 *Pie button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created

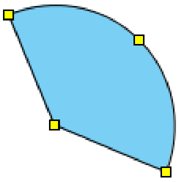


Figure 3-90 *freehand shape created*

Activate fine editing selector can modify the length.

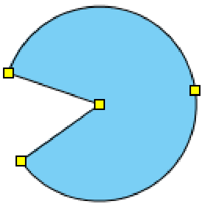


Figure 3-91 *Other pie outline*

Drawing Hexagon

1. Click Hexagon button from diagram toolbar.

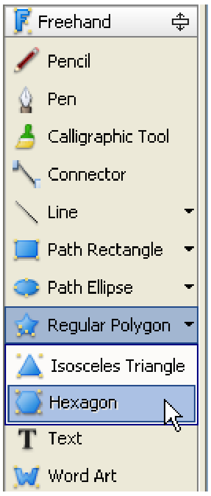


Figure 3-92 Hexagon button on diagram toolbar

2. Press empty space on diagram pane, new freehand shape is created.

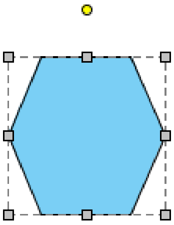


Figure 3-93 Freehand shape created

Drawing Regular Polygon

1. Click Hexagon button from diagram toolbar.

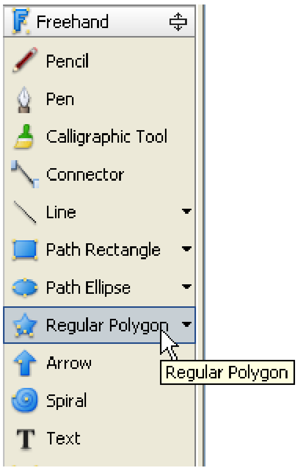


Figure 3-94 Regular Polygon button on diagram toolbar

2. Click empty space on diagram pane, new freehand shape is created.

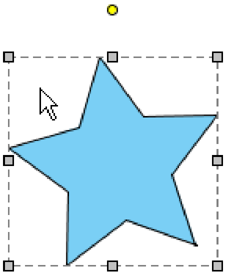


Figure 3-95 *Freehand shape created*

Activate fine editing selector can modify the outline.

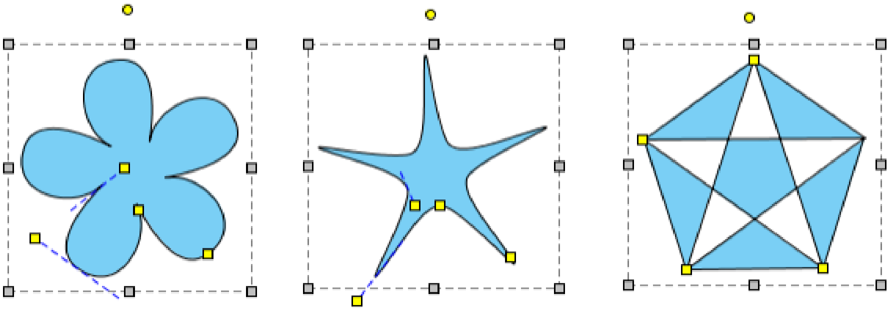


Figure 3-96 *Other Regular Polygon outline*

Drawing Isosceles Traingle

1. Click Isosceles Traingle button from diagram toolbar.

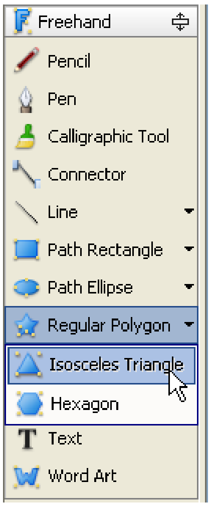


Figure 3-97 *Isosceles Triangle button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

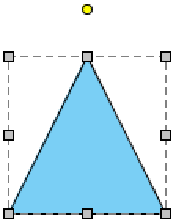


Figure 3-98 *Freehand shape created*

Drawing Single Head Arrow

1. Click Arrow button from diagram toolbar.

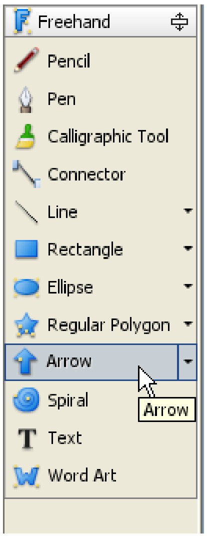


Figure 3-99 *Arrow button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

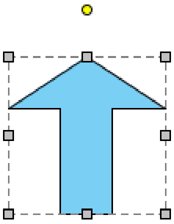


Figure 3-100 *Freehand shape created*

Activate fine editing selector can modify the outline.

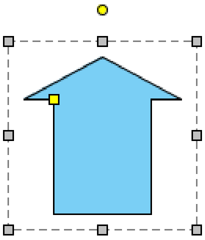


Figure 3-101 *Other Arrow Outline*

Drawing Two Head Arrow

1. Click Two Head Arrow button from diagram toolbar.

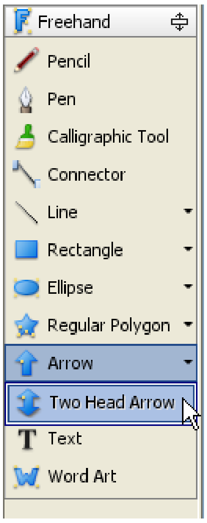


Figure 3-102 Two Head Arrow button on diagram toolbar

2. Click empty space on diagram pane, new freehand shape is created.

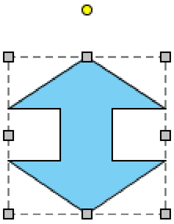


Figure 3-103 Freehand shape created

Activate fine editing selector can modify the outline.

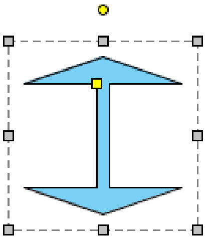


Figure 3-104 Other two head arrow outline

Drawing Spiral

1. Click Spiral button from diagram toolbar.

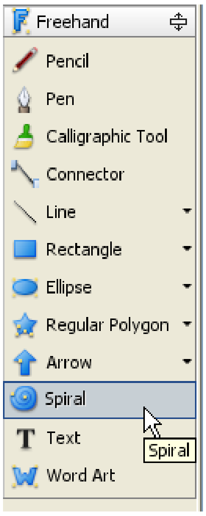


Figure 3-105 *Spiral button on diagram toolbar*

2. Click empty space on diagram pane, new freehand shape is created.

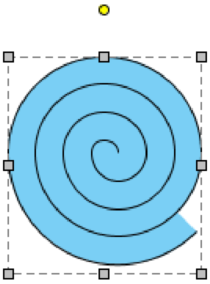


Figure 3-106 *Freehand shape created*

Activate fine editing selector can modify the outline.

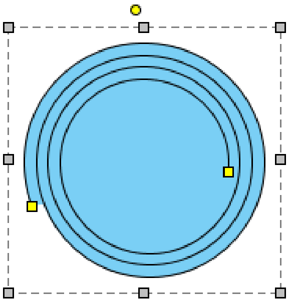


Figure 3-107 *Other Spiral outline*

Inserting Text

1. Click Text button from diagram toolbar.

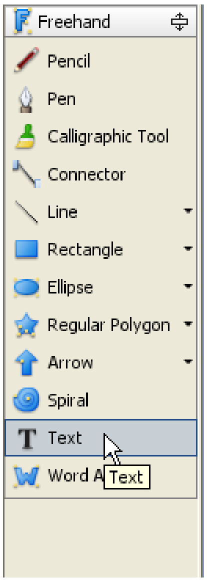
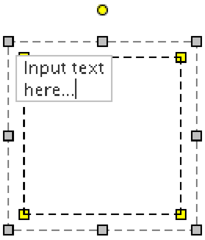


Figure 3-108 *Text button on diagram toolbar*

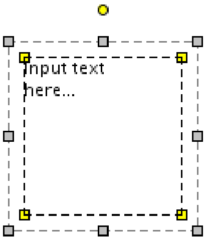
2. Click empty space on diagram pane, input text on textfield, press enter to line break.



Input text
here...

Figure 3-109
Inputing text

3. Press ctrl enter to confirm.



Input text
here...

Figure 3-110 *Freehand shape created*

Inserting Word Art

1. Click Text button from diagram toolbar.

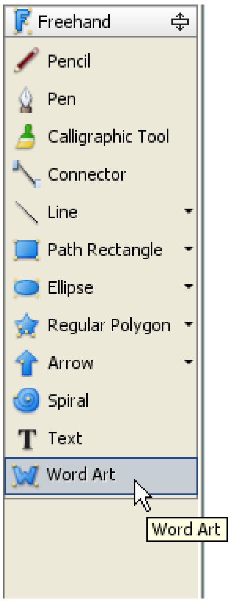


Figure 3-111 *Word Art button on diagram toolbar*

2. Press empty space on diagram pane, input text on textfield, press enter to line break.

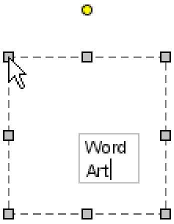


Figure 3-112 *Inputing text*

3. Press ctrl enter to confirm.

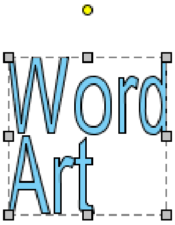


Figure 3-113 *Freehand shape created*

Activate fine editing selector can modify the outline.

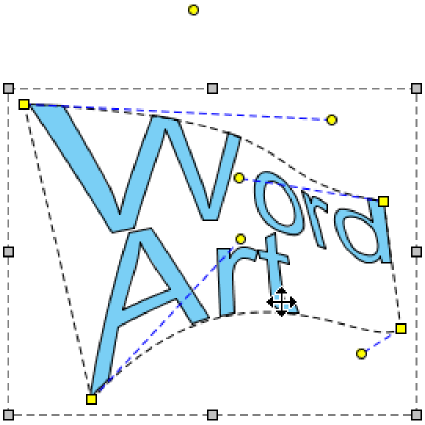


Figure 3-114
Editing Word Art

Here is the result.

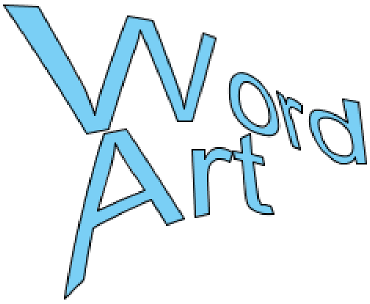


Figure 3-115
Word Art example

Using Sweeper and Magnet to organize diagram

If you have a compact diagram, and wanna to move a set of shapes for increasing/decreasing the distance of 2 set of shapes. You may use Sweeper and Magnet.

Make more space to do modeling with Swapper

Swapper is used to make a set of shapes moving following mouse drag. If mouse starts dragging to right, the shapes on right of mouse will be moved to right. If mouse starts dragging down, the shapes on bottom of mouse will be moved down, vice versa. Swapper seems push the shapes.

1.To use swapper, select **Sweeper** on platelet, and dragging on diagram.

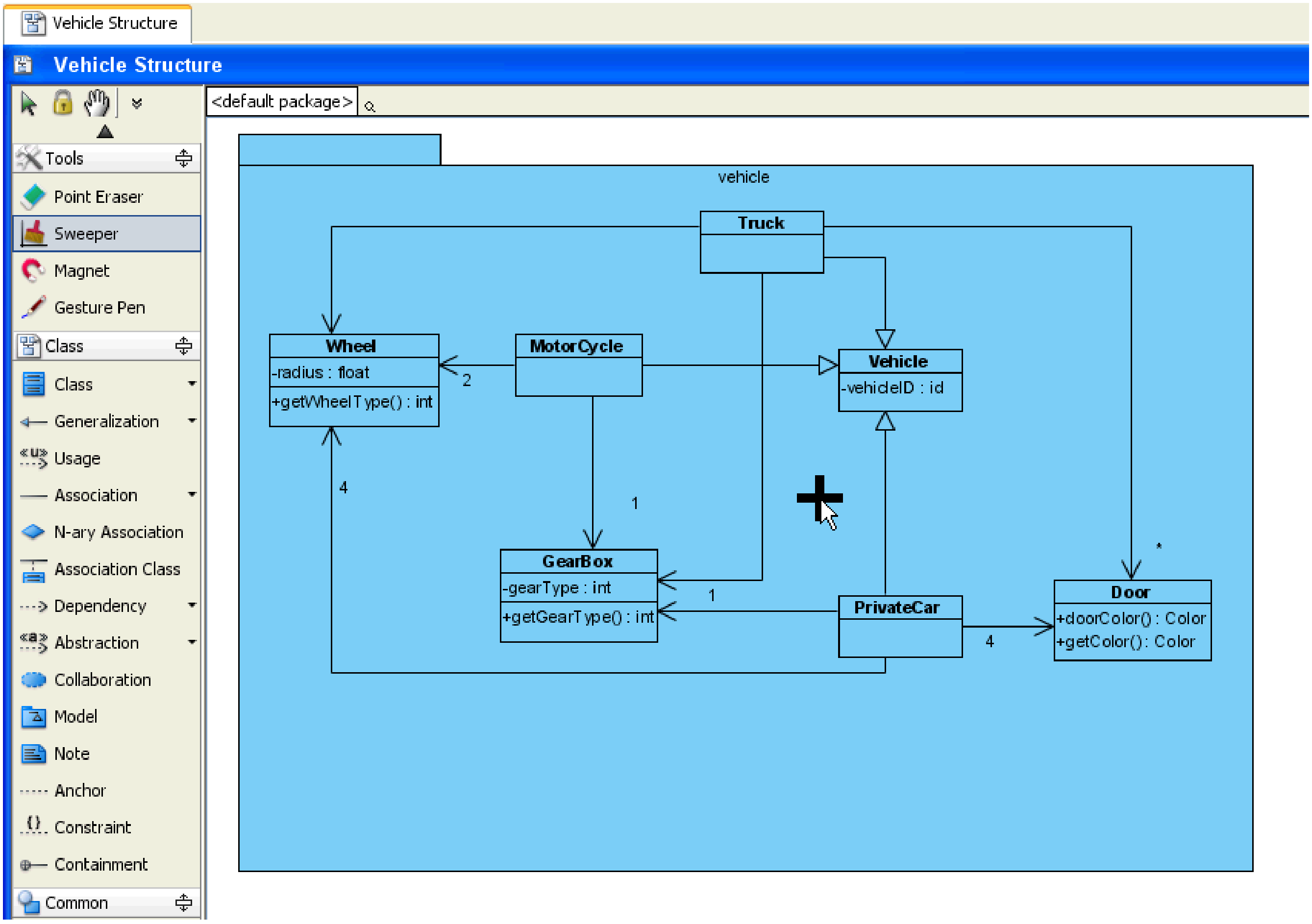


Figure 4-1 Dragging down on diagram

2.The shapes on the bottom are moved down.

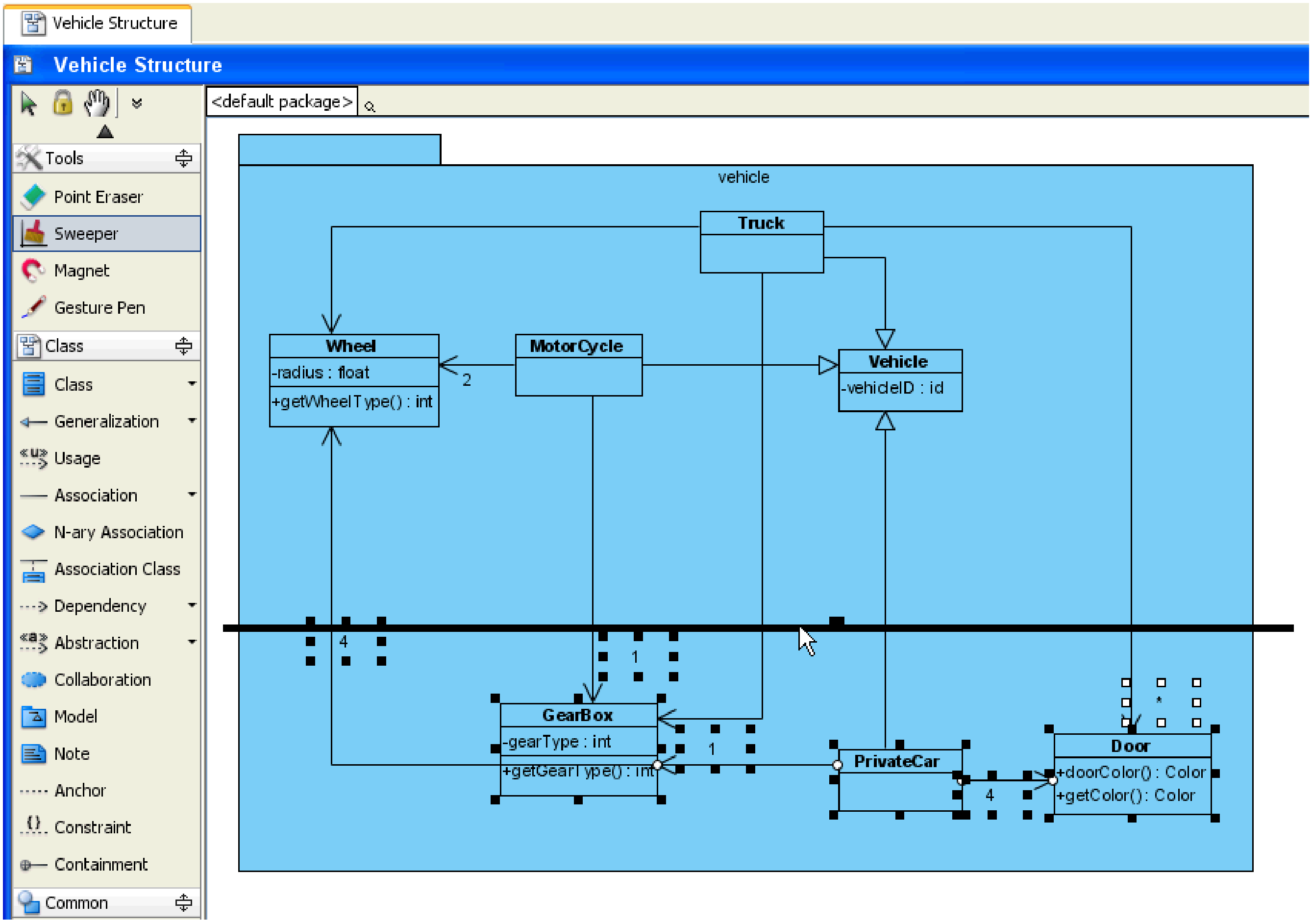


__Figure 4-2__ Shapes are pushed down

Compact Diagram with Magnet

Magnet is used to make a set of shapes moving following mouse drag. If mouse starts dragging to right, the shapes on left of mouse will be moved to right. If mouse starts dragging down, the shapes on top of mouse will be moved down, vice versa. Magnet seems pull the shapes.

1.To use magnet, select **Magnet** on platelet, and dragging on diagram.

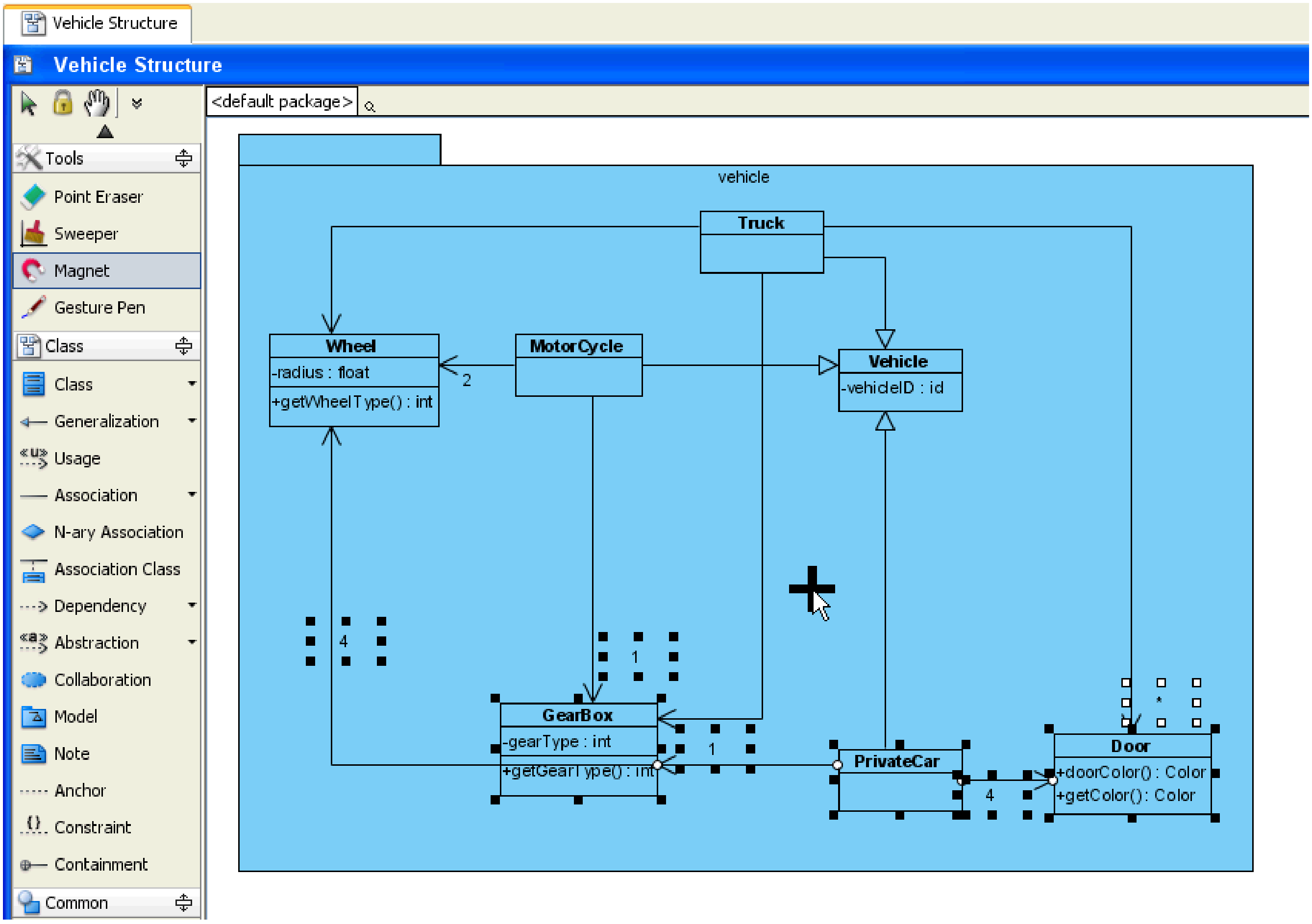


Figure 4-3 *Dragging up on diagram*

2.The shapes on bottom are moved up.

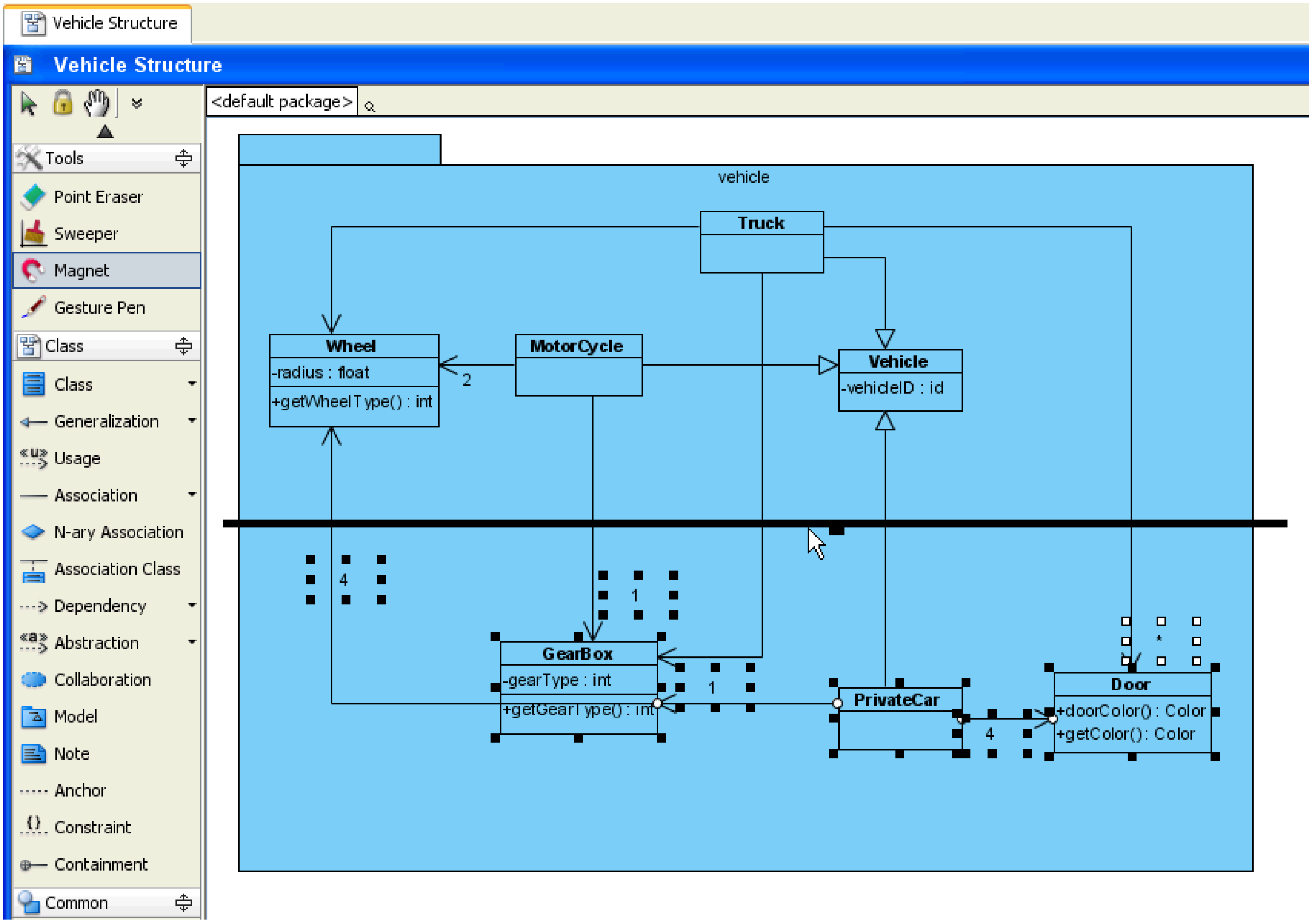


Figure 4-4 Shapes are pulled up

Drawing Diagram with Mouse Gestures

Mouse gestures allow you to execute common commands and create UML models within the diagrams.

Drawing Class with Mouse Gesture

You can create models on diagram by mouse gesture.

1.To create class on diagram, use Mouse Gesture to draw a clockwise rectangle.

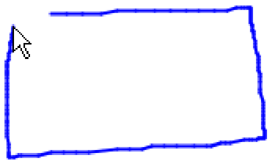


Figure 4-5 *Drawing clockwise rectangle*

2.The class is created after mouse released. The action name **Create Class** will be shown on right-top of diagram.

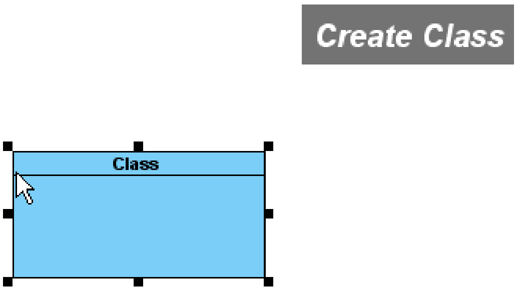


Figure 4-6 *Class Created*

3.You also can create attribute and operation on the class. To create attribute, draw a line from right to left on the class, an attribute is created.

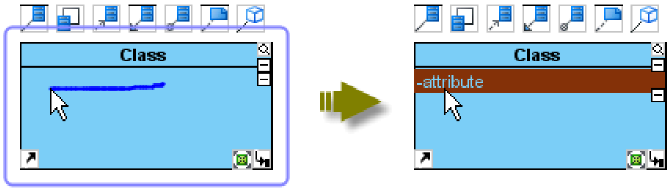


Figure 4-7 *Attribute is created*

4.If drawing the line outside the class, an attribute with <<property>> stereotype will be created.

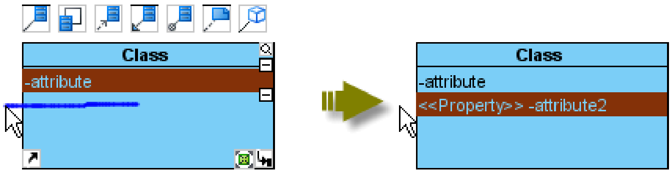


Figure 4-8 *<<Property>> is created*

5.To create operation, draw a line from left to right on the class, an operation with protected visibility is created.

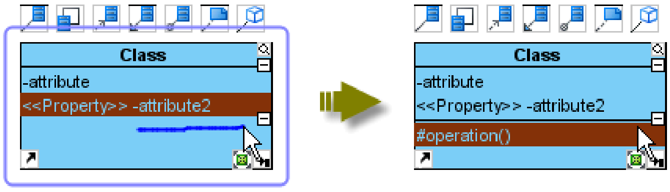


Figure 4-9 *Protected operation is created*

6.If drawing the line outside the class, a public operation will be created.

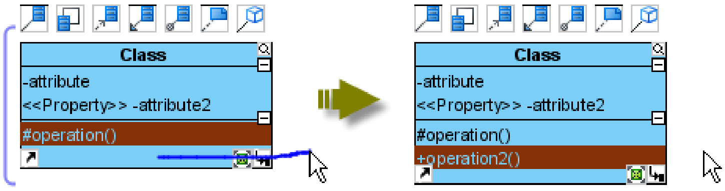


Figure 4-10 Public operation is created

Connecting Classes with Mouse Gesture

You can connect models on diagram by mouse gesture.
1.To do so, drawing from a model to another model.

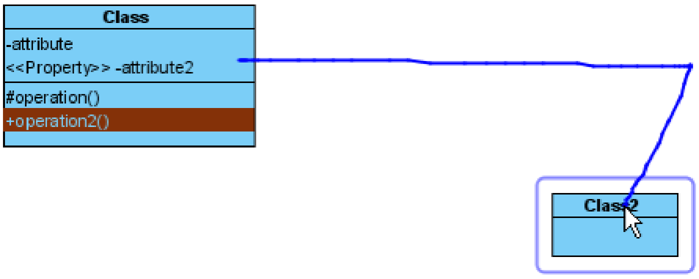


Figure 4-11 Drawing from a model to another model

2.After mouse released, a association is created between the classes.

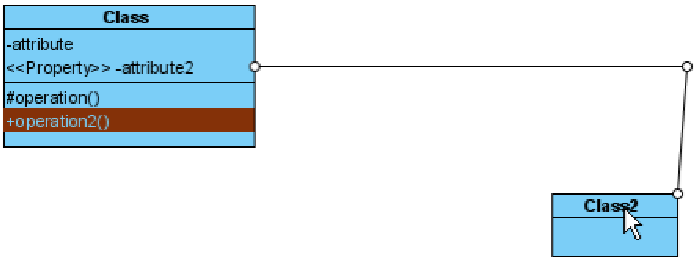


Figure 4-12 Association created

3.You can also create a new shape with connector. To do so, drawing from a model to empty area of the diagram.

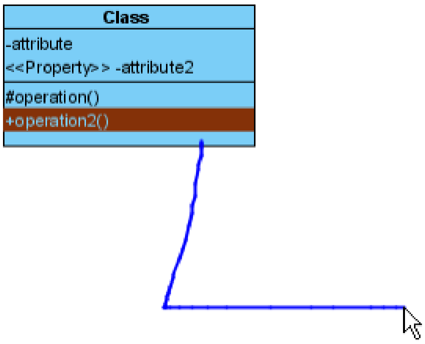


Figure 4-13 Drawing to empty area

4.After mouse released, a popup menu is shown for selecting what kind of connector and model you wanna to create. For example, create a generalization with a class.

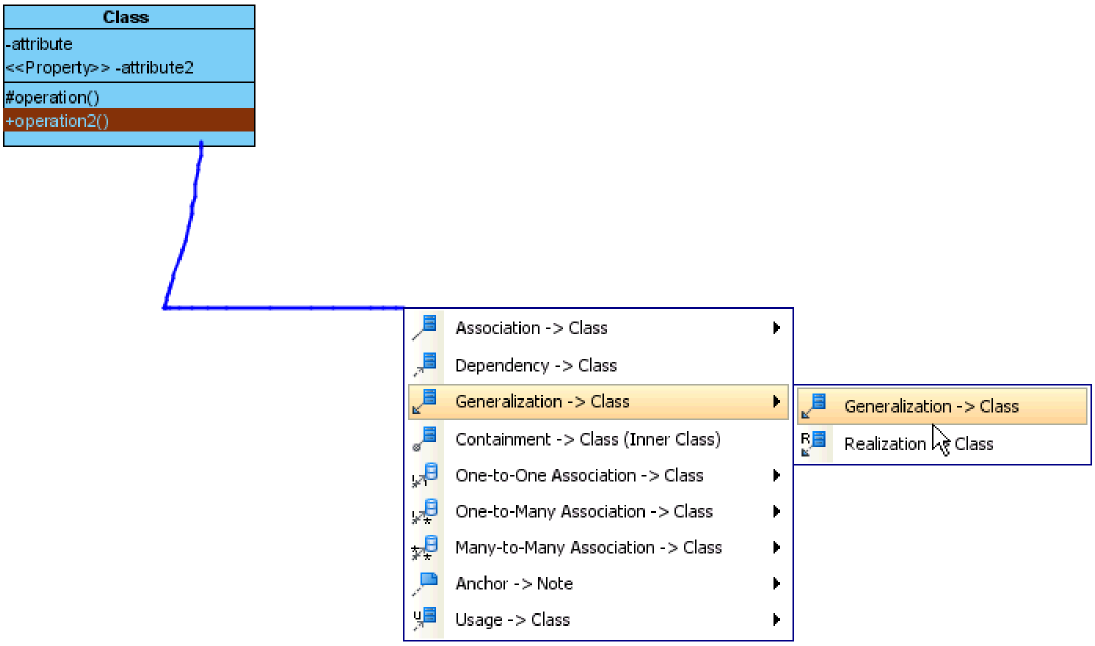


Figure 4-14 *Create generalization with class*

5.The class with generalization is created.

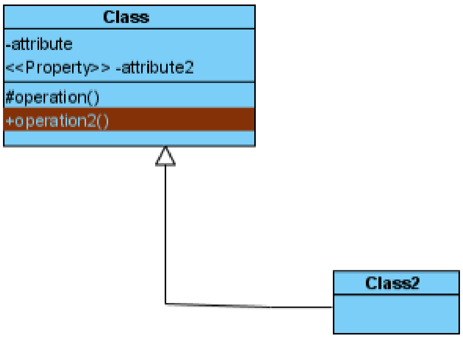
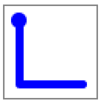
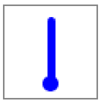


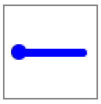
Figure 4-15 *Class with generalization created*

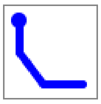
Showing All Supported Mouse Gestures

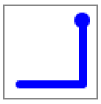
General

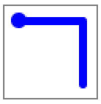

Layout diagram


Open diagram specification

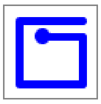

Connect new shape

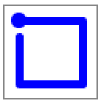

Connect existing shape

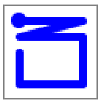

Close Diagram

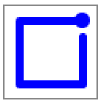

Thumbnail view

Activity Diagram

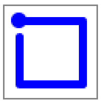

Action


Activity

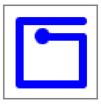

Decision Node


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.)

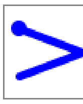
Activity Diagram (UML 1.x)



Action State

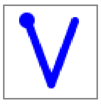


Sub-Activity

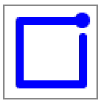


Swimlane

Horizontal Synchro Bar

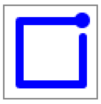


Vertical Synchronization Bar

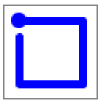


Initial State/Final State (If there is no Initial State, an Initial State will be created. Else if there is no Final State, a Final State will be created.)

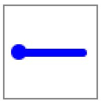
Business Process Diagram



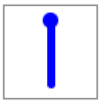
Sub-Process



Pool/Task

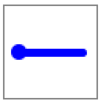


Horizontal Lane

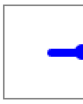


Vertical Lane

Class Diagram

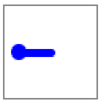


Sync. to ERD



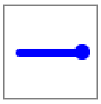
Package

Add attribute (Add an attribute to class. If mouse released outside the class, getter and setter property will be set to true.)

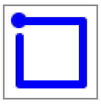


Add operation (Add an operation to class. If mouse released inside the class, visibility will be protected, otherwise it will be public)

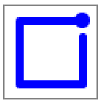
Communication Diagram



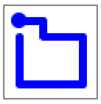
Sync. To Sequence Diagram



Lifeline

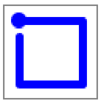


Actor

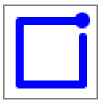


Package

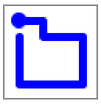
Component Diagram



Component

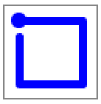


Instance Specification

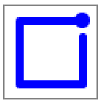


Package

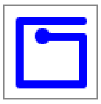
Composite Structure Diagram



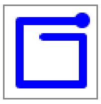
Class



Interface

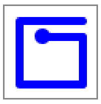


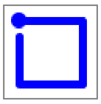
Collaboration

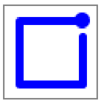


Collaboration Use

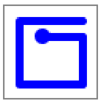
Data Flow Diagram

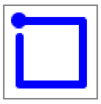

Process

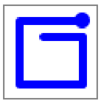

External Entity

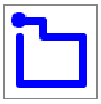

Data Store

Deployment Diagram

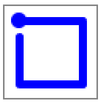

Node

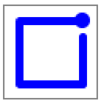

Component

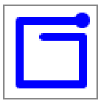

Instance Specification

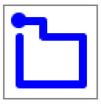

Package

EJB Diagram

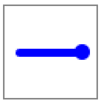

Entity Bean

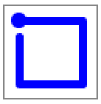

Message-Driven Bean

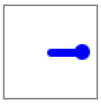

Session Bean


Package

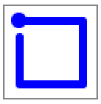
Entity Relationship Diagram

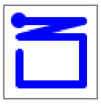

Sync. to Class Diagram

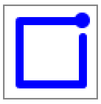

Entity


Add column

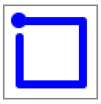
Interaction Overview Diagram


Interaction

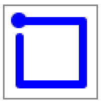

Decision Node

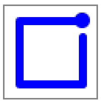

Initial Node/Final Node (If there is no Initial Node, an Initial Node will be created. Else if there is no Final Node, a Final Node will be created.)

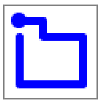
Mind Mapping Diagram


Node

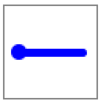
Object Diagram

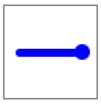

Instance Specification

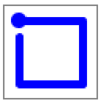

Class

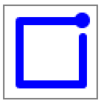

Package

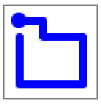
ORM Diagram


Sync. Classes -> Entities


Sync. Entities -> Classes


Class


Entity


Package

Overview Diagram

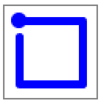
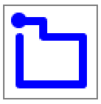
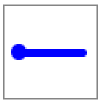


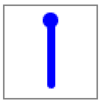
Diagram Overview

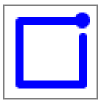
Package Diagram

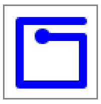

Package

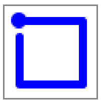
Sequence Diagram


Sync. to Communication Diagram

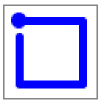

Lifeline

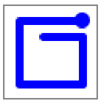

Actor

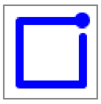

Alt


Loop

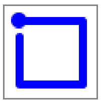
State Machine Diagram

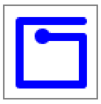

State

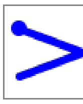

Submachine State

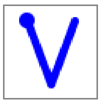

Initial Node/Final Node (If there is no Initial State, an Initial State will be created. Else if there is no Final State, a Final State will be created.)

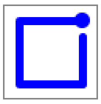
State Machine Diagram (UML 1.x)


State

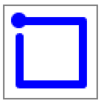

Concurrent State


Submac State
Horizont Synchro Bar

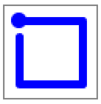

Vertical Synchronization Bar

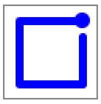

Initial State/Final State (If there is no Initial State, an Initial State will be created. Else if there is no Final State, a Final State will be created.)

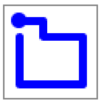
Timing Diagram


Frame

Use Case Diagram


Use Case


Actor


Package

Easily locate Diagram Element with "Jump To"

In order to let you locate the desired shape/model easier and faster, the jump to shape/model facility is introduced. You can select either jump to a shape in the active diagram, or jump to any shape/model in the current project, or jump to diagram in current project.

Jump to Element within Active Diagram

If there is active diagram opened, you can jump to the shape in the active diagram.

1.You can select **Edit > Jump to Element in Active Diagram...** in main menu or pressing **Ctrl+J** to show **Jump to** dialog.

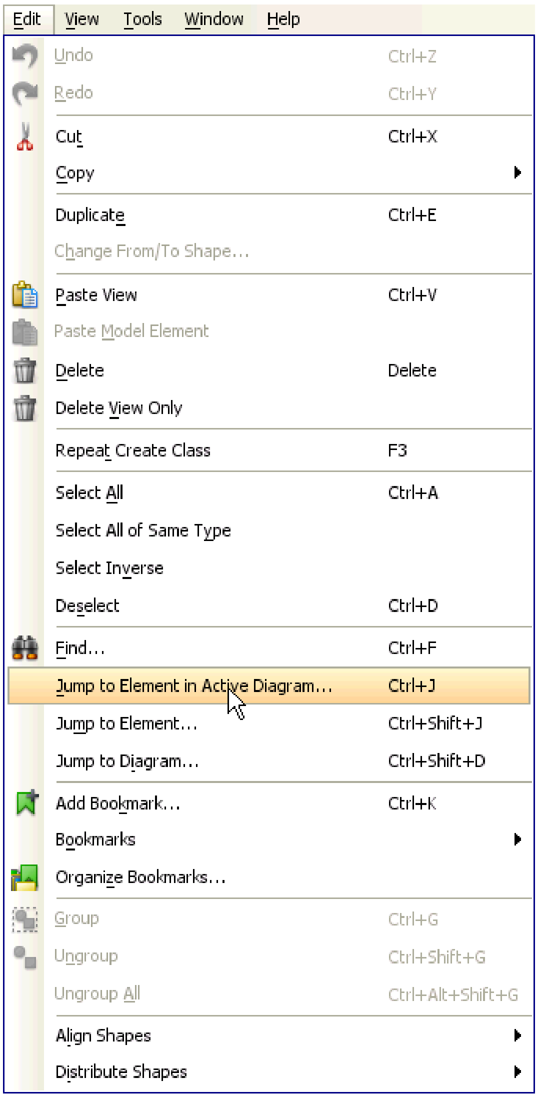
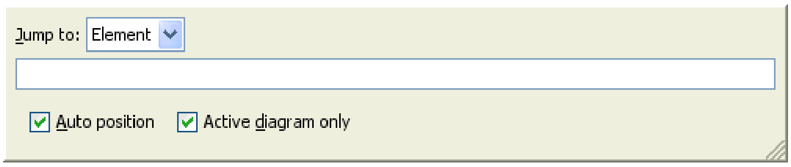


Figure 4-16 *Select Jump to Element in Active Diagram in main menu*

2. **Jump to** dialog is shown.



Jump to: Element

☑ Auto position ☑ Active diagram only

Figure 4-17 *Jump to dialog is shown*

3.Entering *Motor* on the text field, it will show a list of model which name starts with *Motor.*

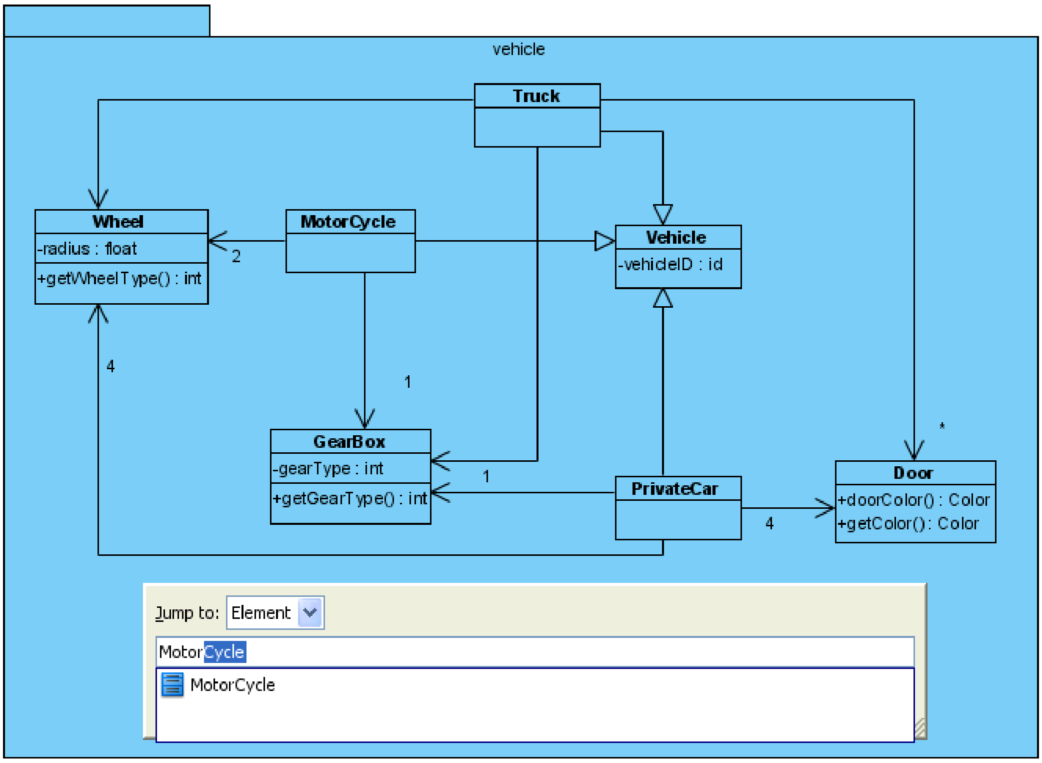


Figure 4-18 *A list of possible models are shown*

4.Pressing **Down** key to select the *MotorCycle,* the *MotorCycle* will be spot-lighted on the active diagram.

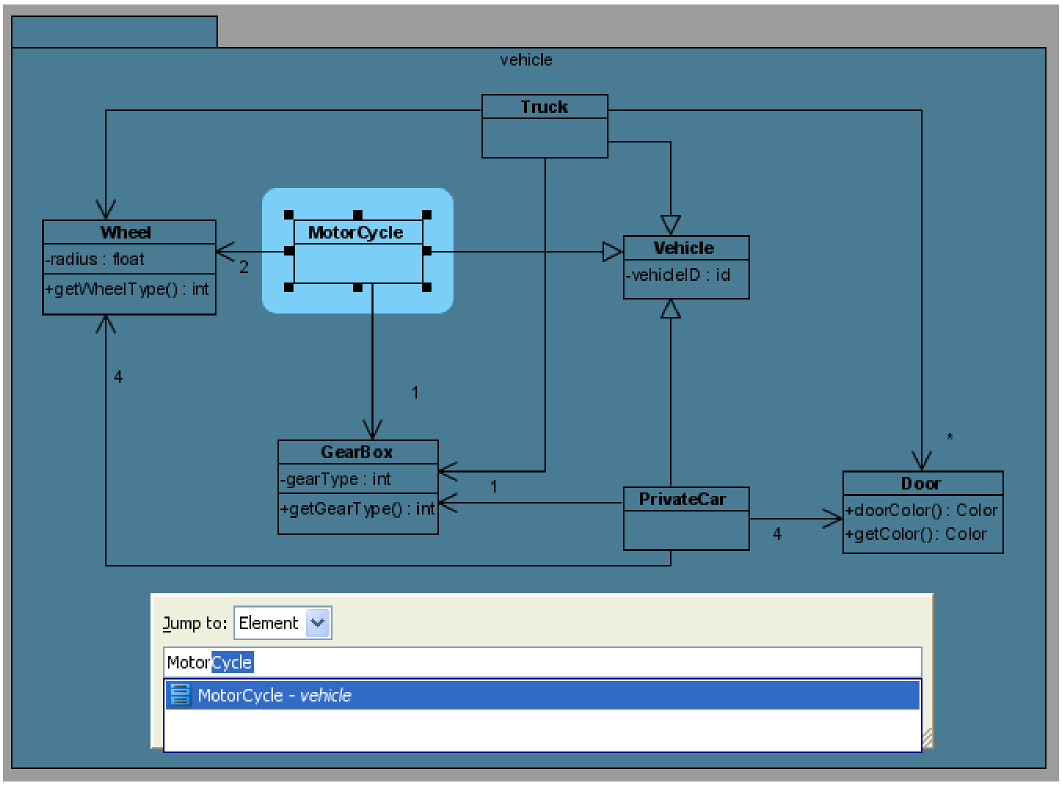


Figure 4-19 *Model is spot-lighted*

5.Pressing **Enter** to confirm jump to the model, the jump to dialog will be invisible and the model is selected on diagram.

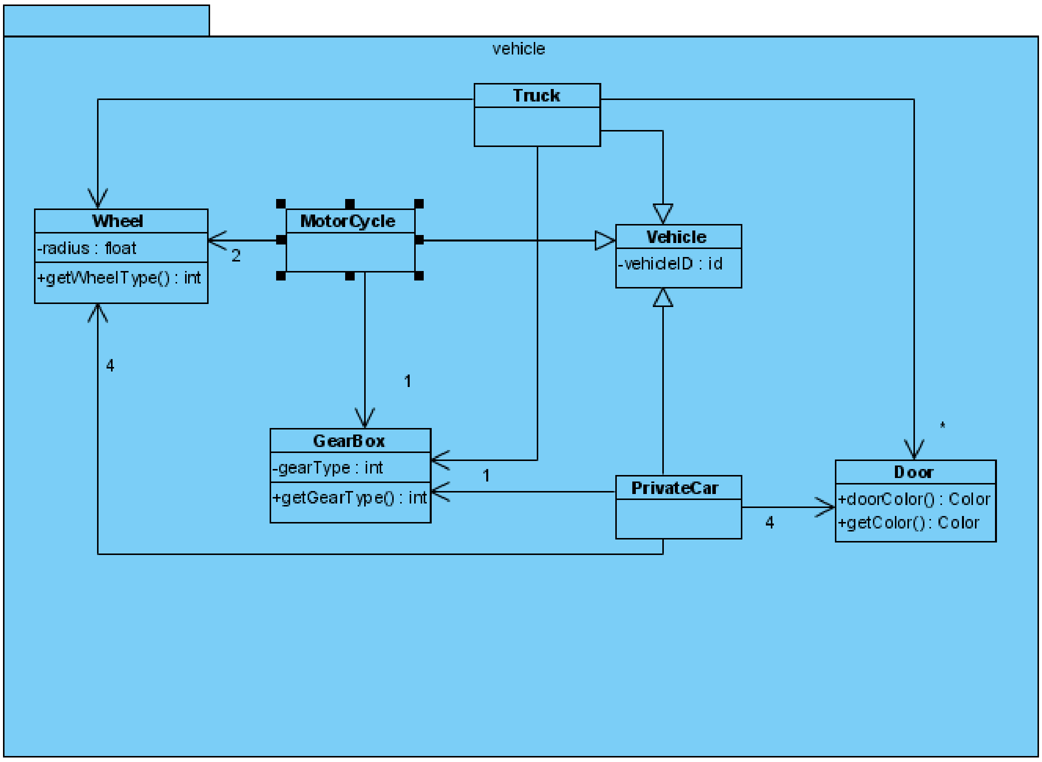
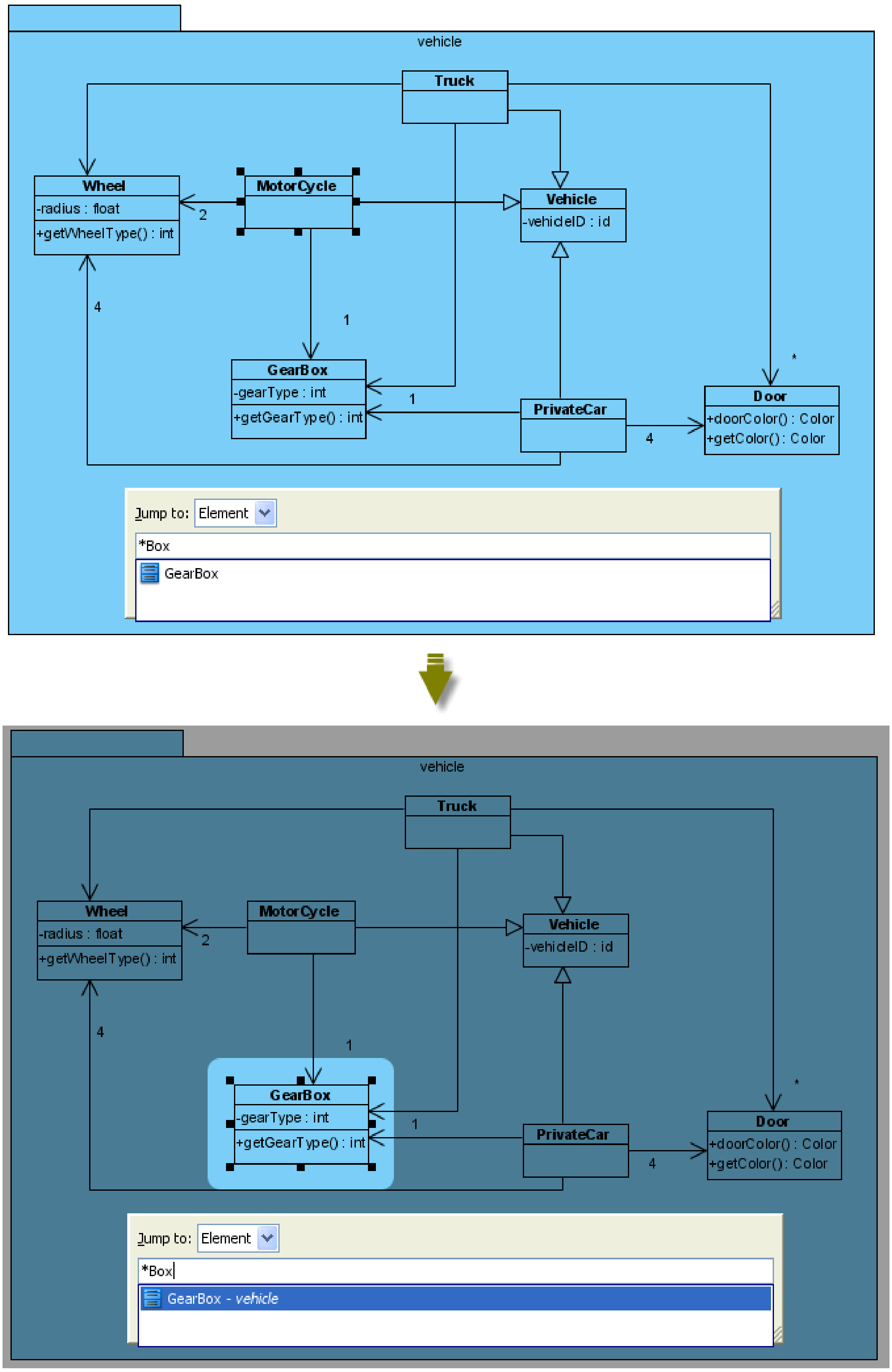


Figure 4-20 *Model is selected after confirm jump to*

6.You may typing * in the text field as any character. For example, typing *Box*, any models' name ends with * will be shown.



Jump to: Element

*Box

GearBox

Jump to: Element

*Box

GearBox - *vehicle*

Figure 4-21 *Entering name with **

Jump to Element within the project

Besides jump to element in active diagram, you can jump to any element within the project.

1.You may select **Edit > Jump to Element...** in main menu or pressing **Ctrl+Shift+J** to show **Jump to** dialog.

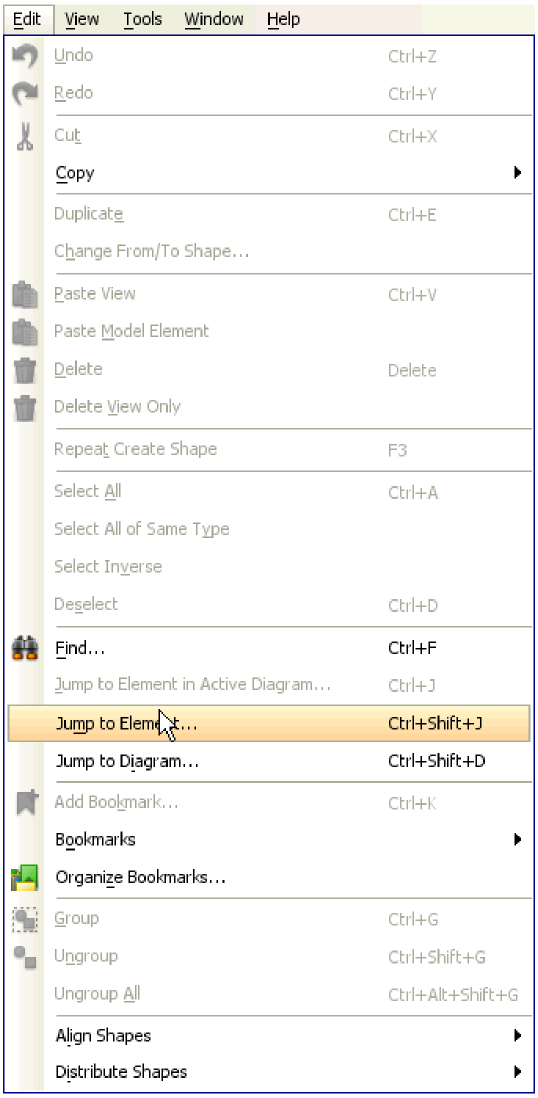
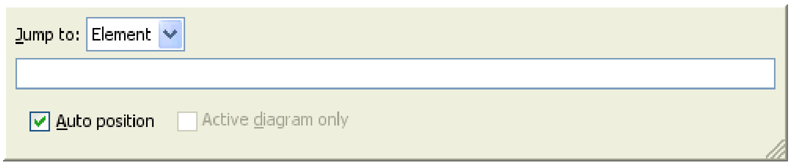


Figure 4-22 *Select Jump to Element... in main menu*

2. **Jump to** dialog is shown. The **Active diagram only** checkbox is unchecked means will search all elements with in project. The checkbox is disabled because no diagram is opened.

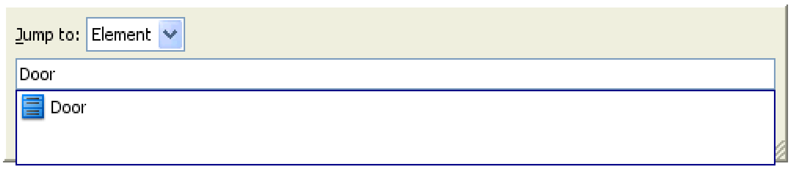


Jump to: Element

☑ Auto position □ Active diagram only

Figure 4-23 *Jump to dialog will search all elements within project*

3.Entering *Door* will show a list of elements names starts with *Door.*



Jump to: Element

Door

Door

Figure 4-24 *A list of possible models are shown*

4.Pressing **Down** to select the model, the diagram name will be shown on on list.

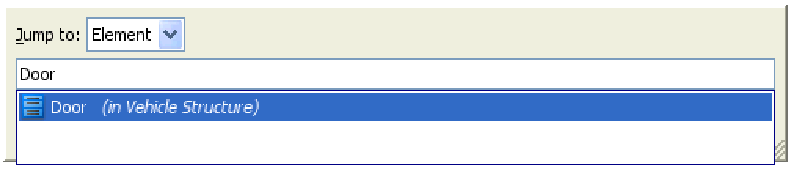


Figure 4-25 Diagram name is shown

5.Pressing **Enter** to open the diagram and select the model on the diagram.

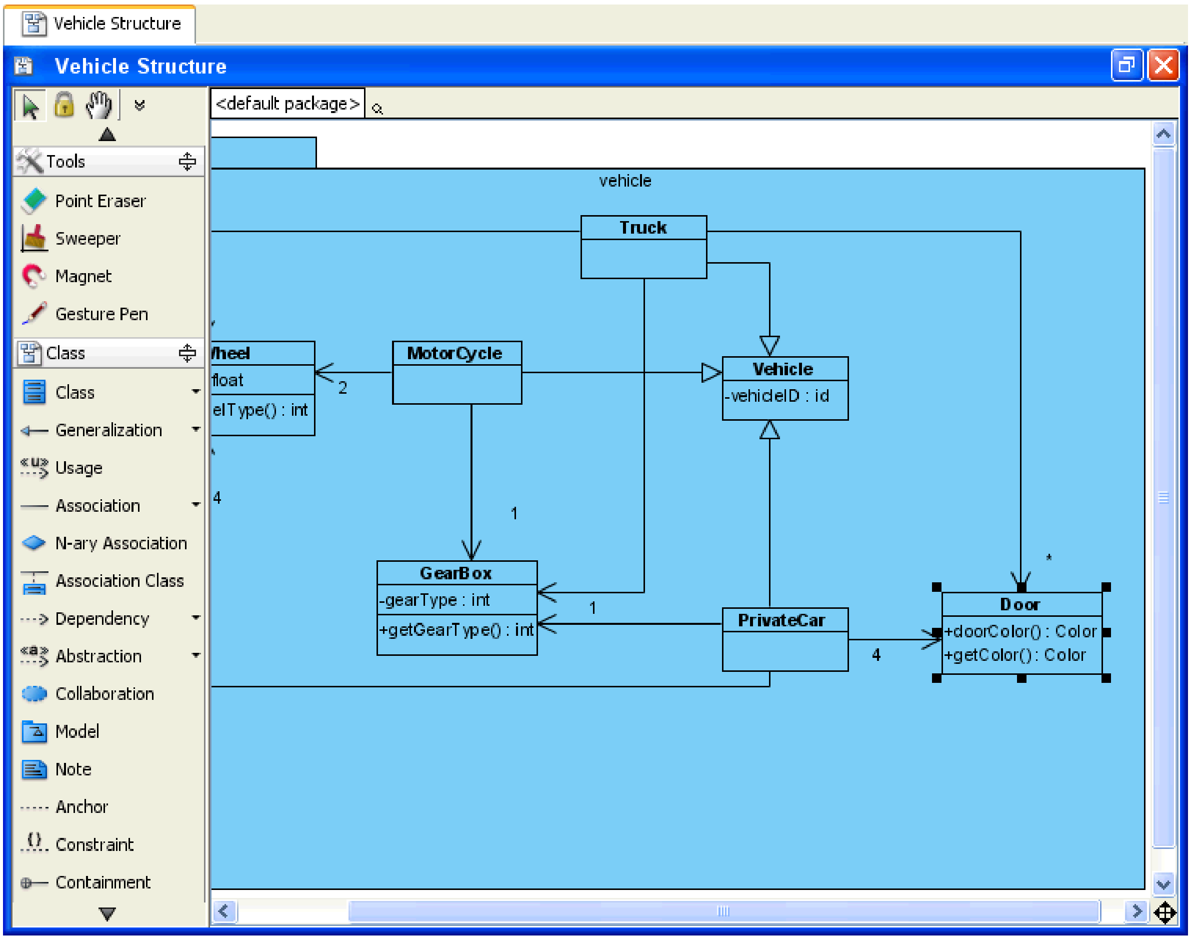


Figure 4-26 Diagram is opened with selecting the model

Jump to Diagram

You also can jump to diagram.

1.To do so, select **Edit > Jump to Diagram...** in main menu or pressing **Ctrl+Shift+D** to show **Jump to** dialog.

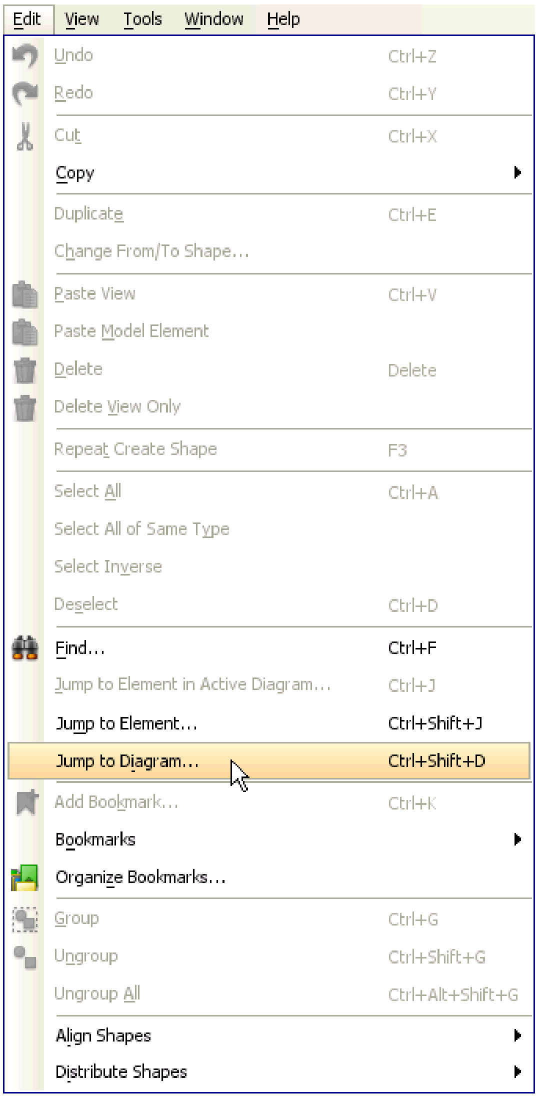
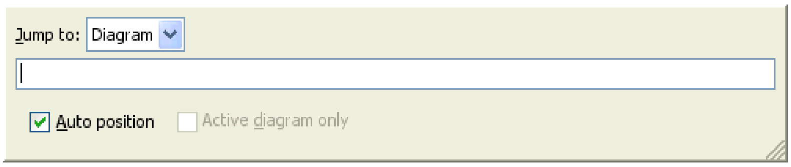


Figure 4-27 *Select Jump to Diagram... in main menu*

2. **Jump to** dialog is shown with selected **Diagram** in combo box. It means **Jump to** dialog will search all diagrams within the project.

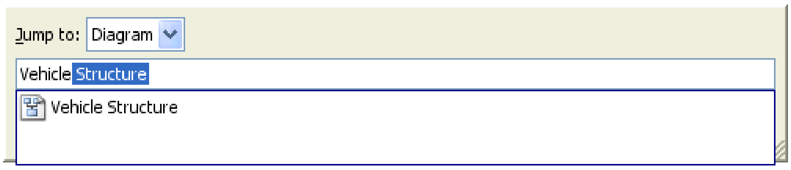


Jump to: Diagram

☑ Auto position □ Active diagram only

Figure 4-28 *Jump to dialog will search all diagrams within the project*

3.Entering *Vehicle* will show a list of diagrams names starts with *Vehicle.* Select the diagram and pressing **Enter** will open the diagram.



Jump to: Diagram

Vehicle Structure

Vehicle Structure

Figure 4-29 *Select the diagram on list*

Easy navigation to connected elements

1.If you have a compact diagram that the from/to models of connector can't be shown on screen. It is difficult to know the from/to models. VP-UML supports **Scroll to** function to scroll to from/to model of a connector. For example, you may want to know which model is the from model that is connected with *Wheel.*

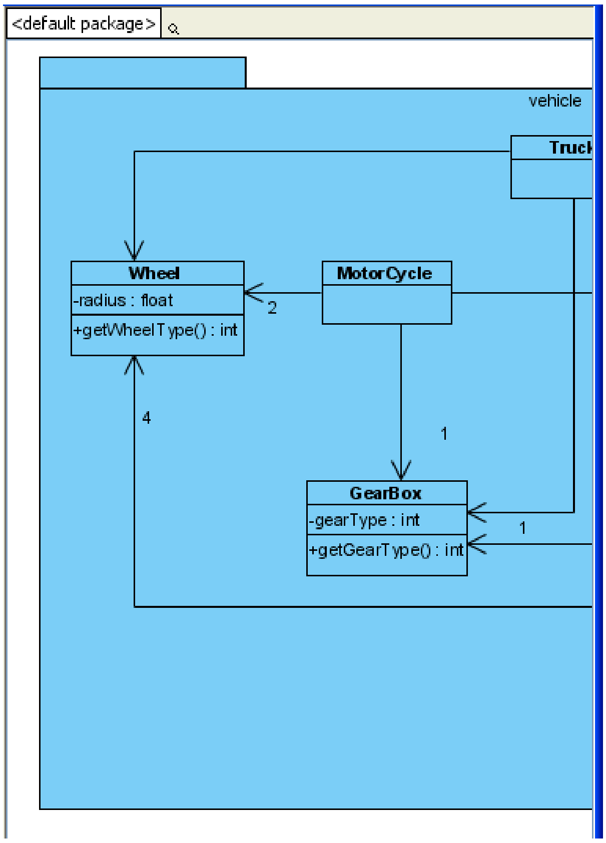


Figure 4-30 *The from model is not shown on screen because the screen is not large enough*

2.Right-click on the connector and select **Scroll to > Source (PrivateCar)** from popup menu. (The *(PrivateCar)* is the name of the from model.)

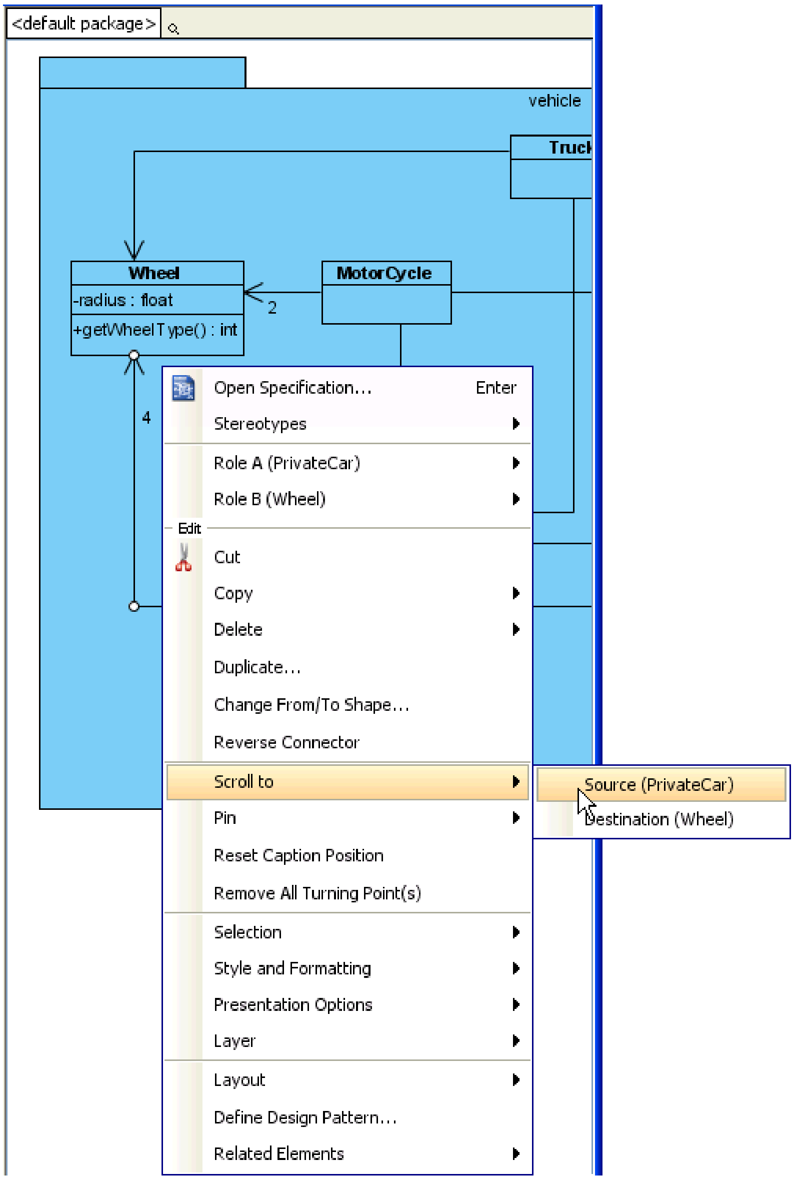


Figure 4-31 *Scroll to from model*

3.The from model is selected on diagram.

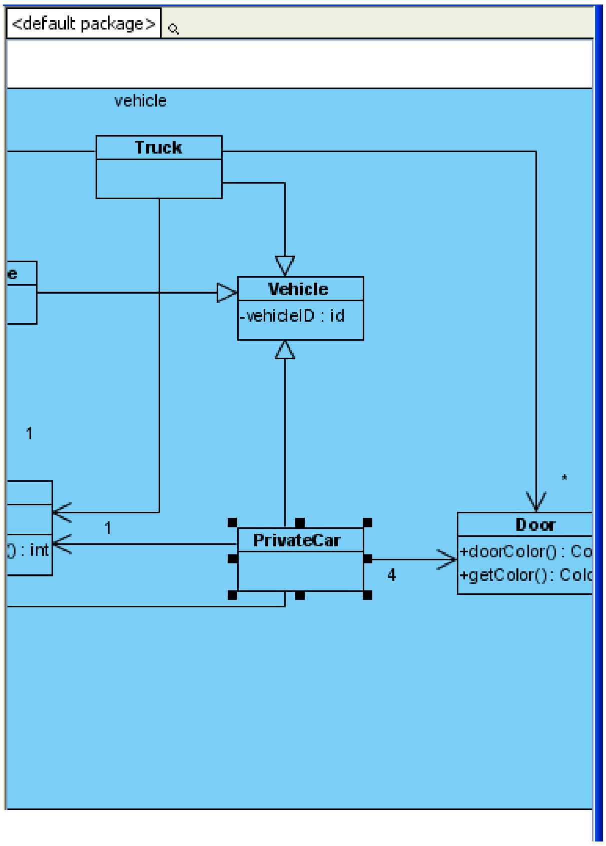


__Figure 4-32__ From model is selected

Grouping Diagram Elements

After you have applied alignment to a group of shapes, you may want to keep its configuration and not want them to get messed up. Grouping feature is designed for this usage. After shapes are grouped, you can treat the grouped shapes as a single shape.

Group Diagram Elements

1.Select more than one shape.

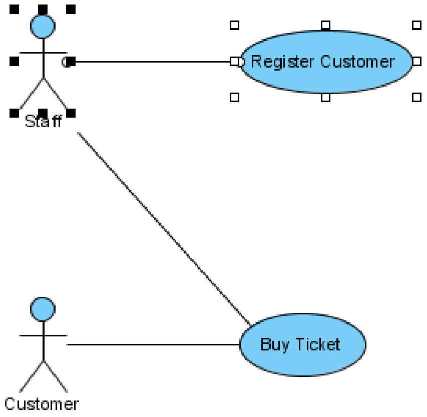


Figure 4-33 *Select several shapes*

2.You may select **Edit > Group** in main menu

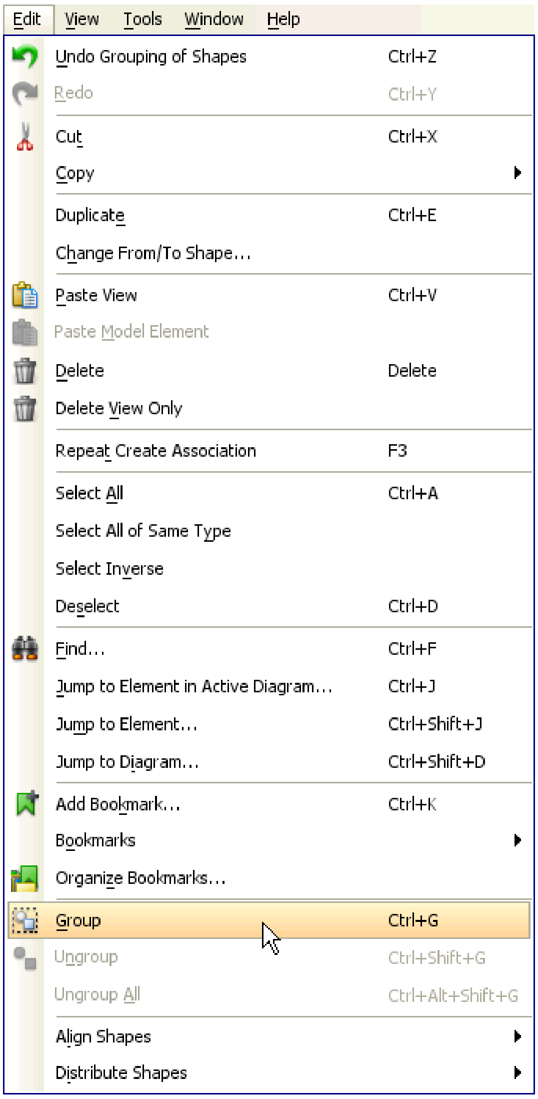


Figure 4-34 *Group by main menu*

3.Or using Group Resources to Group the selected shapes. After grouped, the Group Resources will be changed.

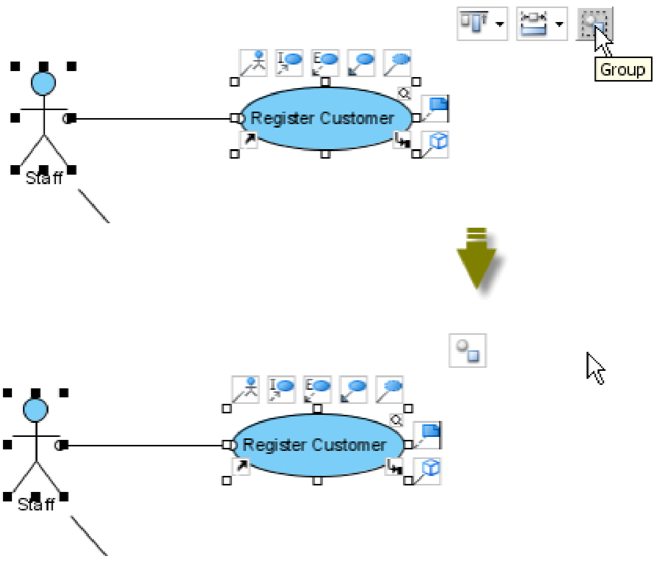


Figure 4-35 *Group by group resources*

4.After shapes are grouped, moving 1 shape, another shapes inside the group will be also moved.

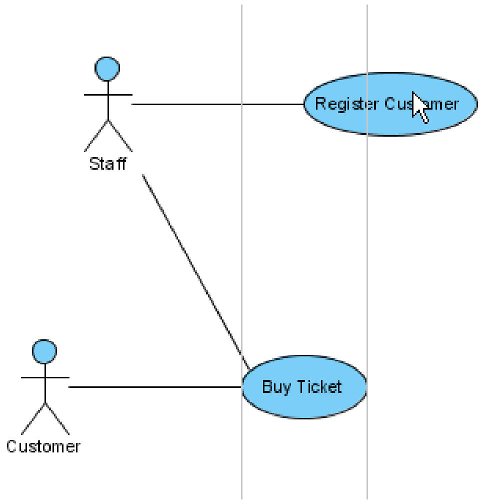


Figure 4-36 *All shapes in group are moved together*

Ungroup Diagram Elements

1.To ungroup the shapes, select the grouped shape, you may select **Edit > Ungroup** in main menu.

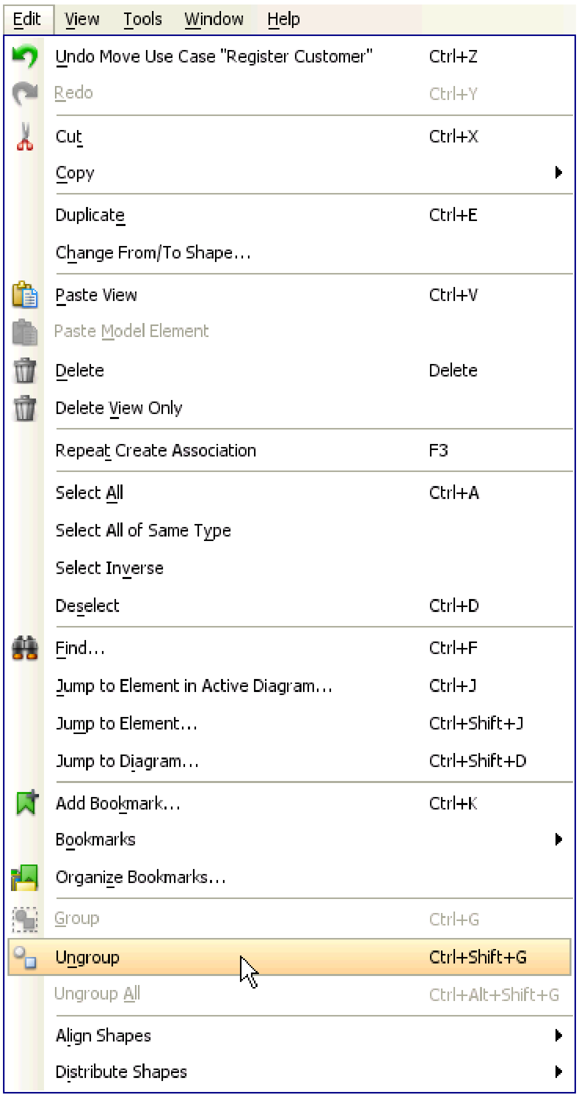


Figure 4-37 Ungroup in main menu

2.You may also use Group Resources to ungroup the shapes.

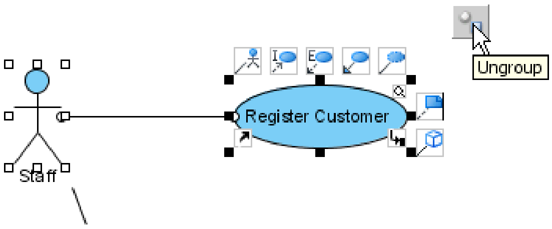


Figure 4-38 Ungroup by group resources

Show/hide Diagram Elements

If you have a compact diagram, you may want to hide/show some of the shapes on the diagram.

Hide Selected Diagram Element(s)

1.Right-click on the selected shape(s), select from popup menu

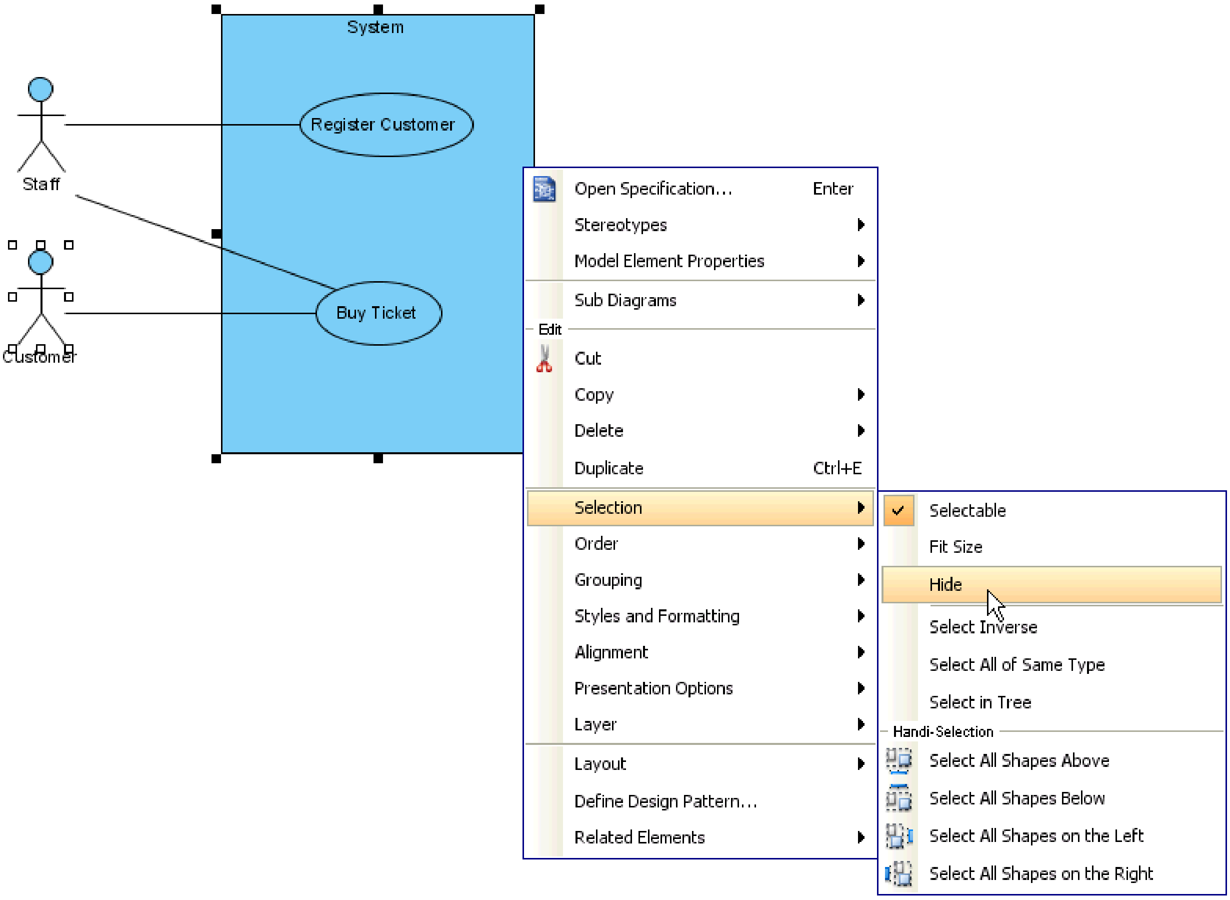


Figure 4-39 *Hide selected shapes from popup menu*

2.Besides the selected shapes are hide, some of the related shapes (e.g. children and relationship) will be hide too.

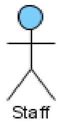


Figure 4-40 *The selected shapes and their related shapes are hide*

Show/Hide Diagram Elements by Shape Type

You can hide all the shapes with same type from the diagram. To do so, right-click on diagram, select **Diagram Content > Show/Hide > Hide by Shape Type > Actor** from popup menu. All the Actor on this diagram will be hide.

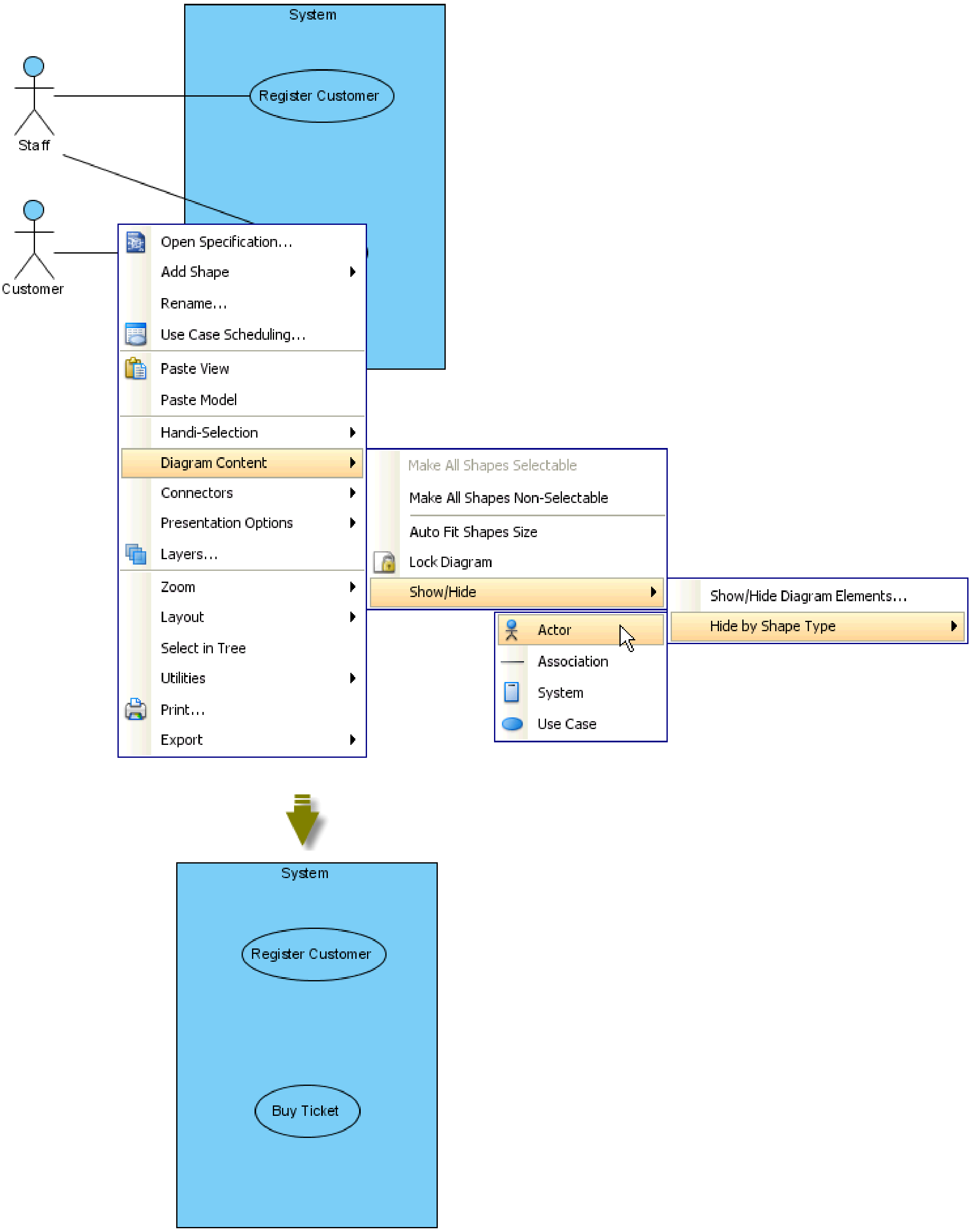


Figure 4-41 *Hide all shapes by shape type*

Show all Diagram Elements

You can show all hided shapes from the diagram. Select **Diagram Content > Show/Hide > Show all Diagram Elements**.

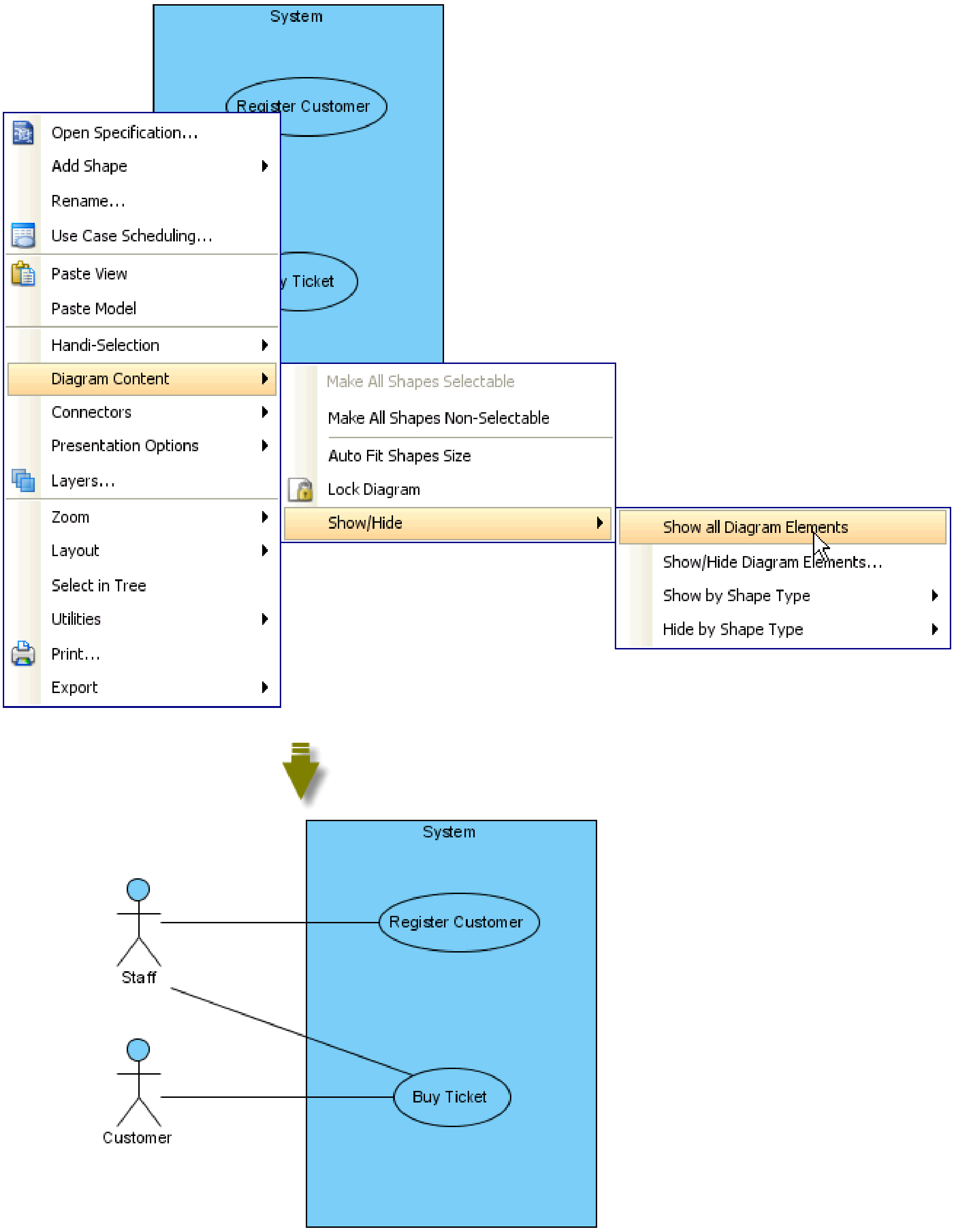


Figure 4-42 *Show all Diagram Elements*

Manage Show/Hide of specific Diagram Element(s)

You also can show/hide the shapes by **Show Hide Diagram Elements** dialog box.

1.Right-click on diagram, select **Diagram Content > Show/Hide > Show/Hide Diagram Elements...** from popup menu.

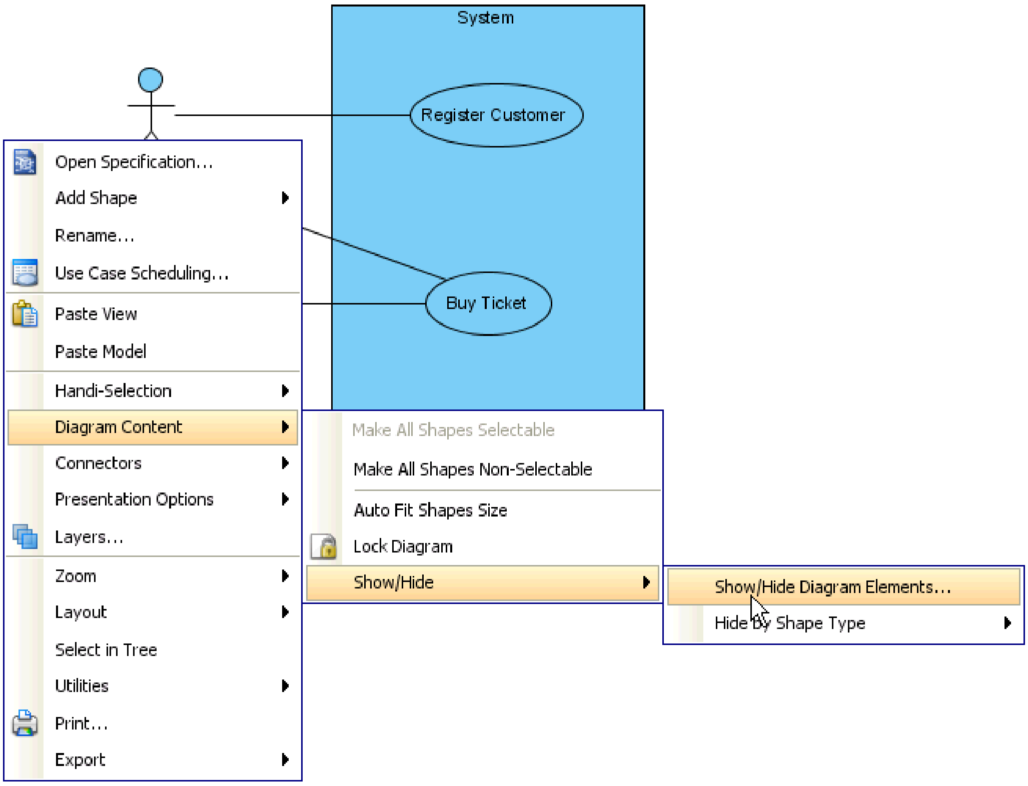


Figure 4-43 *Show/Hide Diagram Elements...*

2. **Show Hide Diagram Elements** dialog box is shown, you may select which shape(s) will be hide, or will be shown. For example, select Register Customer to be hide.

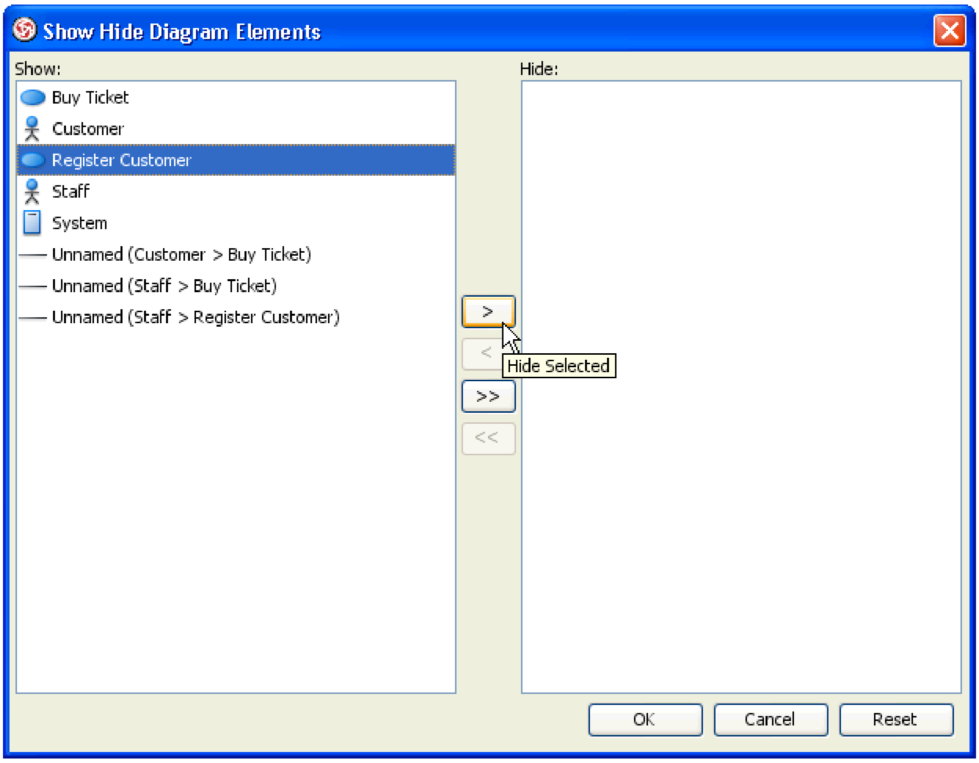

Show Hide Diagram Elements
Show:
Buy Ticket
Customer
Register Customer
Staff
System
Unnamed (Customer > Buy Ticket)
Unnamed (Staff > Buy Ticket)
Unnamed (Staff > Register Customer)
Hide:
>
<
>>
<<
Hide Selected
OK
Cancel
Reset

Figure 4-44 *Select shapes to be show/hide*

3.Selected one shape from **Show** list, the selected shape's related shapes will be also added to **Hide** list. Then select **OK** button to confirm.

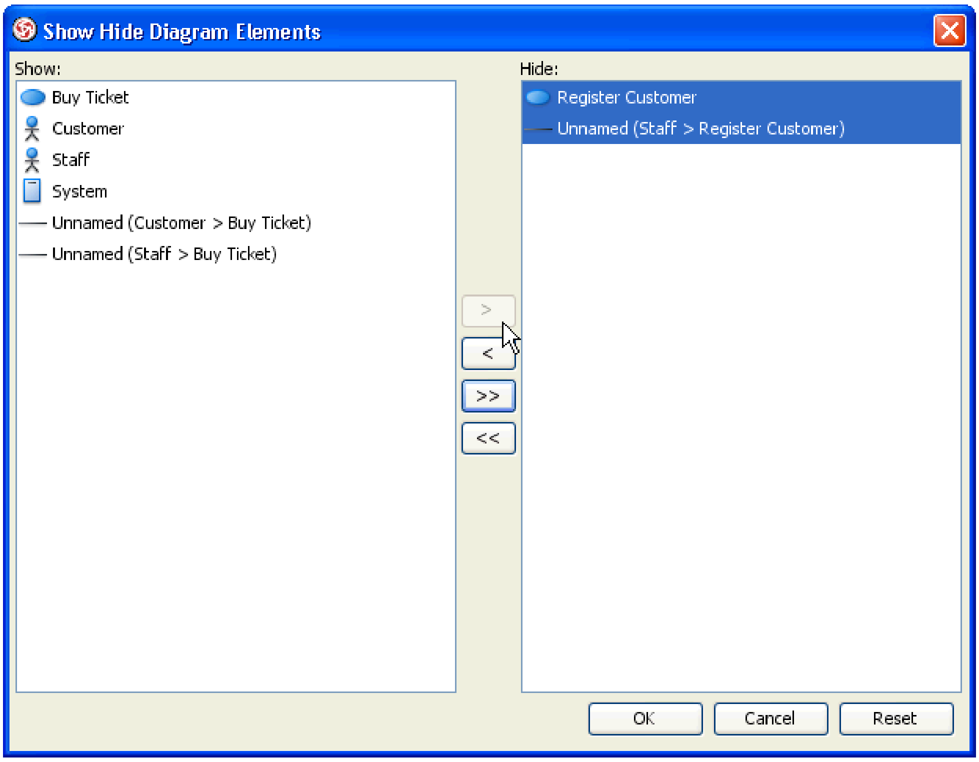

Show Hide Diagram Elements
Show:
Buy Ticket
Customer
Staff
System
Unnamed (Customer > Buy Ticket)
Unnamed (Staff > Buy Ticket)
Hide:
Register Customer
Unnamed (Staff > Register Customer)
>
<
>>
<<
OK
Cancel
Reset

Figure 4-45 Shapes will be hide after select OK button

Another dimension of Diagramming - Layer

Diagram supports multi-layers to contain different shapes.

Creating Layer

1.Right-click on diagram, select **Layers...** from popup menu.

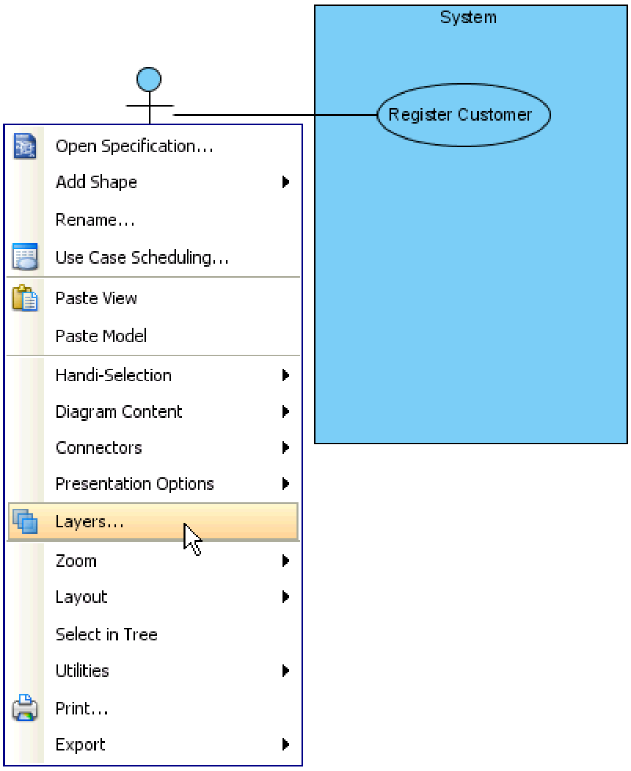


Figure 4-46 Layers

2. **Diagram Layers** dialog box is shown, click **Create new layer** button to create layer.

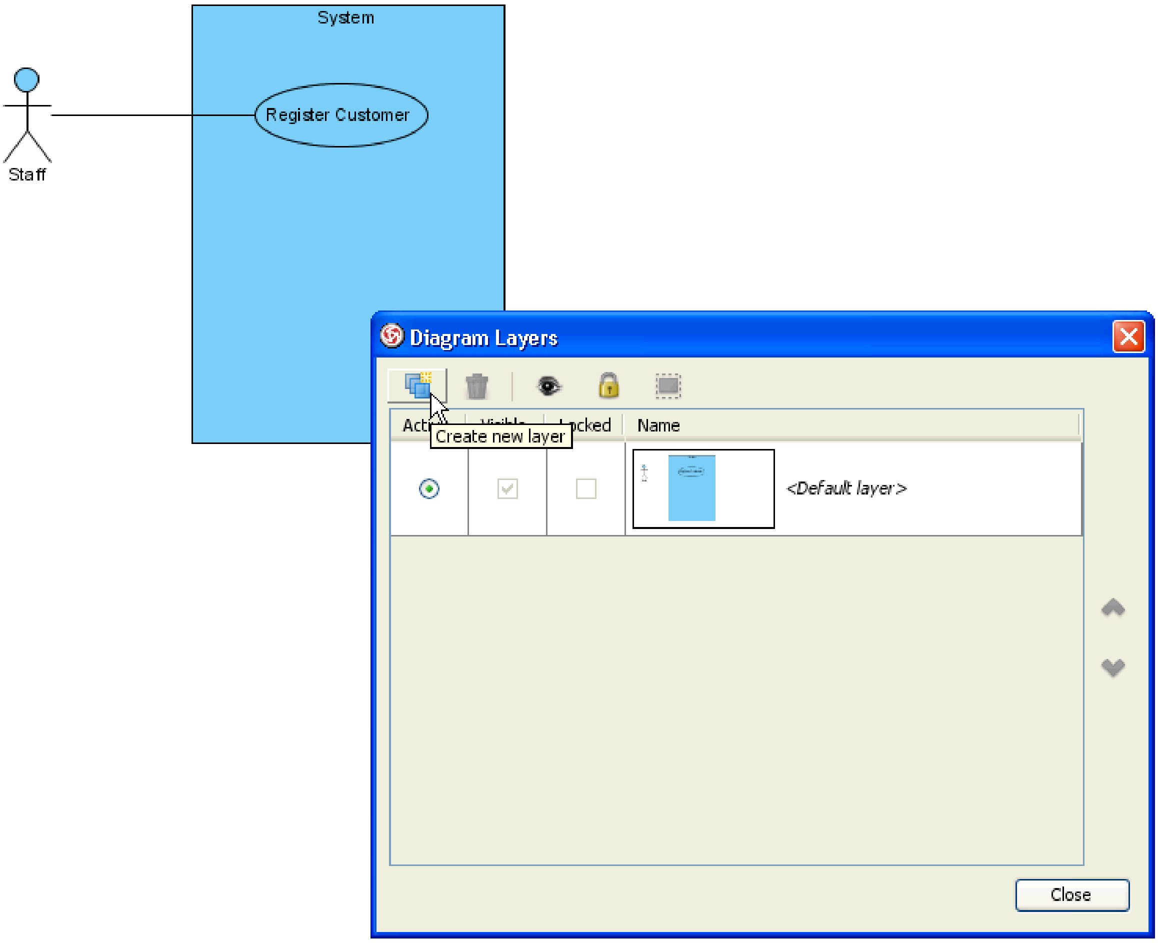
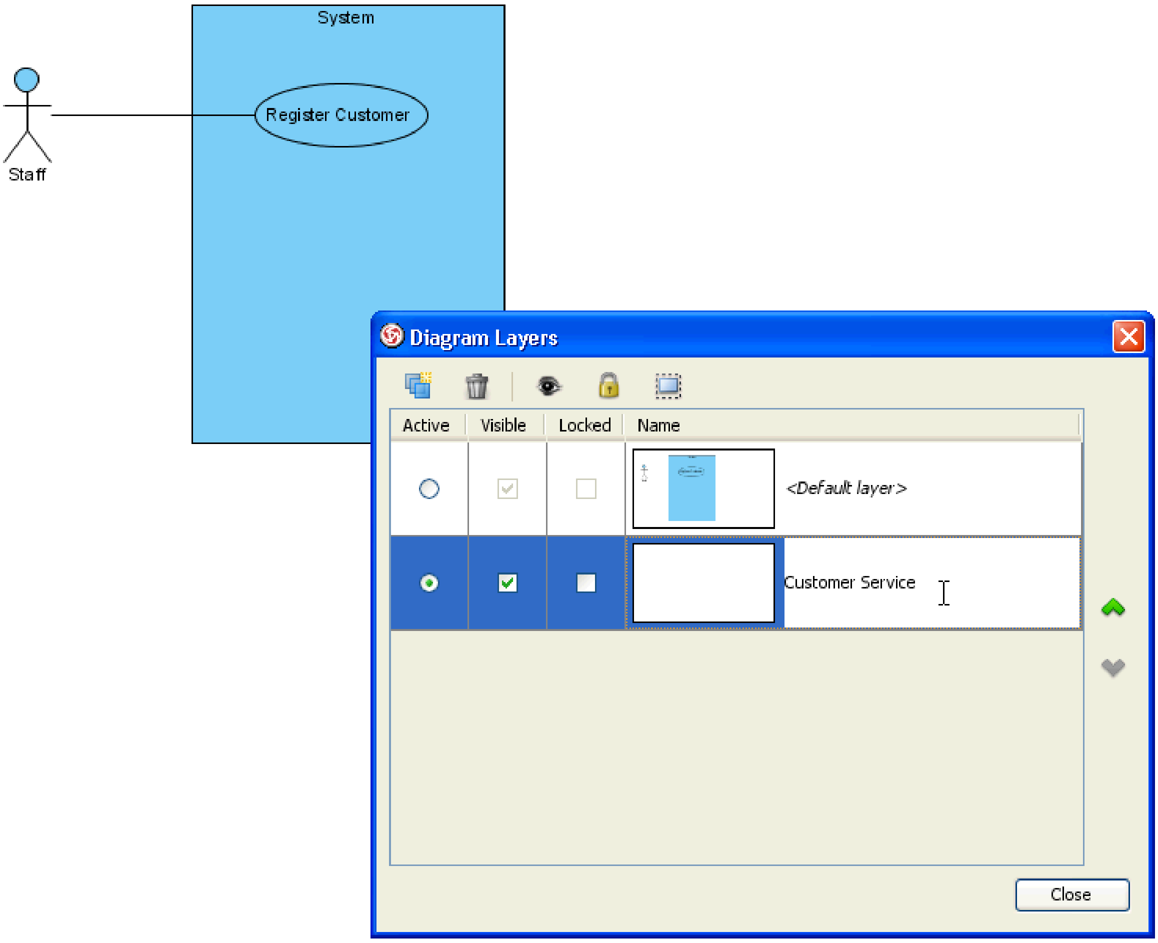


Figure 4-47 Create layer from dialog box

3.Layer is created, you may define the name of the layer and the layer becomes active layer.



Active	Visible	Locked	Name
○	☑	□	<Default layer>
●	☑	□	Customer Service

Figure 4-48 *New layer will be active layer*

Drawing Shapes in Layer

After select the active layer of the diagram, the new shapes drawn on diagram will be added into the active layer
1.Add some shapes on diagram. (You may or may not close the **Diagram Layers** dialog box.)

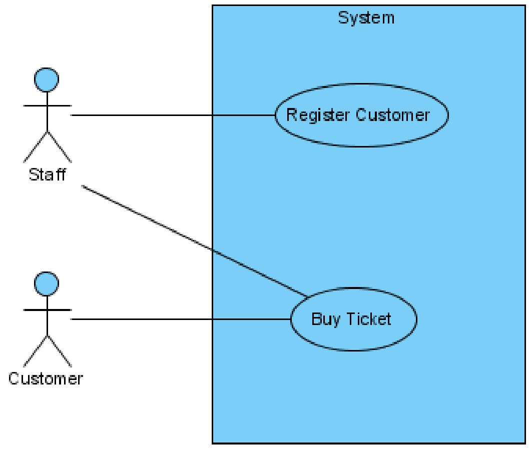
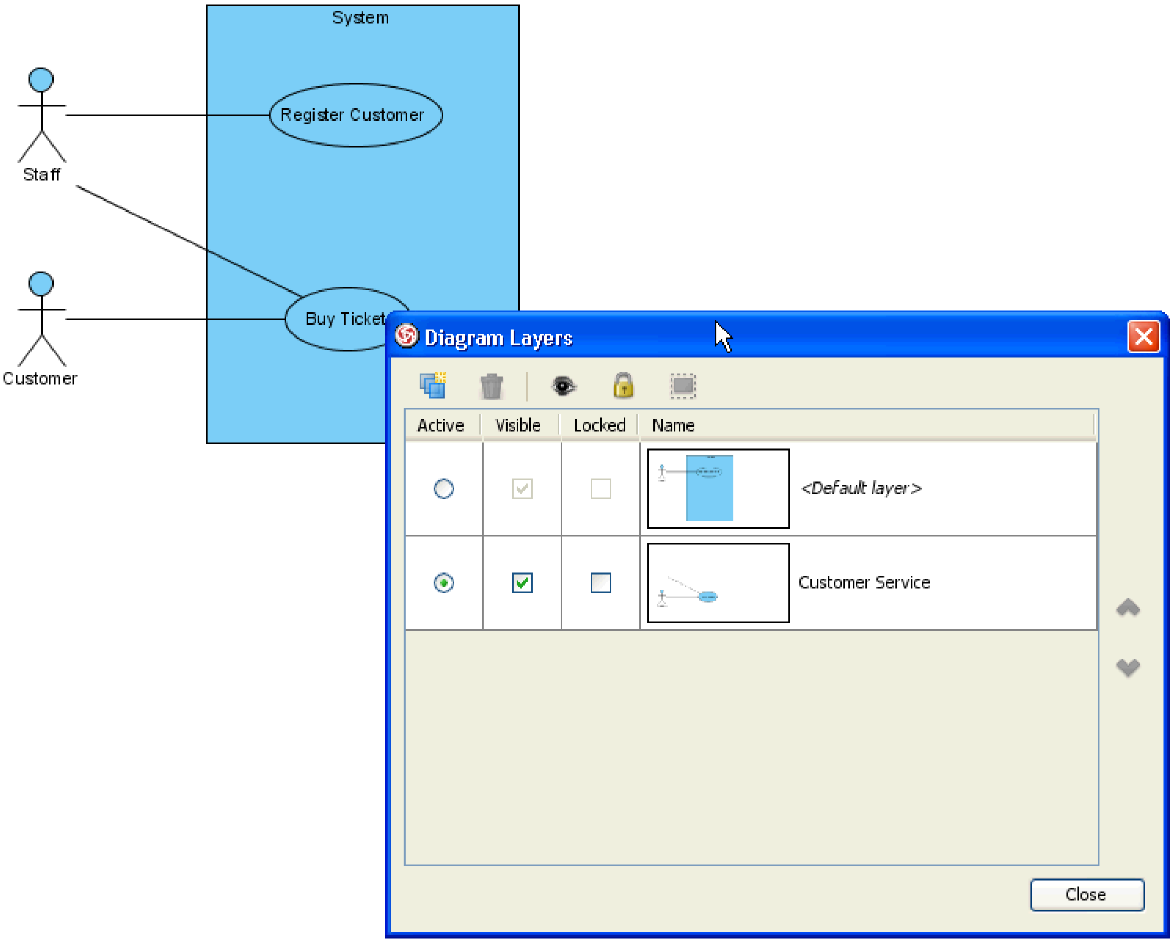


Figure 4-49 *New shapes are added on diagram*

2.The new shapes are added into the active layer, **Diagram Layers** dialog can show the preview of the layer.

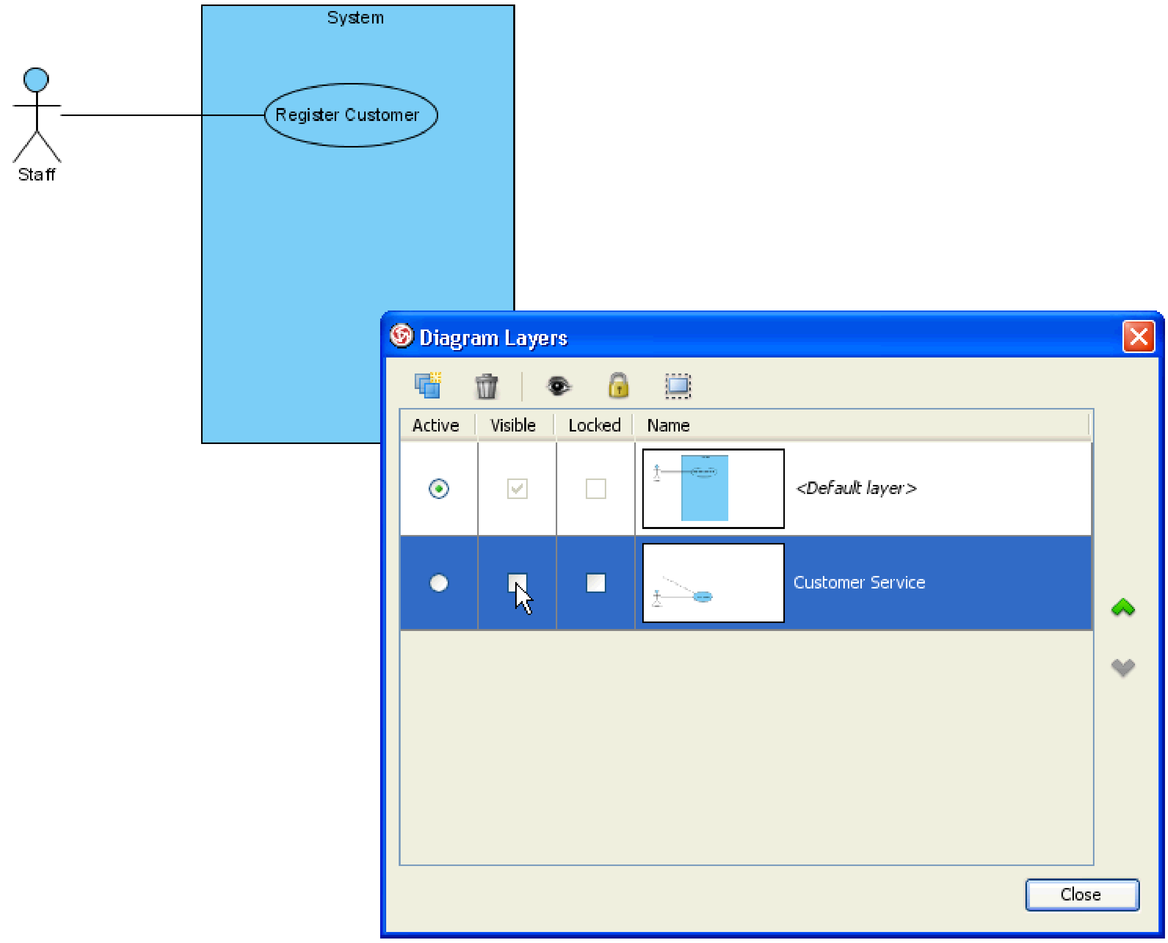


Active	Visible	Locked	Name
○	☑	□	*<Default layer>*
●	☑	□	Customer Service

Figure 4-50 *Shapes added to layer and shown on preview of the layer*

Make Layer Visible and Invisible

On **Diagram Layers** dialog box, you can make the layer visible or invisible on diagram. To do so, select **Visible** check box of the layer.

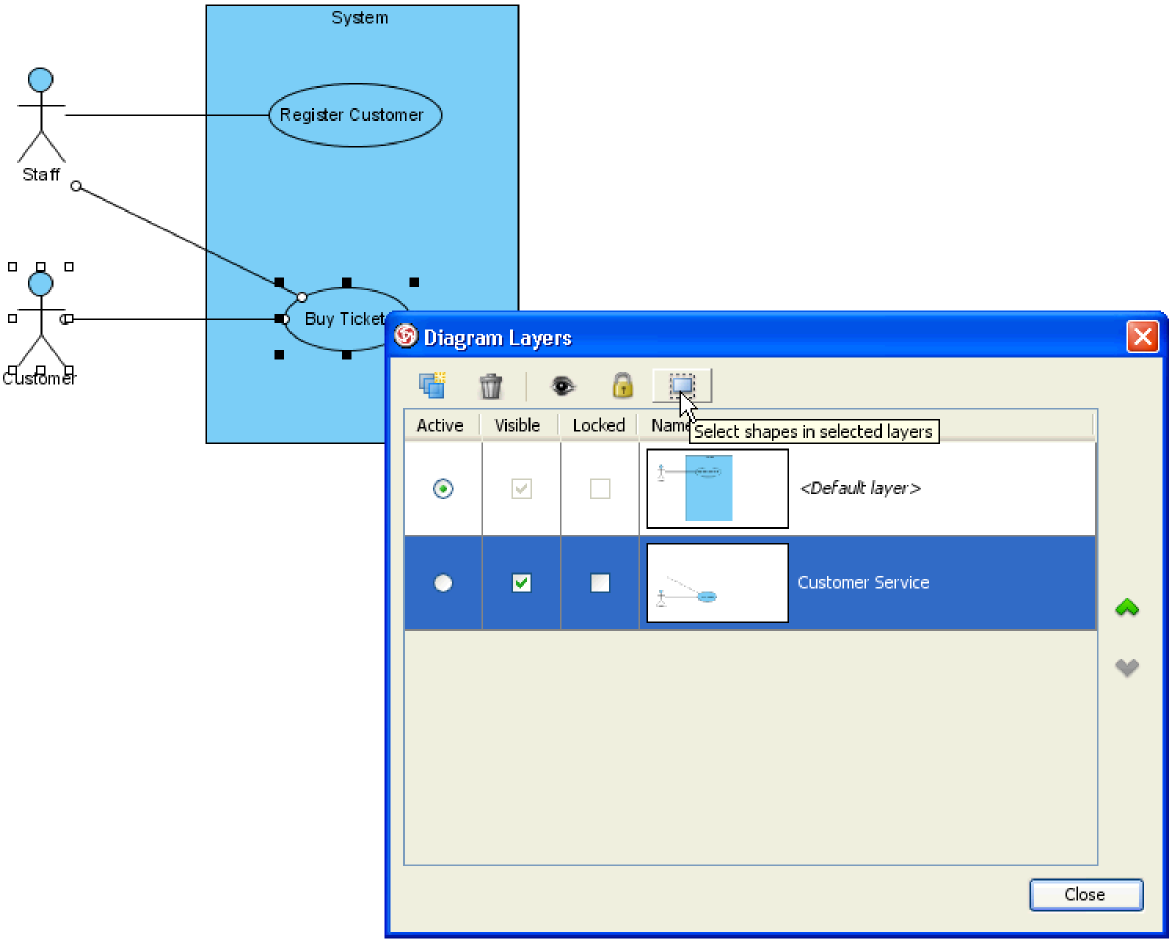


Active	Visible	Locked	Name
◉	☑	□	<Default layer>
○	□	□	Customer Service

Figure 4-51 *Shapes on the layers are invisible on diagram*

Select all Diagram Elements in Layer

You also can select all the shapes of the selected layer. To do so, select **Select shapes in selected layers** button on dialog box.



Active	Visible	Locked	Name
●	☑	□	<Default layer>
○	☑	□	Customer Service

Figure 4-52 *Shapes of the layer are selected on diagram*

Avoid accidentally moved Diagram Element with Selectable option

You may make some shapes to be selectable/non-selectable to avoid accidentally moved the shape.

Change the shape to Non-Selectable

1.Right-click on the selected shapes, uncheck **Selection > Selectable** from popup menu.

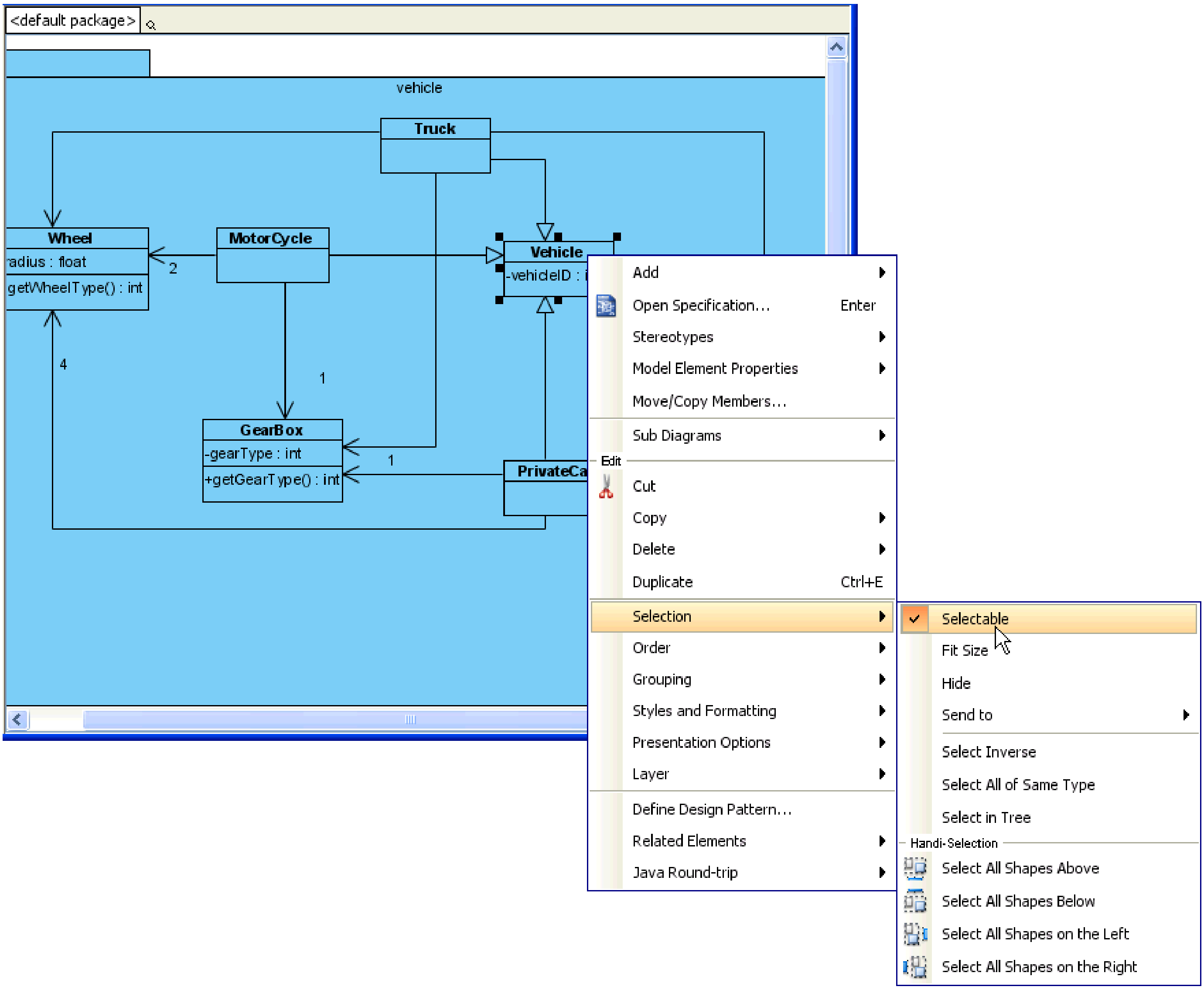


Figure 4-53 *Selected shape to be non-selectable*

2.After the shape is non-selectable, mouse click on the shape cannot select it.

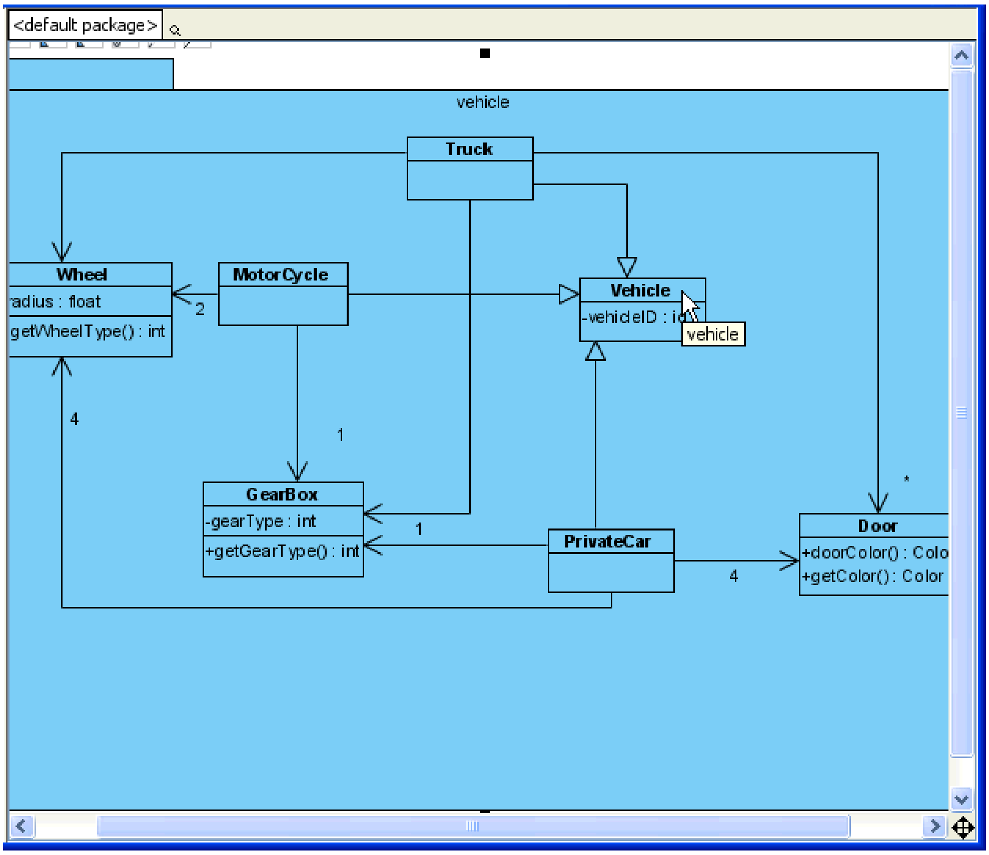


Figure 4-54 Shape is non-selectable

3.The shape will not be select by mouse drag on diagram too.

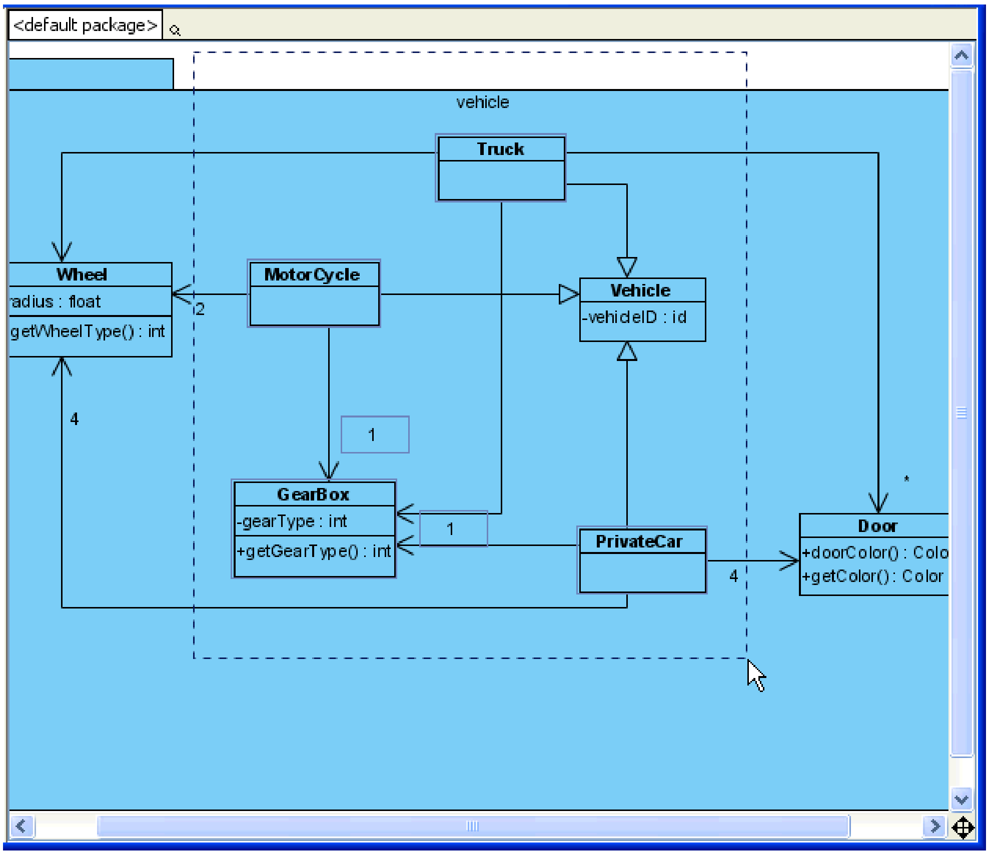


Figure 4-55 Not be selected

4.So, the non-selectable shape won't be moved following others.

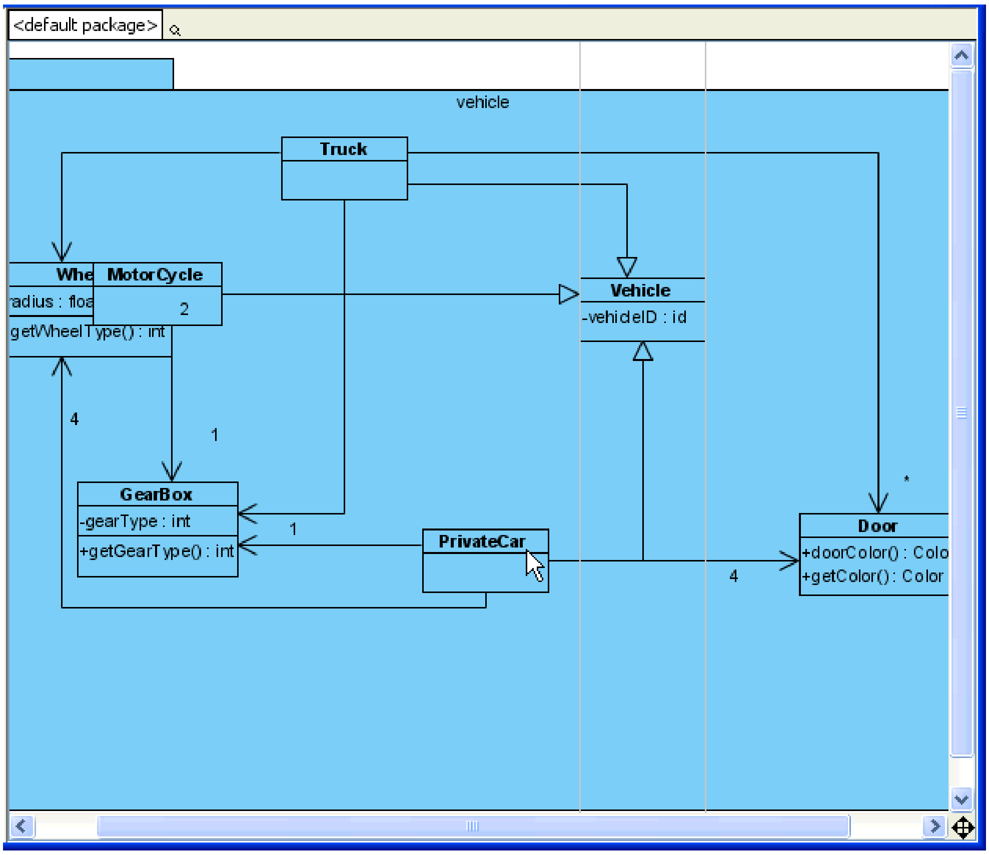


Figure 4-56 Not be moved

Change the shape to Selectable again

To make the shape selectable again, right-click on the non-selectable shape, select **Selectable** from popup menu.

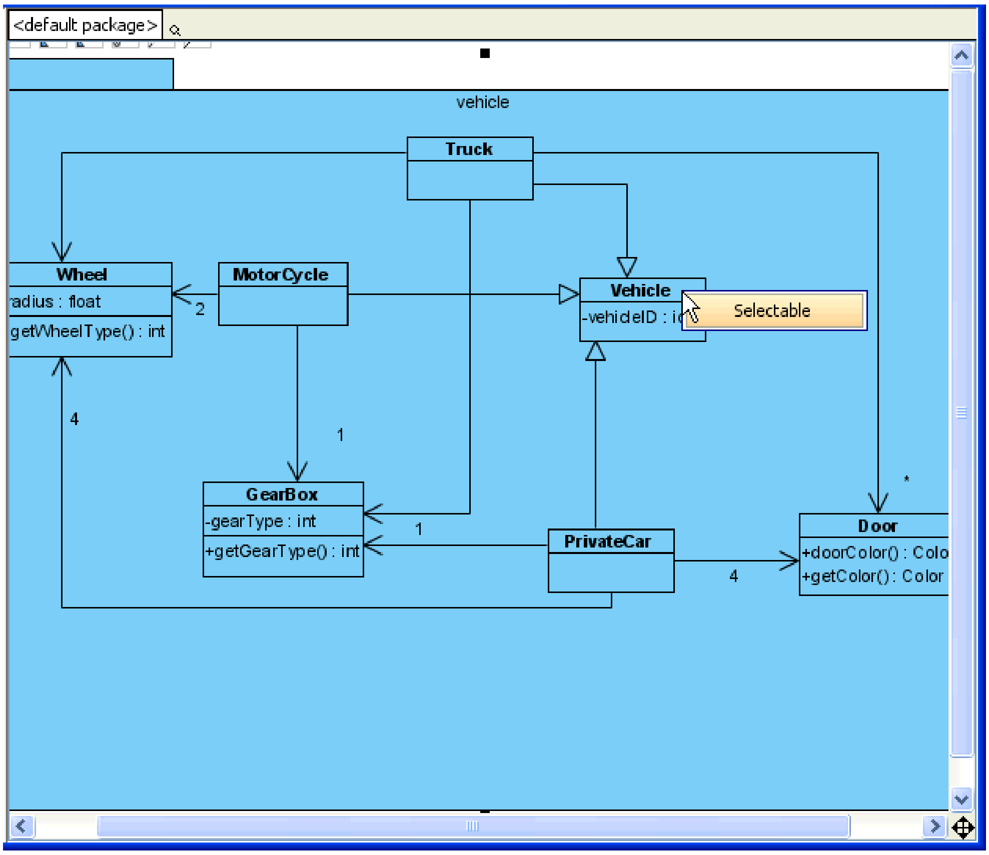


Figure 4-57 Select Selectable

Set 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, select **Diagram Content > Make All Shapes Non-Selectable** from popup menu.

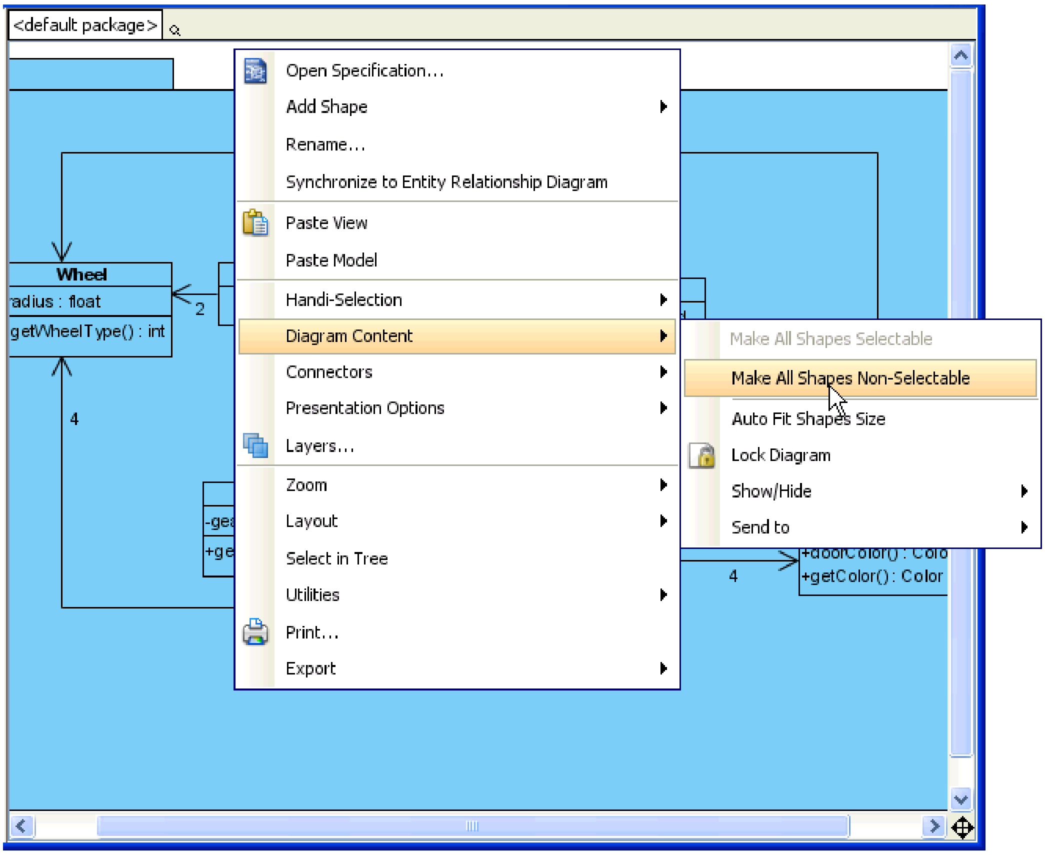


__Figure 4-58__ Make all shapes selectable/non-selectable

Make Diagram Read only by Diagram Locking

Diagram can be locked to be read-only. After locked, cannot add, update or delete the shapes on the diagram.

Lock the Diagram

1.To lock the diagram, right-click on the diagram, select **Diagram Content > Lock Diagram** from popup menu.

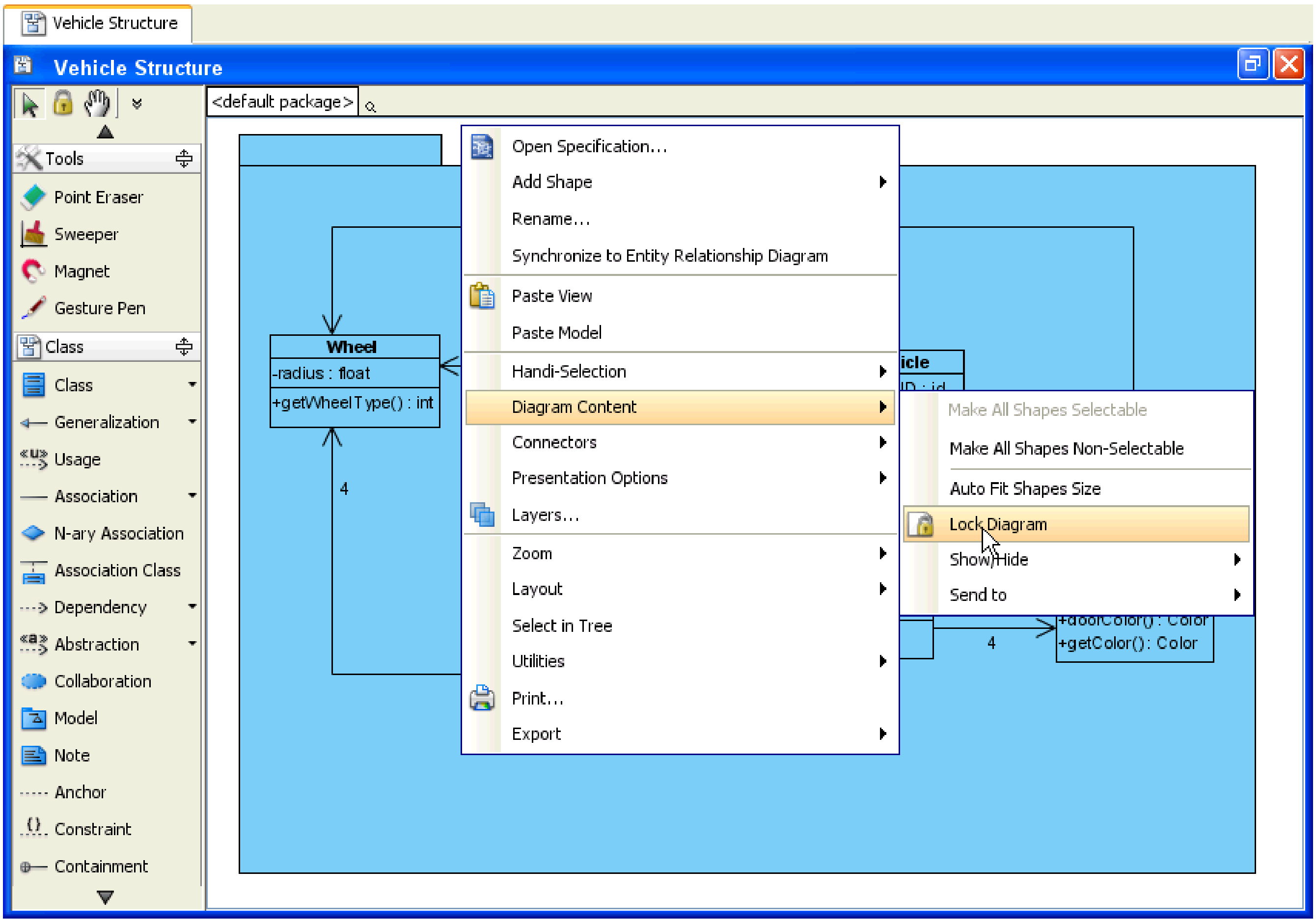


Figure 4-59 *Lock the diagram*

2.After locked, the platelet is disabled. No shapes can be selected. Diagram become read-only. The **Diagram Locked** dialog box is shown, closing this dialog box won't unlock the diagram.

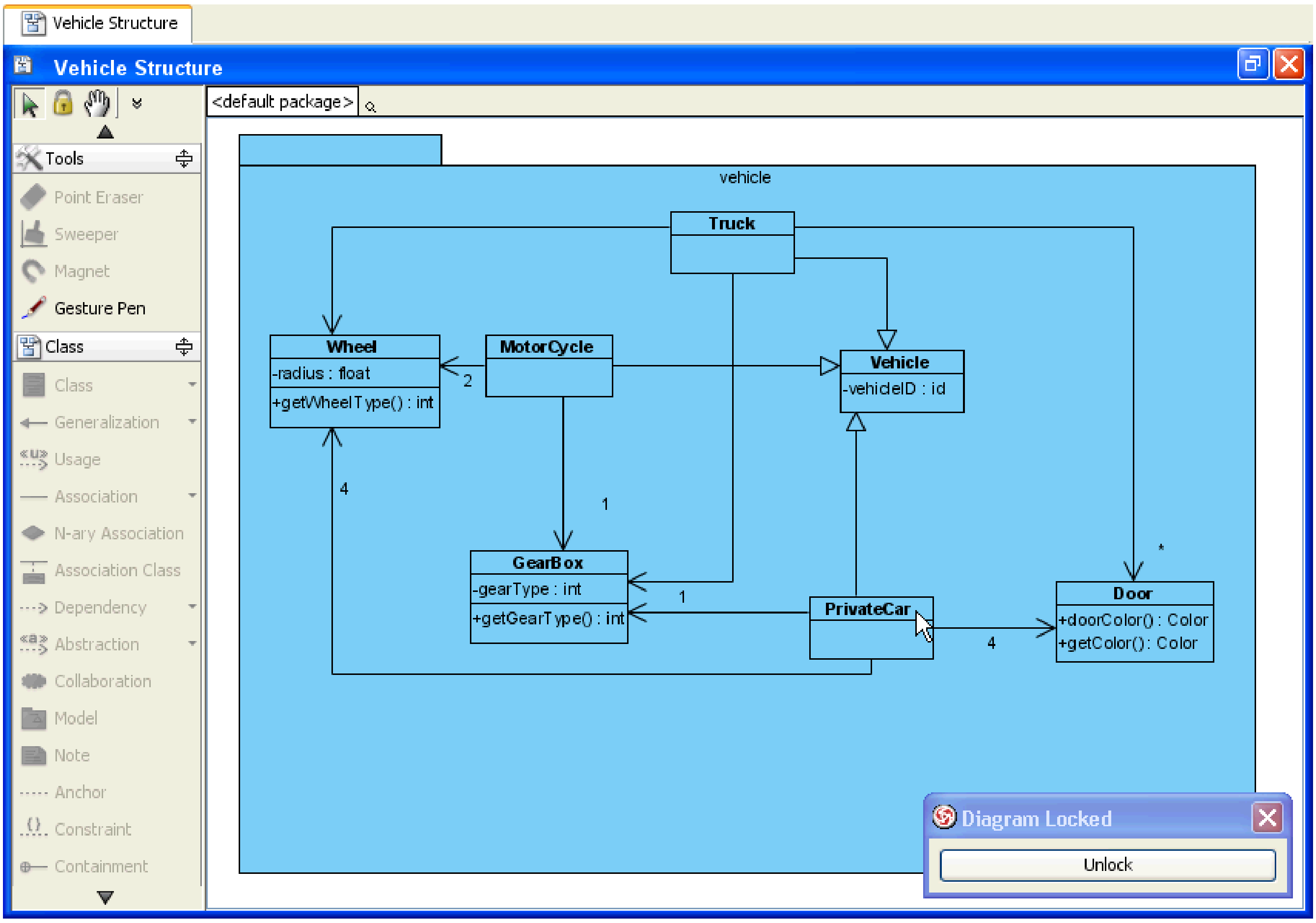


Figure 4-60 Diagram is locked

Unlock the Diagram

1.To unlock the diagram, you may click **Unlock** button on **Diagram Locked** dialog box.

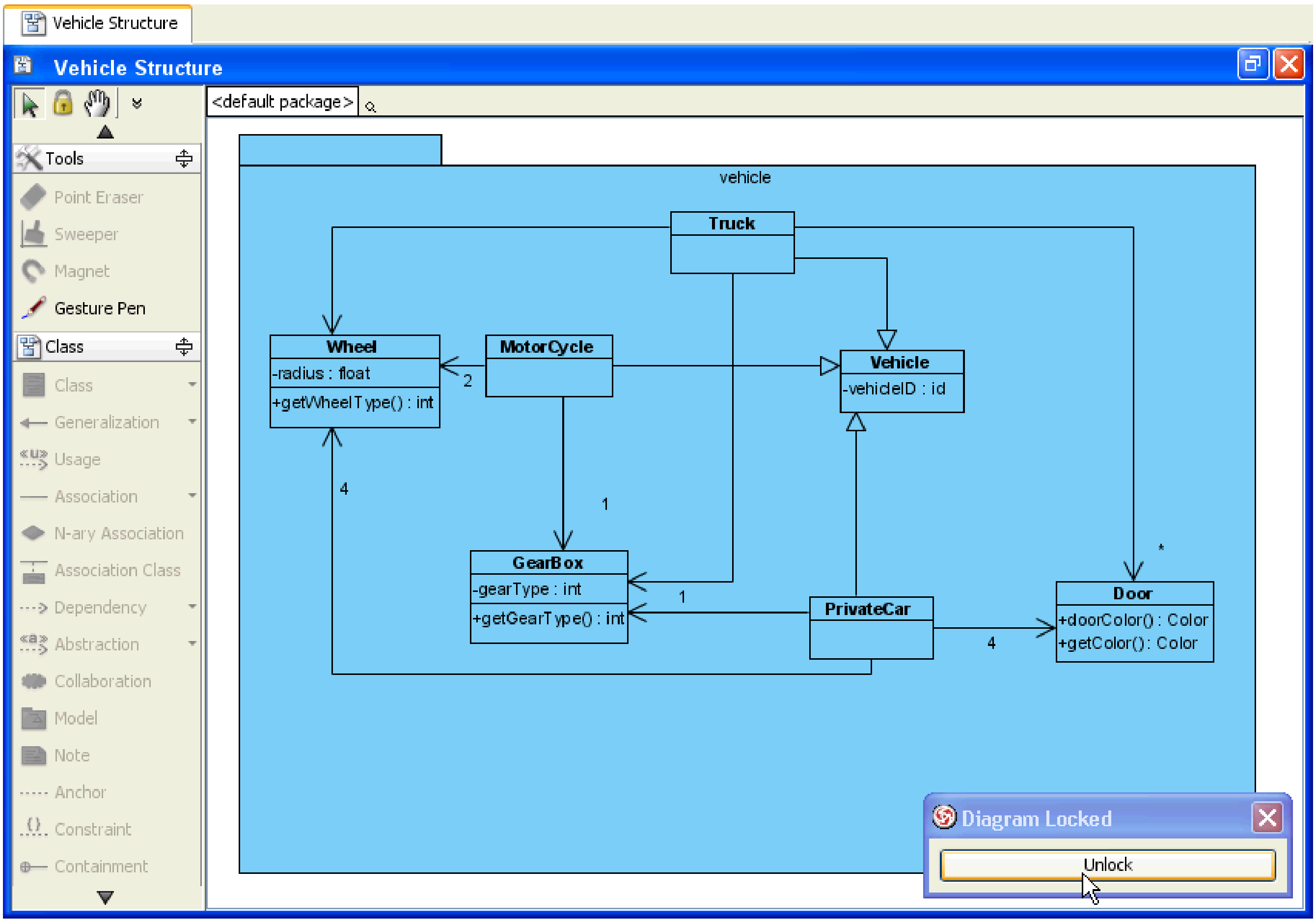


Figure 4-61 Unlock

2.Or right-click on diagram, select **Diagram Content > Lock Diagram** from popup menu.

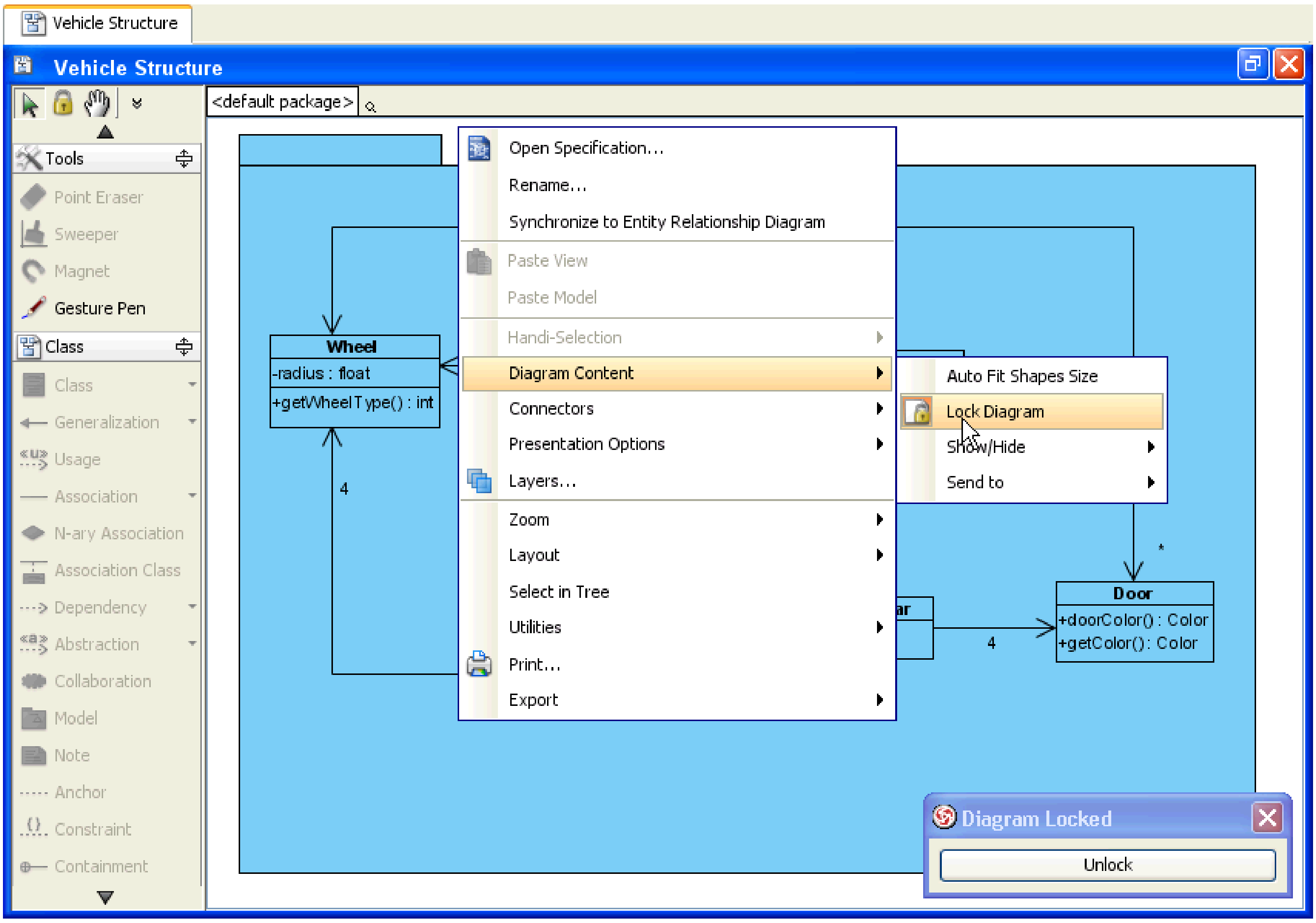


Figure 4-62 Unlock from popup menu

Showing model element in multiple diagrams (Context base modeling)

Models can be shown on more than one diagram. To show models on different diagram. You may drag and drop the model from tree to diagram, or copy and paste the model view. For example, you have a diagram contains some classes may be reused on other diagram.

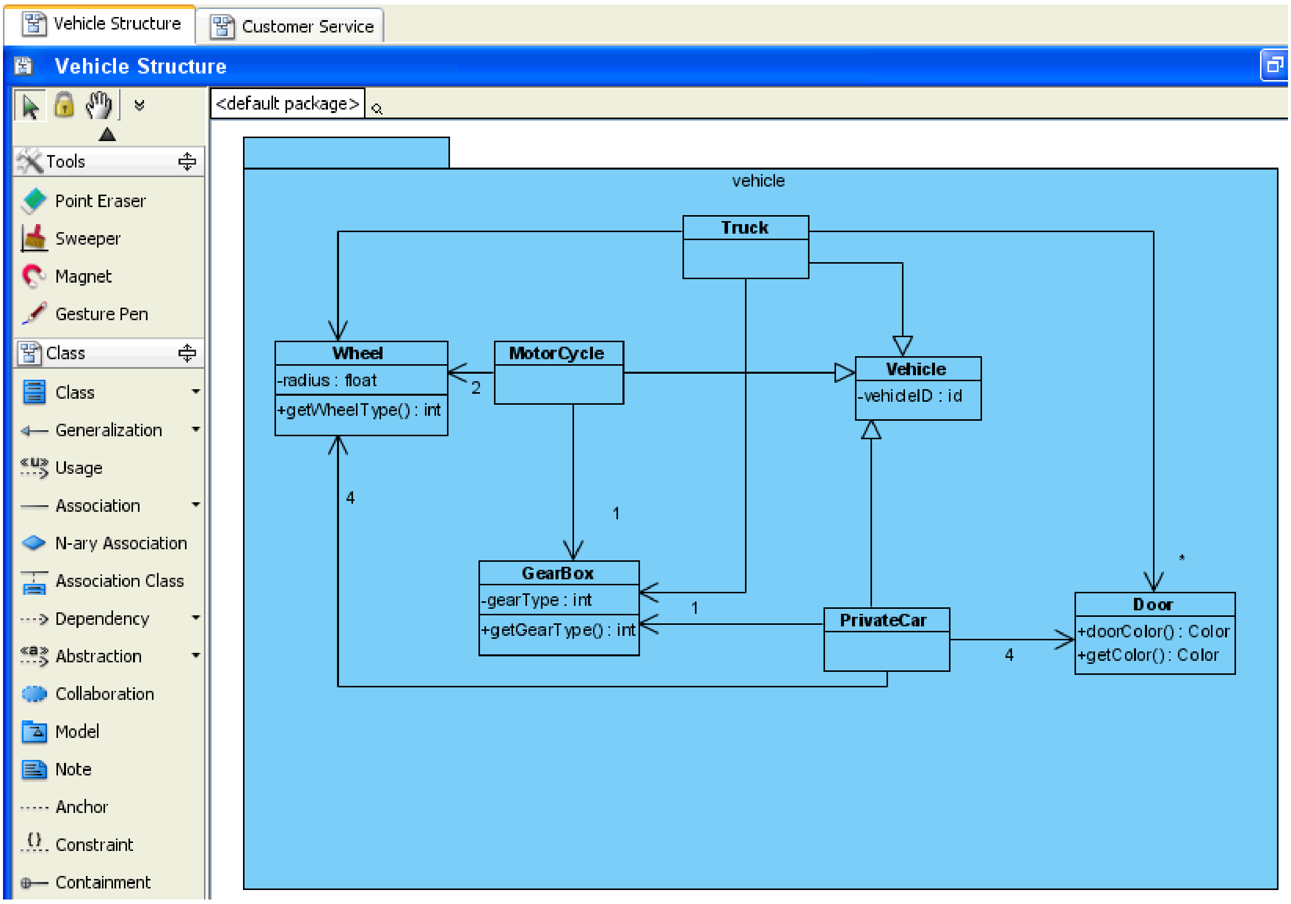


Figure 4-63 *Classes may be reused*

And have another diagram that will reuse the classes.

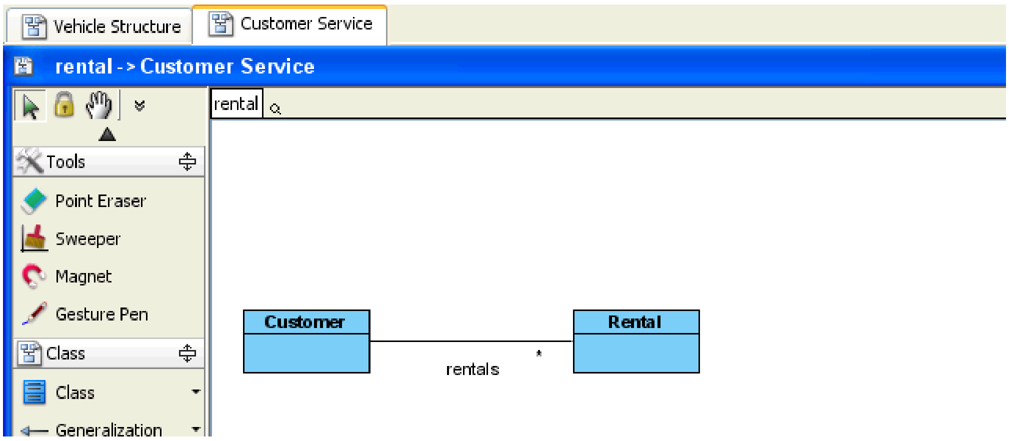


Figure 4-64 *Diagram will reuse classes*

Reusing Model Elements

1.You may drag and drop the models from tree to the diagram.

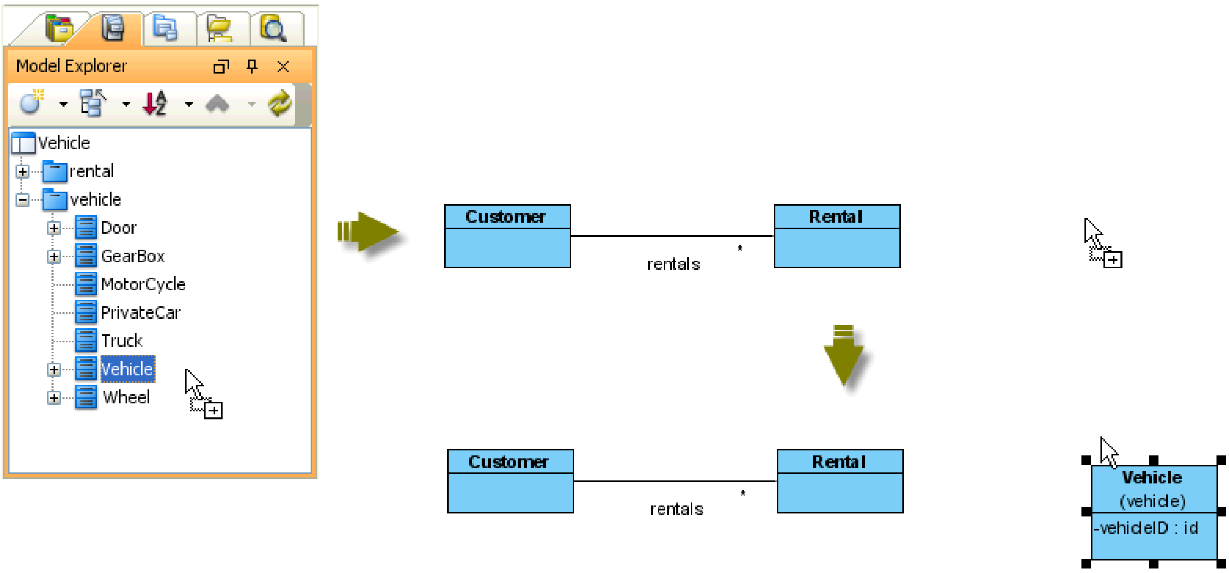


Figure 4-65 *Model shown on diagram after drag and drop*

2.You may also copy the model first.

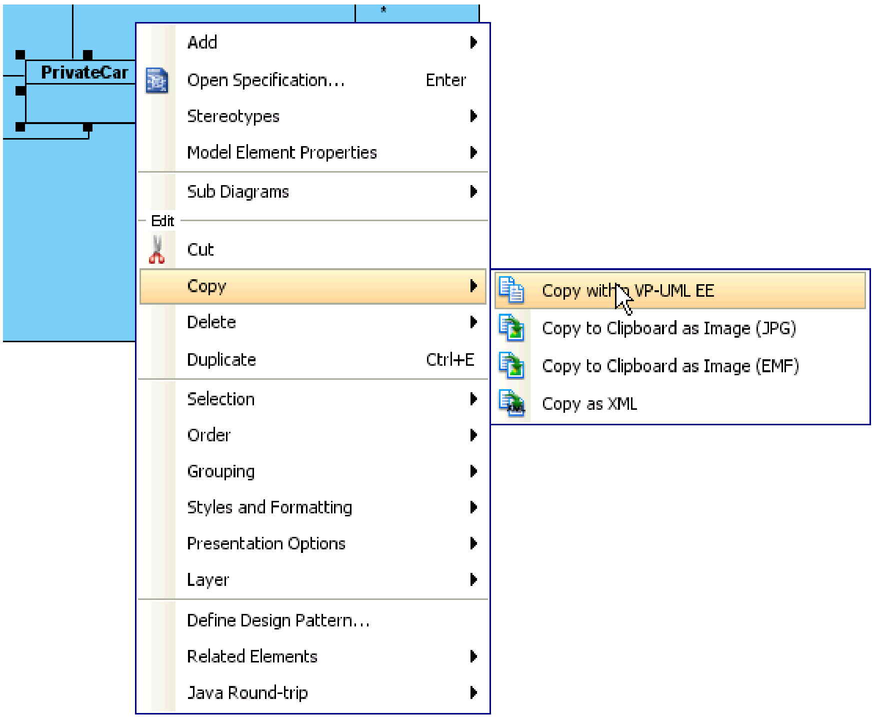


Figure 4-66 *Copy the model*

3.And then **Paste View** on the diagram. (**Paste Model** will create a new model.)

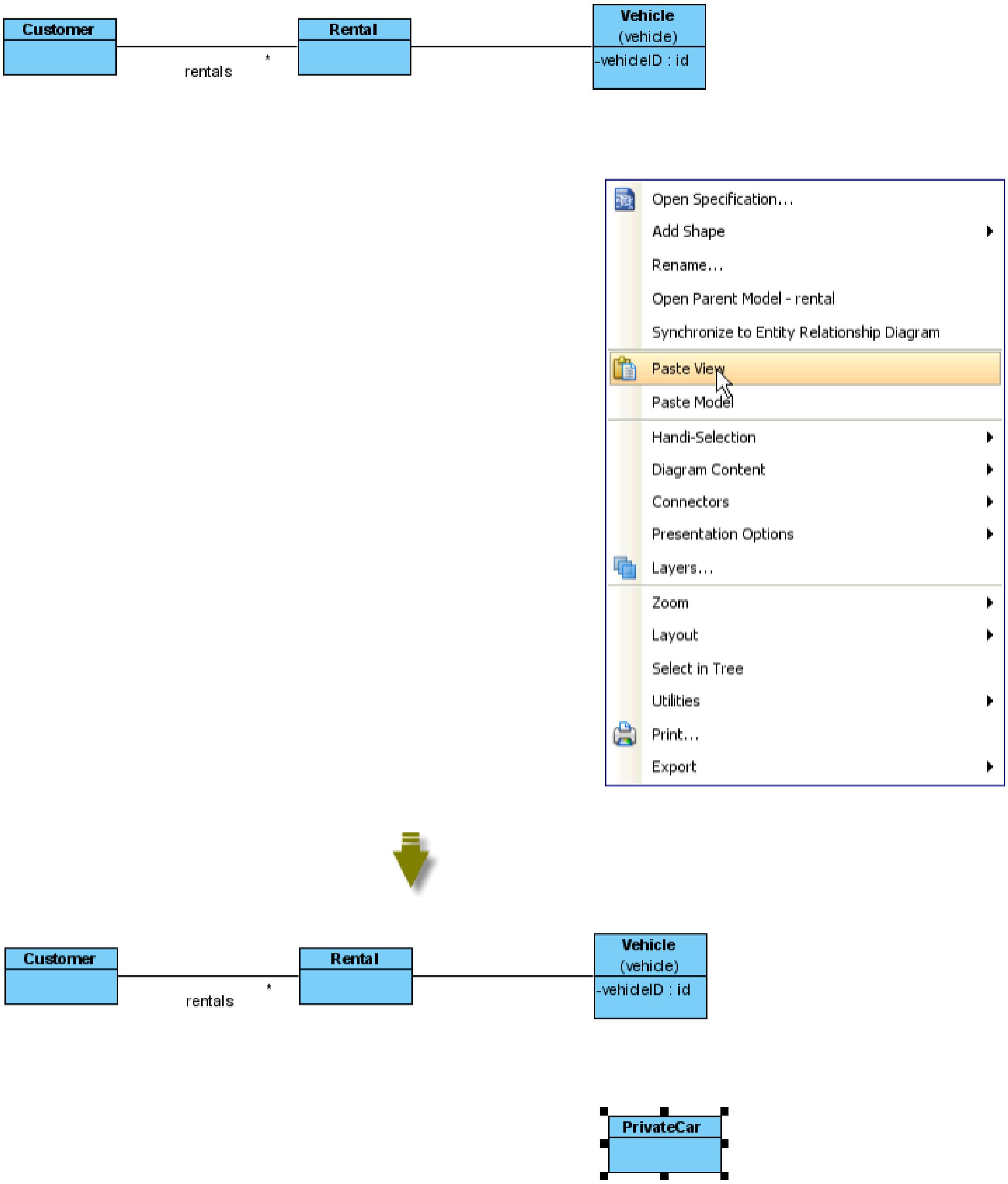


Figure 4-67 *Paste view*

4.The relationship between the models may not be shown on diagram, to show the relationship, right-click on the model, select **Related Elements > Visualize Related Model Element...** from popup menu.

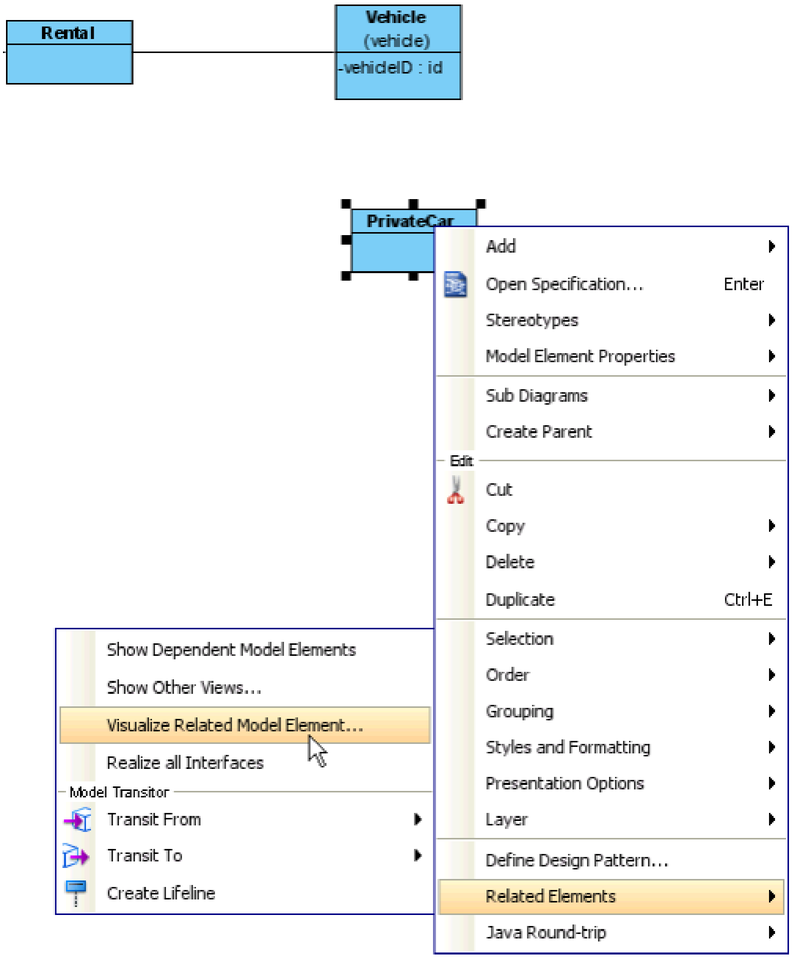
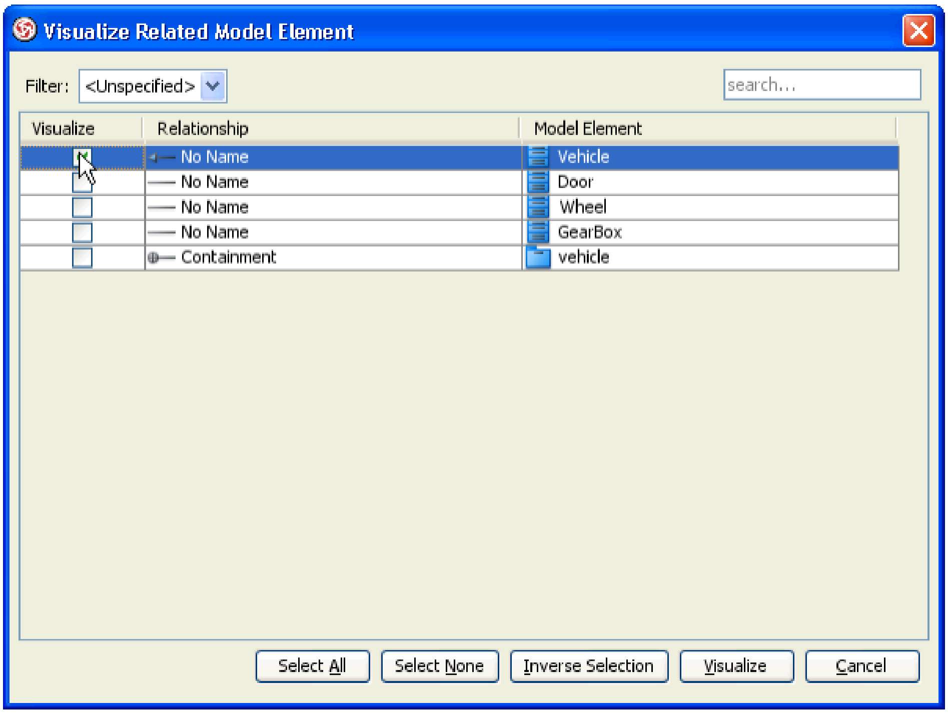


Figure 4-68 Visualize Related Model Element

5.On **Visualize Related Model Element** dialog box, select the relationships between the models, and select **Visualize** button.



Filter: <Unspecified>

Visualize	Relationship	Model Element
☑	No Name	Vehicle
□	No Name	Door
□	No Name	Wheel
□	No Name	GearBox
□	Containment	vehicle

Figure 4-69 Visualize the relationship

6.The relationship is shown on diagram.

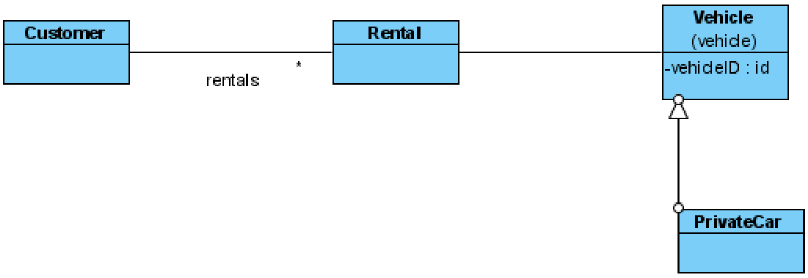


Figure 4-70 *Relationship is visualized*

Open Other Views

You can know the model is shown on which diagram(s).

1.To do so, right-click on the model, select **Related Elements > Show Other Views...** from popup menu.

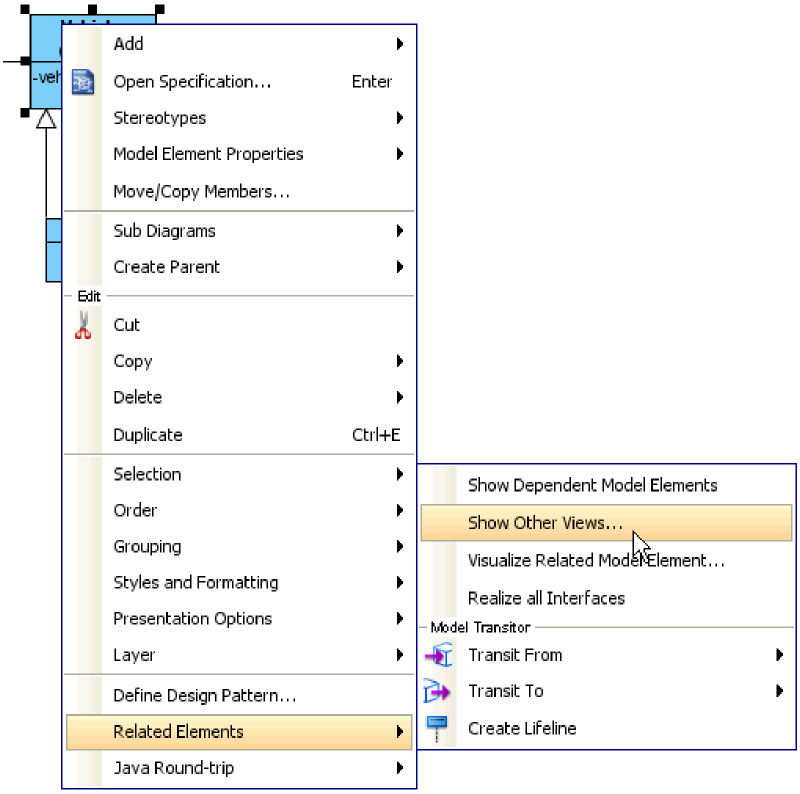


Figure 4-71 *Go to other view*

2.The diagrams, that shows this model (besides the current diagram), are listed and shown as preview. You can select the diagram and select **Go to View** button to open the diagram.

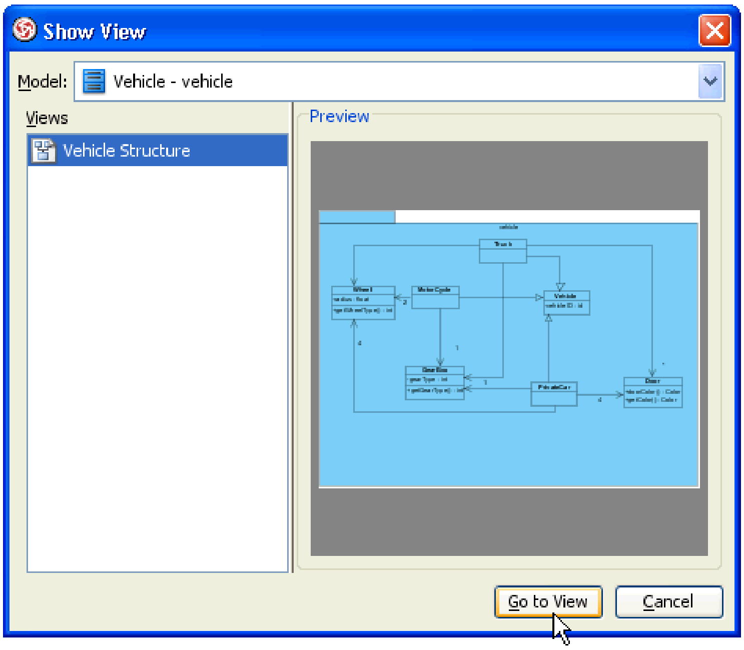


Figure 4-72 Go to other view

3.The diagram is opened and the model is selected on the diagram.

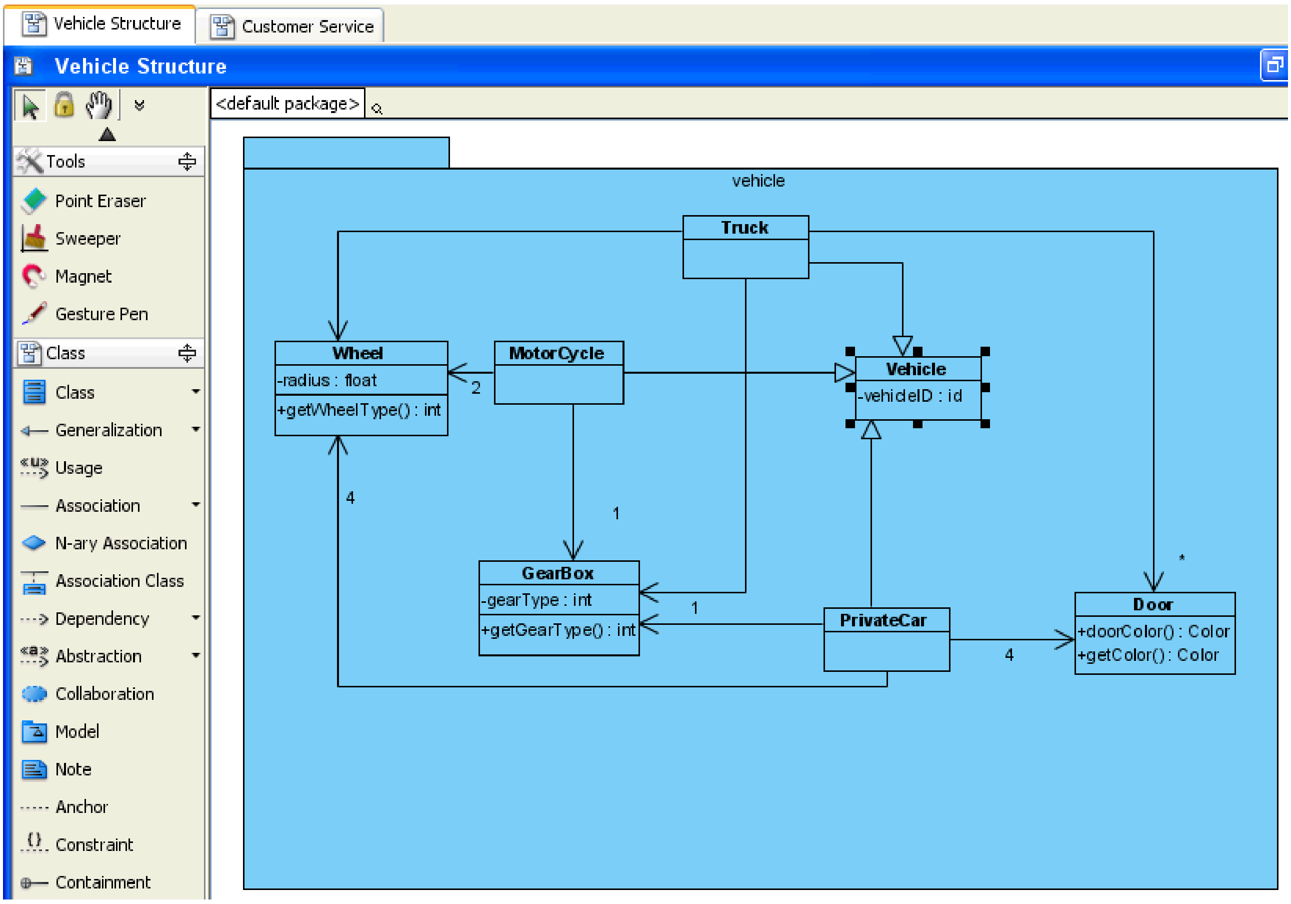


Figure 4-73 Diagram is opened with selecting the model

Enforcing master view between model element and shape

If a model is shown on different diagram. You can specify which shape is the master view of the model. To do so, right-click on the shape, select Selection > Set as Master View from popup menu. (If the shape you right-clicked already is the master view, the Set as Master View won't be shown on popup menu)

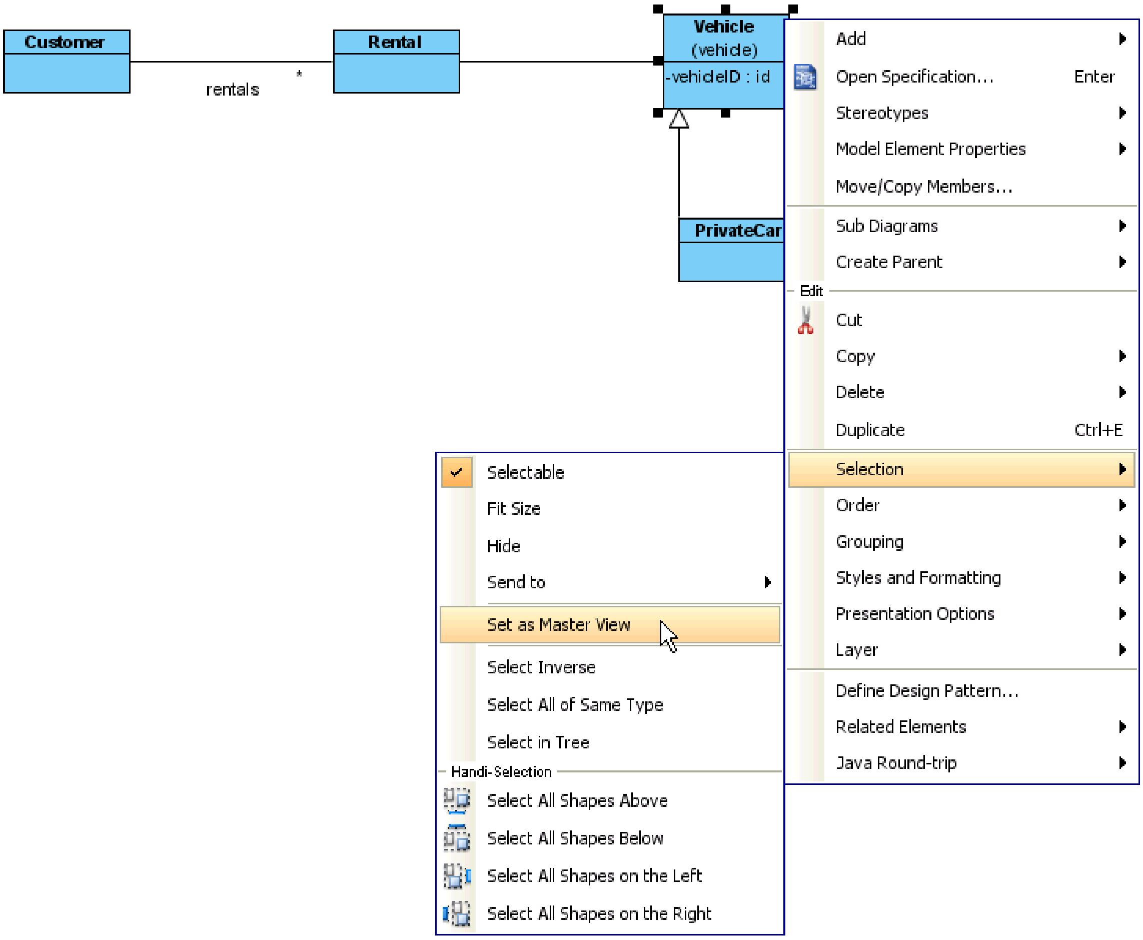


Figure 4-74 *Set the shape as Master View of the model*

Using Overview Diagram

Overview Diagram can show a set on diagrams as preview, and shows the relationships between the diagrams.

Creating Overview Diagram

1.To create Overview Diagram, you may select **File > New Diagram > Others > Overview Diagram** from main menu.

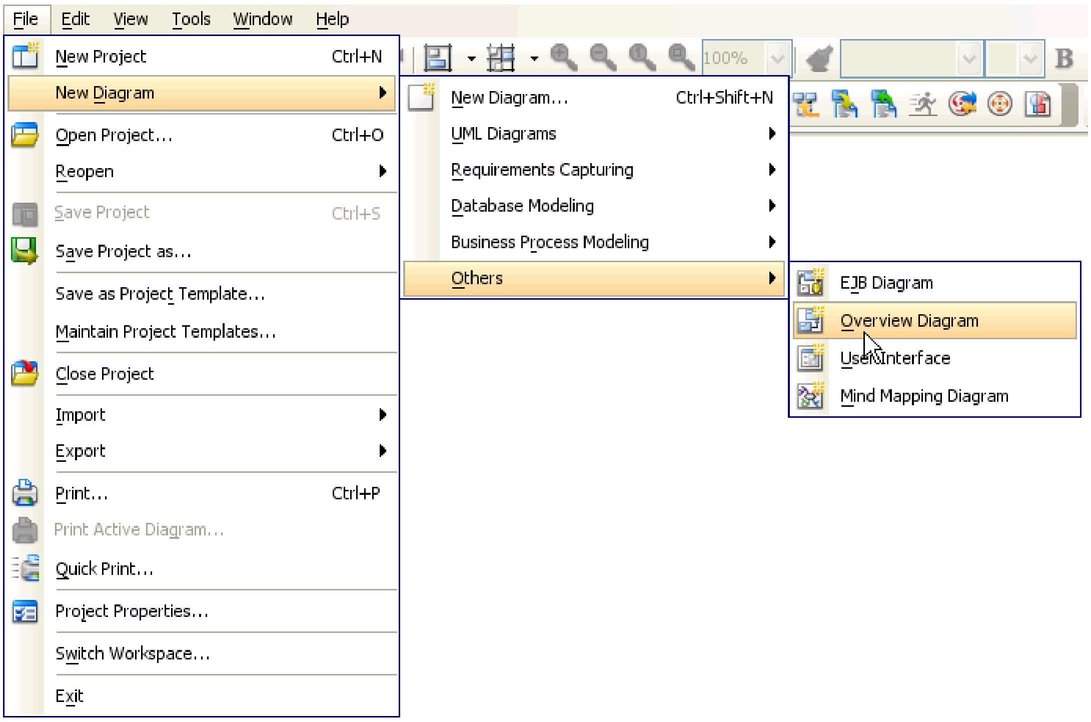


Figure 4-75 Create Overview Diagram

2.The overview diagram is created.

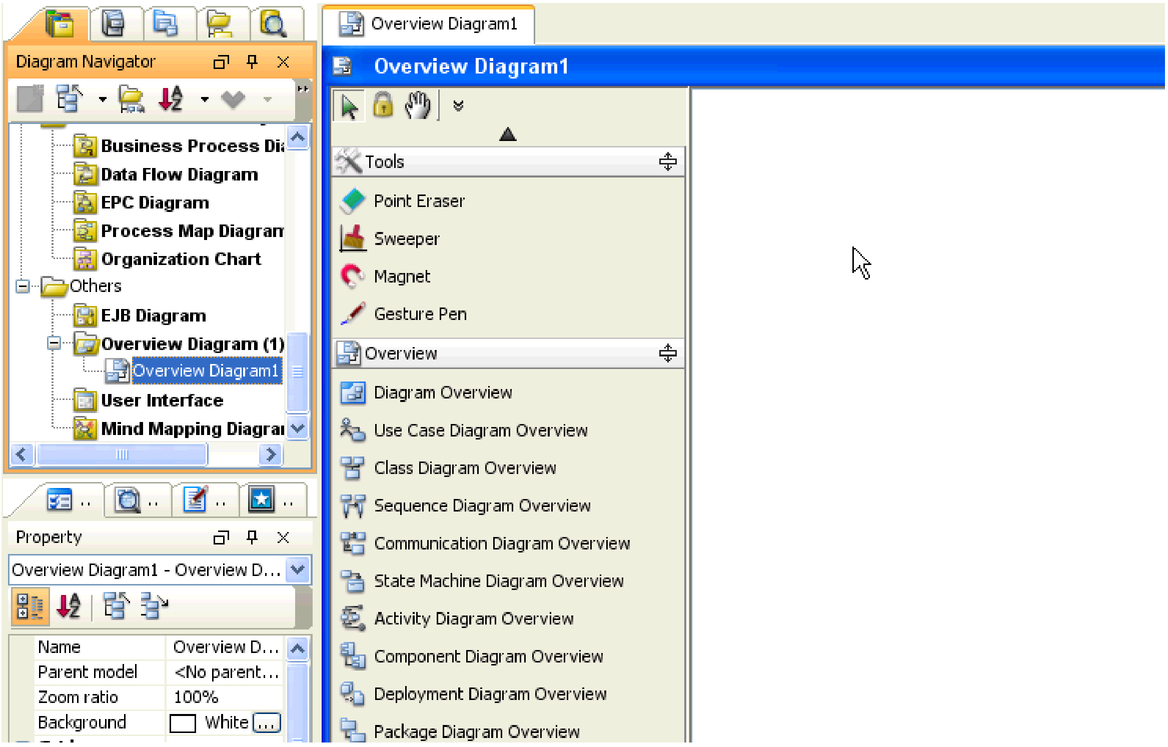


Figure 4-76 Overview Diagram is created

Creating Use Case Diagram

You can create diagram from overview diagram.

1.To create an Use Case Diagram from overview diagram, select **Use Case Diagram Overview** from platelet.

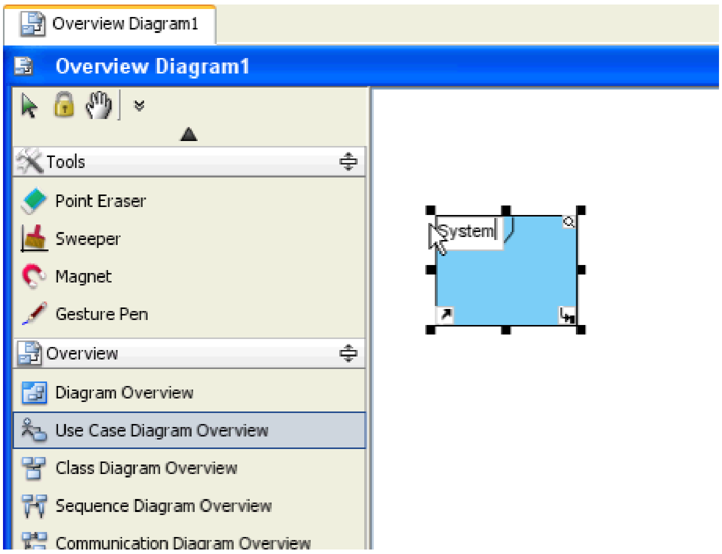


***Figure 4-77** You may name the use case diagram*

2.The use case diagram will be auto opened to let you draw the content.

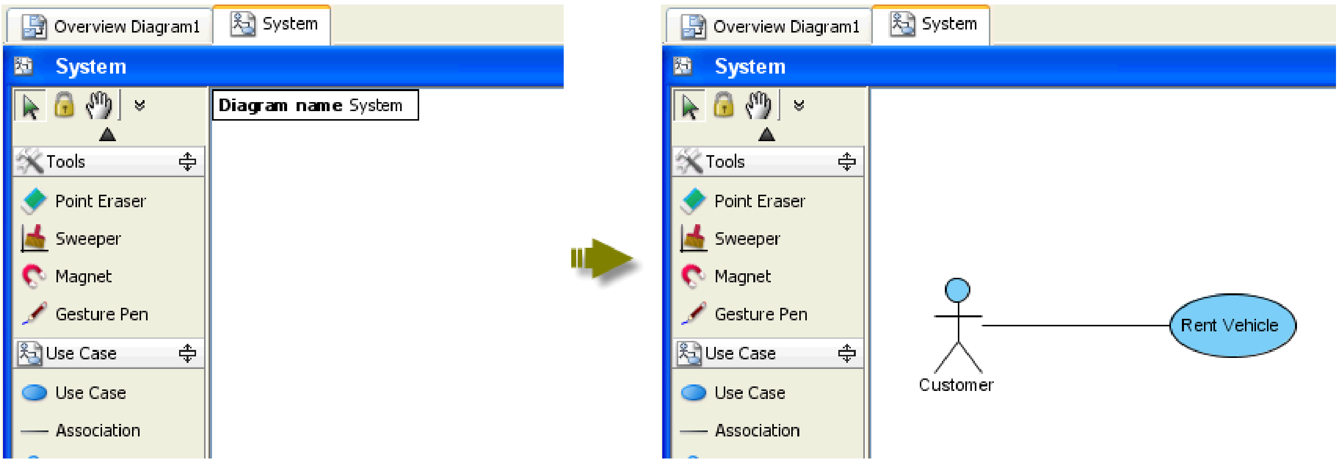


***Figure 4-78** Draw the use case diagram*

3.Switch back to overview diagram, you can see the preview of the use case diagram is shown.

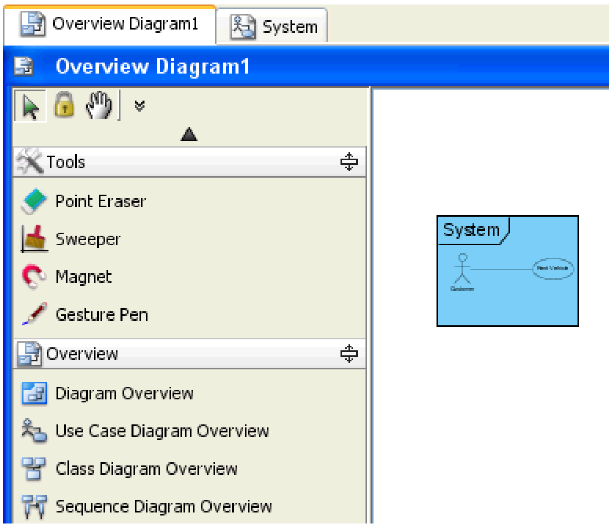


***Figure 4-79** Overview diagram shows the use case diagram*

Showing Existing Diagram

Besides creating a new diagram from overview diagram, you can also show an existing diagram on overview diagram.

1.Assume you have an sequence diagram.

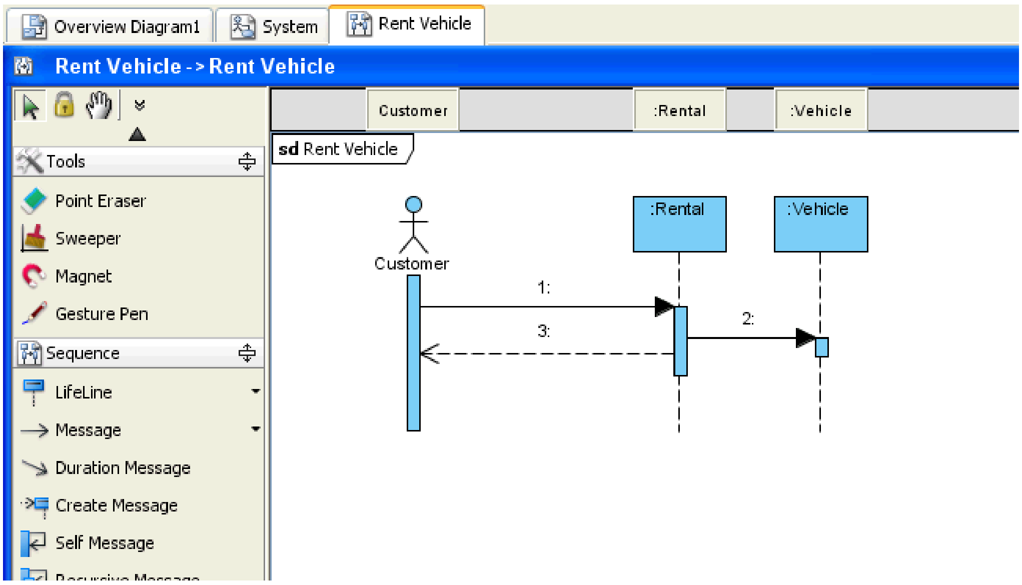


Figure 4-80 *Sequence diagram is already exists*

2.On overview diagram, create a **Diagram Overview**.

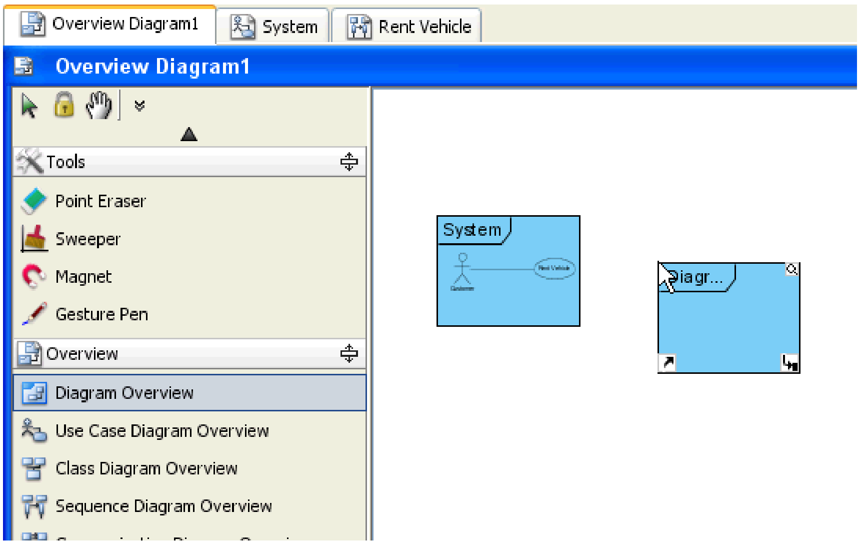


Figure 4-81 *Create Diagram Overview*

3.Right-click on the diagram overview and select **Associate to Diagram...** from popup menu.

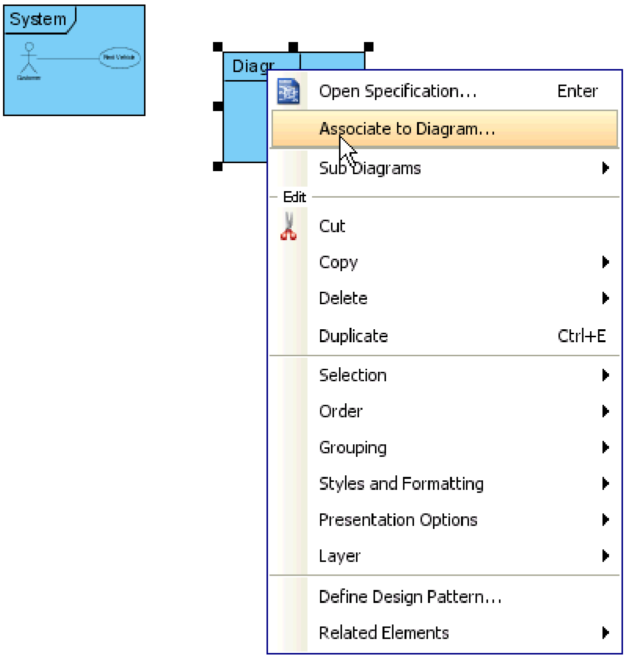


Figure 4-82 *Associate to diagram*

4.Select the sequence diagram and select **OK** button.

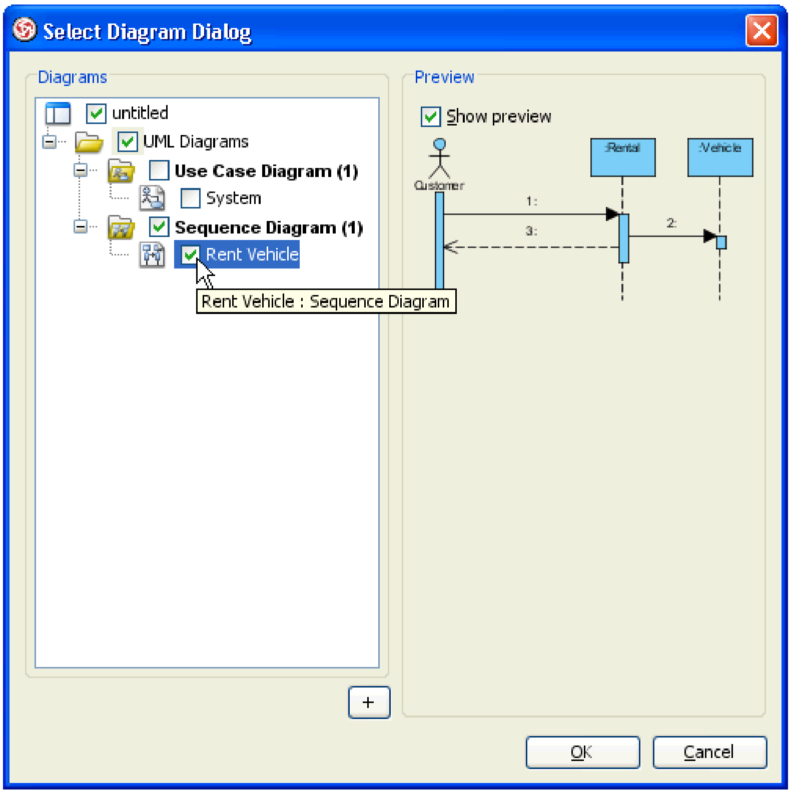


Figure 4-83 *Select existing diagram*

5.The diagram overview will show the sequence diagram. You also can resize the diagram overview to preview the diagram clearly. And add relationship between the diagram overview.

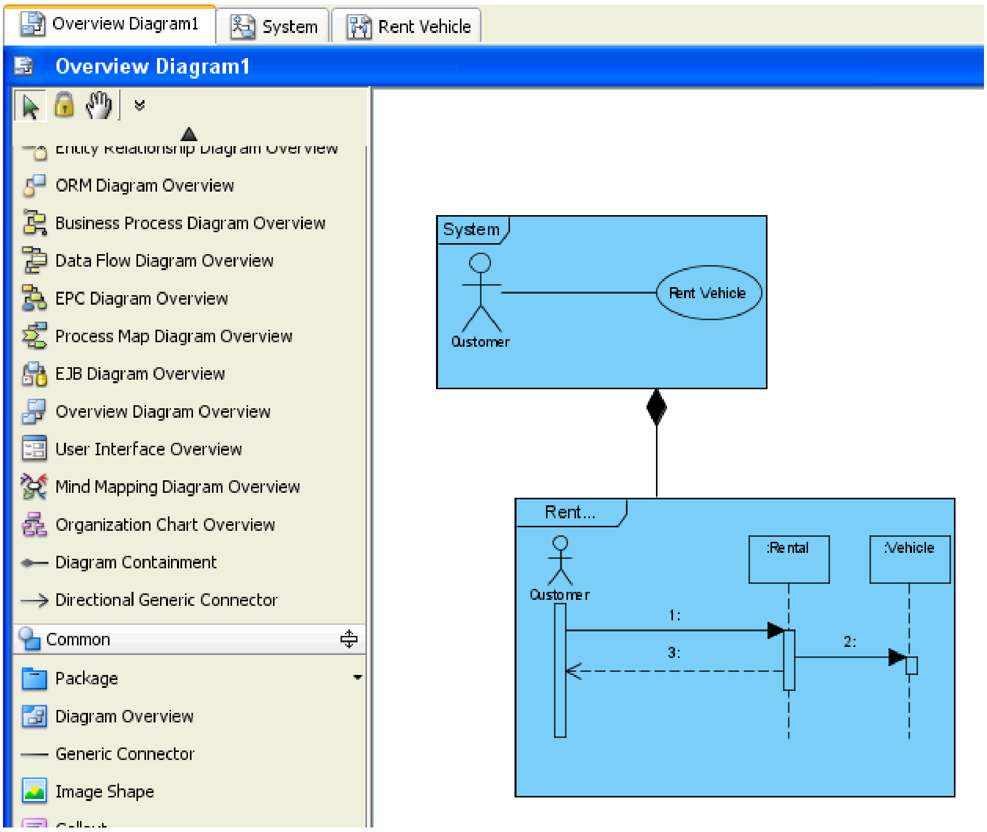


Figure 4-84 Diagrams are shown

Reference to External Resources

Reference to Files

1. Mouse over shape and click on References resource.

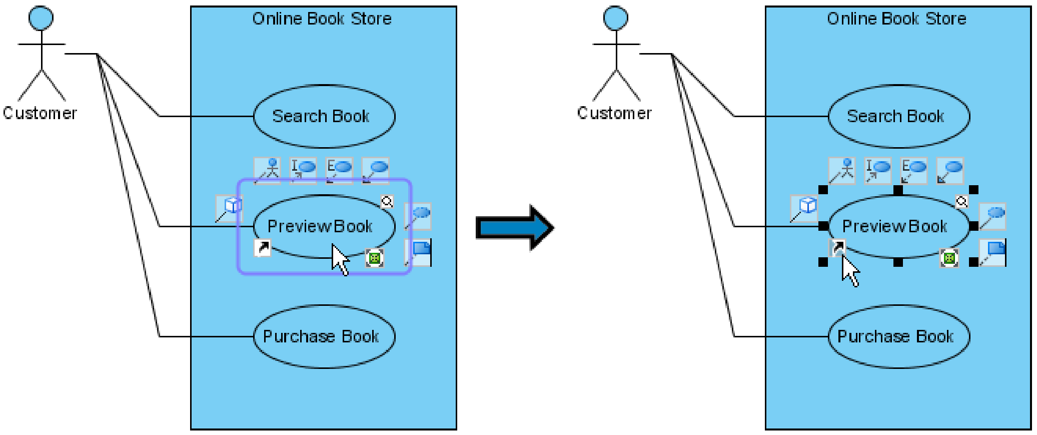


Figure 5-1 *Mouse over references resource*

2. Click on Edit References... menu item.

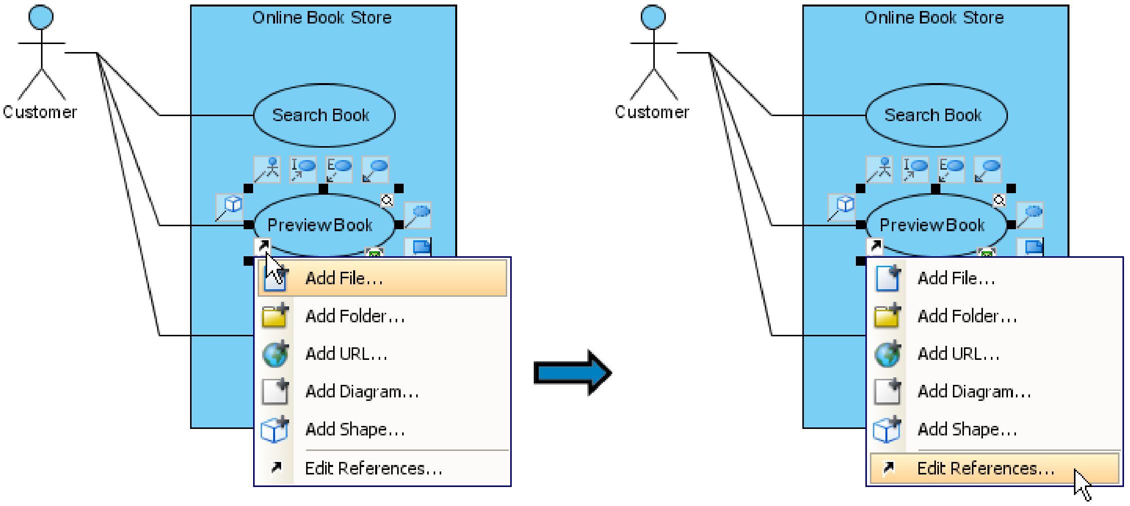


Figure 5-2 *Edit references from resource*

3. Press Add File... button.

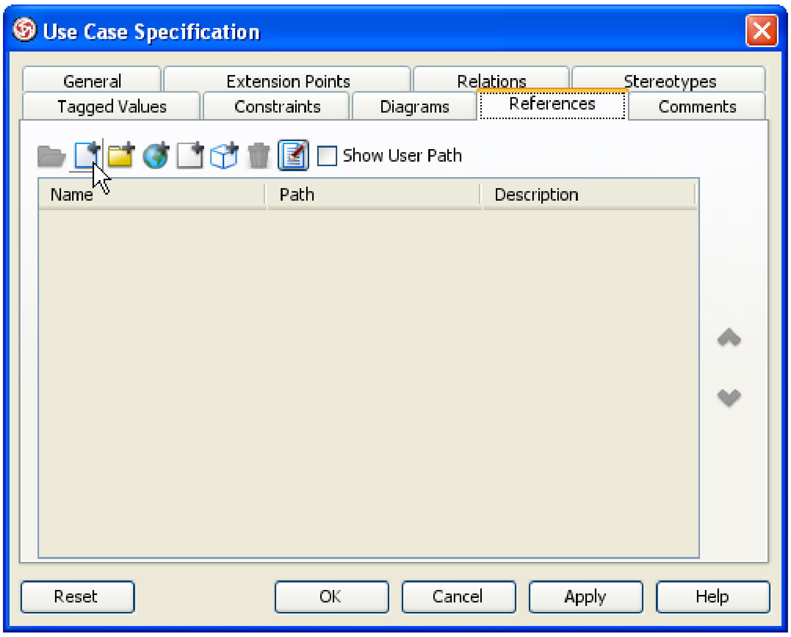
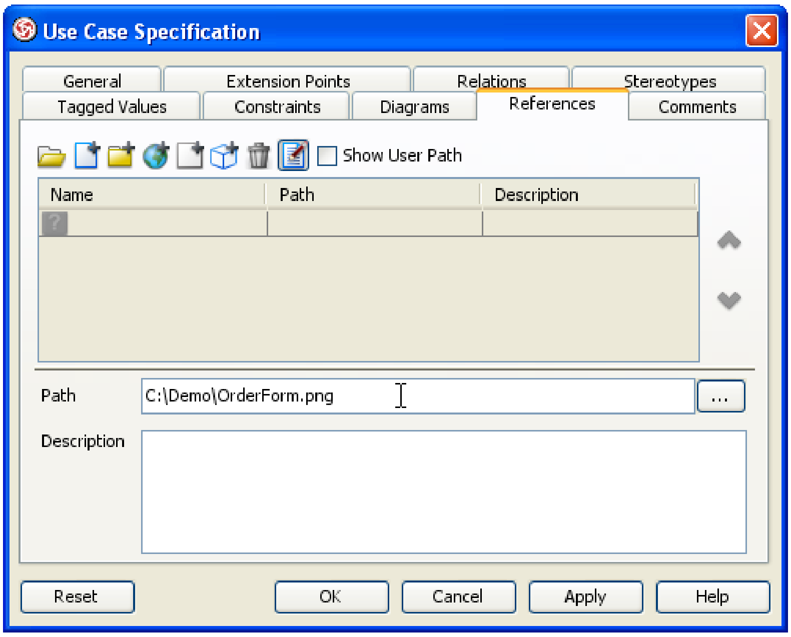


Figure 5-3 *Adding file reference*

4. Input path of reference file.

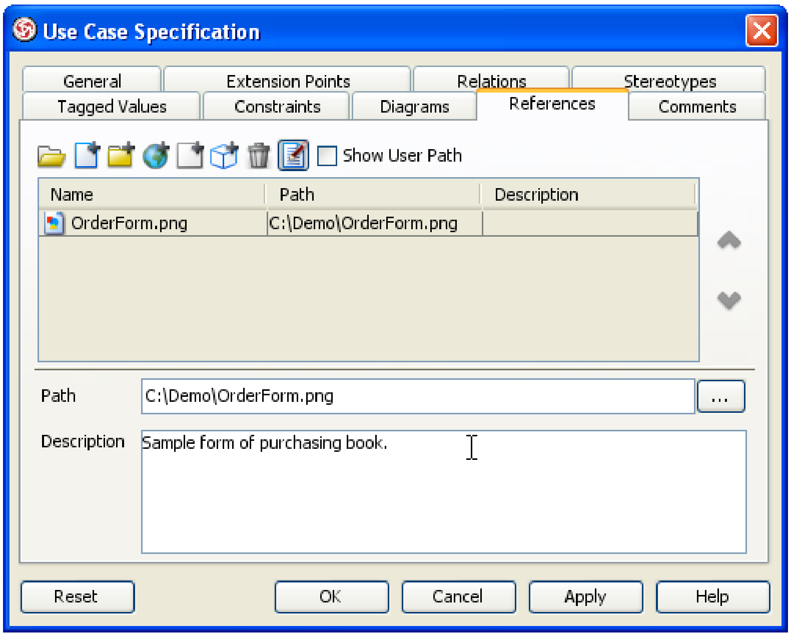


Path: C:\Demo\OrderForm.png

Description:

Figure 5-4 *Inputting file reference path*

5. Input description of the file.



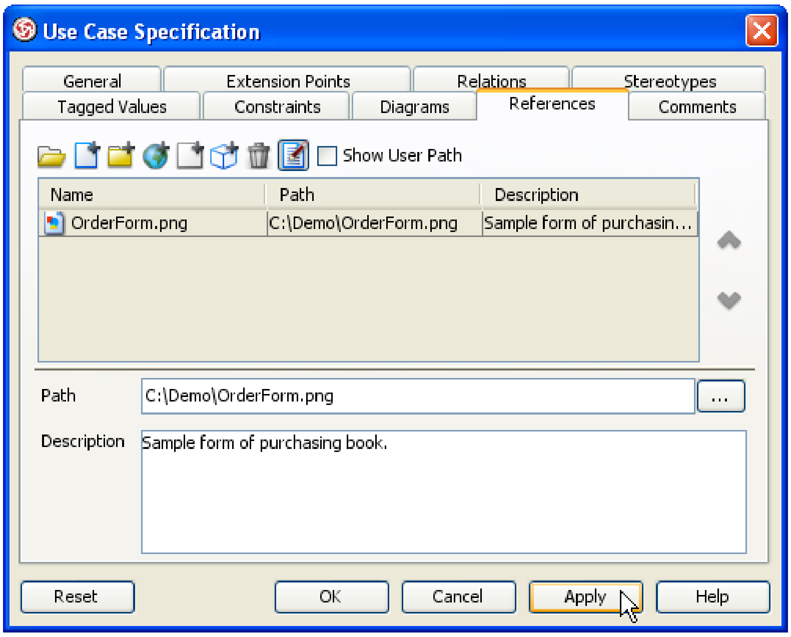
Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	

Path: C:\Demo\OrderForm.png

Description: Sample form of purchasing book.

Figure 5-5 Inputting file reference description

6. Press Apply button to confirm the reference creation.



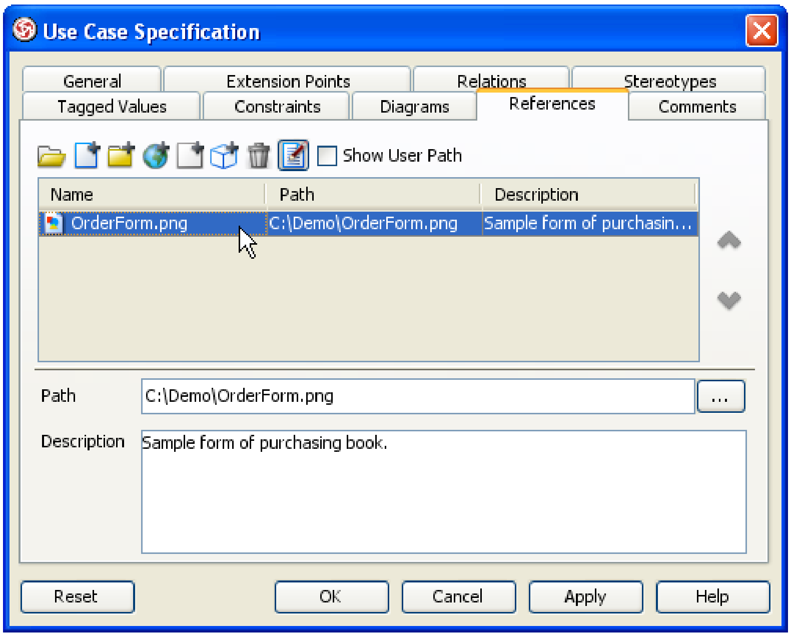
Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	Sample form of purchasin...

Path: C:\Demo\OrderForm.png

Description: Sample form of purchasing book.

Figure 5-6 Apply file reference

7. Select the file reference to be opened.



□ Show User Path

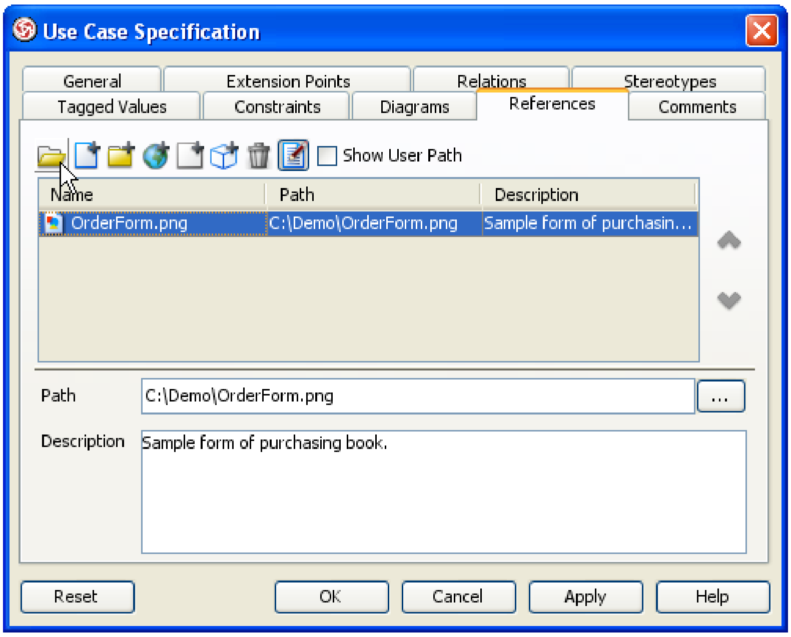
Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	Sample form of purchasin...

Path: C:\Demo\OrderForm.png

Description: Sample form of purchasing book.

Figure 5-7 *Select file reference*

8. Press Open... button.



□ Show User Path

Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	Sample form of purchasin...

Path: C:\Demo\OrderForm.png

Description: Sample form of purchasing book.

Figure 5-8 *Open file reference*

Reference to Folder

1. Mouse over shape and click on References resource.

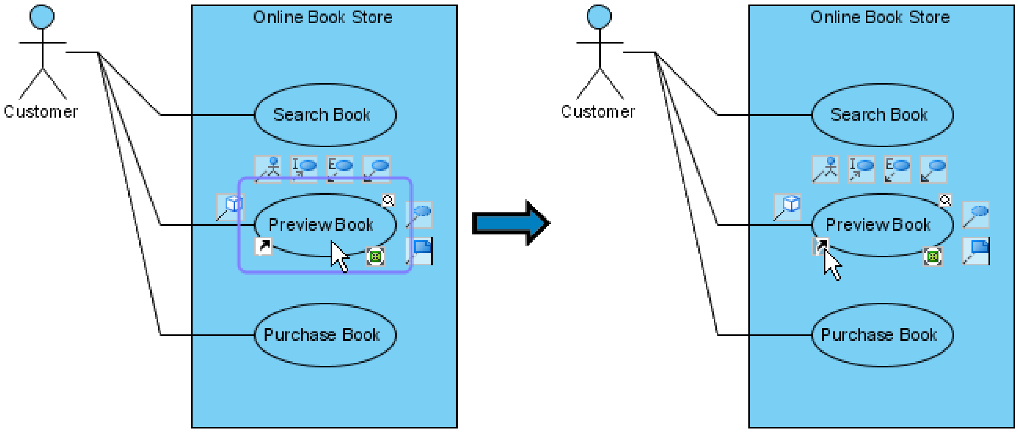


Figure 5-9 *Mouse over references resource*

2. Click on Add Folder... menu item.

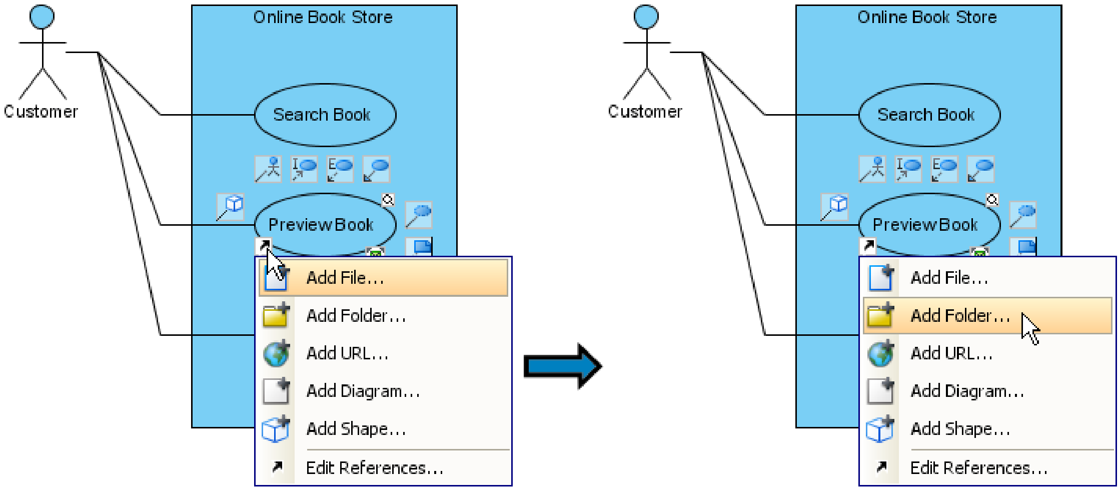


Figure 5-10 *Add folder from resource*

3. Press ... button.

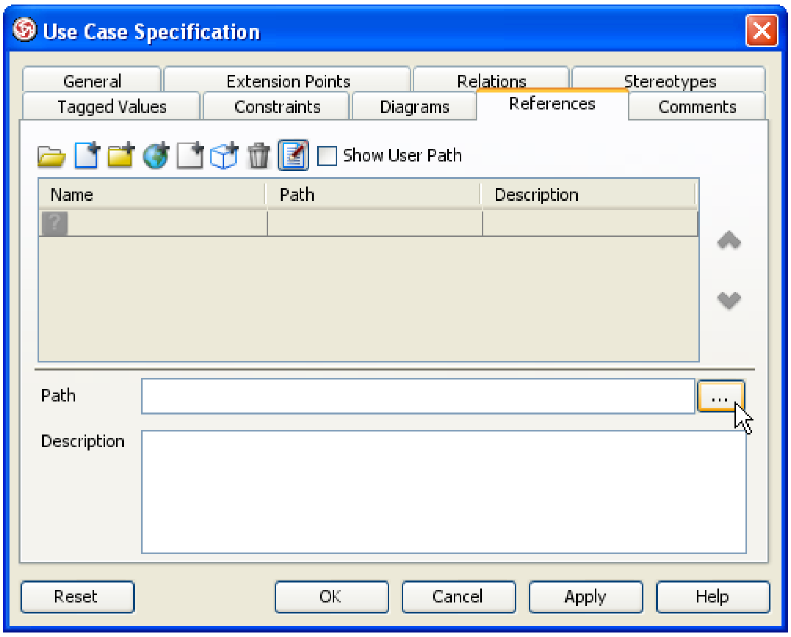
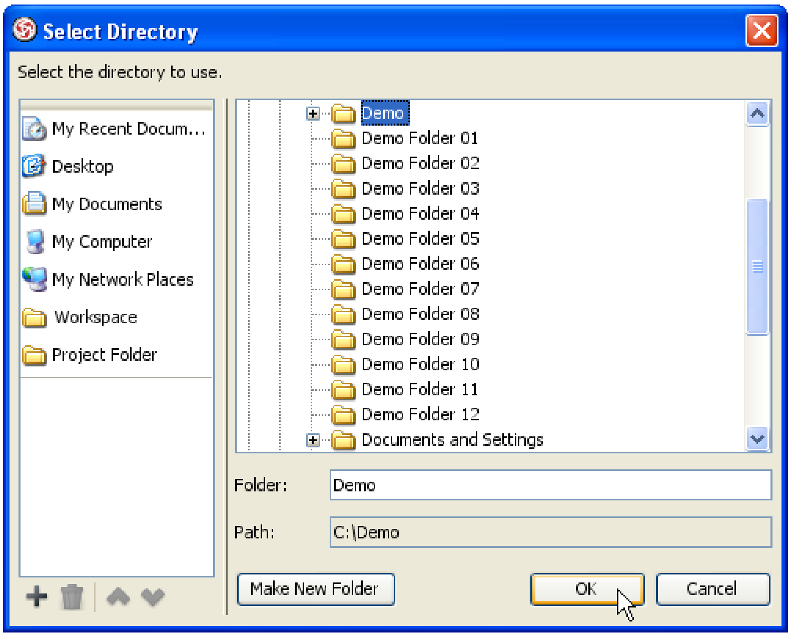


Figure 5-11 *Browse folder*

4. Select folder to be referenced.

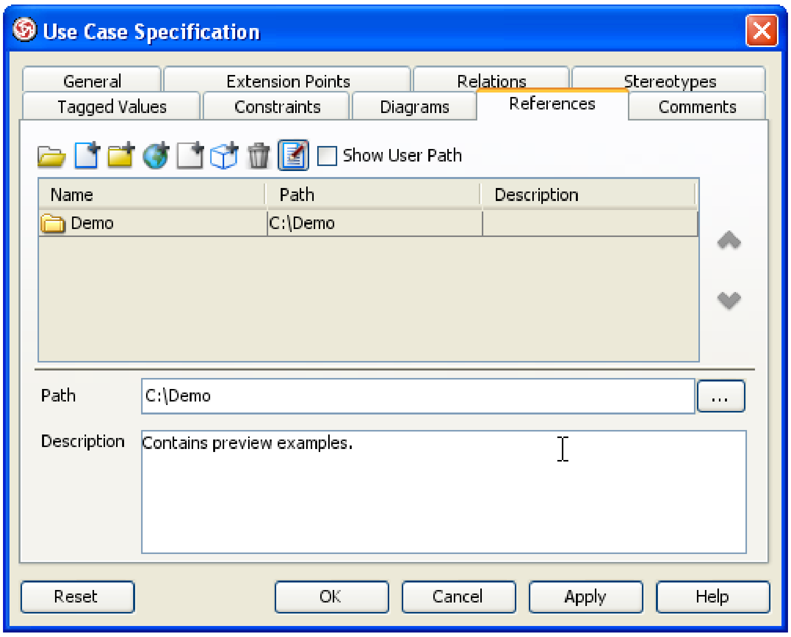


Folder: Demo

Path: C:\Demo

***Figure 5-12** Select reference folder*

5. Input description of the folder.



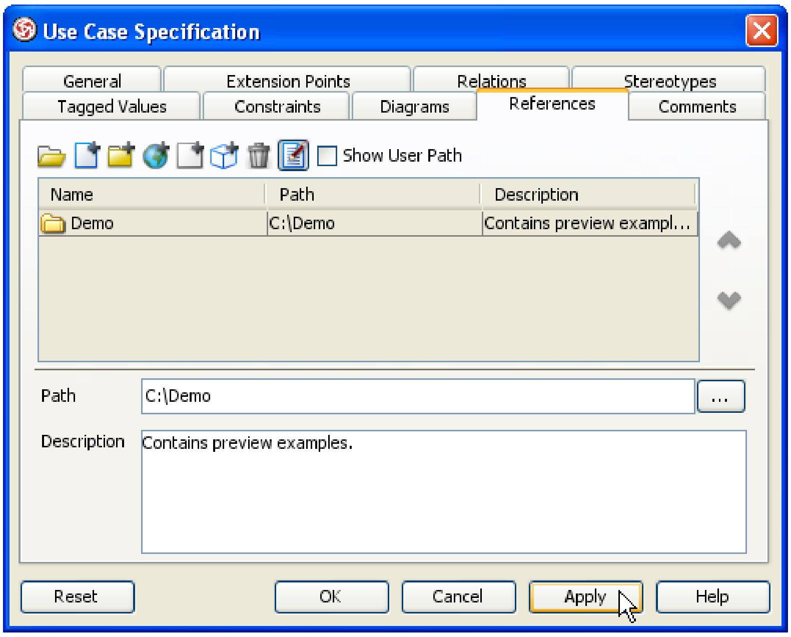
Name	Path	Description
Demo	C:\Demo	

Path: C:\Demo

Description: Contains preview examples.

***Figure 5-13** Inputting folder reference description*

6. Press Apply button to confirm the reference creation.



Use Case Specification

General | Extension Points | Relations | Stereotypes

Tagged Values | Constraints | Diagrams | References | Comments

□ Show User Path

Name	Path	Description
Demo	C:\Demo	Contains preview exampl...

Path: C:\Demo ...

Description: Contains preview examples.

Reset | OK | Cancel | Apply | Help

Figure 5-14 Apply folder reference

7. Mouse over shape and click on References resource.

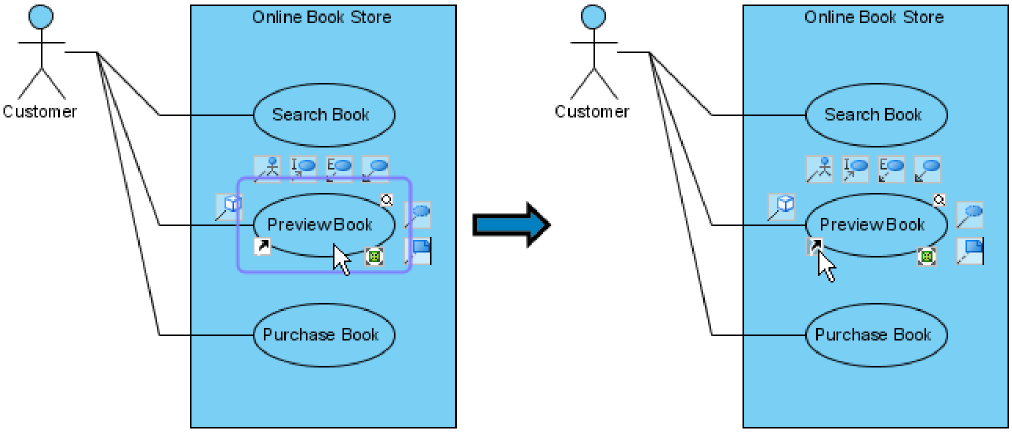


Figure 5-15 Mouse over references resource

8. Click on folder resource menu item.

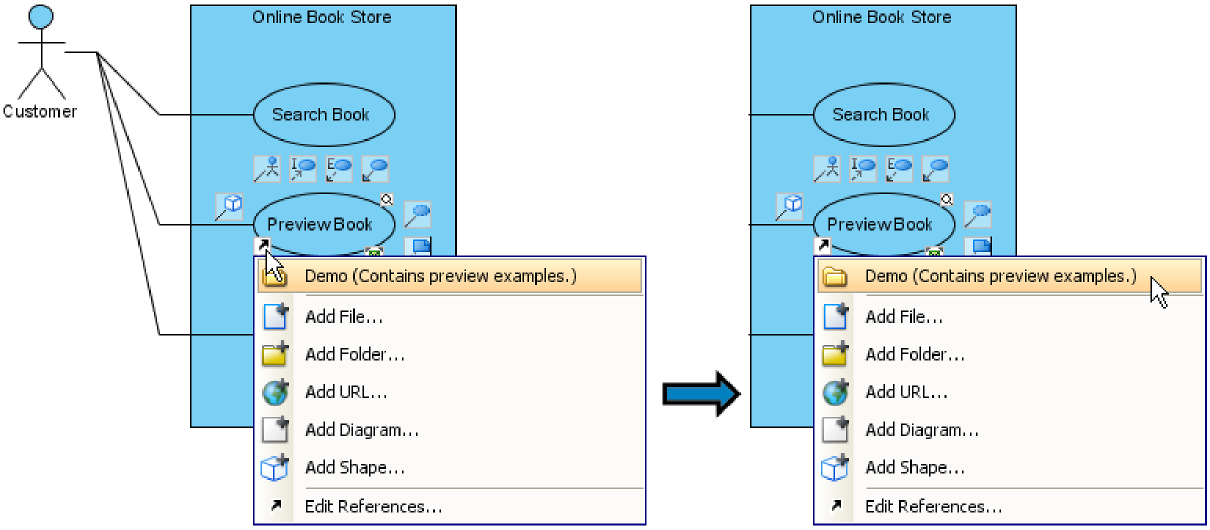


Figure 5-16 Open folder reference

Reference to URL

1. Right click on shape and click on Open Specification... menu item.

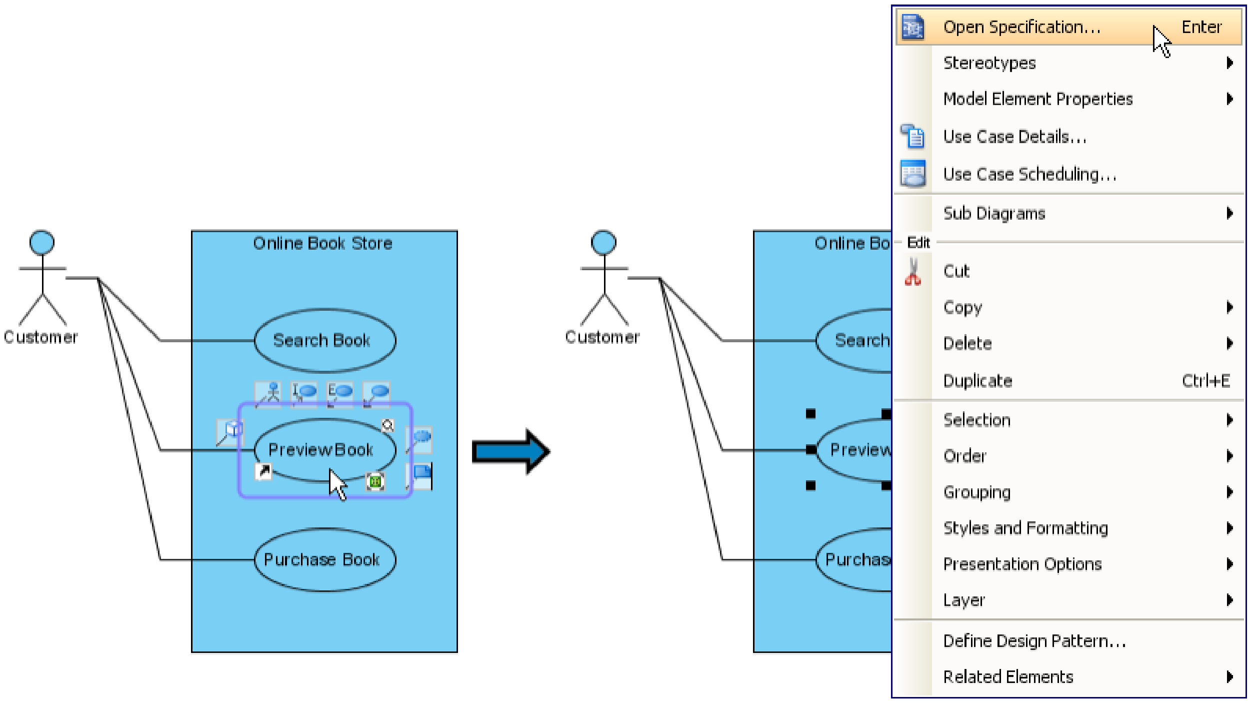
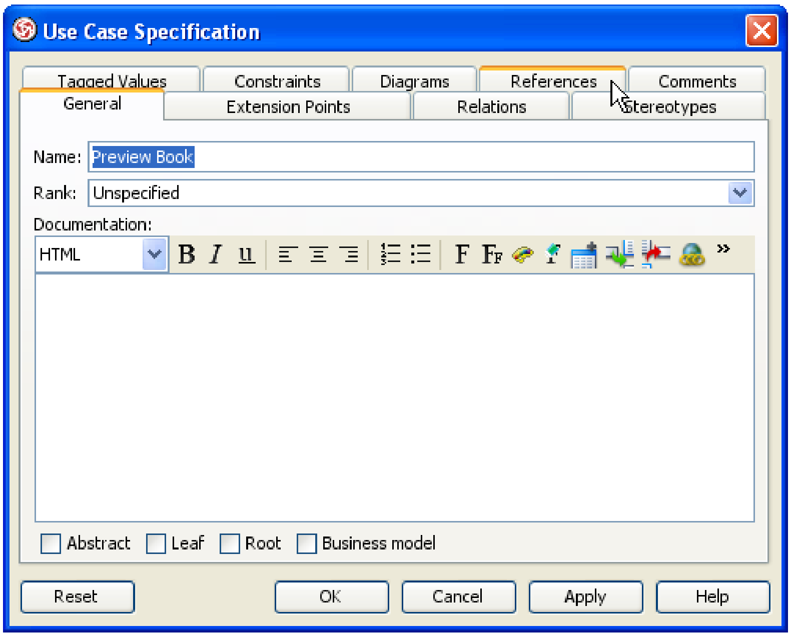


Figure 5-17 *Show popup of shape*

2. Switch to References tab.



Name: Preview Book

Rank: Unspecified

□ Abstract □ Leaf □ Root □ Business model

Figure 5-18 *Switch to references tab*

3. Press Add URL... button.

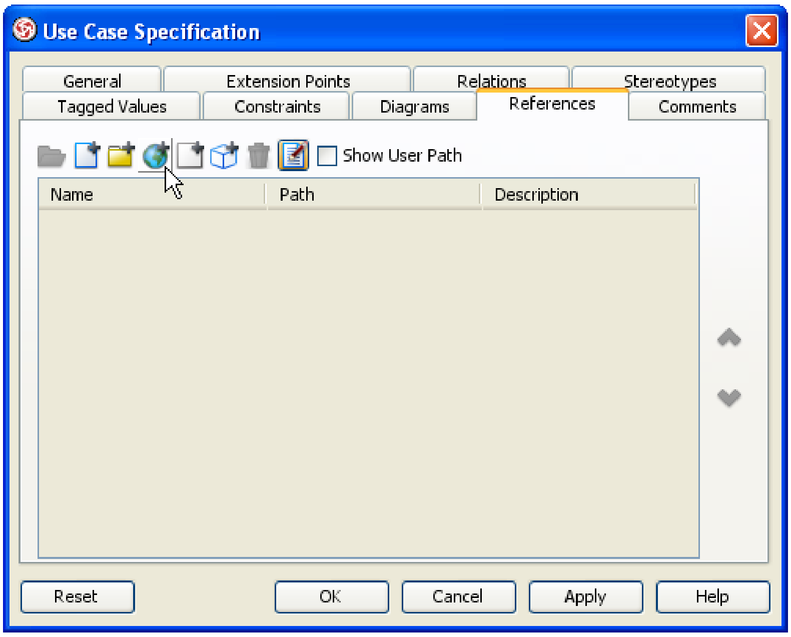
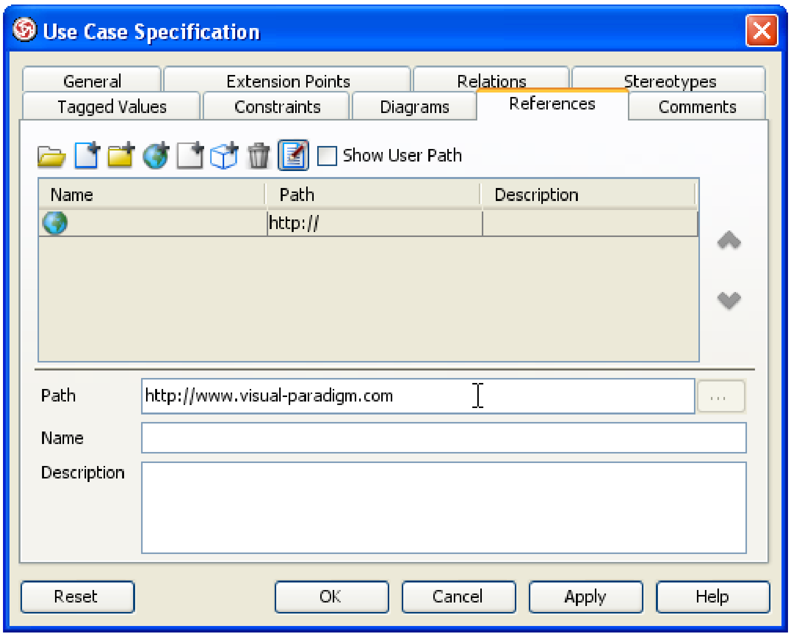


Figure 5-19 Adding URL reference

4. Input path of URL.



Name | Path | Description
http://

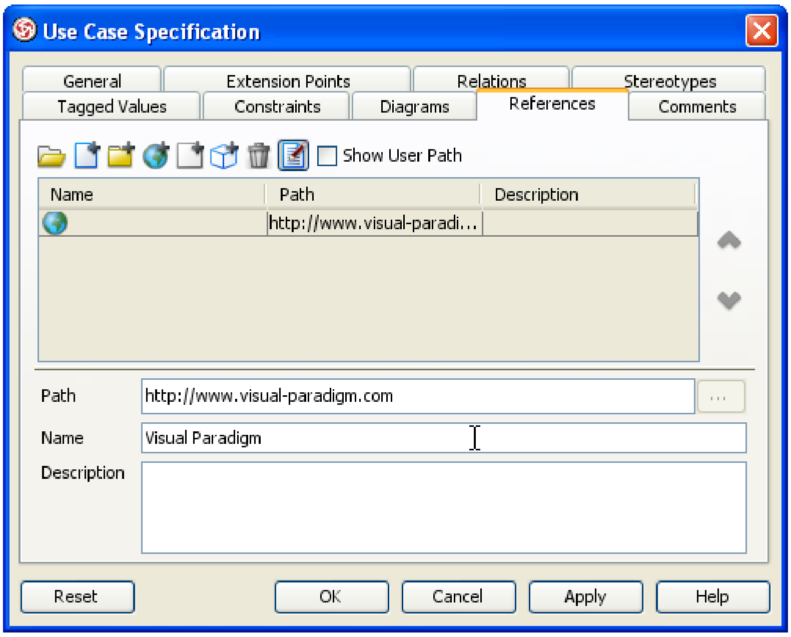
Path: http://www.visual-paradigm.com

Name

Description

Figure 5-20 Inputting URL reference path

5. Input URL name.



Name	Path	Description
	http://www.visual-paradi...	

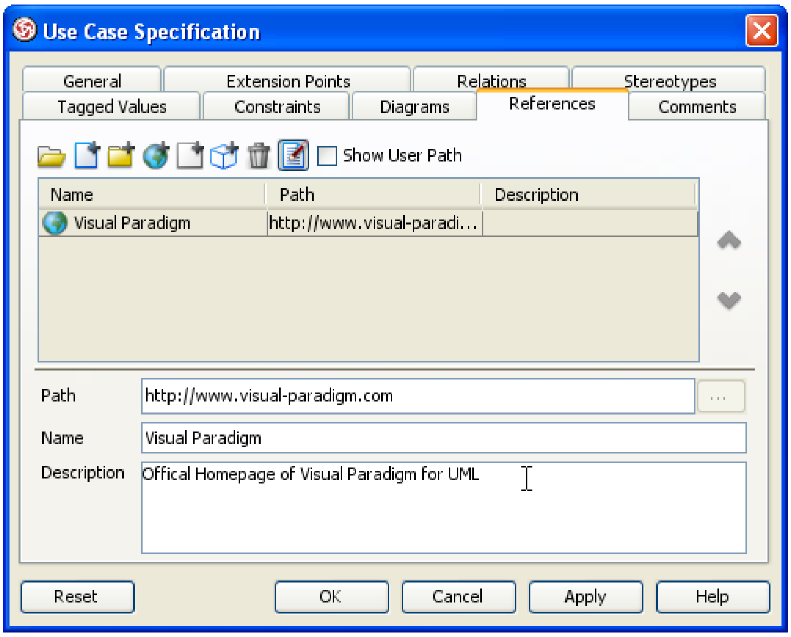
Path: http://www.visual-paradigm.com

Name: Visual Paradigm

Description:

Figure 5-21 *Inputting URL reference name*

6. Input description of the URL.



Name	Path	Description
Visual Paradigm	http://www.visual-paradi...	

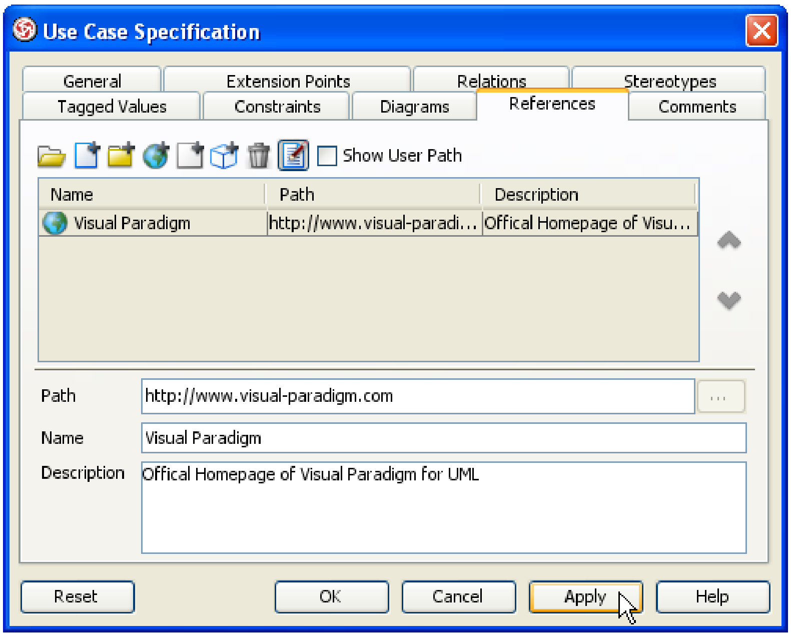
Path: http://www.visual-paradigm.com

Name: Visual Paradigm

Description: Offical Homepage of Visual Paradigm for UML

Figure 5-22 *Inputting URL reference description*

7. Press Apply button to confirm the reference creation.



☐ Show User Path

Name	Path	Description
Visual Paradigm	http://www.visual-paradi...	Offical Homepage of Visu...

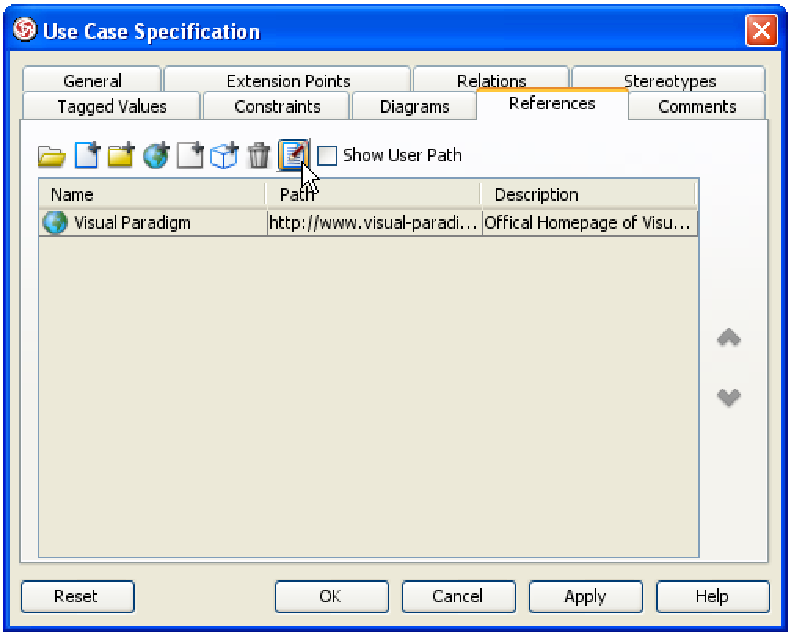
Path: http://www.visual-paradigm.com

Name: Visual Paradigm

Description: Offical Homepage of Visual Paradigm for UML

Figure 5-23 *Apply URL reference*

8. Press Details button.

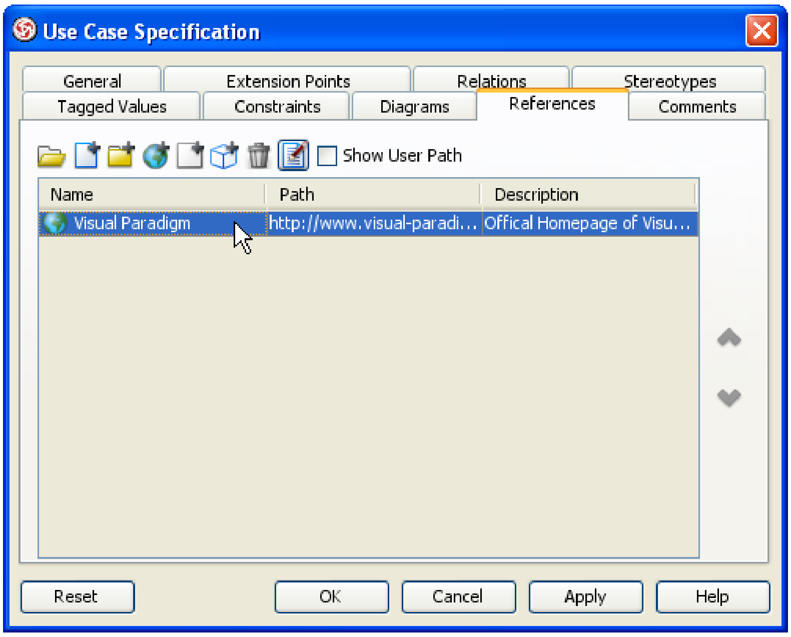


☐ Show User Path

Name	Path	Description
Visual Paradigm	http://www.visual-paradi...	Offical Homepage of Visu...

Figure 5-24 *Hide reference details*

9. Select the URL reference to be opened.

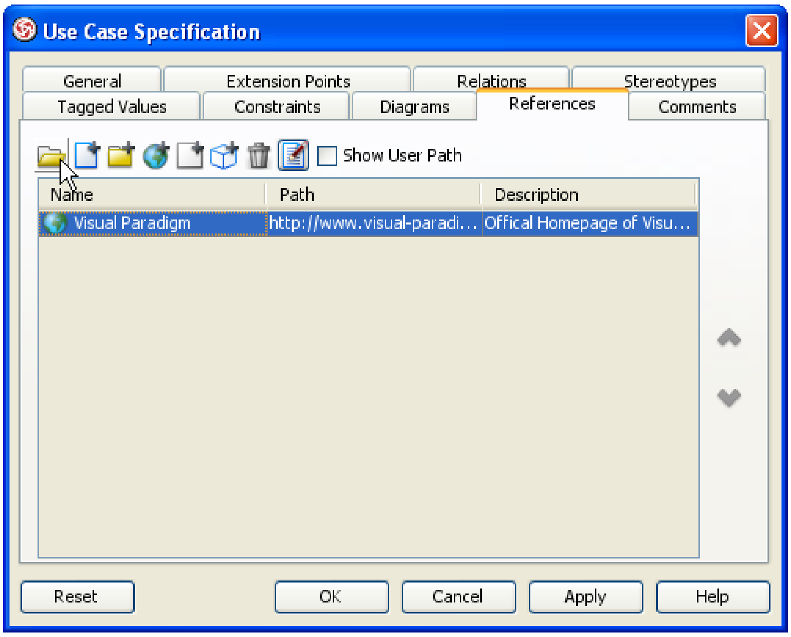


☐ Show User Path

Name	Path	Description
Visual Paradigm	http://www.visual-paradi...	Offical Homepage of Visu...

Figure 5-25 Select URL reference

10. Press Open... button.



☐ Show User Path

Name	Path	Description
Visual Paradigm	http://www.visual-paradi...	Offical Homepage of Visu...

Figure 5-26 Open URL reference

Reference to Diagrams and Shapes

Reference to Diagrams

1. Mouse over shape and click on References resource.

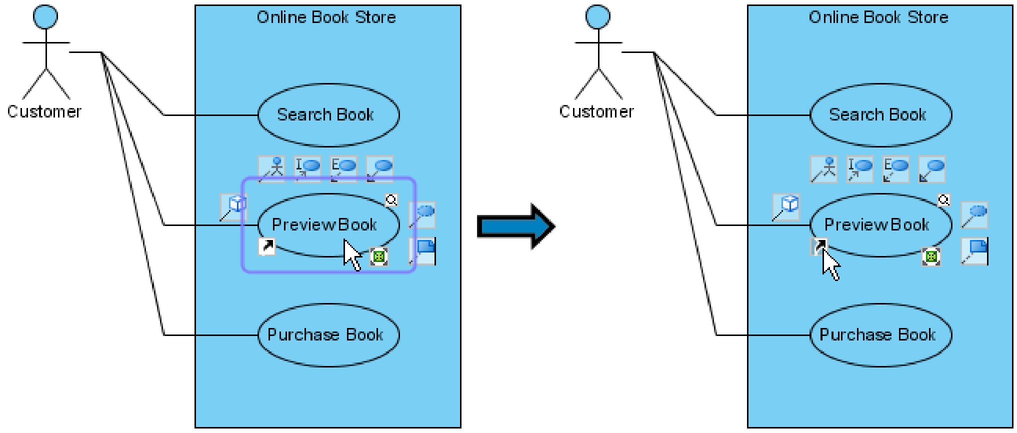


Figure 5-27 *Mouse over references resource*

2. Click on Edit References... menu item.

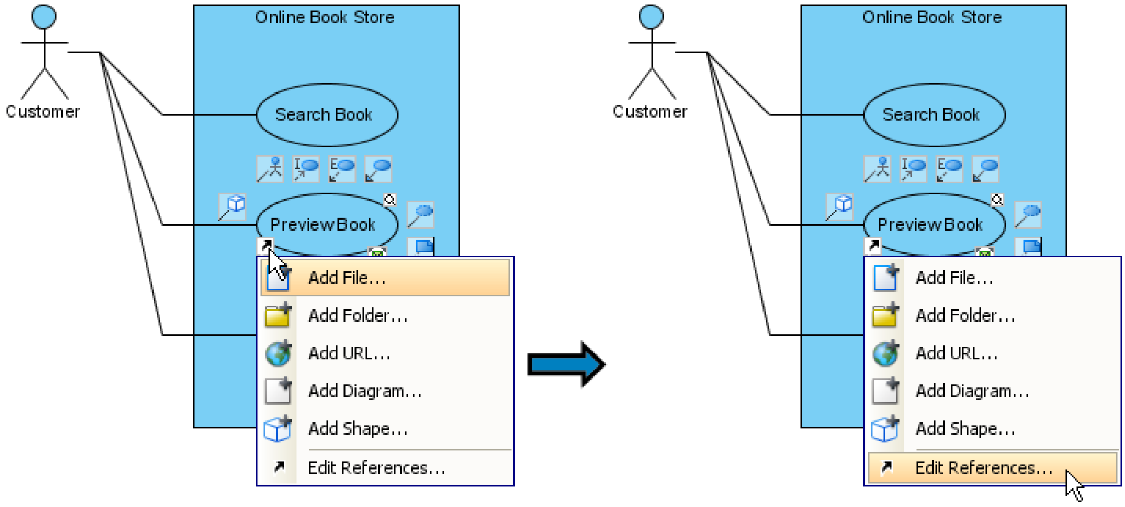


Figure 5-28 *Edit references from resource*

3. Press Add Diagram... button.

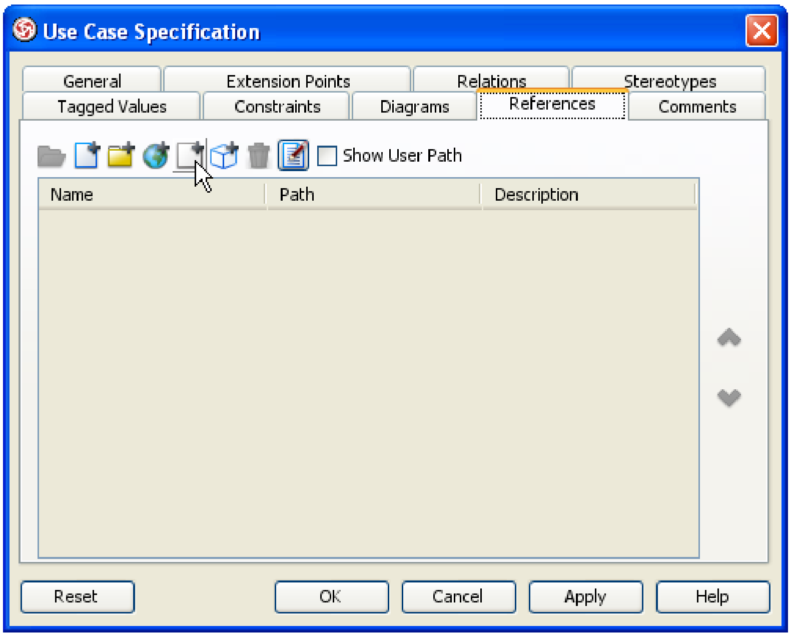


Figure 5-29 *Adding diagram reference*

4. Select reference diagram.

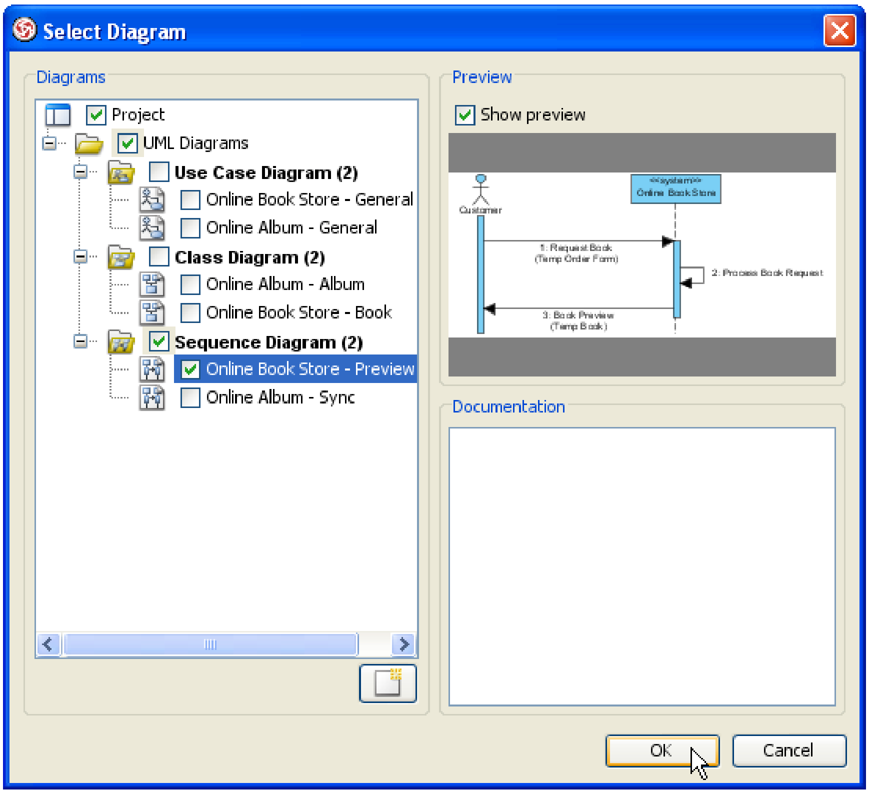
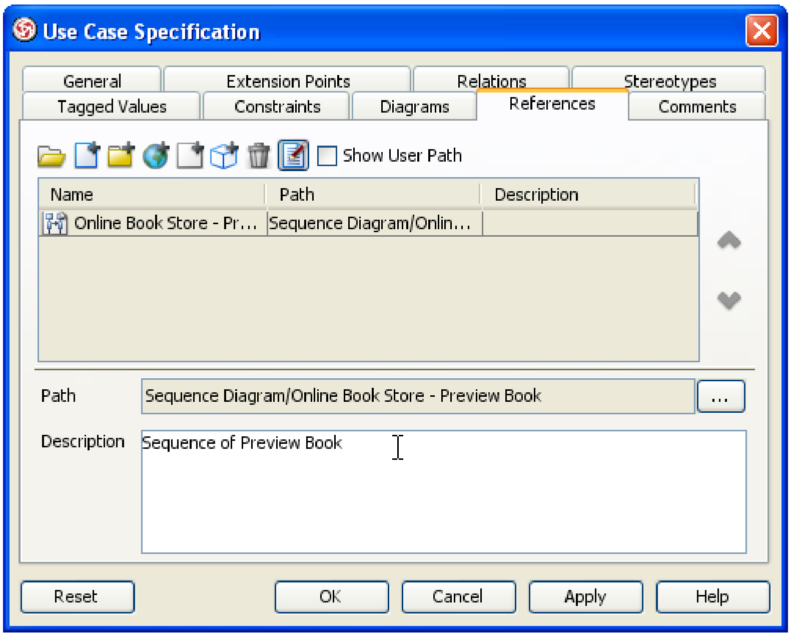


Figure 5-30 *Selecting reference diagram*

5. Input description of the diagram.



□ Show User Path

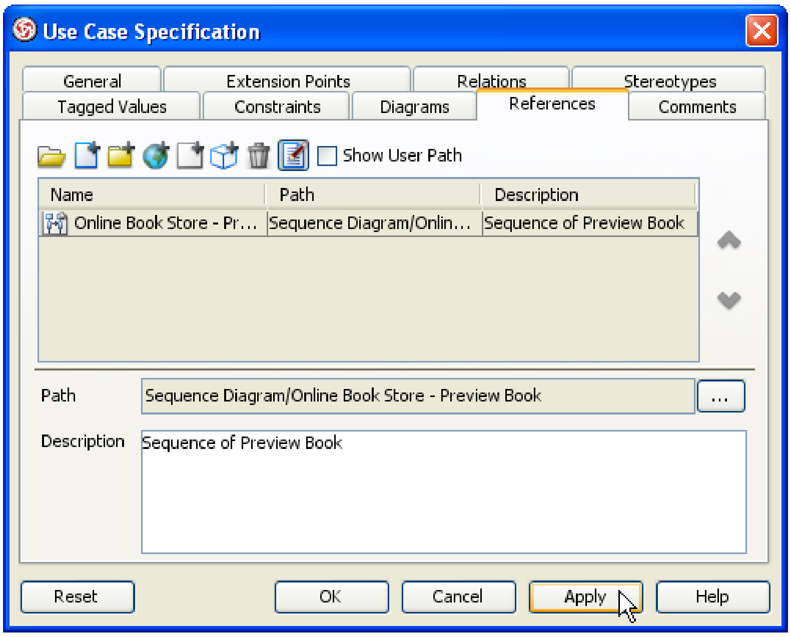
Name	Path	Description
Online Book Store - Pr…	Sequence Diagram/Onlin…	

Path: Sequence Diagram/Online Book Store - Preview Book

Description: Sequence of Preview Book

Figure 5-31 *Inputting diagram reference description*

6. Press Apply button to confirm the reference creation.



□ Show User Path

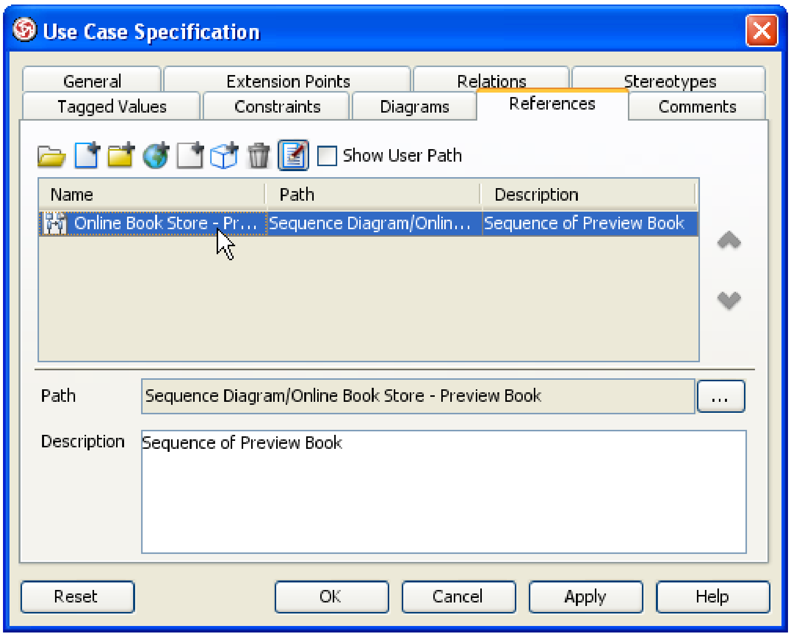
Name	Path	Description
Online Book Store - Pr…	Sequence Diagram/Onlin…	Sequence of Preview Book

Path: Sequence Diagram/Online Book Store - Preview Book

Description: Sequence of Preview Book

Figure 5-32 *Apply diagram reference*

7. Select the diagram reference to be opened.



□ Show User Path

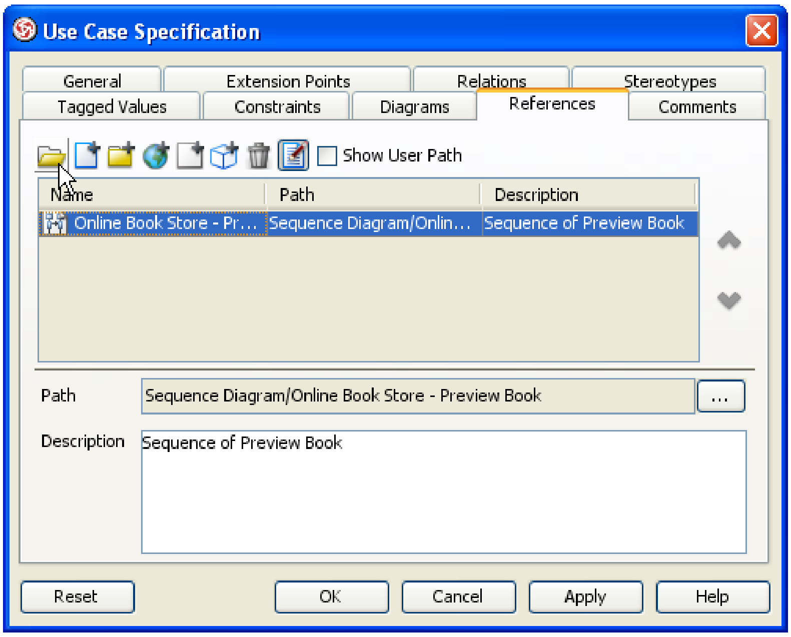
Name	Path	Description
Online Book Store - Pr...	Sequence Diagram/Onlin...	Sequence of Preview Book

Path: Sequence Diagram/Online Book Store - Preview Book ...

Description: Sequence of Preview Book

Figure 5-33 *Select diagram reference*

8. Press Open... button.



□ Show User Path

Name	Path	Description
Online Book Store - Pr...	Sequence Diagram/Onlin...	Sequence of Preview Book

Path: Sequence Diagram/Online Book Store - Preview Book ...

Description: Sequence of Preview Book

Figure 5-34 *Open diagram reference*

Reference to Shapes

1. Mouse over shape and click on References resource.

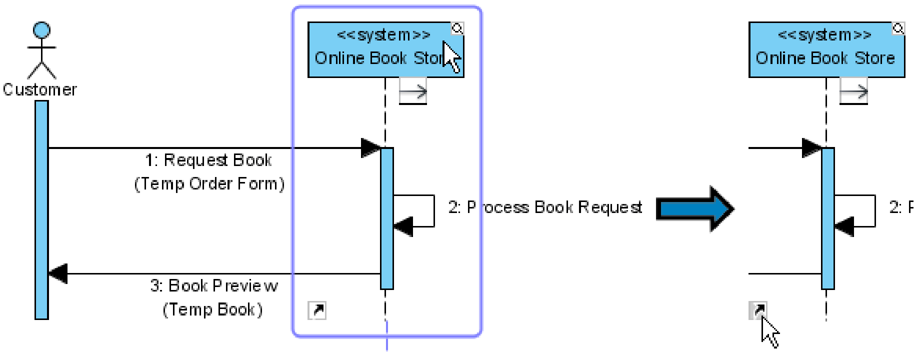


Figure 5-35 Mouse over references resource

2. Click on Add Shape... menu item.

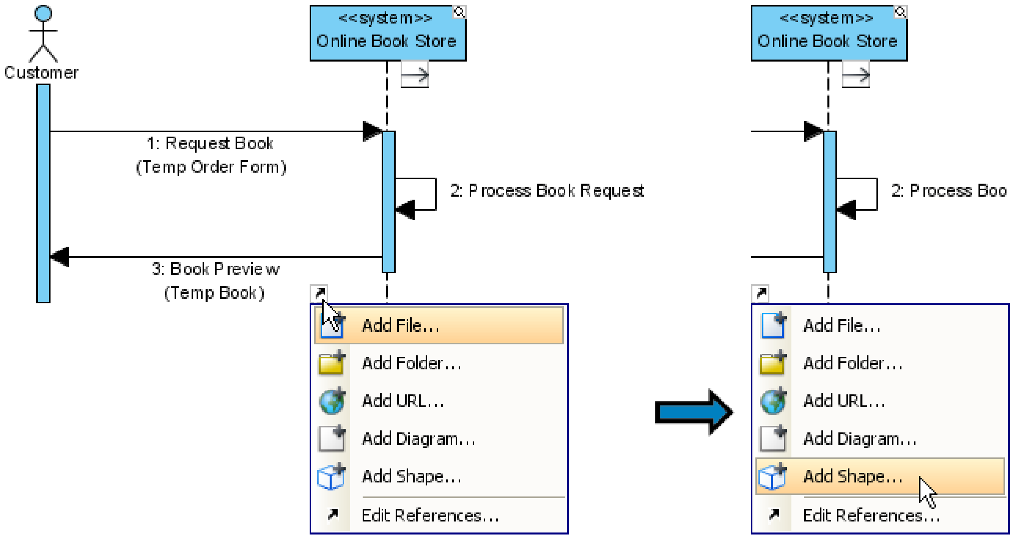


Figure 5-36 Add shape from resource

3. Select shape to be referenced.

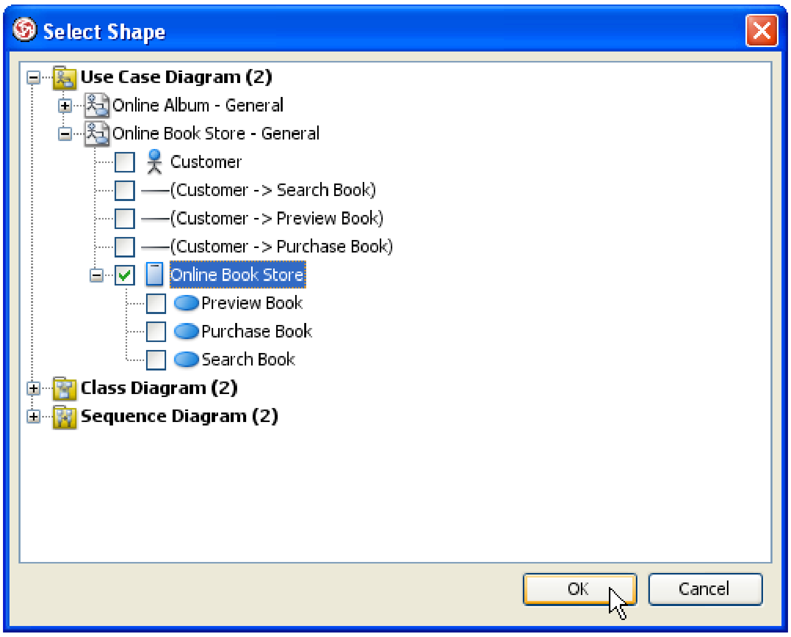
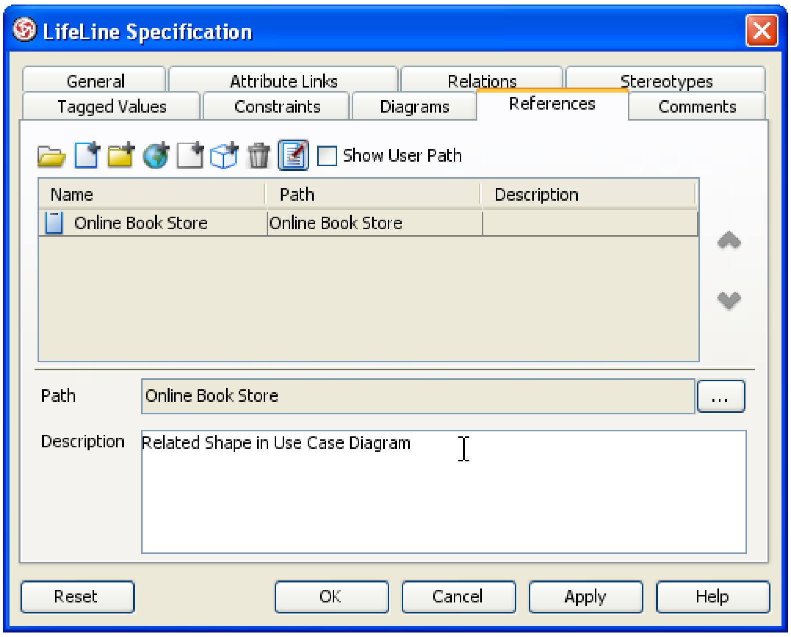


Figure 5-37 Select reference shape

4. Input description of the shape.



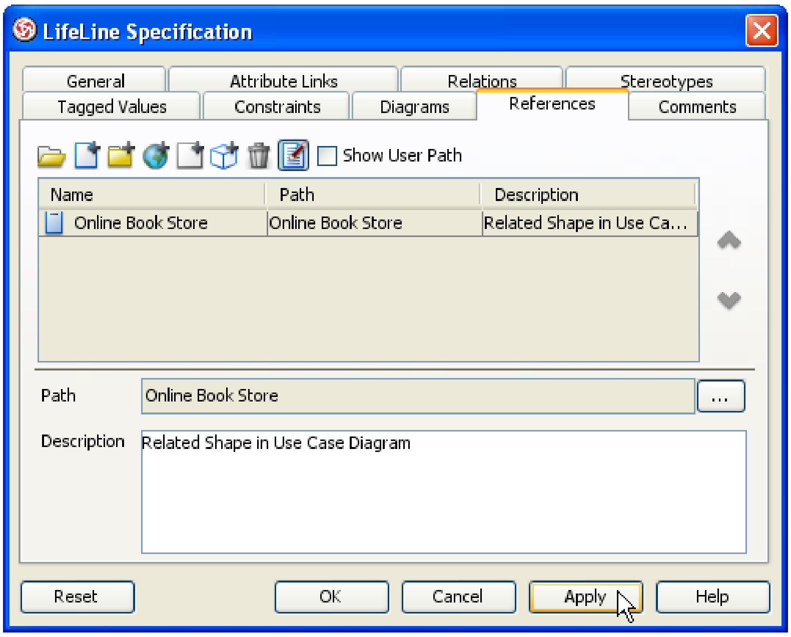
Name	Path	Description
Online Book Store	Online Book Store	

Path: Online Book Store

Description: Related Shape in Use Case Diagram

Figure 5-38 *Inputting shape reference description*

5. Press Apply button to confirm the reference creation.



Name	Path	Description
Online Book Store	Online Book Store	Related Shape in Use Ca...

Path: Online Book Store

Description: Related Shape in Use Case Diagram

Figure 5-39 *Apply folder reference*

6. Mouse over shape and click on References resource.

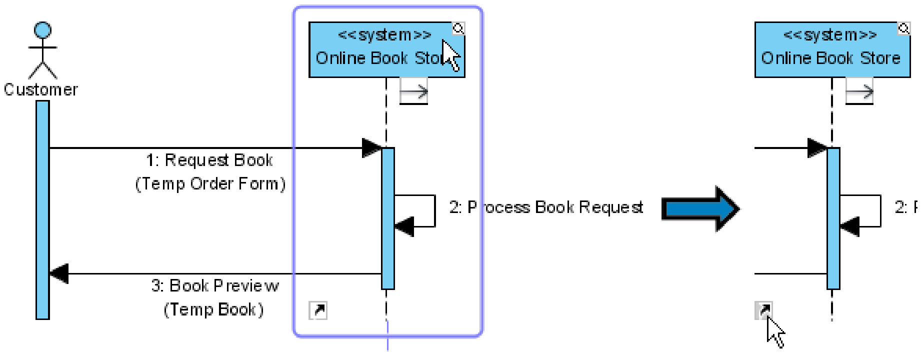


Figure 5-40 *Mouse over references resource*

7. Click on shape reference menu item.

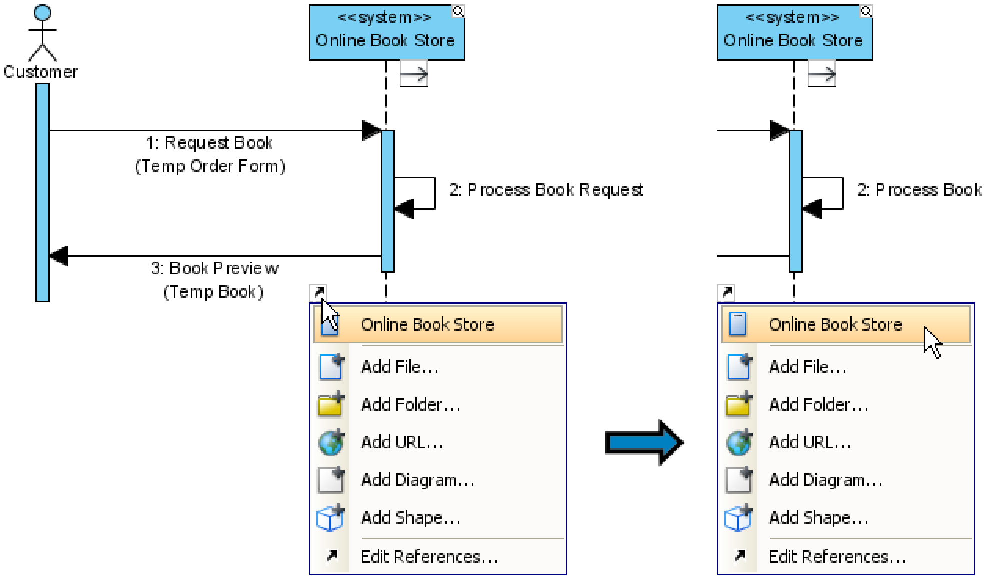


Figure 5-41 *Open shape reference*

Managing References

Edit References

1. Mouse over shape and click on References resource.

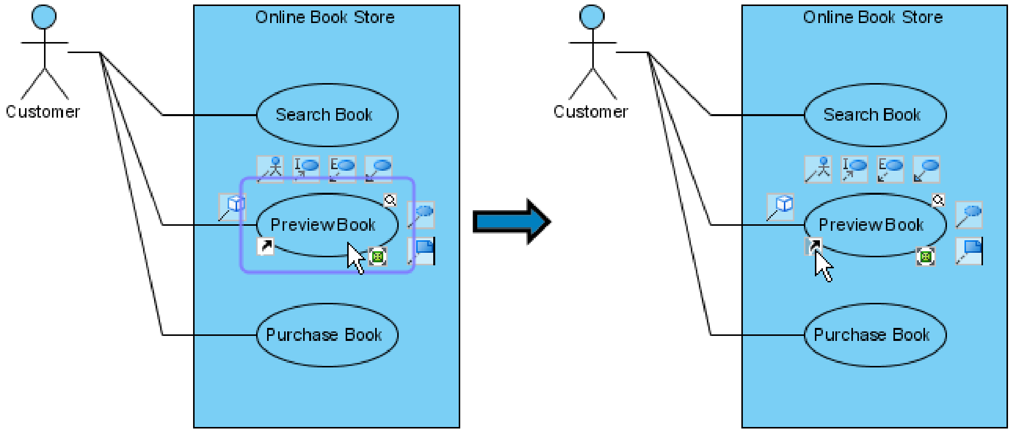


__Figure 5-42__ Mouse over references resource

2. Click on Edit References... menu item.

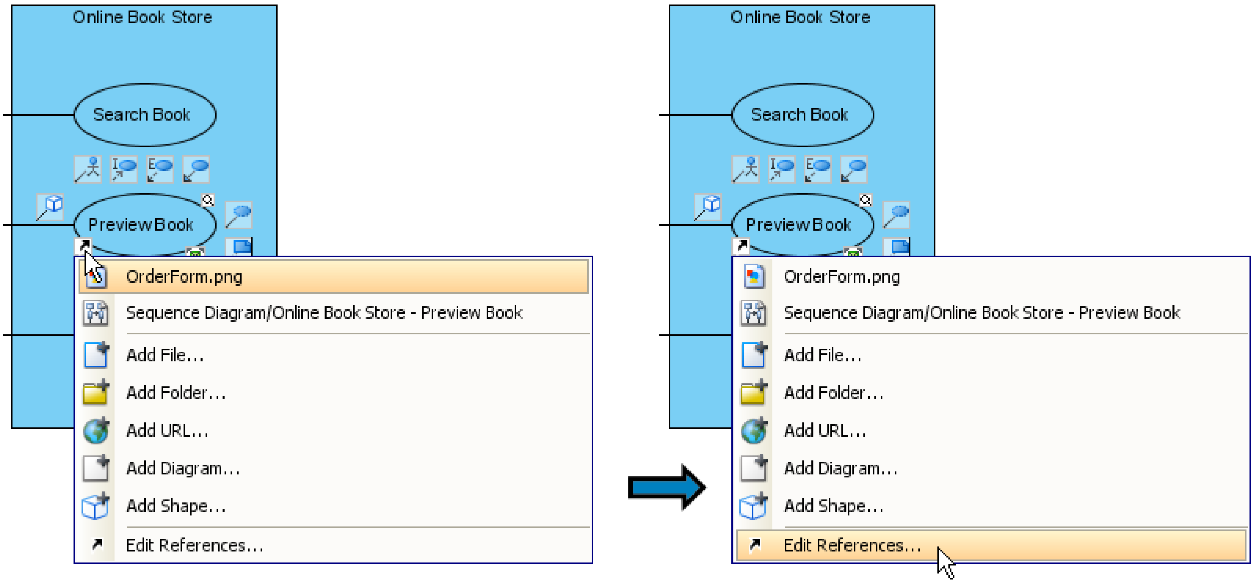
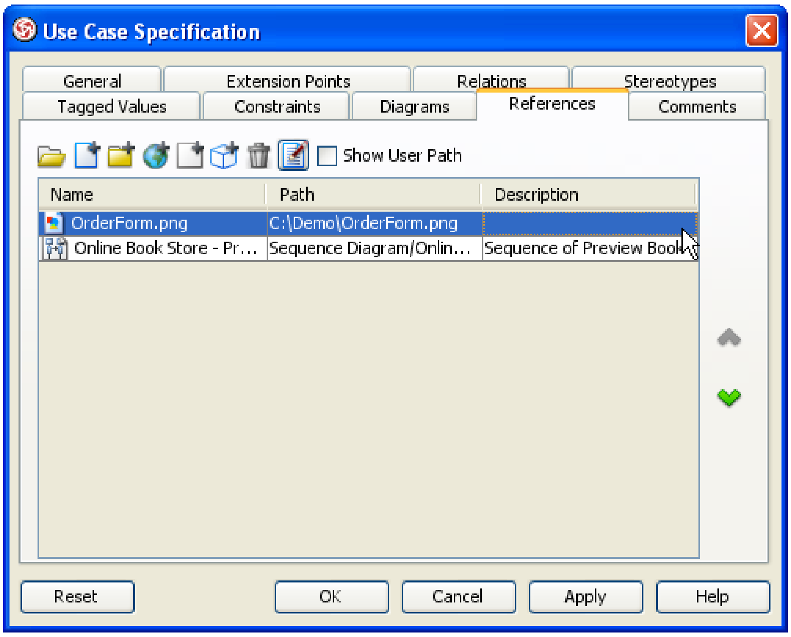


__Figure 5-43__ Edit references from resource

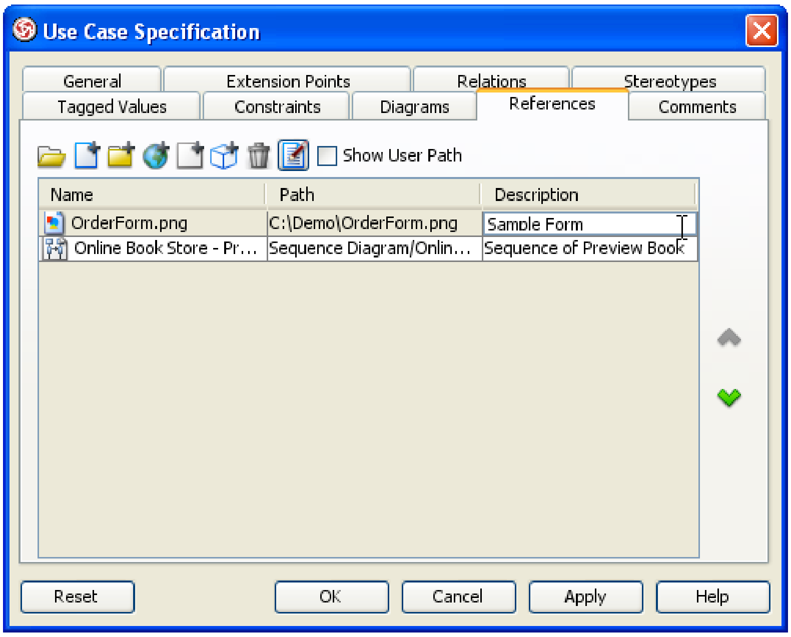
3. Double-click description of file reference.



Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	
Online Book Store - Pr…	Sequence Diagram/Onlin…	Sequence of Preview Book

Figure 5-44 Double click file reference description

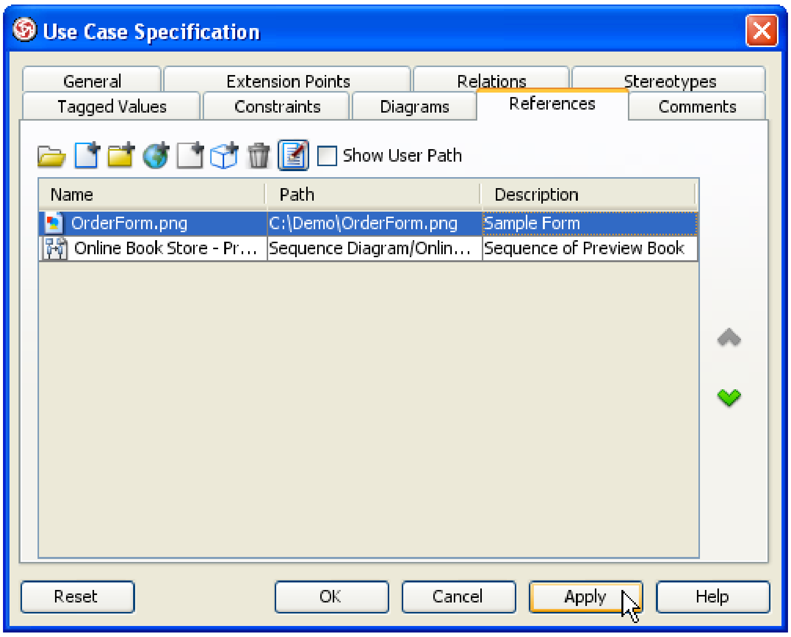
4. Input description of reference file.



Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	Sample Form
Online Book Store - Pr…	Sequence Diagram/Onlin…	Sequence of Preview Book

Figure 5-45 Inputting file reference description

5. Press Enter to confirm description editing.



Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	Sample Form
Online Book Store - Pr...	Sequence Diagram/Onlin...	Sequence of Preview Book

***Figure 5-46** Confirming description editing*

Add Reference

1. Mouse over shape and click on References resource.

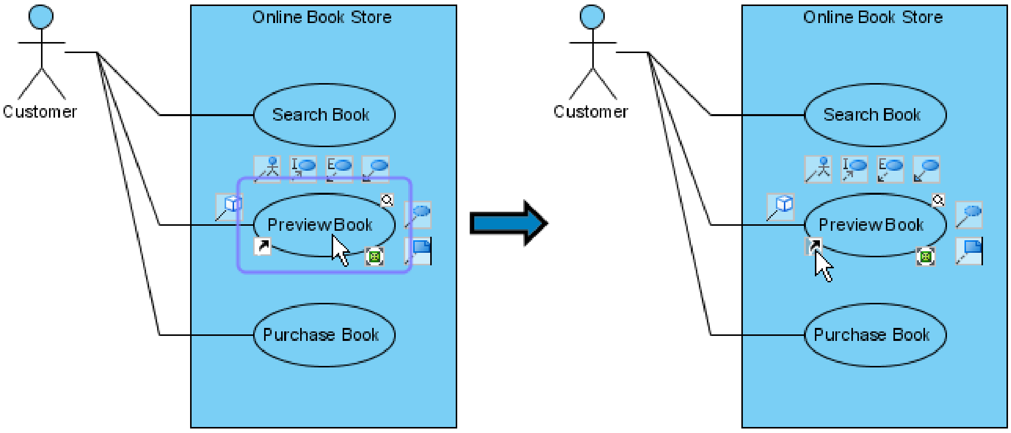


***Figure 5-47** Mouse over references resource*

2. Click on Add URL... menu item.

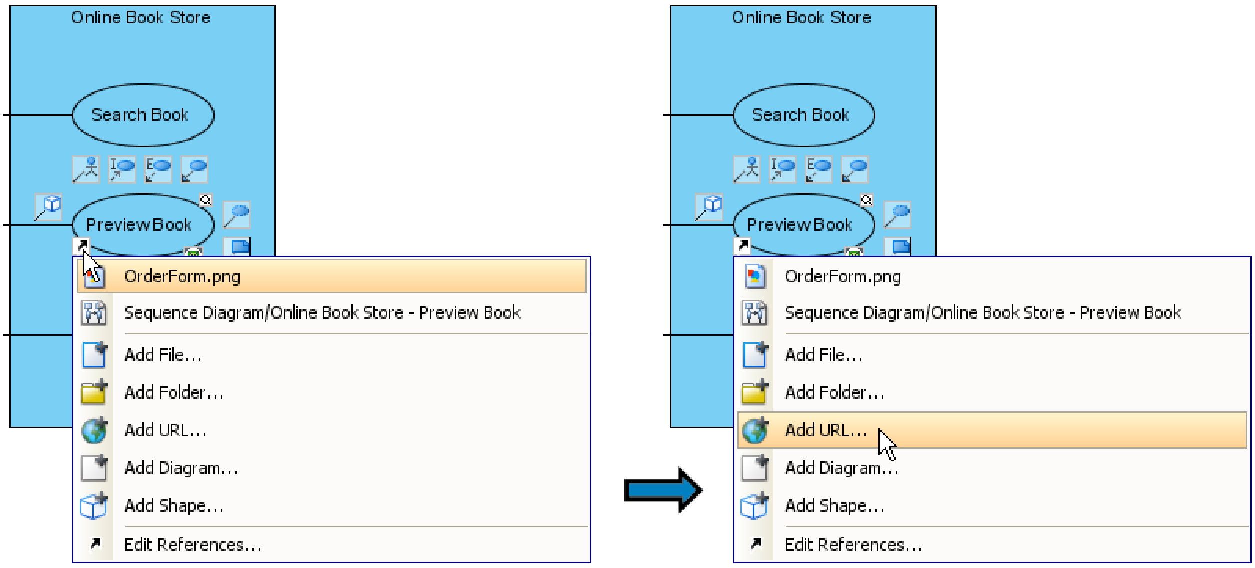
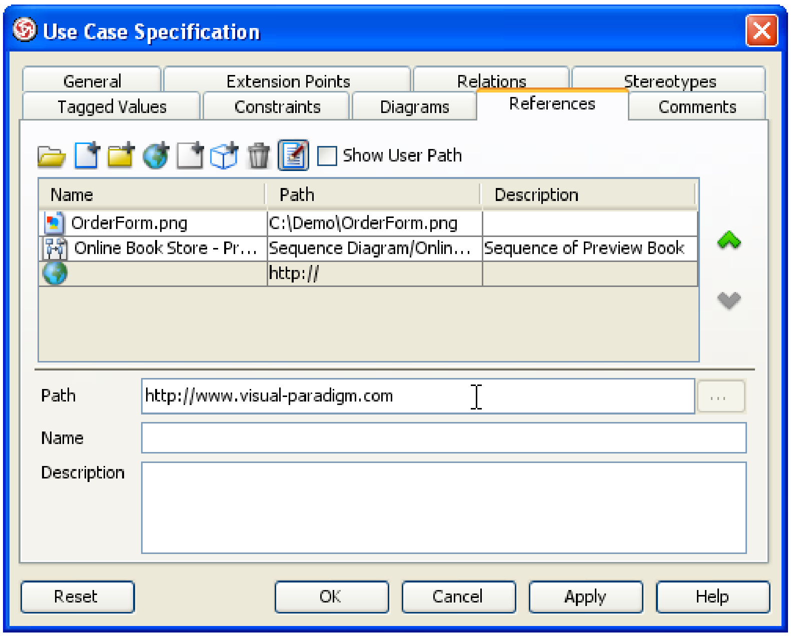


Figure 5-48 *Add URL from resource*

3. Input path of the URL.



□ Show User Path

Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	
Online Book Store - Pr...	Sequence Diagram/Onlin...	Sequence of Preview Book
	http://	

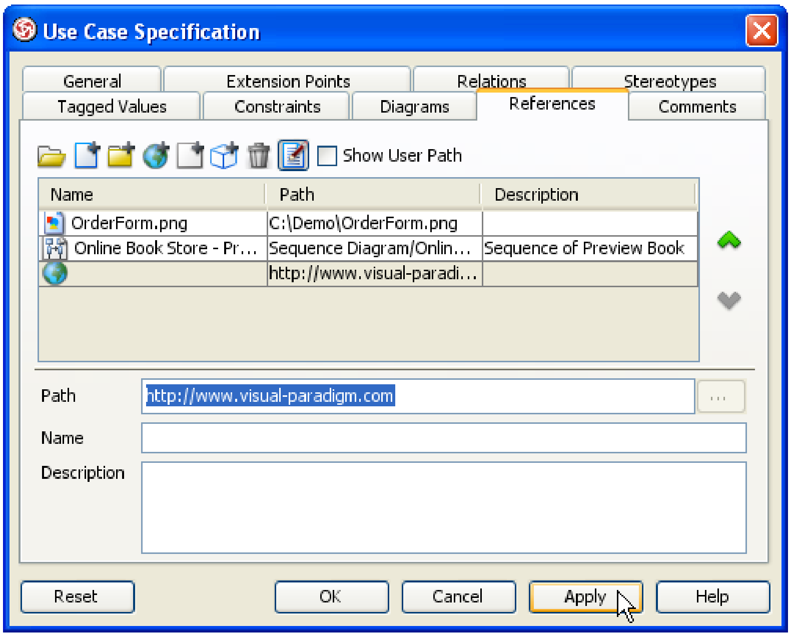
Path: http://www.visual-paradigm.com

Name:

Description:

Figure 5-49 *Inputting URL reference path*

4. Press Apply button to confirm the reference creation.



□ Show User Path

Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	
Online Book Store - Pr...	Sequence Diagram/Onlin...	Sequence of Preview Book
	http://www.visual-paradi...	

Path: http://www.visual-paradigm.com

Name:

Description:

Figure 5-50 Apply URL reference

Remove Reference

1. Mouse over shape and click on References resource.

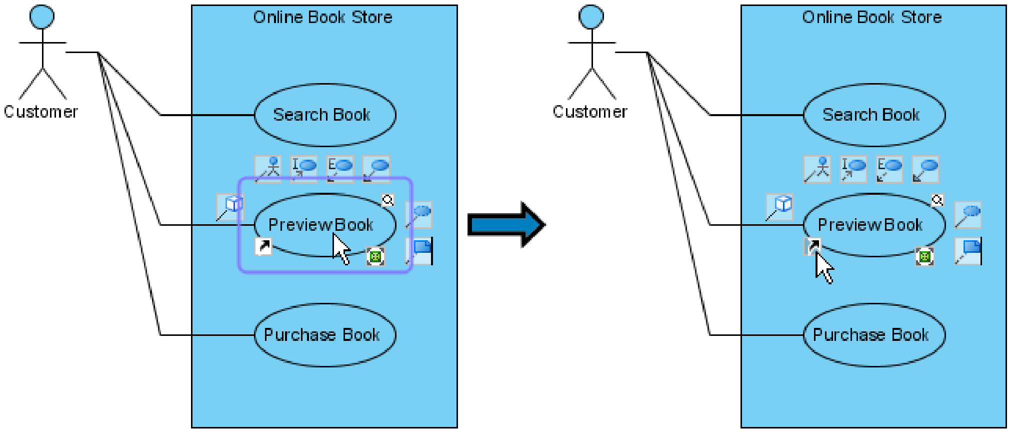


Figure 5-51 Mouse over references resource

2. Click on Edit References... menu item.

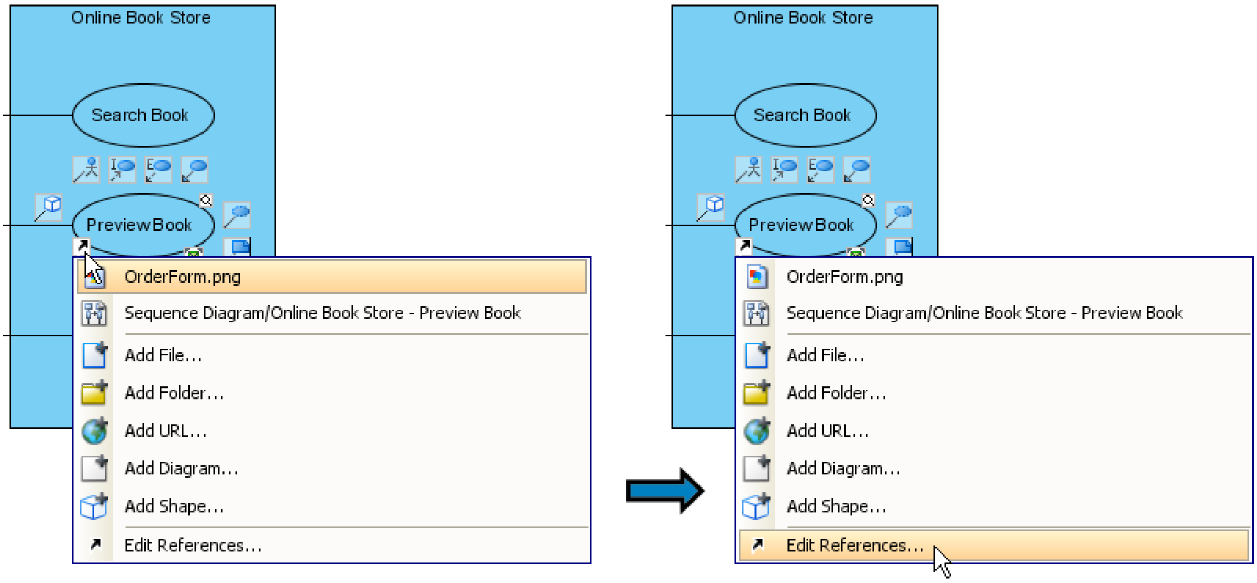
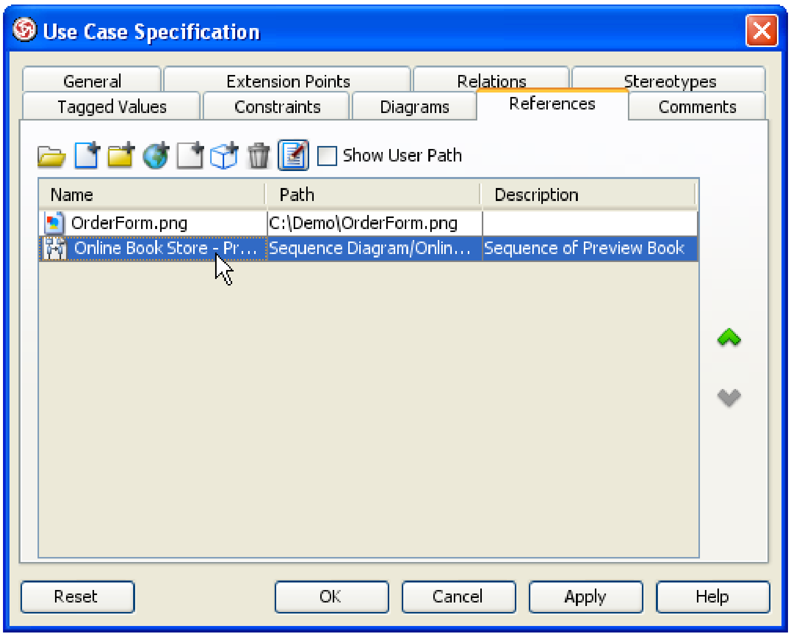


***Figure 5-52** Edit references from resource*

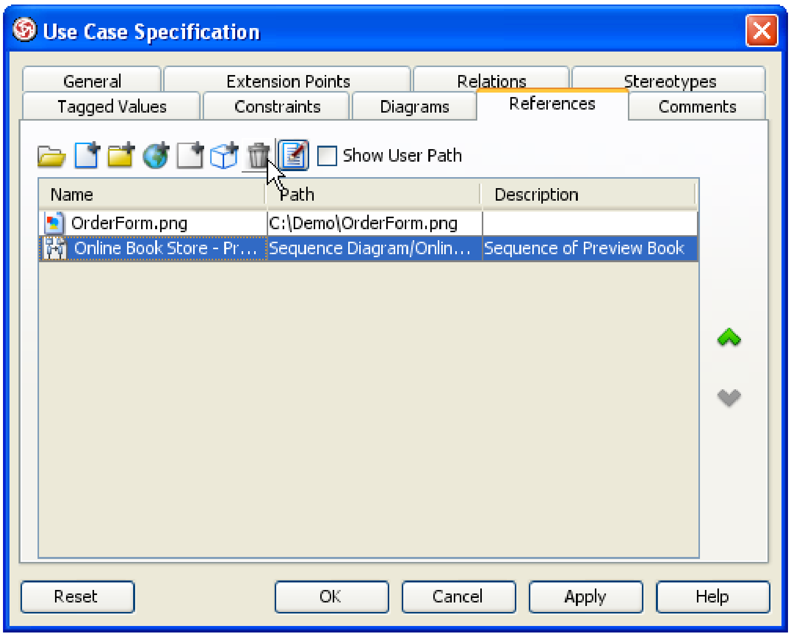
3. Select the diagram reference to be removed.



Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	
Online Book Store - Pr...	Sequence Diagram/Onlin...	Sequence of Preview Book

***Figure 5-53** Select diagram reference*

4. Press Remove button.

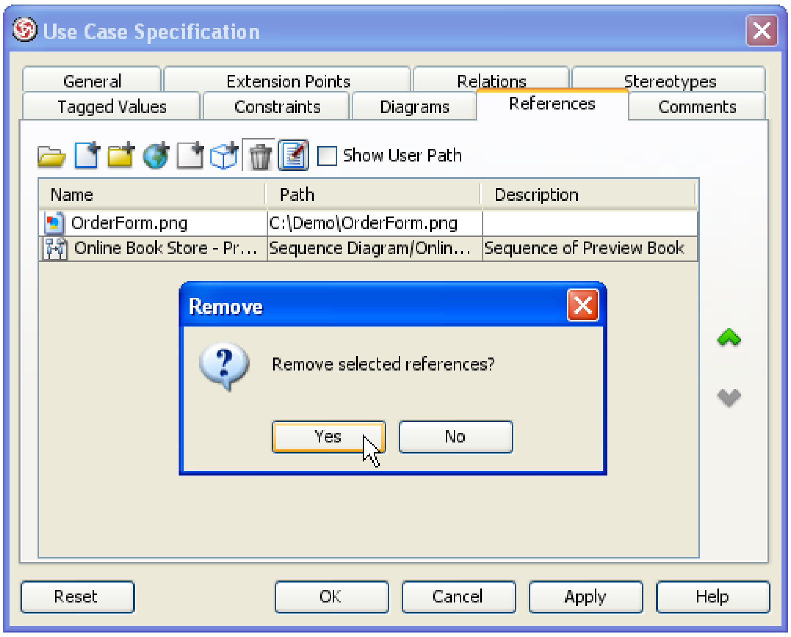


□ Show User Path

Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	
Online Book Store - Pr…	Sequence Diagram/Onlin…	Sequence of Preview Book

Figure 5-54 Remove diagram reference

5. Press Yes to confirm removing diagram reference.

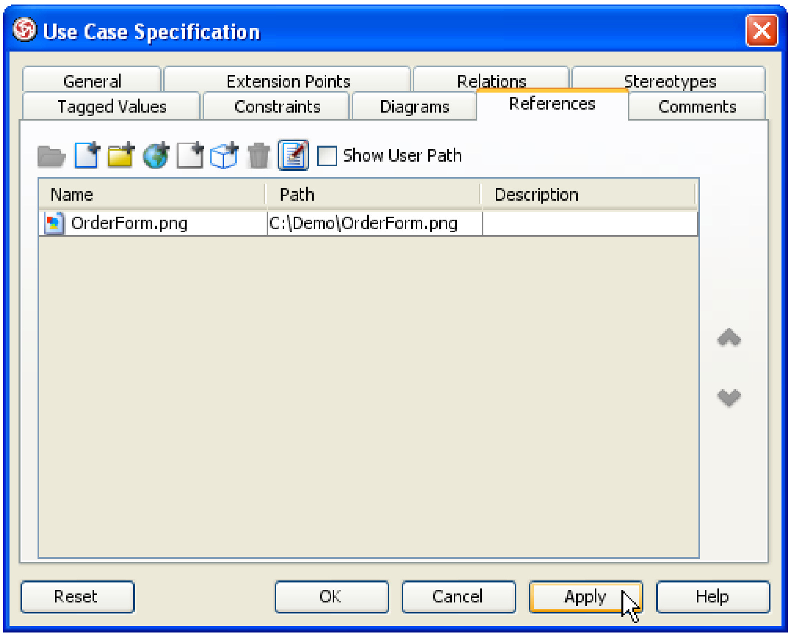


Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	
Online Book Store - Pr…	Sequence Diagram/Onlin…	Sequence of Preview Book

Remove selected references?

Figure 5-55 Confirm removing diagram reference

6. Press Apply button to confirm the reference removal.



□ Show User Path

Name	Path	Description
OrderForm.png	C:\Demo\OrderForm.png	

Figure 5-56 *Apply diagram reference removal*

Elaborate Model Element with Sub Diagram

Creating Sub Diagram

1. Mouse over shape and click on Sub Diagrams resource.

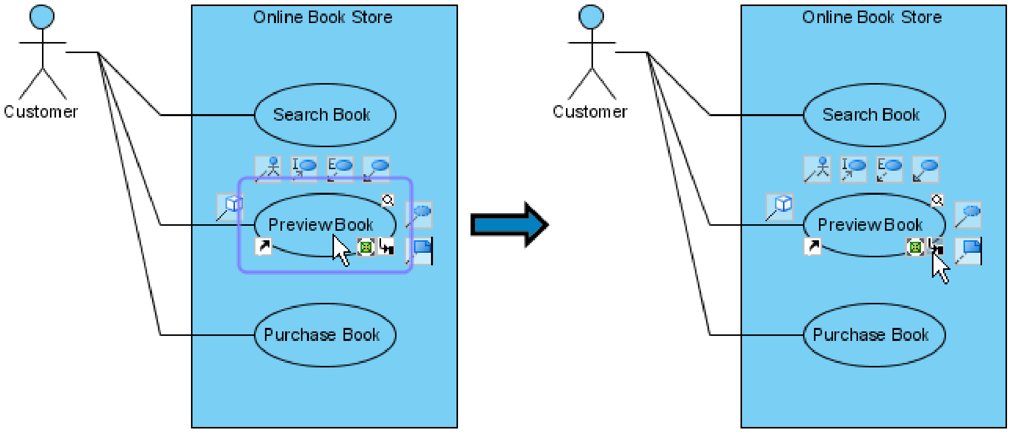


Figure 5-57 *Mouse over sub-diagrams resource*

2. Click on Add > Sequence Diagram menu item.

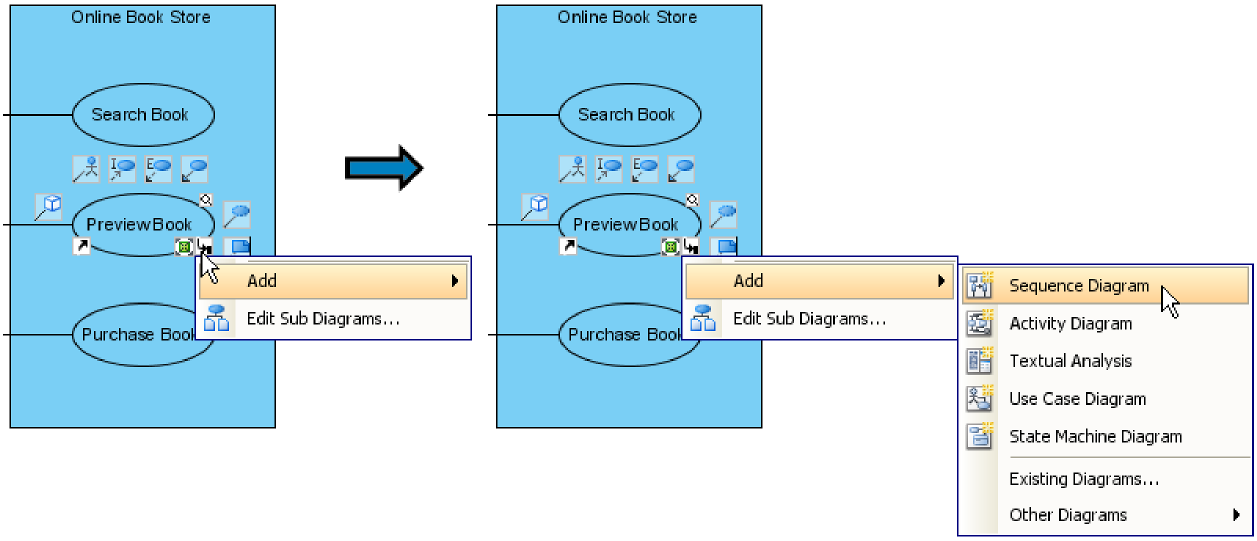


Figure 5-58 *Add sub diagram from resource*

Adding Existing Diagram as Sub Diagram

1. Right click on shape.

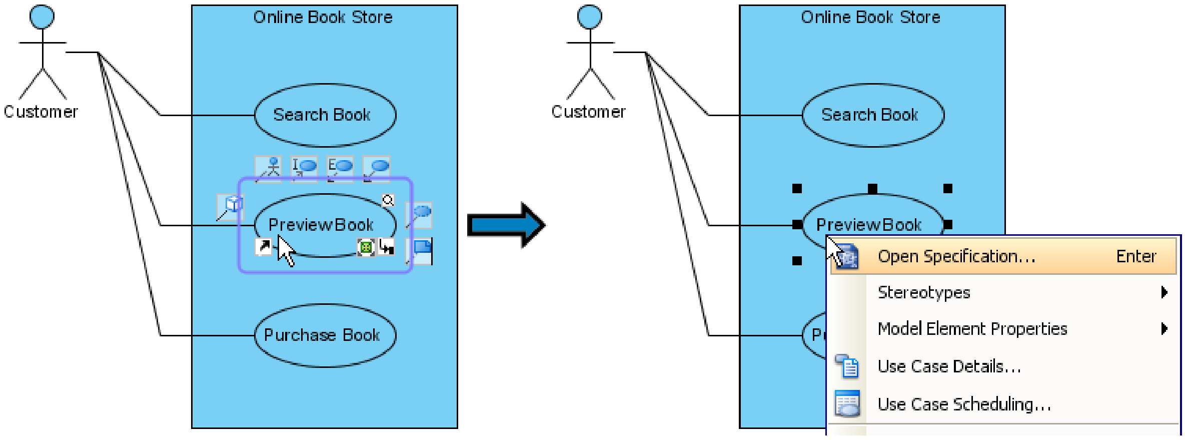


Figure 5-59 *Show popup of shape*

2. Click on Sub Diagrams > Add Existing Diagrams... menu item.

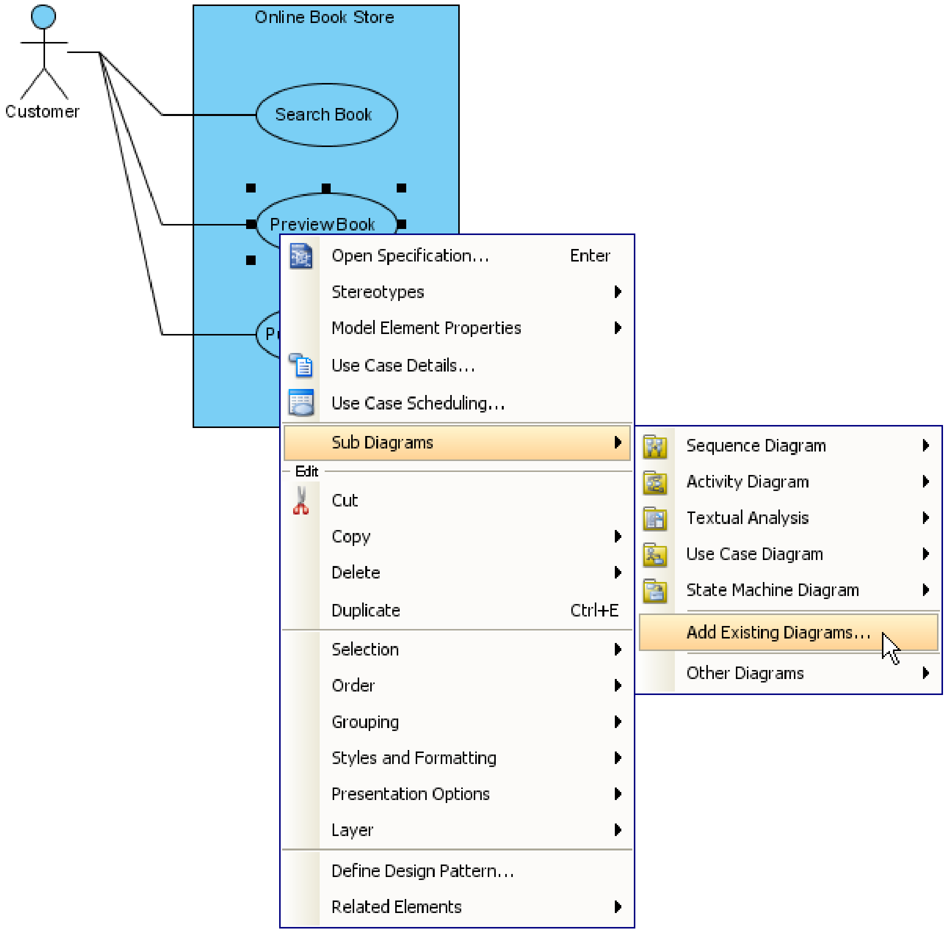
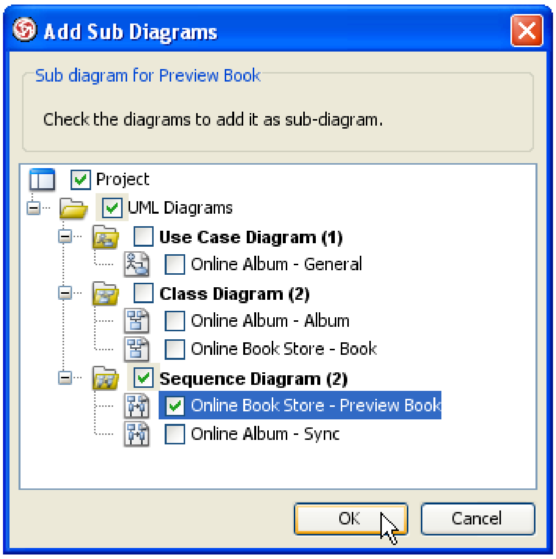


Figure 5-60 *Adding existing diagram as sub diagram*

3. Select sub diagram.



Check the diagrams to add it as sub-diagram.

- ☑ Project
  - ☑ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - ☑ **Sequence Diagram (2)**
      - ☑ Online Book Store - Preview Book
      - □ Online Album - Sync

Figure 5-61 *Selecting sub diagram*

Remove Sub Diagram

1. Mouse over shape and click on Sub Diagrams resource.

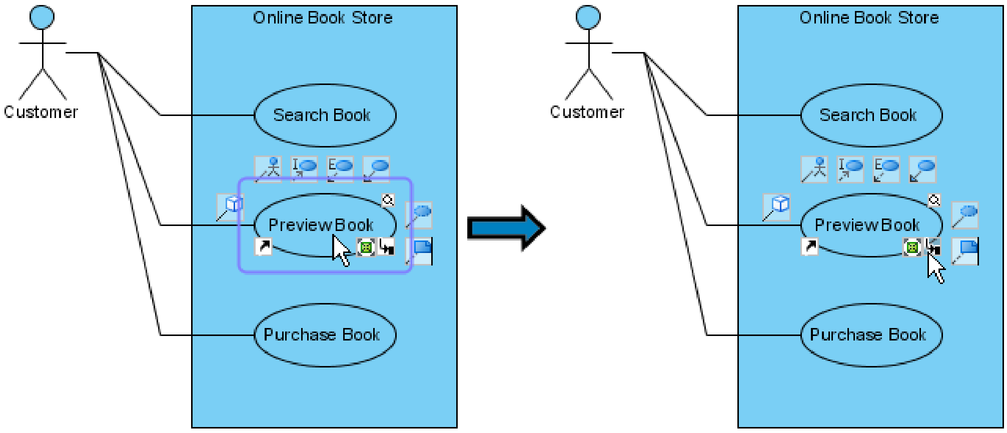


Figure 5-62 *Mouse over sub diagrams resource*

2. Click on Edit Sub Diagrams... menu item.

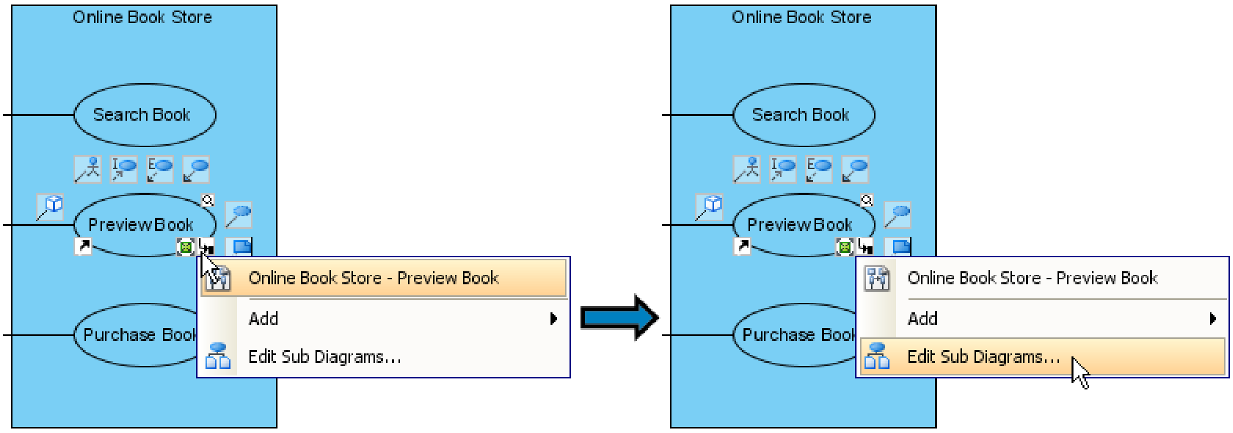
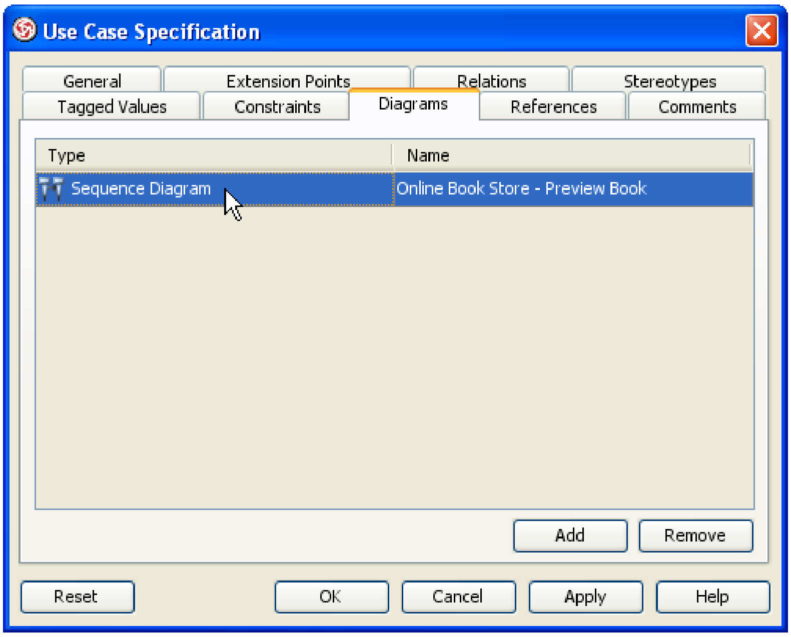


Figure 5-63 *Edit sub diagrams from resource*

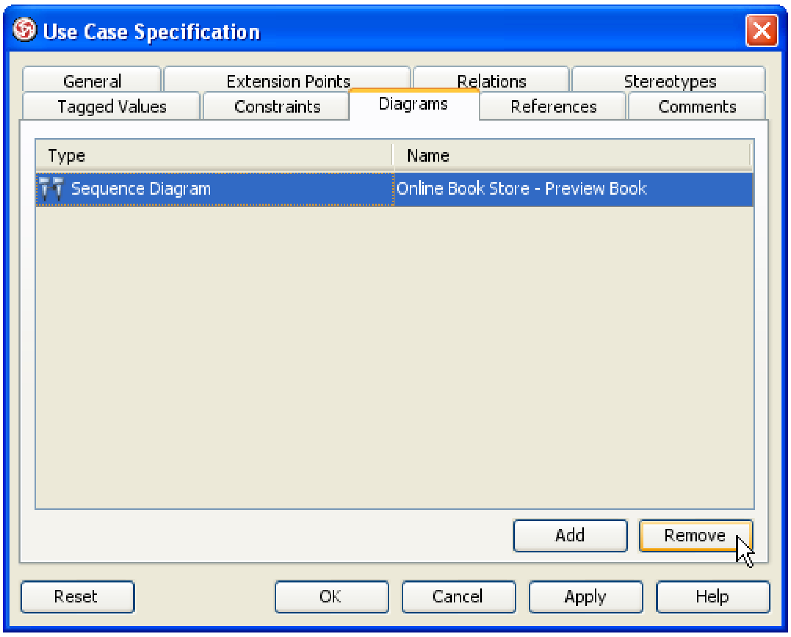
3. Select sub diagram to be removed.



Type	Name
Sequence Diagram	Online Book Store - Preview Book

Figure 5-64 *Select sub diagram*

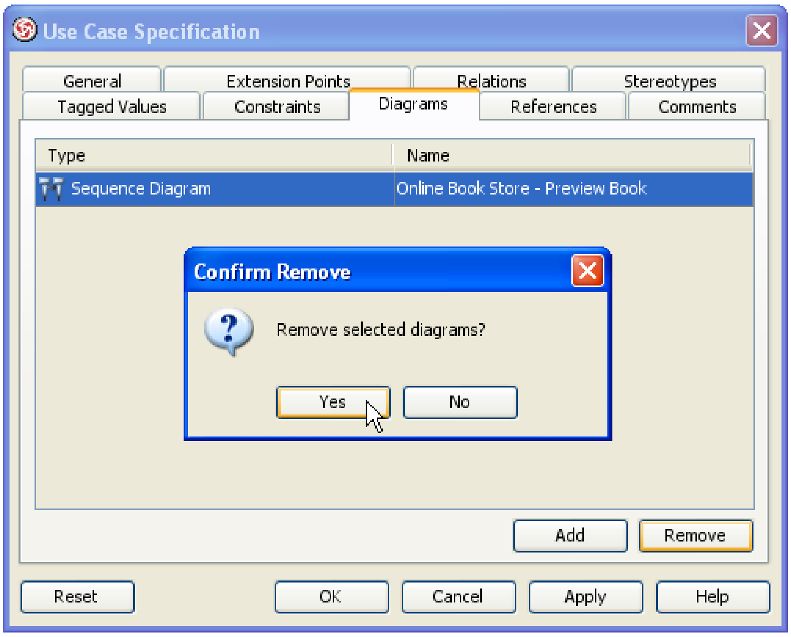
4. Press Remove button.



Type	Name
Sequence Diagram	Online Book Store - Preview Book

Figure 5-65 Remove sub diagram

5. Press Yes to confirm removing sub diagram.



Type	Name
Sequence Diagram	Online Book Store - Preview Book

Remove selected diagrams?

Figure 5-66 Confirm removing sub diagram

6. Press Apply button to confirm the sub diagram removal.

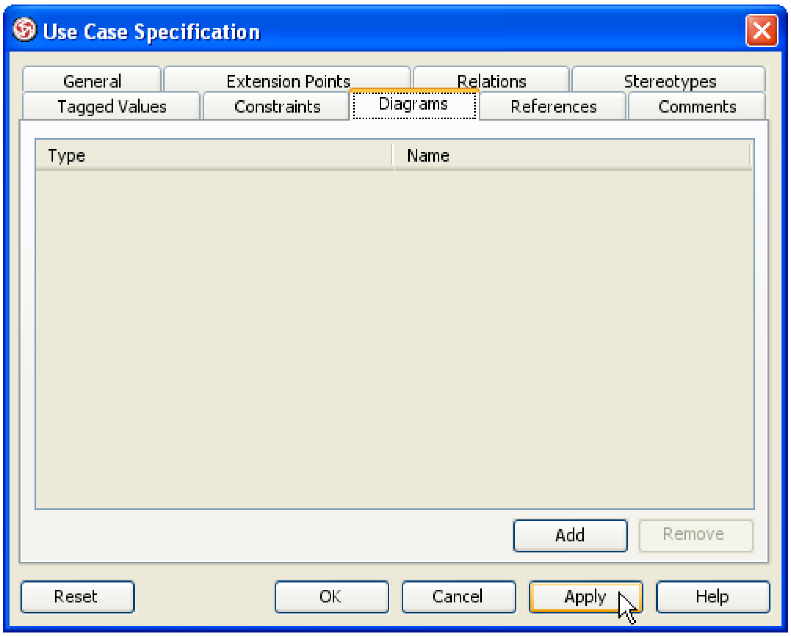


Figure 5-67 *Apply sub diagram removal*

Sub Diagram and it's Diagram Element

Adding sub diagram will add all model element as child.

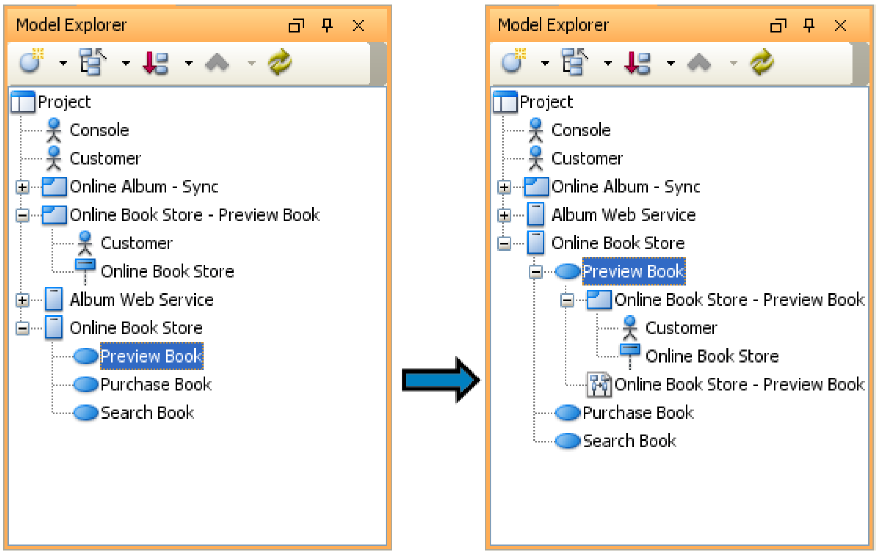


Figure 5-68 *Different after added sub diagram*

Showing Sub Diagrams and reference indicators

1. Right click on diagram and click on Presentation Options > Always Show Reference and Sub Diagram Resource menu item.

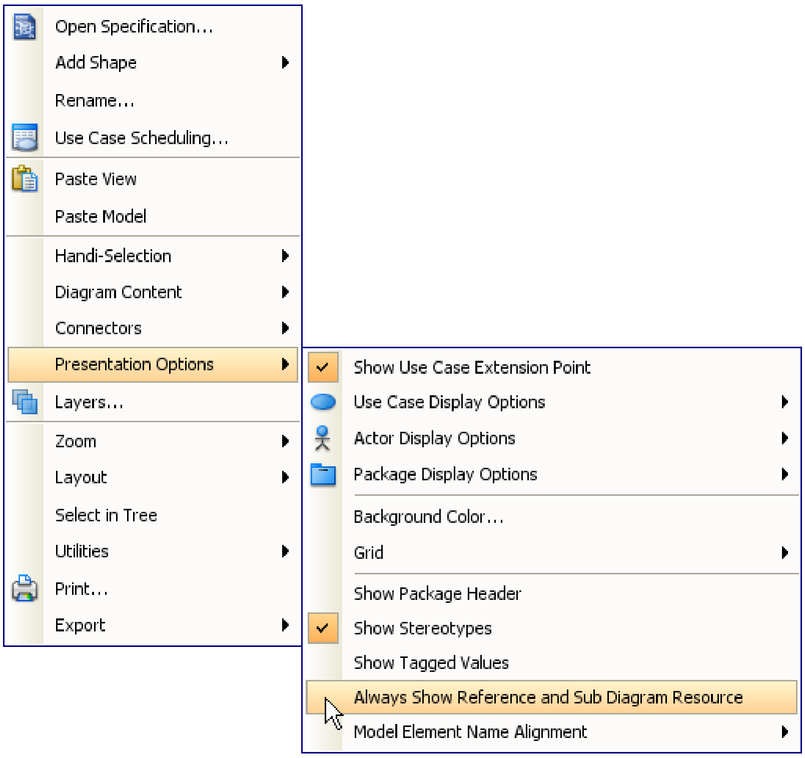


Figure 5-69 Show popup of diagram

2. References resource and Sub Diagrams resources are shown.

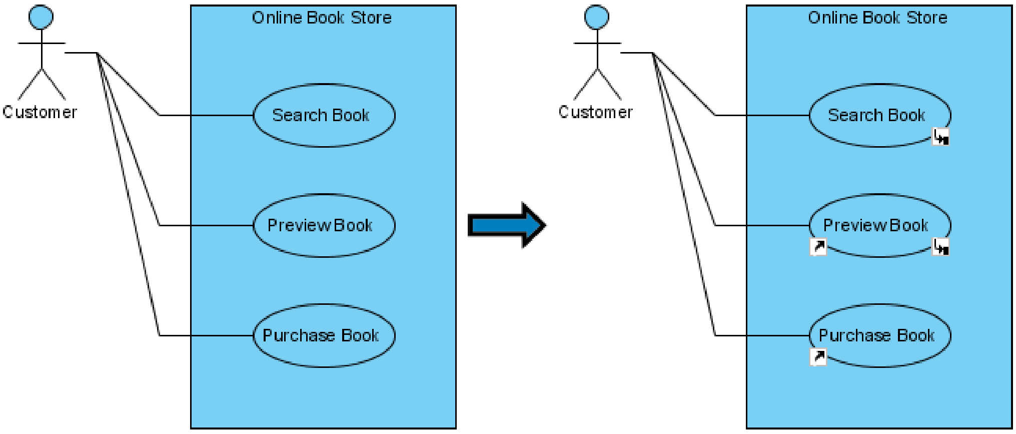


Figure 5-70 Show references resource and sub diagrams resource

Managing Traceability with Model Transitor

Transit Sequence Life Line to Class

1. Mouse over shape.

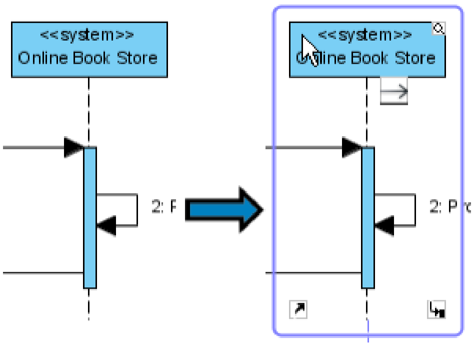


Figure 5-71 Mouse over shape

2. Right click on shape

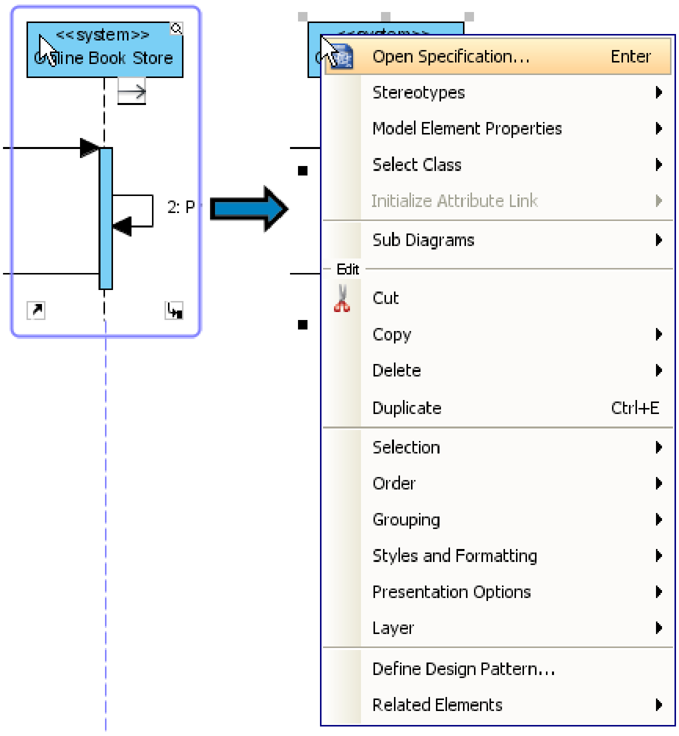


Figure 5-72 Show popup of shape

3. Click on Related Elements > Transit To > Manage Transit To... menu item.

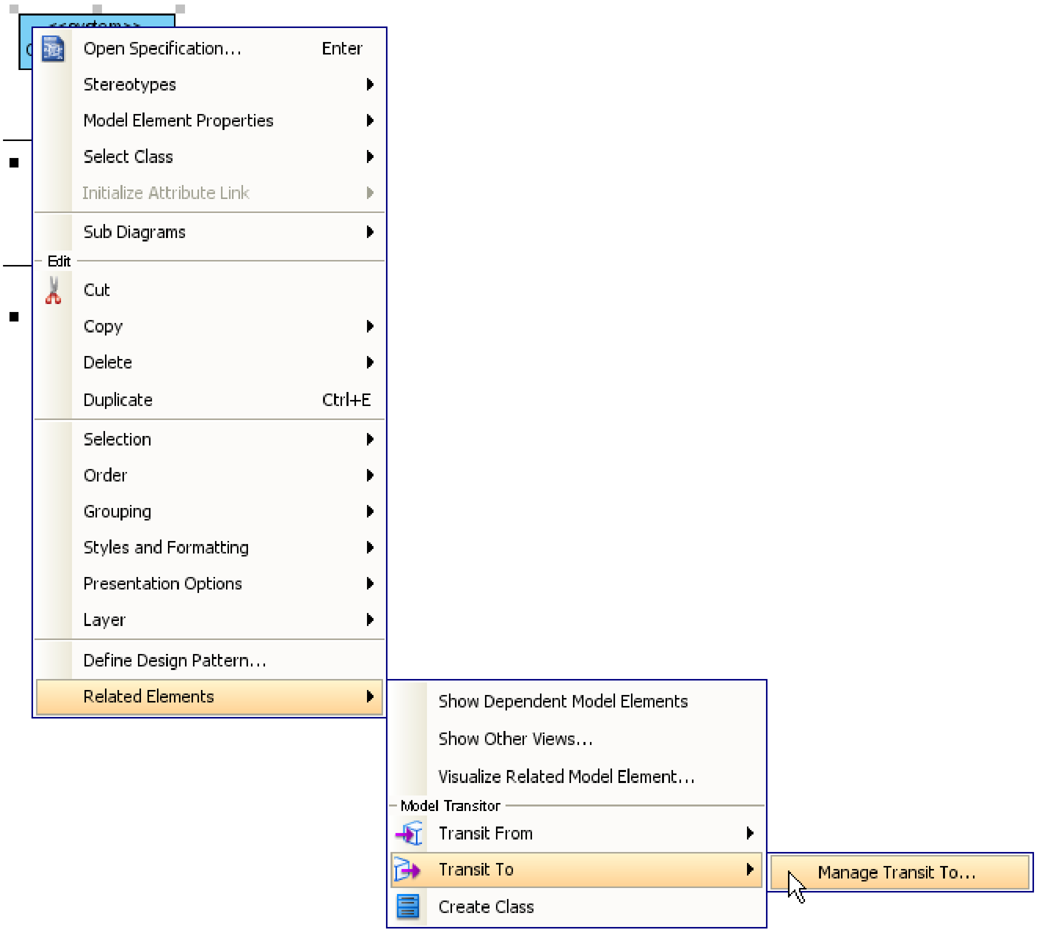


Figure 5-73 *Manage transit to*

4. Select model element and press OK button.

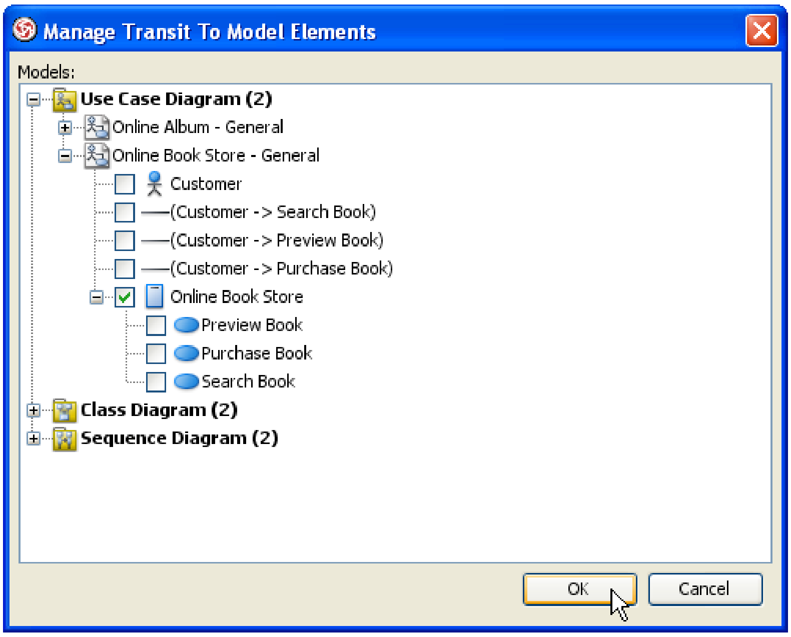


Figure 5-74 *Apply manage transit to*

Navigate to Transit From/To Model Element

1. Mouse over shape.

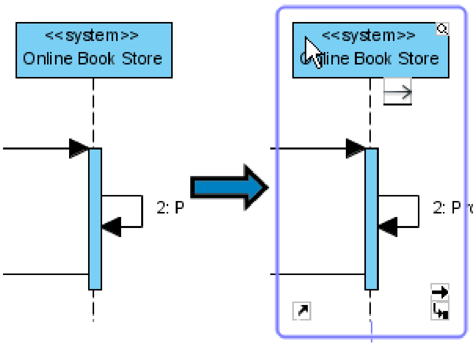


Figure 5-75 Mouse over shape

2. Right click on shape.

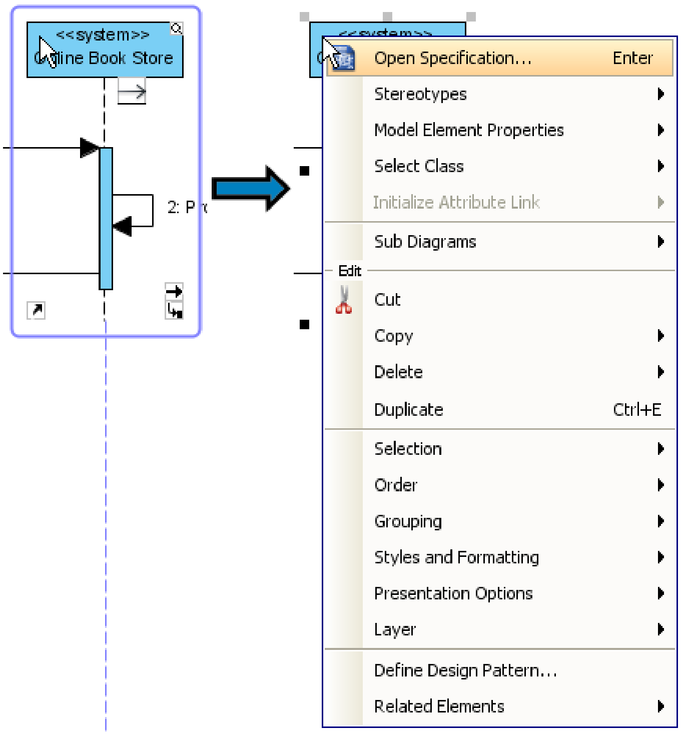


Figure 5-76 Show popup of shape

3. Click on Related Elements > Transit To > *Online Book Store - General Online Book Store* menu item.

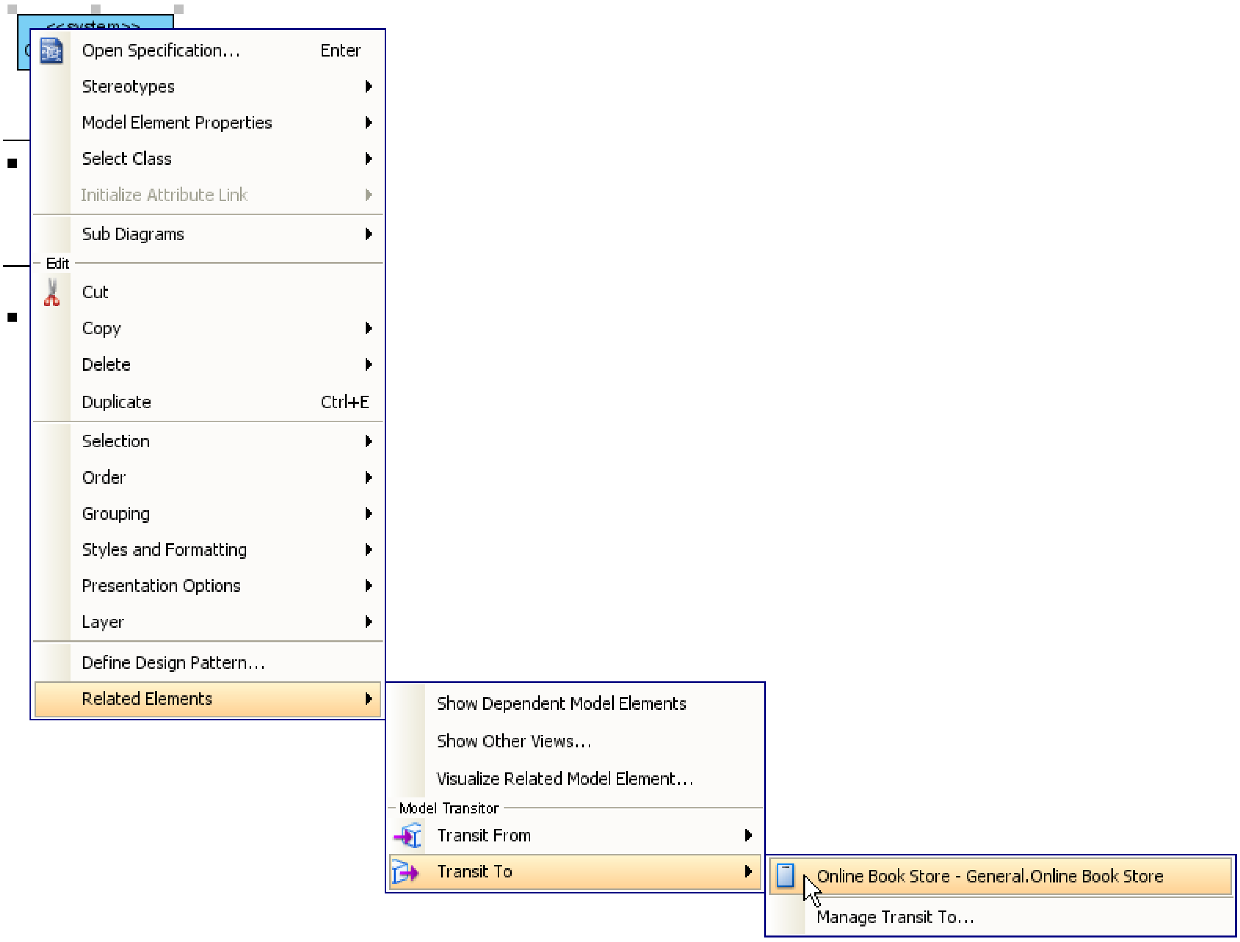


Figure 5-77 *Navigate to transit to model element*

4. Mouse over shape.

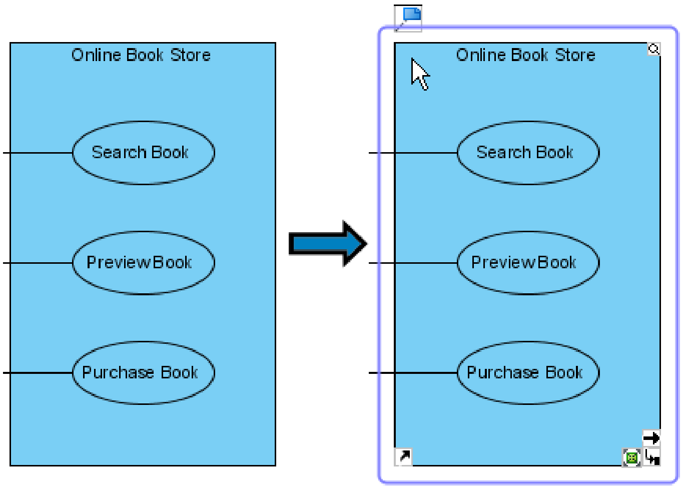


Figure 5-78 *Mouse over shape*

5. Right click on shape.

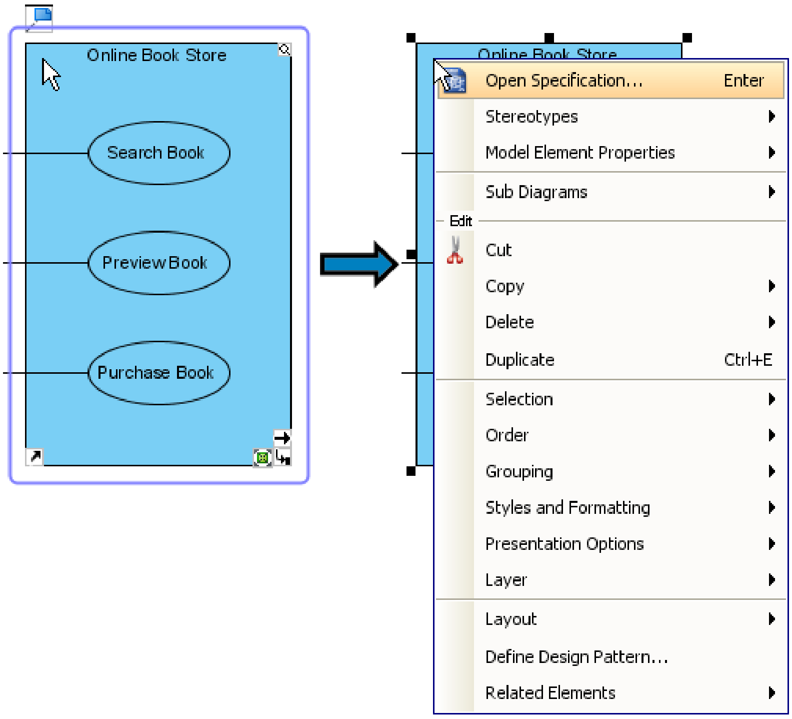


Figure 5-79 *Show popup of shape*

6. Click on Related Elements > Transit From > *Online Book Store - Preview Book.Online Book Store* menu item.

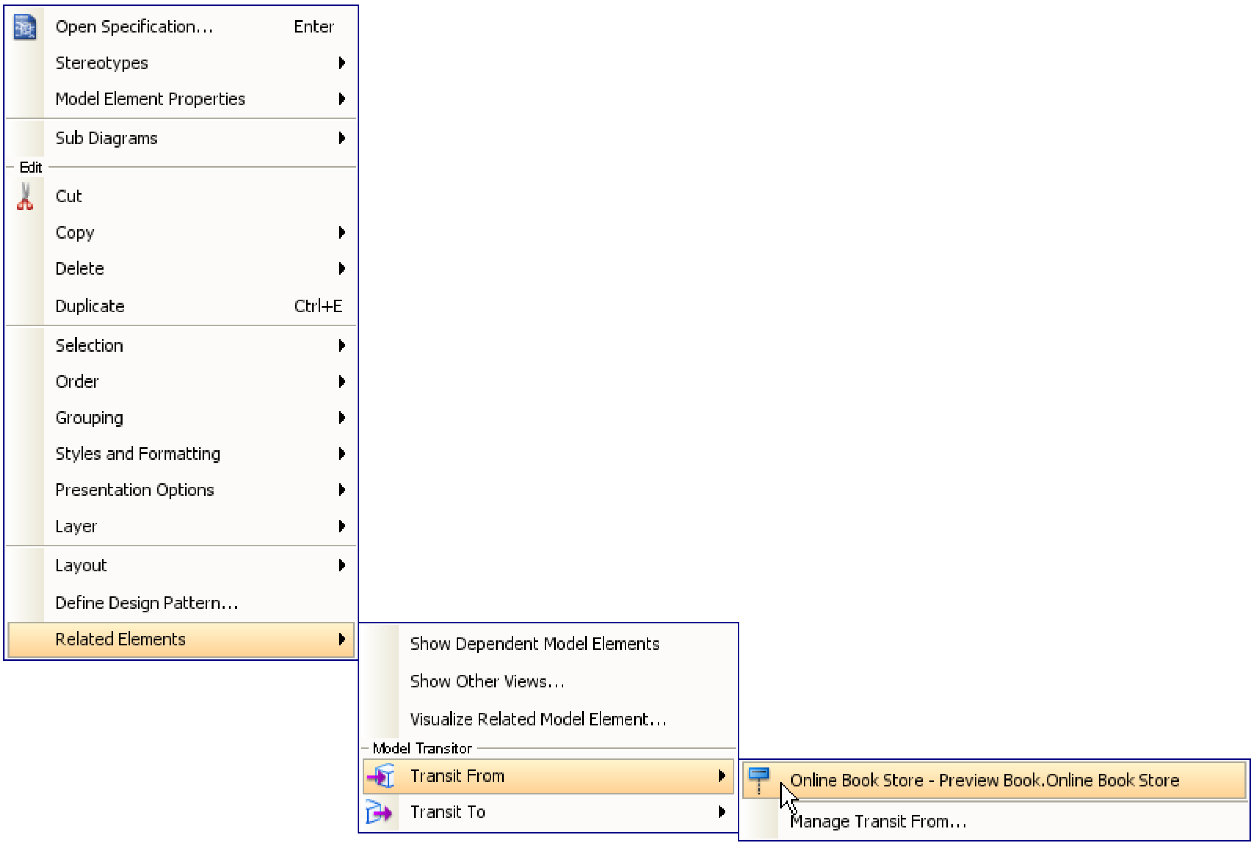


Figure 5-80 *Navigate to transit from model*

Manage Transit To Models

1. Mouse over shape.

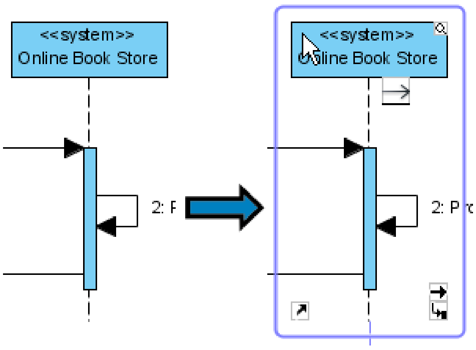


Figure 5-81 Mouse over shape

2. Right click on shape.

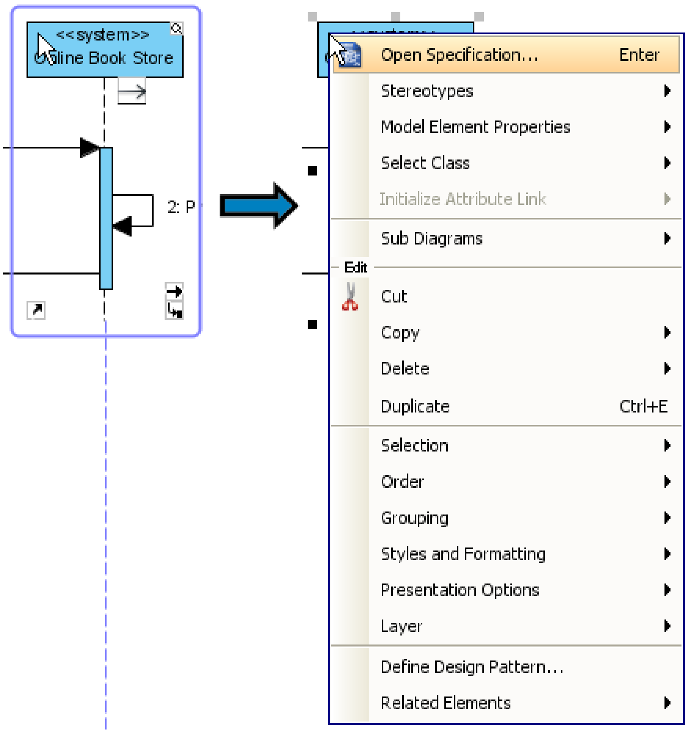


Figure 5-82 Show popup of shape

3. Click on Related Elements > Transit To > Manage Transit To... menu item.

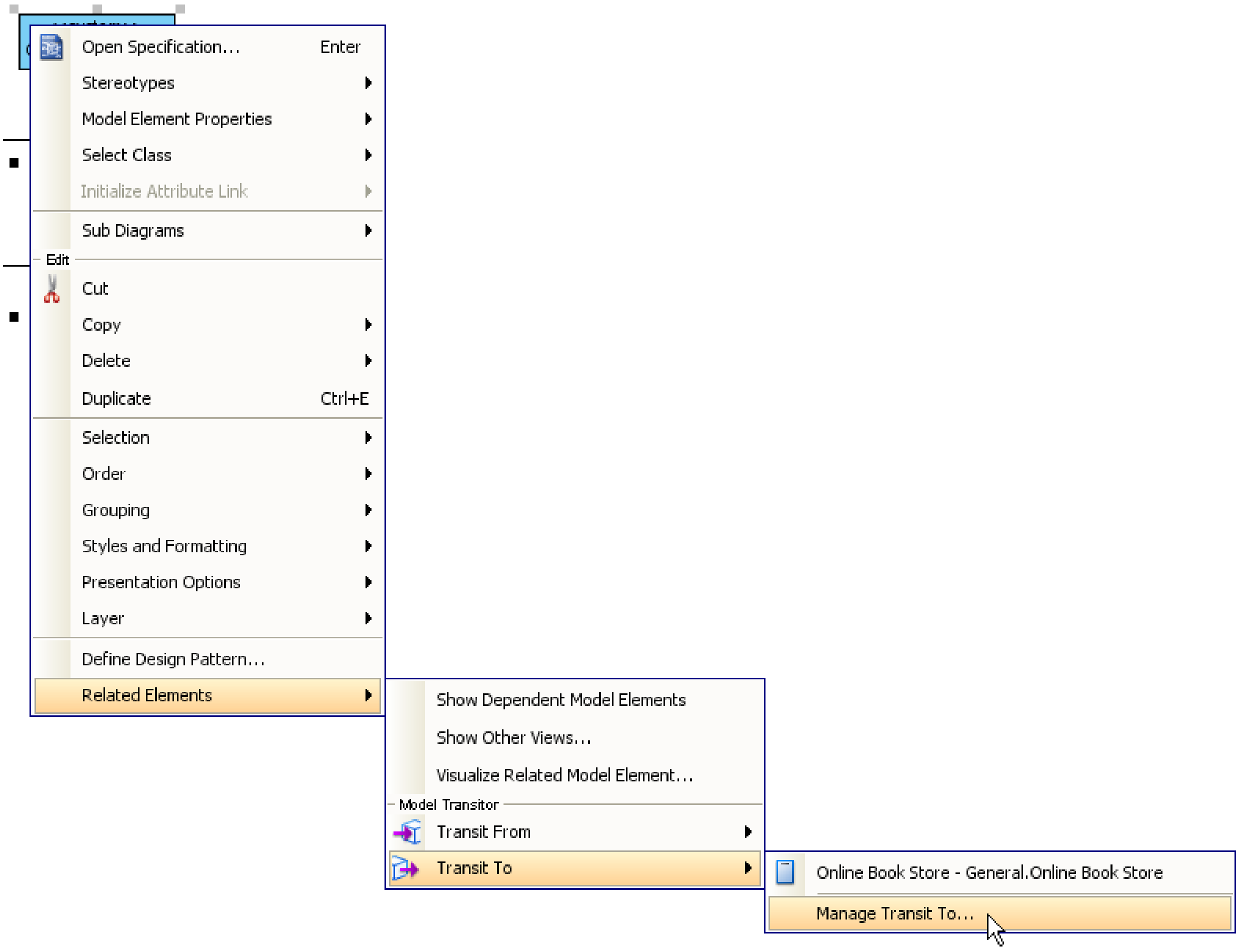
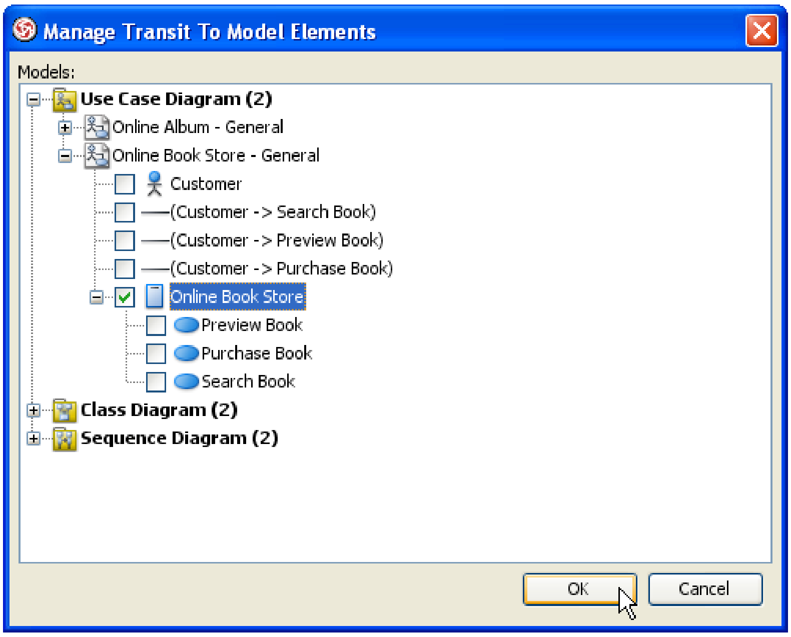


Figure 5-83 Manage transit to

4. Select or deselect model elements and press OK button.


Manage Transit To Model Elements

Models:

- Use Case Diagram (2)
  - Online Album - General
  - Online Book Store - General
    - □ Customer
    - □ (Customer -> Search Book)
    - □ (Customer -> Preview Book)
    - □ (Customer -> Purchase Book)
    - ☑ Online Book Store
      - □ Preview Book
      - □ Purchase Book
      - □ Search Book
- Class Diagram (2)
- Sequence Diagram (2)

Figure 5-84 Apply manage transit to

Manage Transit From Models

1. Mouse over shape.

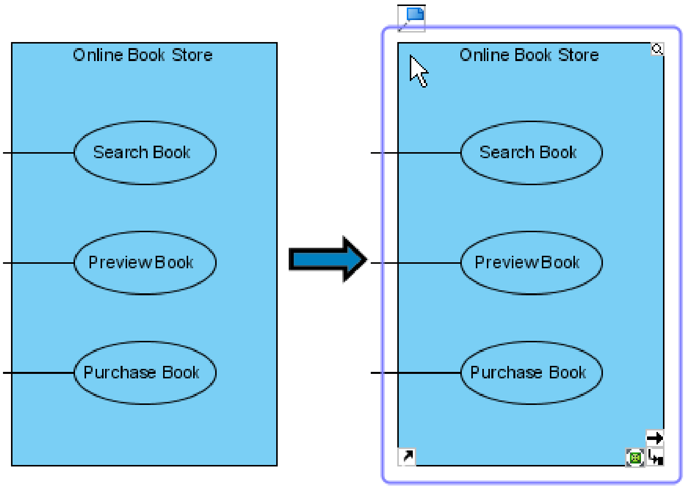


__Figure 5-85__ Mouse over shape

2. Right click on shape.

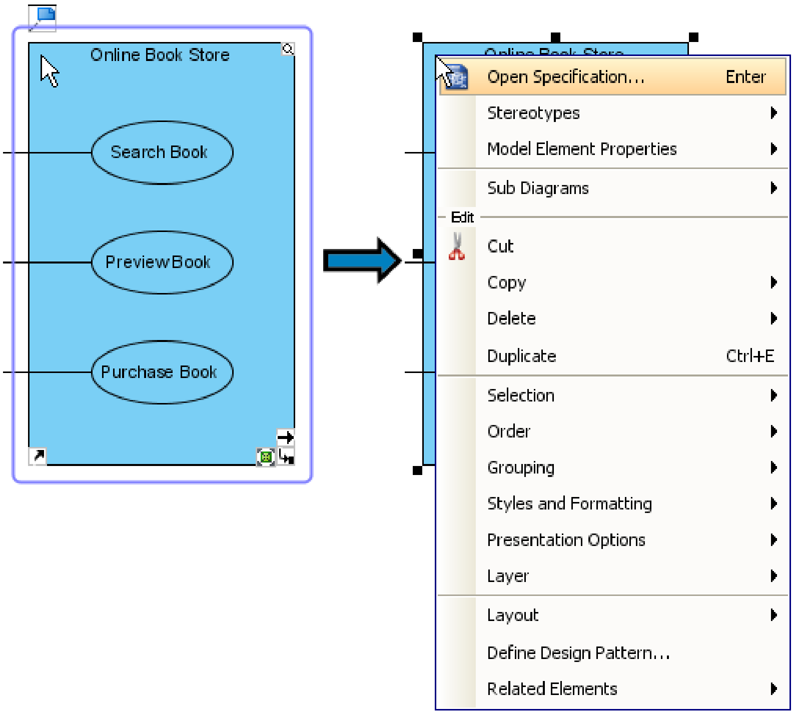


__Figure 5-86__ Show popup of shape

3. Click on Related Elements > Transit From > Manage Transit From... menu item.

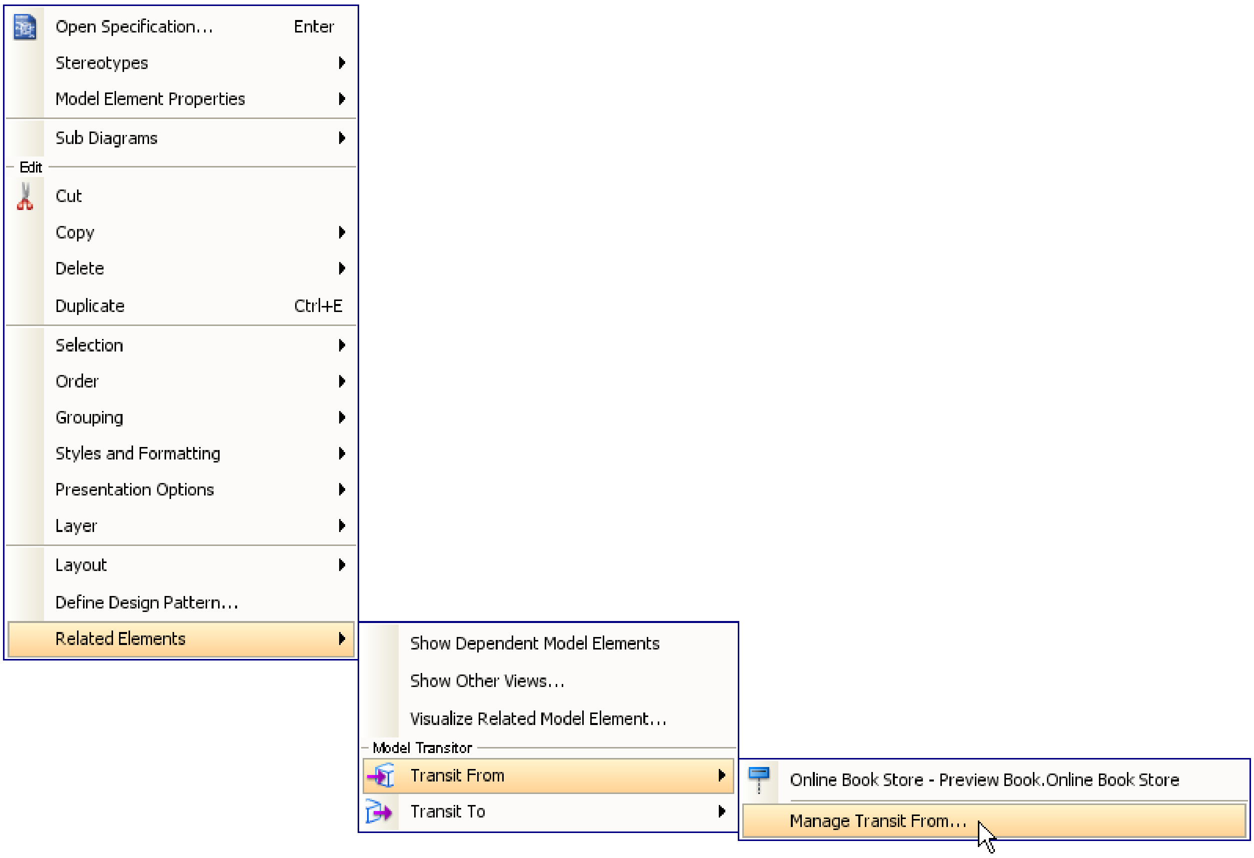
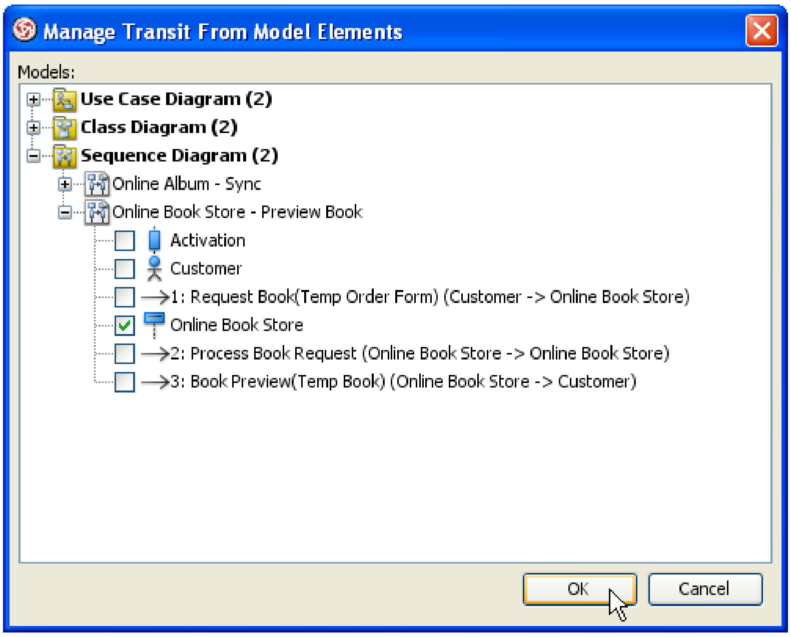


Figure 5-87 Manage transit from

4. Select or deselect model elements and press OK button.


Manage Transit From Model Elements

Models:

- **Use Case Diagram (2)**
- **Class Diagram (2)**
- **Sequence Diagram (2)**
  - Online Album - Sync
  - Online Book Store - Preview Book
    - □ Activation
    - □ Customer
    - □ 1: Request Book(Temp Order Form) (Customer -> Online Book Store)
    - ☑ Online Book Store
    - □ 2: Process Book Request (Online Book Store -> Online Book Store)
    - □ 3: Book Preview(Temp Book) (Online Book Store -> Customer)

Figure 5-88 Apply manage transit from

Changing Diagram Elements Styles

You can changing the diagram element's style in the **Format** dialog. To open the Format dialog, right click on the shape and select **Styles and Formatting** > **Formats...** from the popup menu.

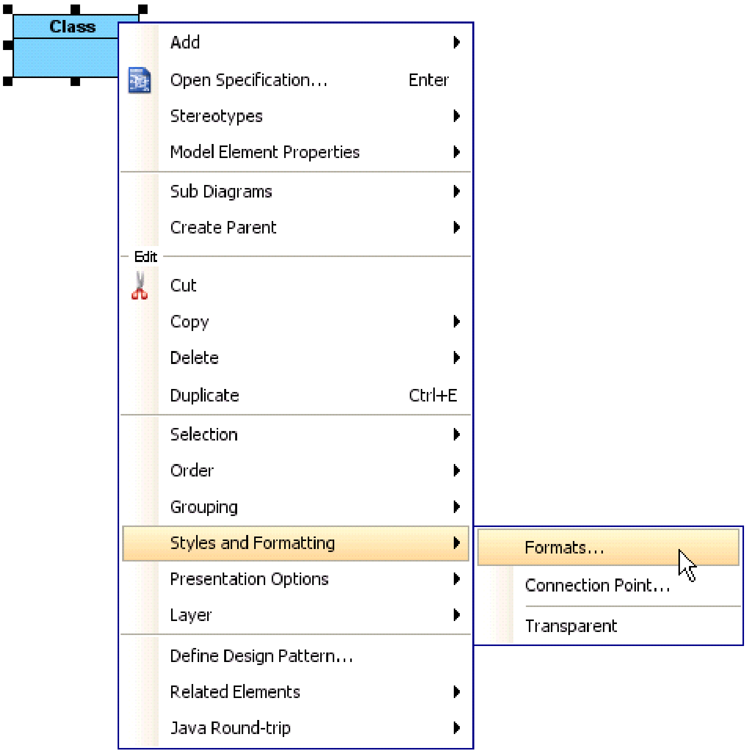
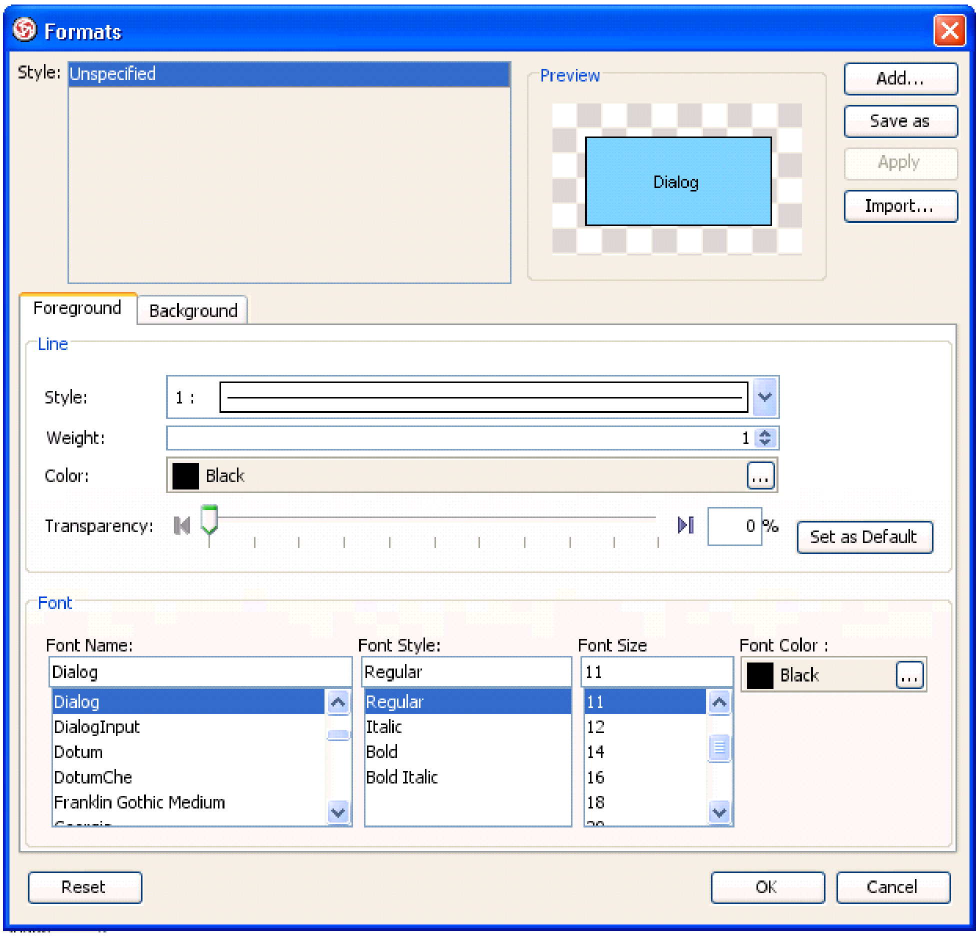


Figure 6-1 *Open Format dialog from popup menu*



Style: Unspecified

Style: 1 :

Weight: 1

Color: Black

Transparency: 0 %

Font Name: Dialog

Font Style: Regular

Font Size 11

Font Color : Black

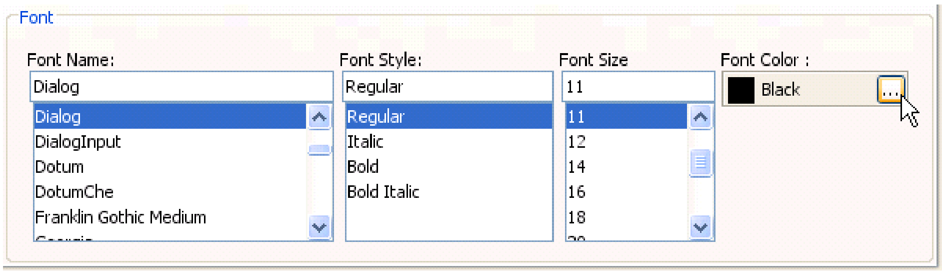
***Figure 6-2** Format dialog*

You can change the followings setting from the Format dialog:

- Changing Shapes Foreground Style
- Changing Shapes Font Style
- Changing Shapes Background Style

Changing Shapes Foreground Style

In the **Format** dialog, you can changing the foreground style in the **Font** section.



Font

Font Name: Dialog

Font Style: Regular

Font Size 11

Font Color : Black

***Figure 6-3** Font section*

Just click on the **...** button beside 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).

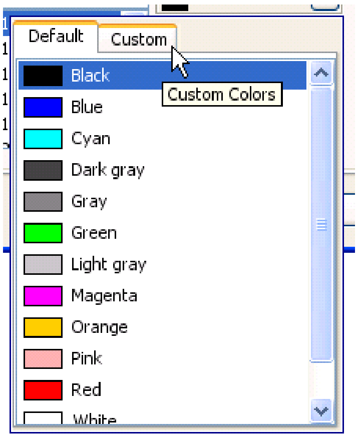


***Figure 6-4** Color pane*

If you want to specify a custom color, you can switch to the **Custom** pane

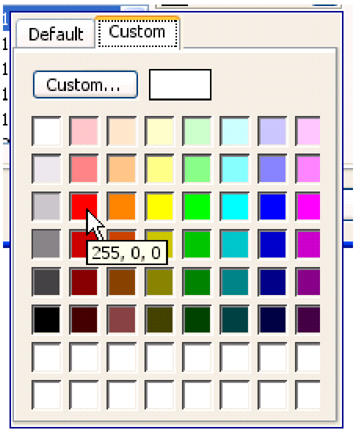


***Figure 6-5** Custom color pane*

After change the font color, the preview will update automatically.

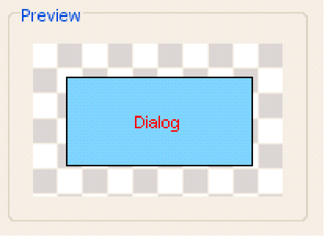
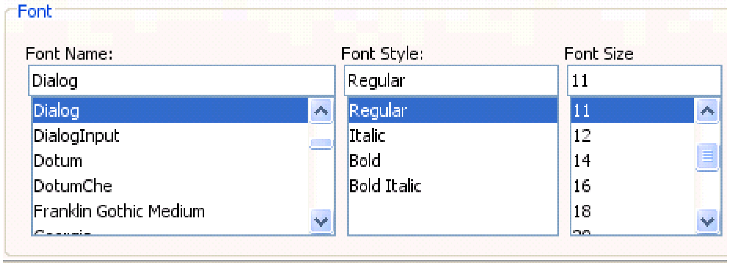


***Figure 6-6** Change font color preview*

Changing Shapes Font Style

In the Font section, you can also change the font family, style and size.


Font

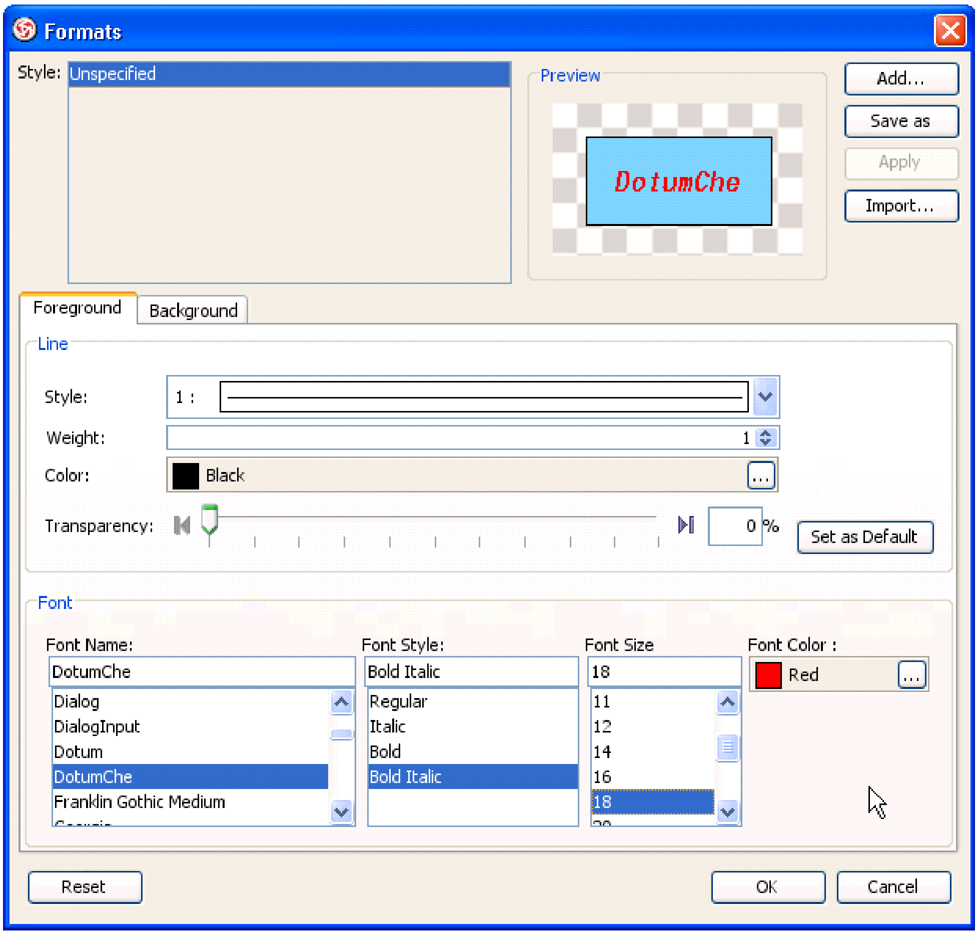
Font Name: Dialog

Font Style: Regular

Font Size 11

***Figure 6-7** Changing font style*

Field	Description
Font Name	Select different types of font. The number of fonts depends on the fonts available in your computer.
Font Style	Select the style of font. You can select one of the 4 styles, a preview will be shown for each of the style items.
Font Size	Select the size of font. You may either click on the default sizes or enter the font size in the text fields.


Formats

Style: Unspecified

Preview: DotumChe

Add... | Save as | Apply | Import...

Foreground | Background

Line

Style: 1 :

Weight: 1

Color: Black

Transparency: 0 % | Set as Default

Font

Font Name: DotumChe (Dialog, DialogInput, Dotum, DotumChe, Franklin Gothic Medium)

Font Style: Bold Italic (Regular, Italic, Bold, Bold Italic)

Font Size: 18 (11, 12, 14, 16, 18)

Font Color : Red

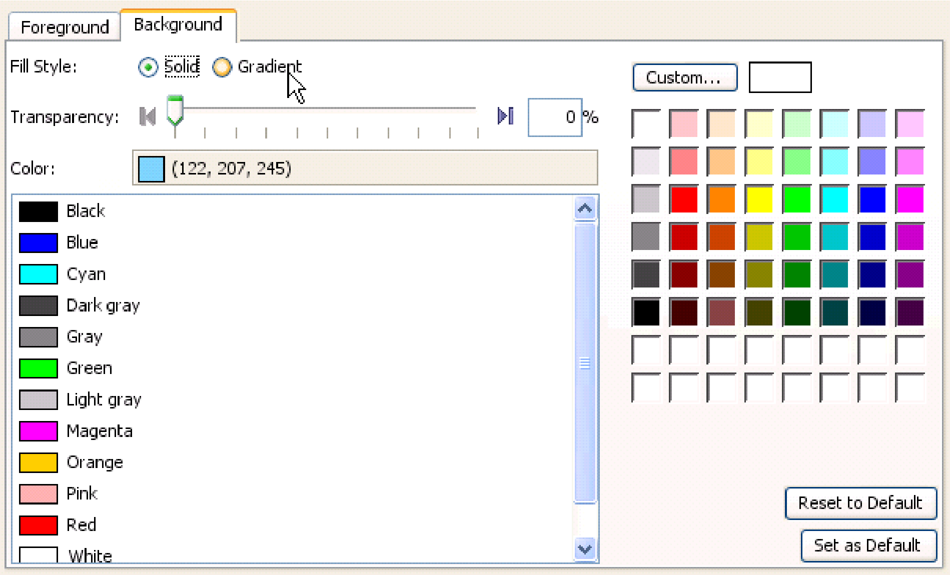
Reset | OK | Cancel

Figure 6-8 Change font style

The Preview pane displays the selected font format.

Changing Shapes Background Style

You can click on the **Background** tab to custom the background style.


Foreground | Background

Fill Style: ● Solid ○ Gradient

Transparency: 0 %

Color: (122, 207, 245)

Black, Blue, Cyan, Dark gray, Gray, Green, Light gray, Magenta, Orange, Pink, Red, White

Custom...

Reset to Default | Set as Default

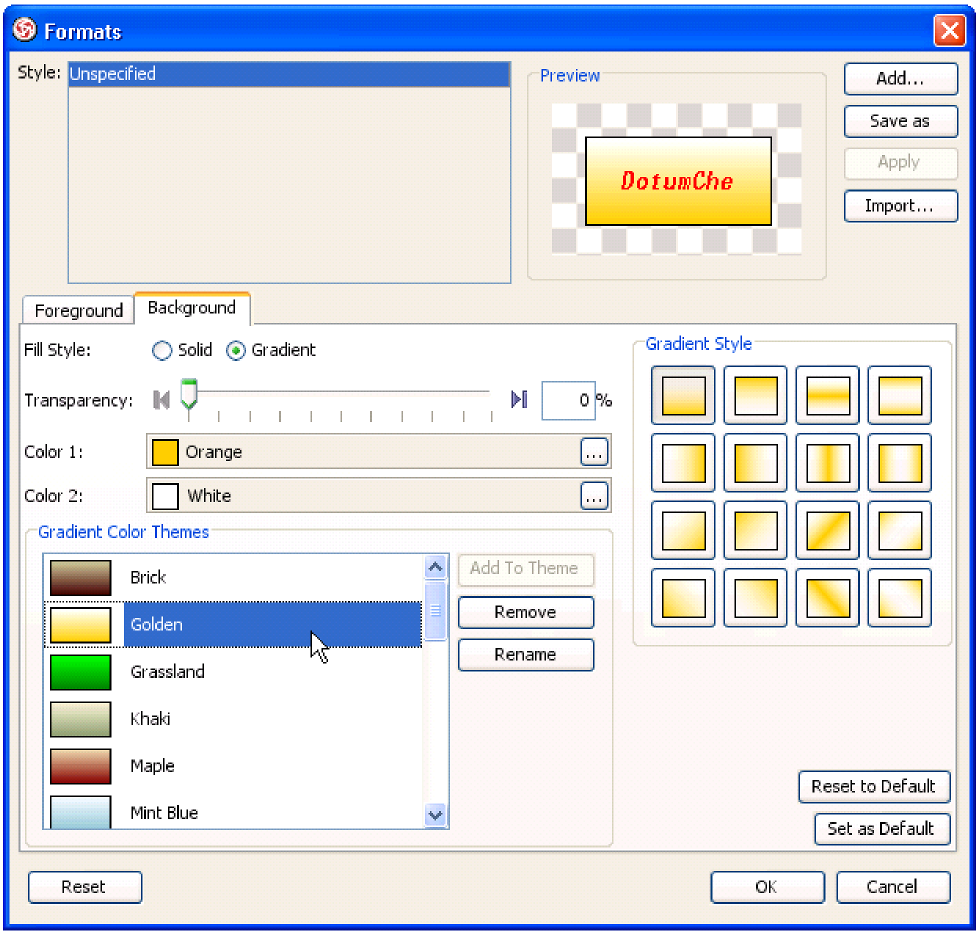
Figure 6-9 Changing Background style

In the **Background** tab, it allows you to select a soild 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 **Soild** (a single color) or **Gradient** (a fill color that is mixed by two colors).

Transparency	Specify the transparency of the fill color. The greater the value, the more transparent is the shape. 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 selecting **Gradient** from the **Fill style** field you will see the detail pane for formatting a gradient fill color.



Formats

Style: Unspecified

Fill Style: ○ Solid ● Gradient

Transparency: 0 %

Color 1: Orange

Color 2: White

Gradient Color Themes: Brick, Golden (selected), Grassland, Khaki, Maple, Mint Blue

Figure 6-10 Select 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 done, click **OK** to apply to the selected shape.

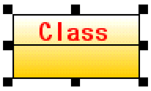


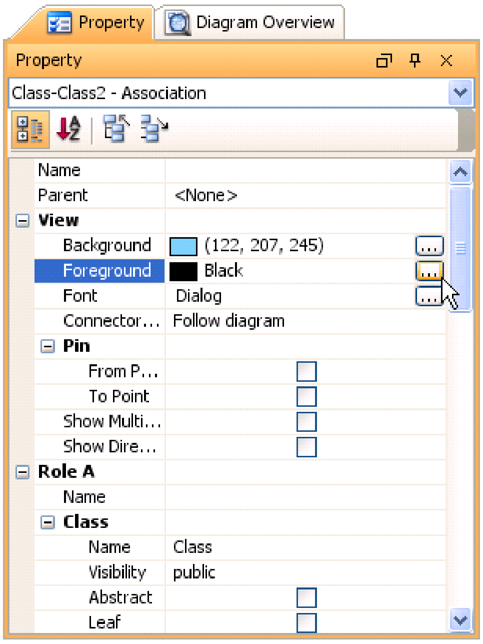
Figure 6-11 Change shape background style result

Changing Connector Line Style

To change the connector line style via **Property table**:

1. Select the connector.
2. Find the **Foreground** row in the Property table.

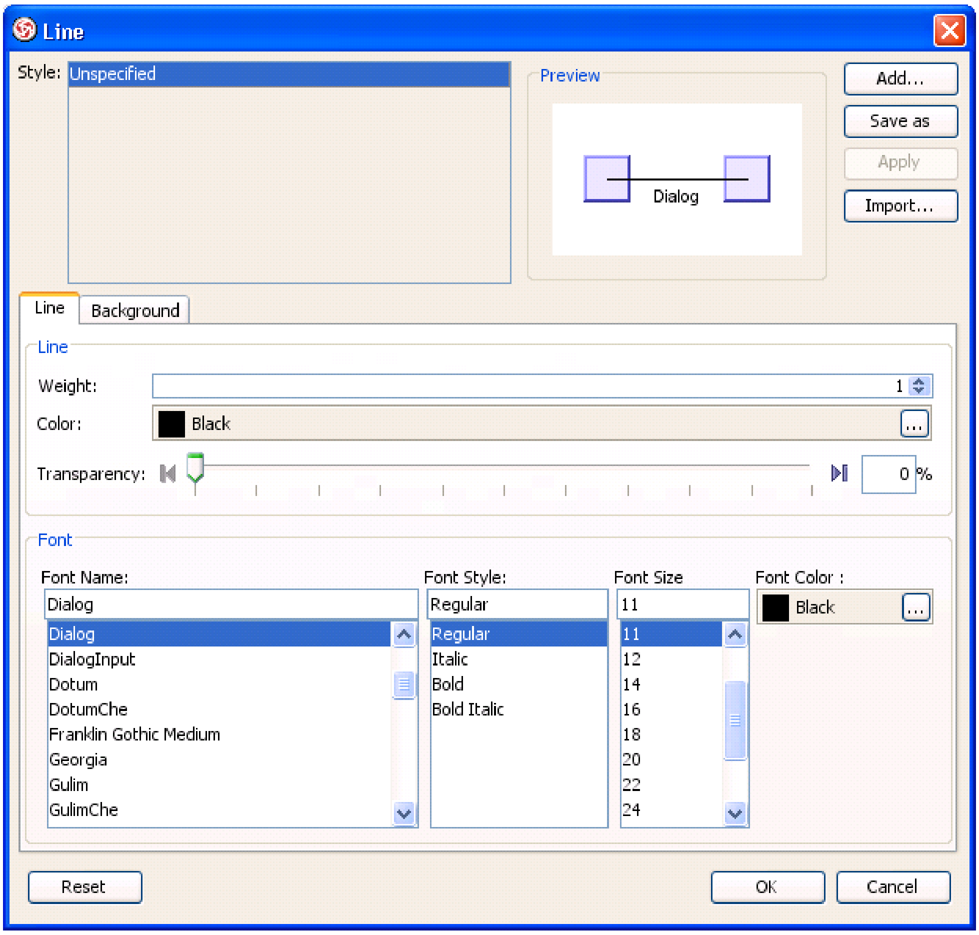
3. Click the **...** button.



Class-Class2 - Association

Name	
Parent	<None>
⊟ View	
Background	(122, 207, 245) ...
Foreground	Black ...
Font	Dialog ...
Connector…	Follow diagram
⊟ Pin	
From P…	□
To Point	□
Show Multi…	□
Show Dire…	□
⊟ Role A	
Name	
⊟ Class	
Name	Class
Visibility	public
Abstract	□
Leaf	□

Figure 6-12 Change connector line style on properties table



Style: Unspecified

Line

Weight: 1

Color: Black ...

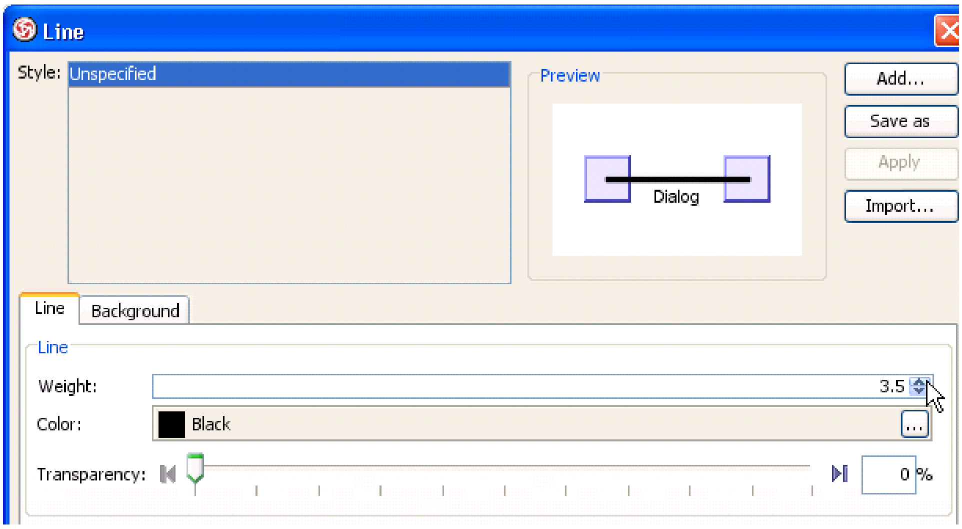
Transparency: 0 %

Font

Font Name:	Font Style:	Font Size	Font Color :
Dialog	Regular	11	Black ...
Dialog	Regular	11	
DialogInput	Italic	12	
Dotum	Bold	14	
DotumChe	Bold Italic	16	
Franklin Gothic Medium		18	
Georgia		20	
Gulim		22	
GulimChe		24	

Figure 6-13 Line style dialog

You can format the line style in the line section. It allows you to adjust weight (thickness), color and transparency.



Style: Unspecified

Weight: 3.5

Color: Black

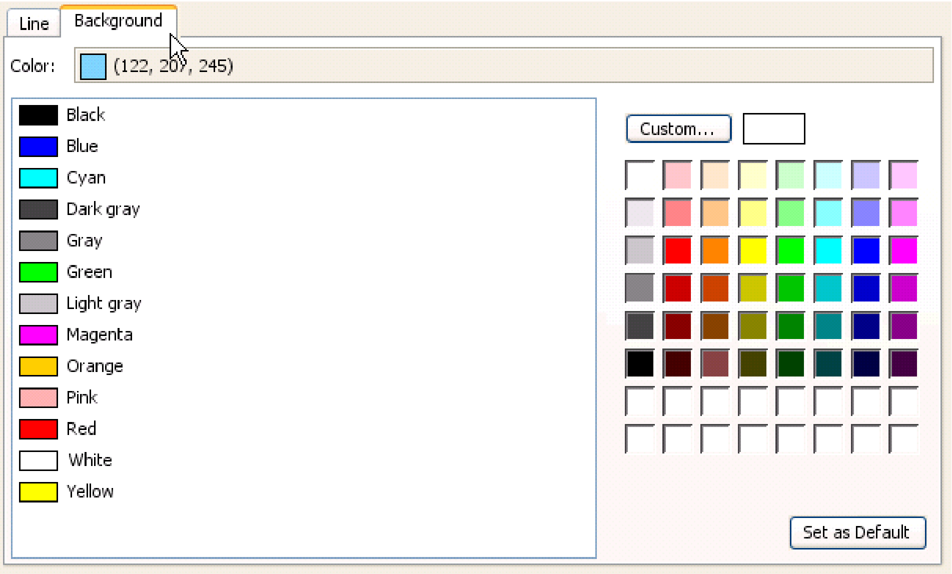
Transparency: 0 %

Figure 6-14 *Line section*

Field	Description
Weight	Adjust the weight (thickness) of a line. The greater the value, the thicker the line. You can use the up/down button to increase/decrease 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 beside 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:** Only integer values can be used 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 completety 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 a rectangle surrounded by the line with the selected line format applied.

Changing Connector Background Style

Upon selecting the **Background** tab, it allows you to specify the background fill style of the line.



Color: (122, 207, 245)

Figure 6-15 *Change connector background style*

Click OK to apply the settings.

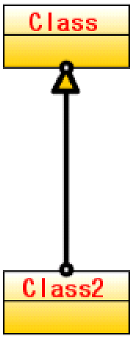


Figure 6-16 *Change connector style result*

Manage and Apply Styles

VP-UML allows you to define you own style and apply to other shape by simple steps.

Adding Style

To open the Style dialog, Select **View > Styles...** from the main menu.

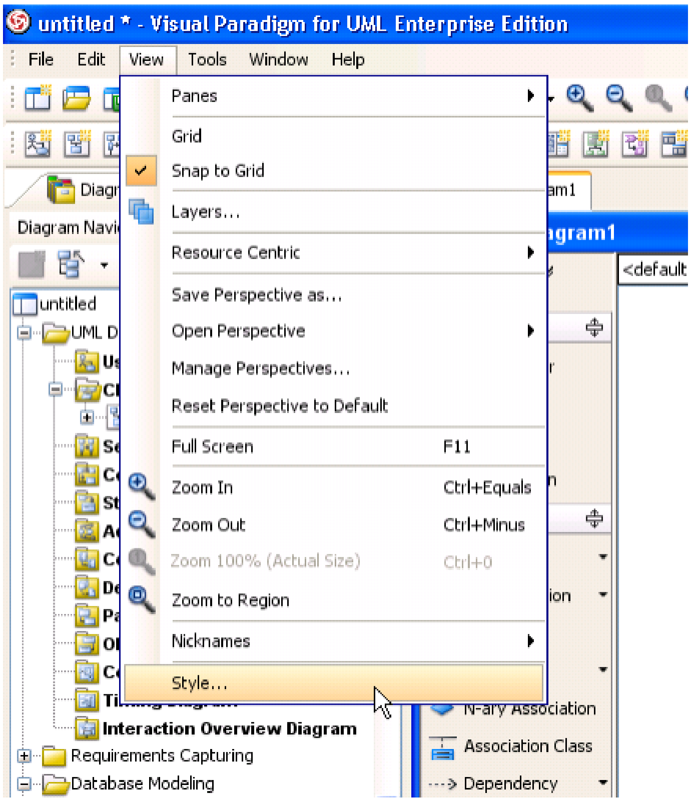


Figure 6-17 *Open Style dialog*

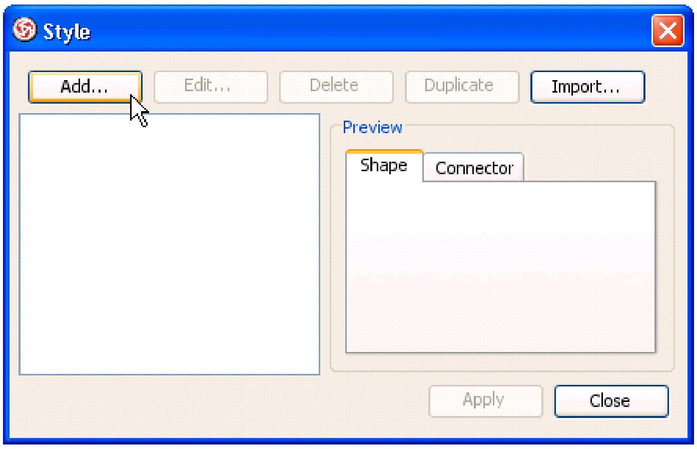
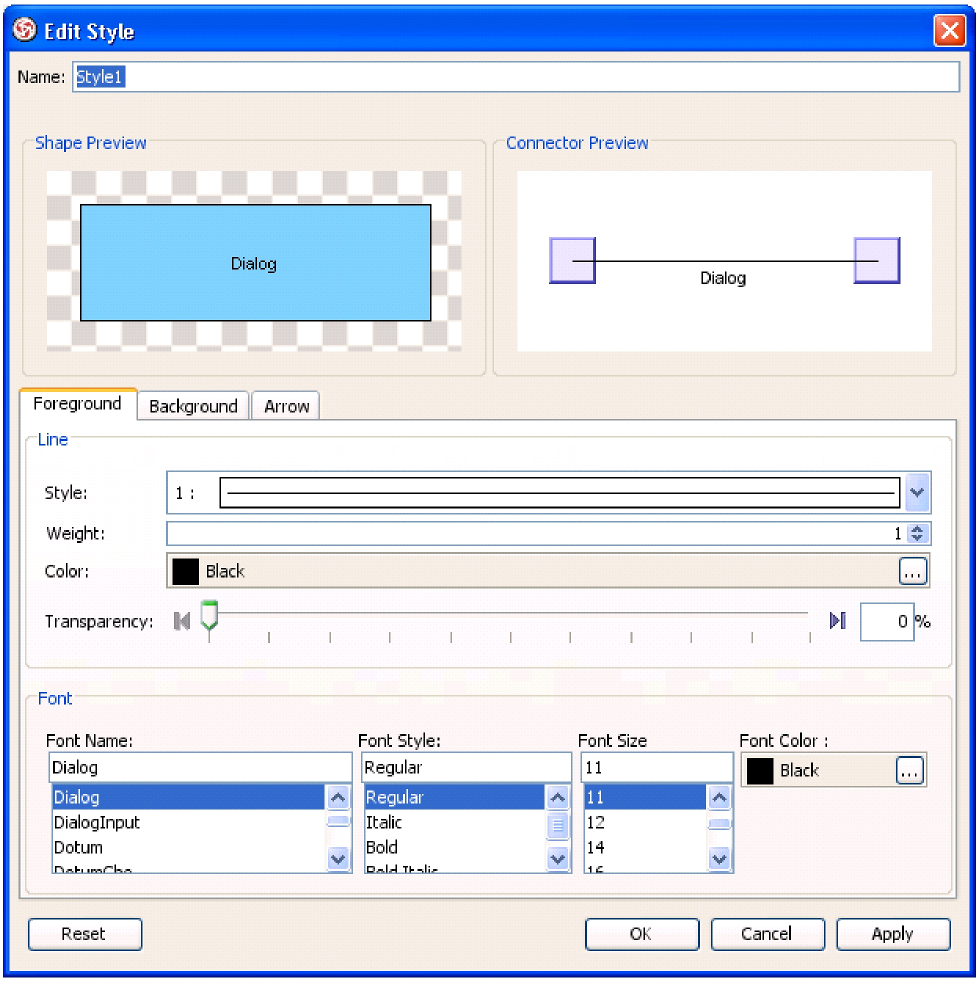


Figure 6-18 *Style dialog*

In the Style dialog, click **Add...** to create and edit a new style.


Edit Style

Name: Style1

Line

Style: 1 :

Weight: 1

Color: Black

Transparency: 0 %

Font

Font Name: Dialog

Font Style: Regular

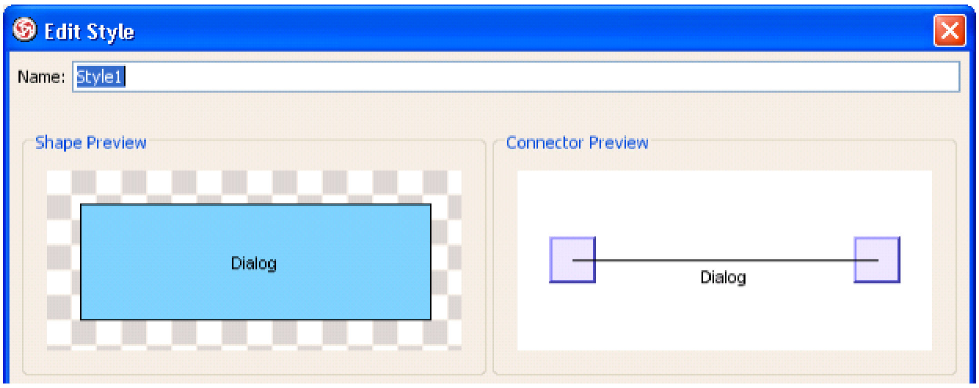
Font Size 11

Font Color : Black

__Figure 6-19__ Edit Style dialog

In the **Edit Style** dialog, you can change the followings setting of the style:

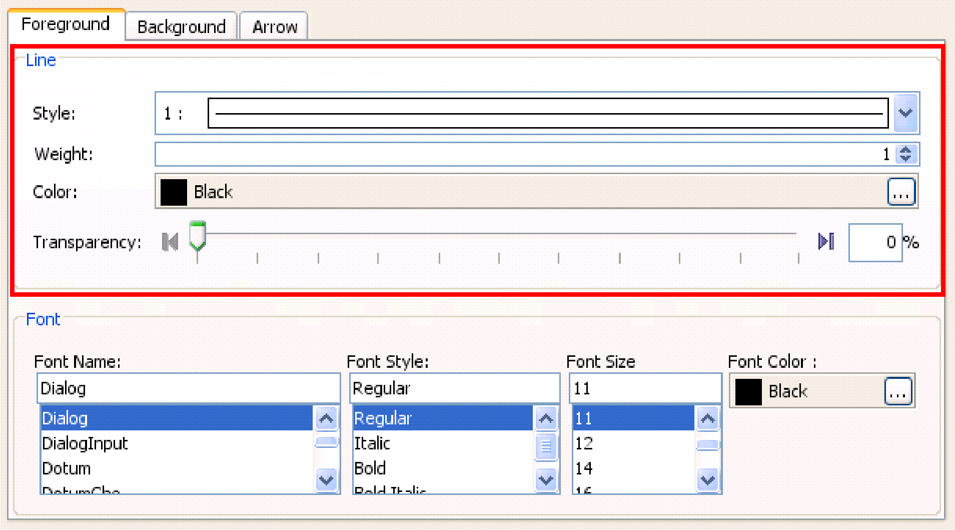
- Name


Edit Style

Name: Style1

__Figure 6-20__ edit name

- Foreground Line Style



Style: 1 :

Weight: 1

Color: Black

Transparency: 0 %

Font Name: Dialog

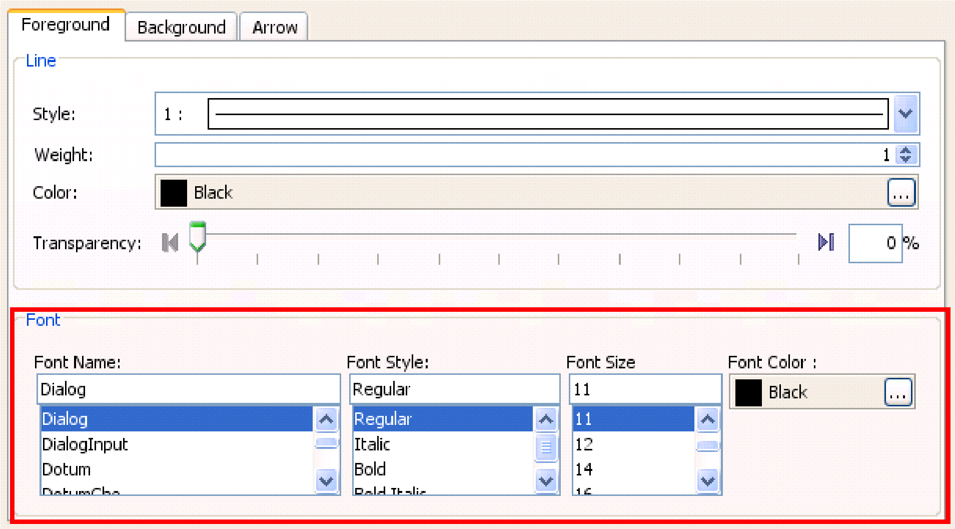
Font Style: Regular

Font Size 11

Font Color : Black

***Figure 6-21** edit line style*

- Font Style



Style: 1 :

Weight: 1

Color: Black

Transparency: 0 %

Font Name: Dialog

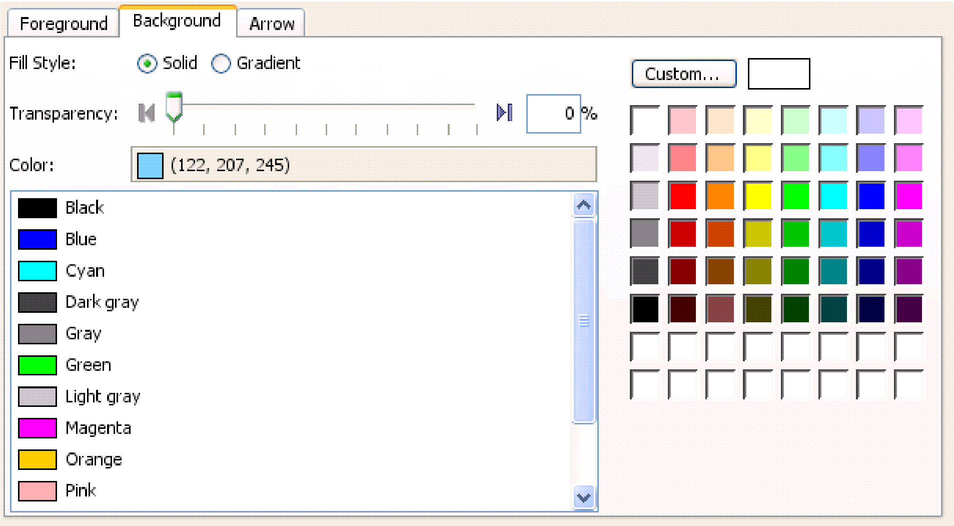
Font Style: Regular

Font Size 11

Font Color : Black

***Figure 6-22** edit font style*

- Background Style



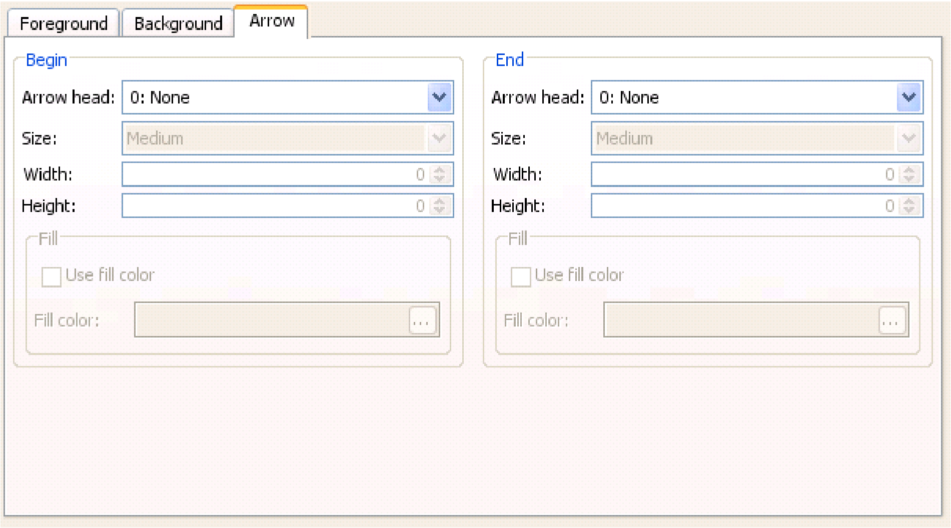
Fill Style: ● Solid ○ Gradient

Transparency: 0 %

Color: (122, 207, 245)

***Figure 6-23** edit background style*

- Arrow Style



Begin

Arrow head: 0: None

Size: Medium

Width: 0

Height: 0

Fill

☐ Use fill color

Fill color: ...

End

Arrow head: 0: None

Size: Medium

Width: 0

Height: 0

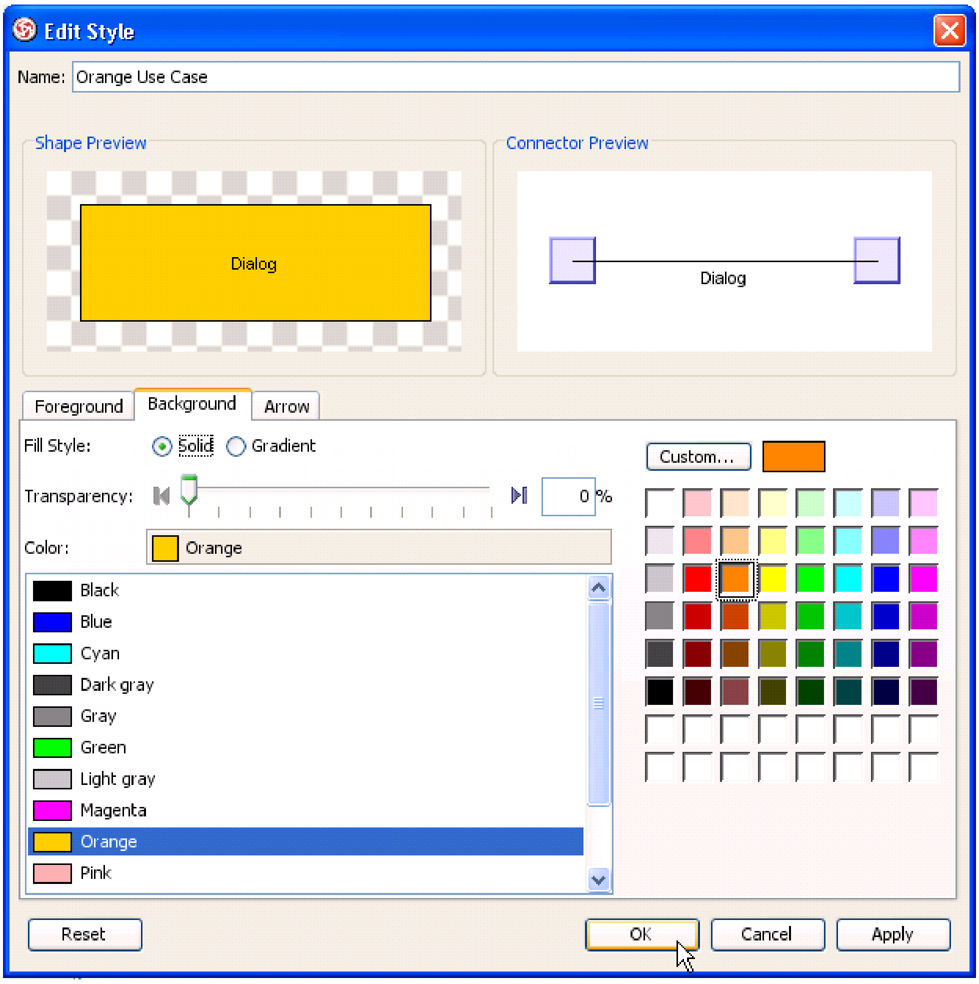
Fill

☐ Use fill color

Fill color: ...

Figure 6-24 edit arrow style

After change the settings, select **OK** to add the style.



Name: Orange Use Case

Fill Style: ● Solid ○ Gradient

Transparency: 0 %

Color: Orange

- Black
- Blue
- Cyan
- Dark gray
- Gray
- Green
- Light gray
- Magenta
- Orange
- Pink

Figure 6-25 style settings

The style was added to the project.

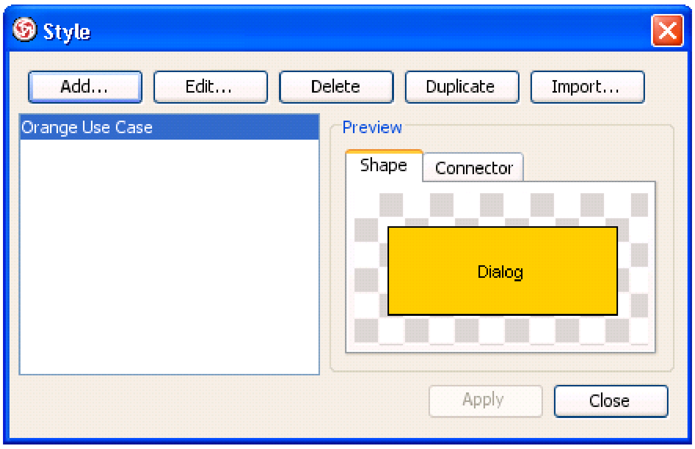


Figure 6-26 *style added*

Applying Style

Upon keeping the Style dialog open, create a new use case diagram together with a new use case.

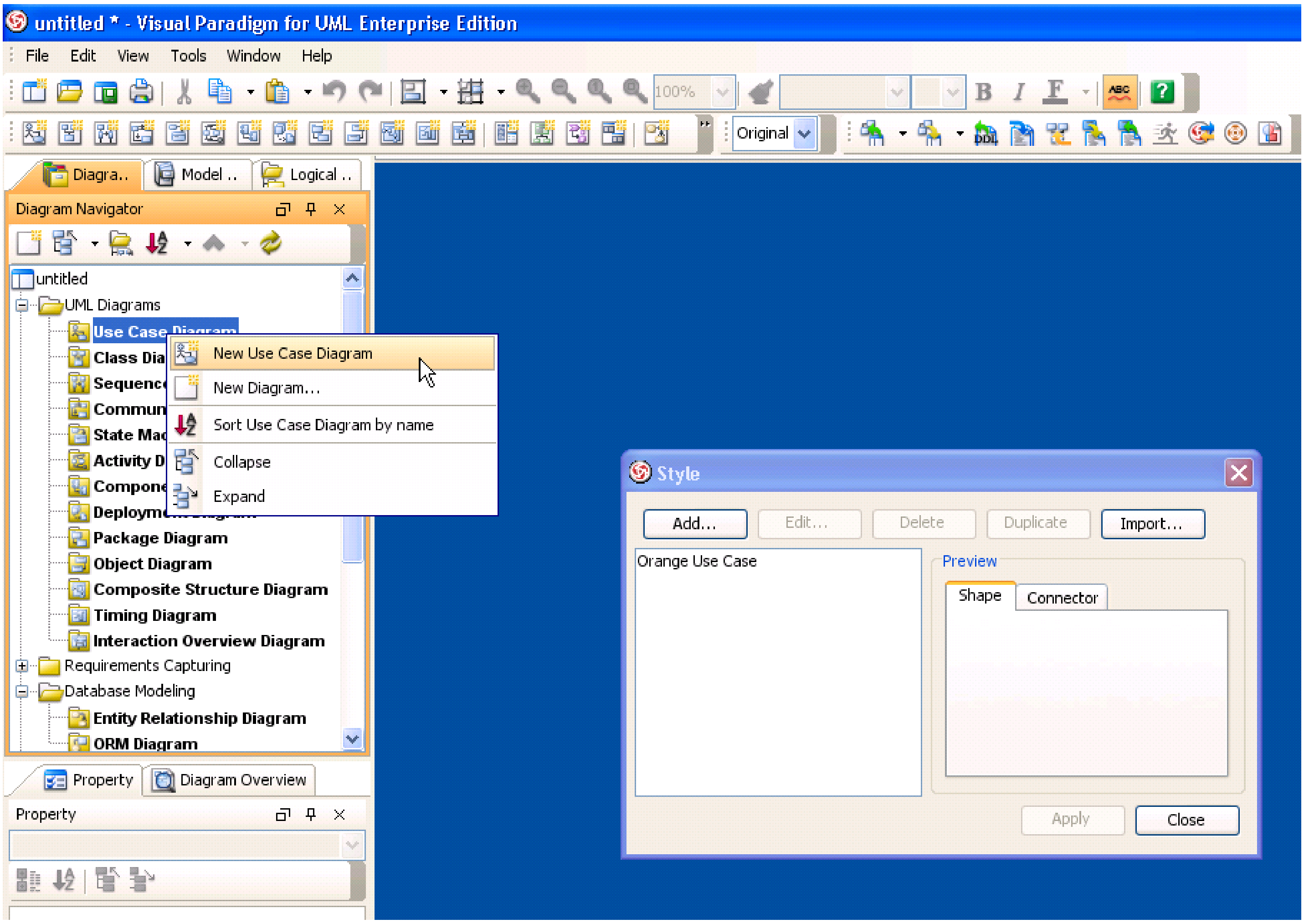


Figure 6-27 *New use case diagram*

Select the shape, and click **Apply** in the **Style** dialog.

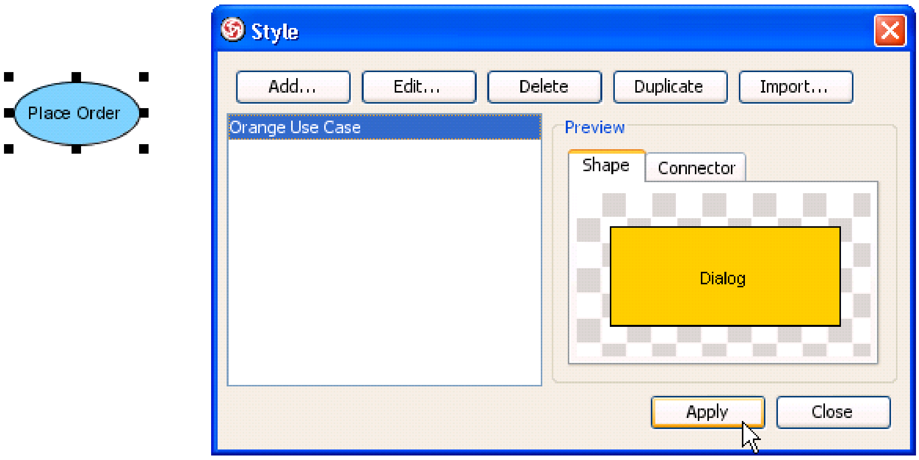


Figure 6-28 *Apply style to shape*

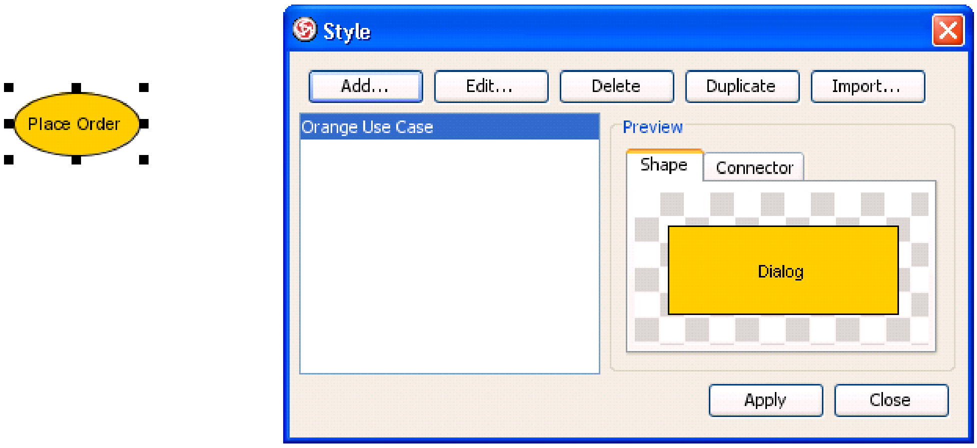


Figure 6-29 *Apply style result*

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, VP-UML provides five connector styles to help you handle the connectors, namely **Rectilinear**, **Oblique**, **Curve**, **Round Oblique** and **Round Rectilinear**.

Setting Connector Line Style

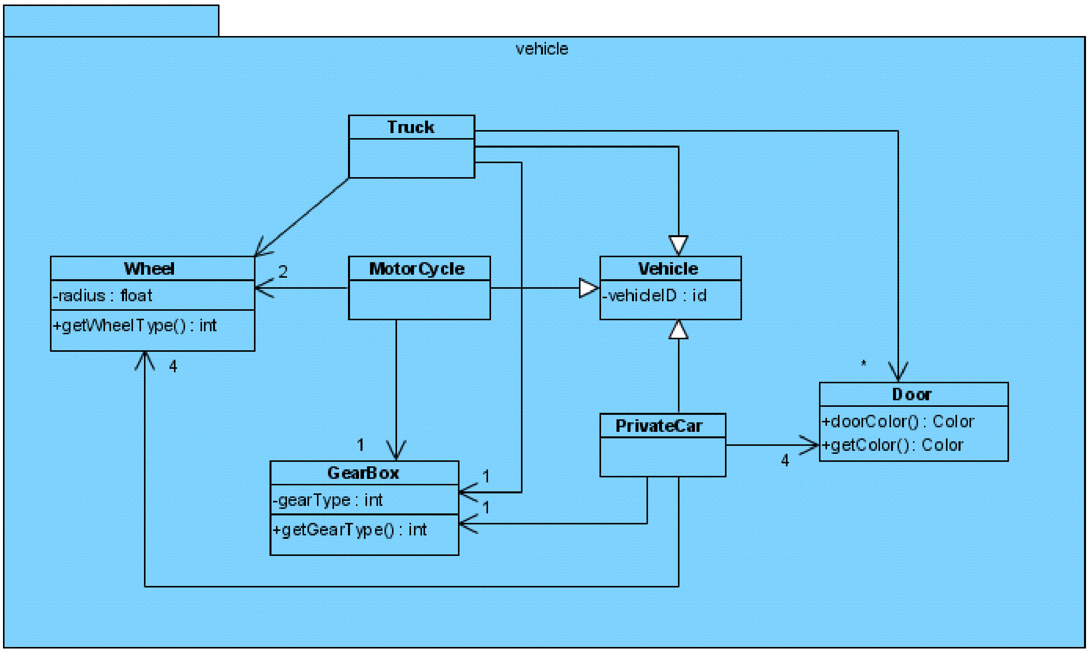


Figure 6-30 *Sample class diagram*

To change the line style, select the connector and select **Style and Formatting > Connector Style > Rectilinear** from the popup menu.

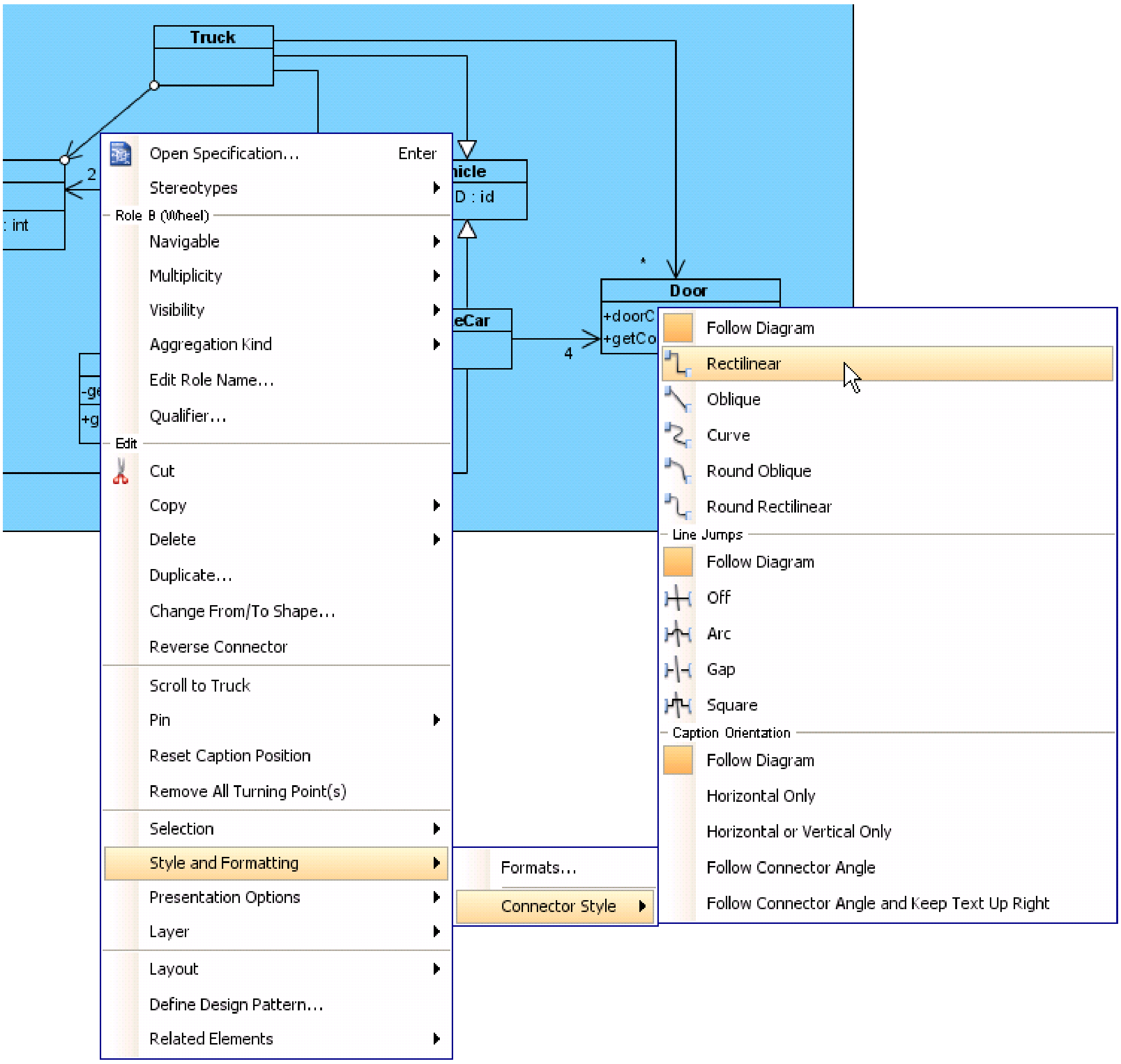


Figure 6-31 *Change line style*

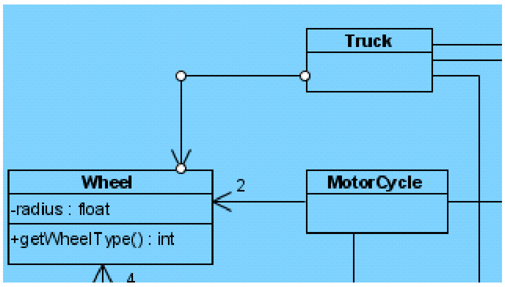
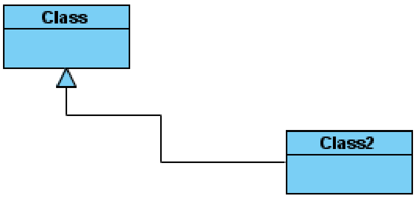
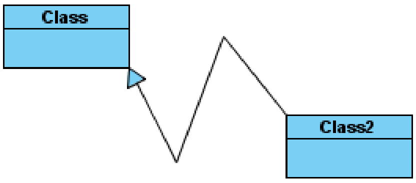
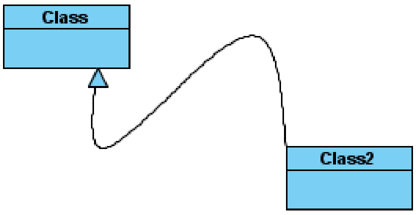
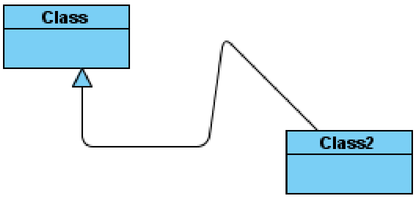
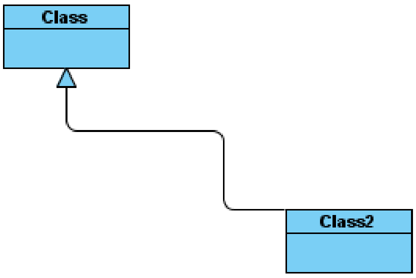


Figure 6-32 *Rectilinear style*

Line Style Options

Name	Sample
Rectilinear	

Oblique	
Curve	
Round Oblique	
Round Rectilinear	

Setting Diagram base Line Style

Beside the 5 style mentioned above, there also have **Follow Diagram** feature, you don't need to set it one by one if you want to change all connectors in the diagrams which defined as **Follow Diagram**.

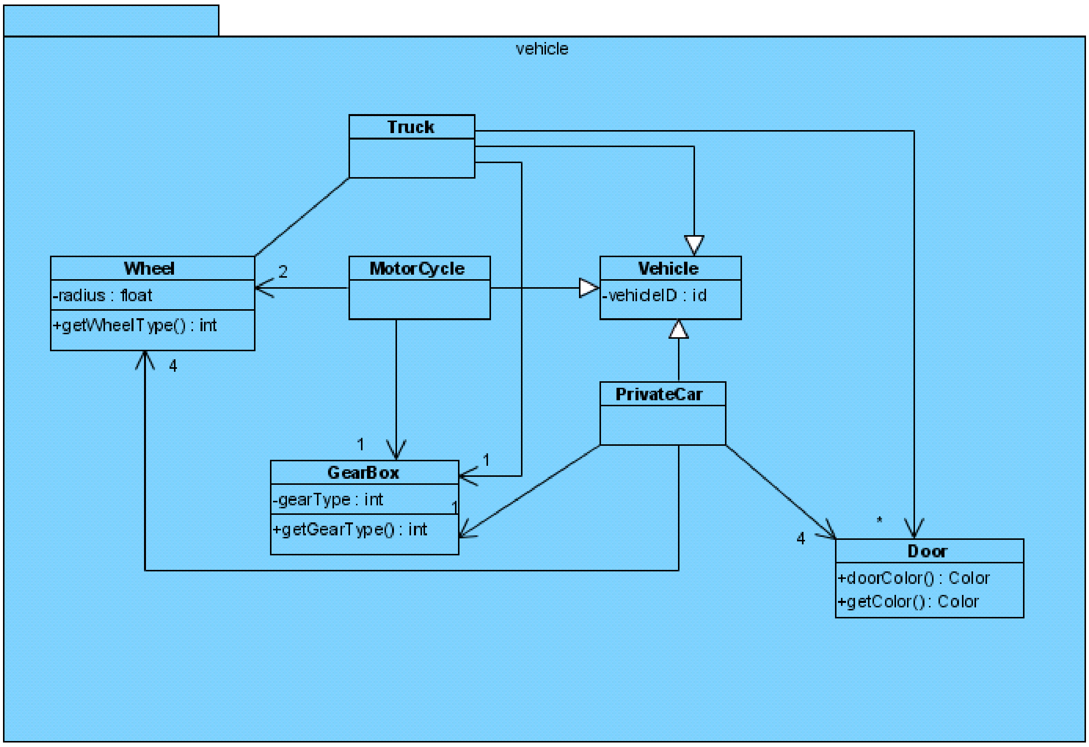


__Figure 6-33__ Sample class diagram 2

To change the diagram line style, right click on the diagram, and select **Connectors > Curve** from the popup menu.

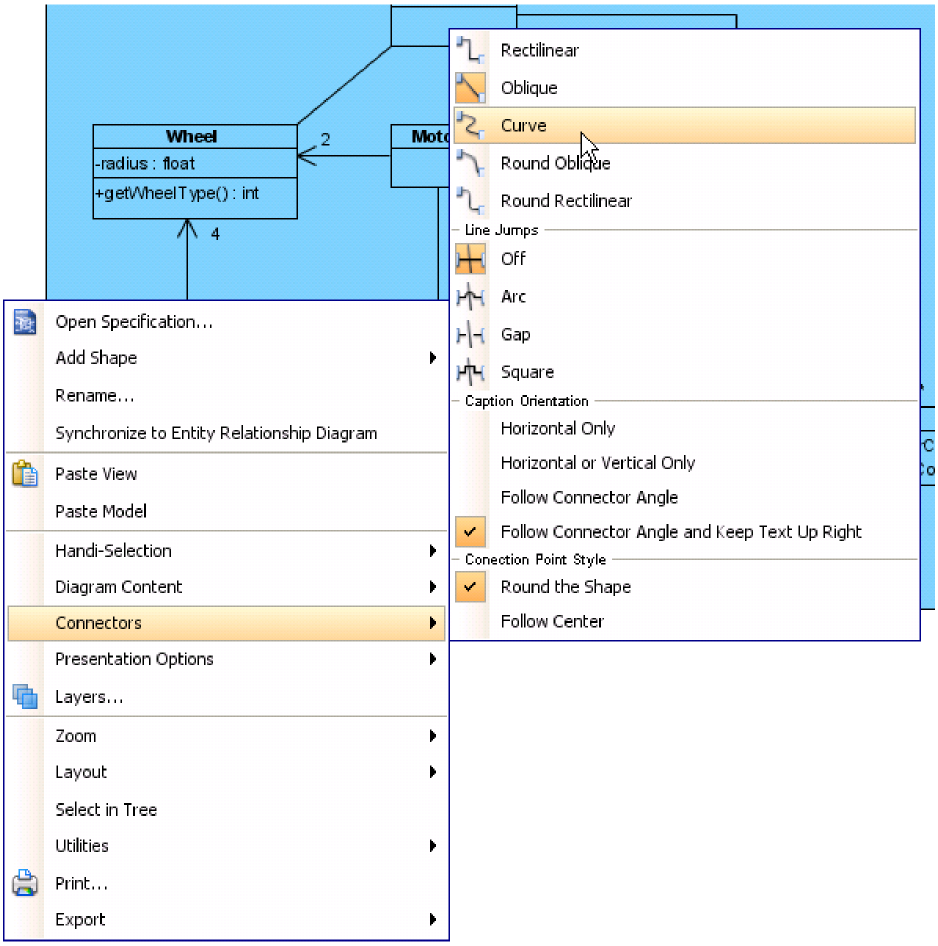


Figure 6-34 Change diagram line style

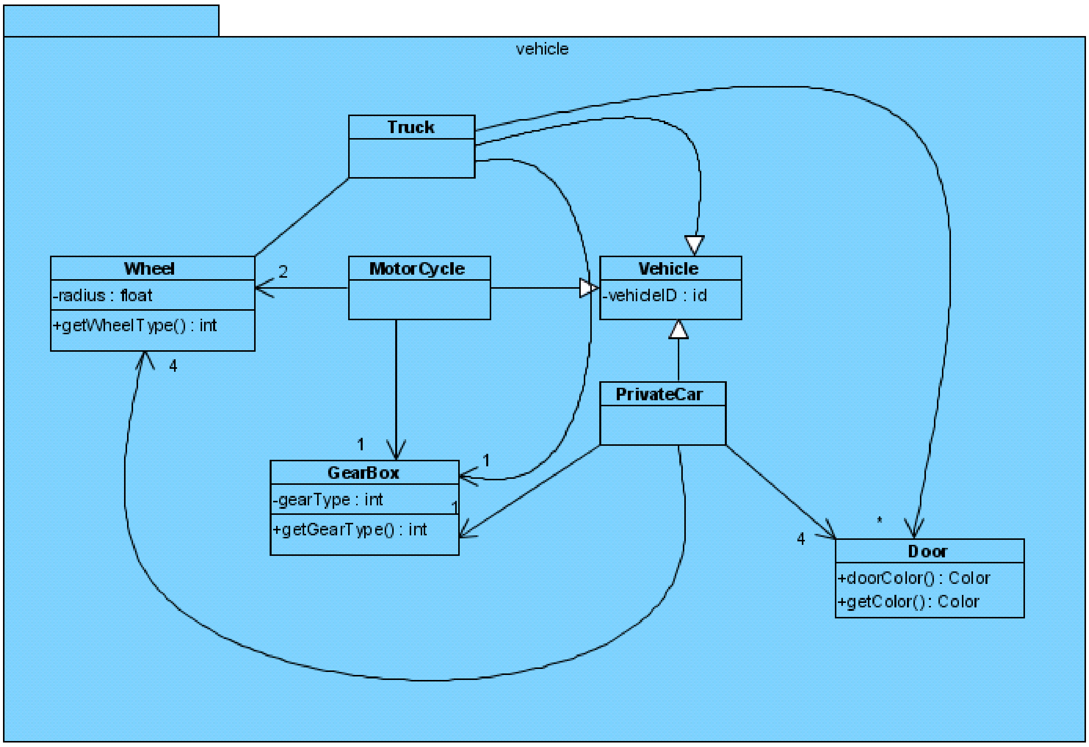


Figure 6-35 Curve line style

Setting Line Jumps Options

The line jumps option used for indicate the presentation if two connector was crossed.

Setting Connector Line Jumps Options

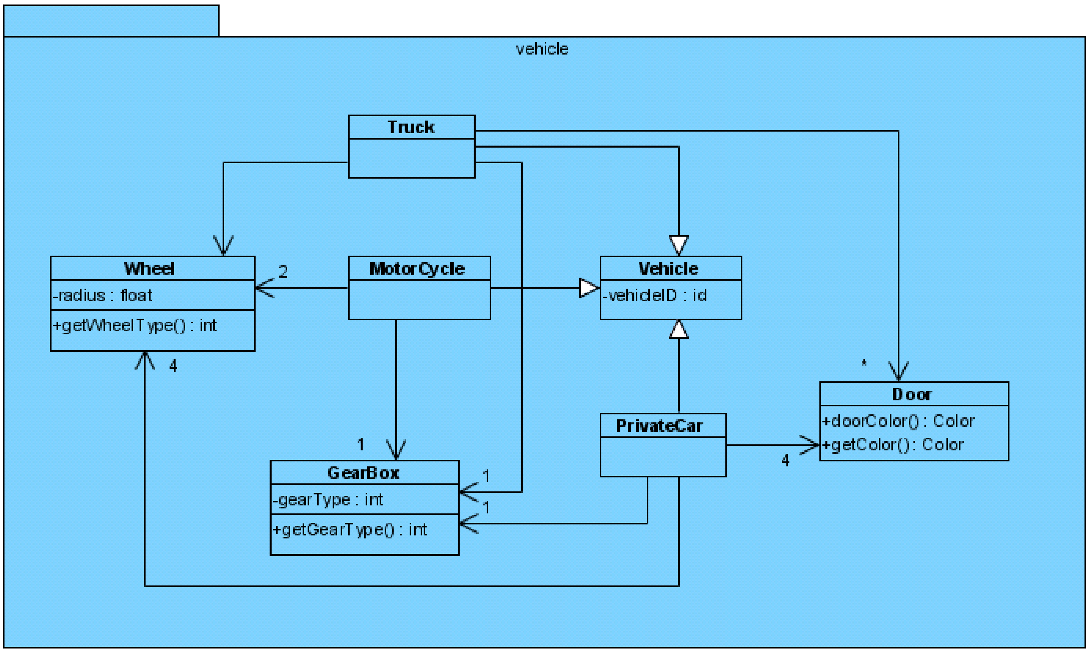


Figure 6-36 Sample class diagram

To change the jumps option of a connector, right click on the connector, and select **Style and Formatting > Connector Style > Arc**.

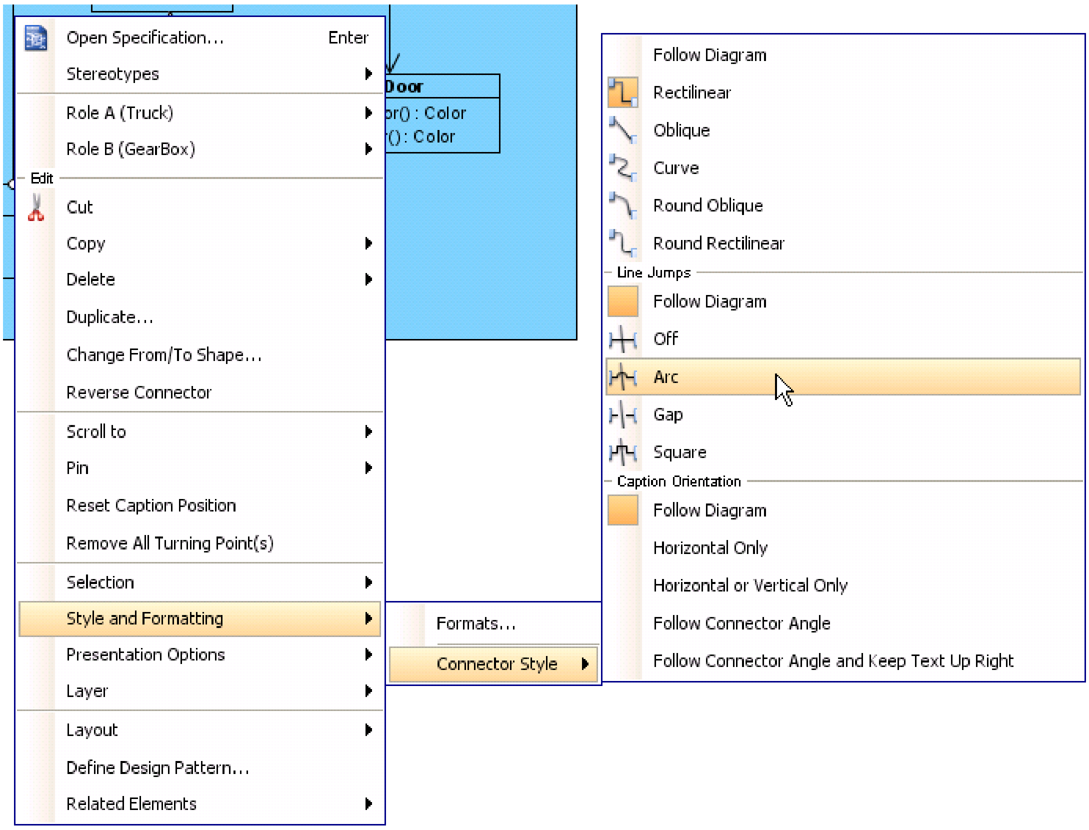


Figure 6-37 Change arc line jumps

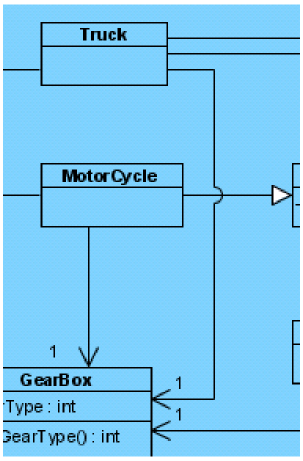
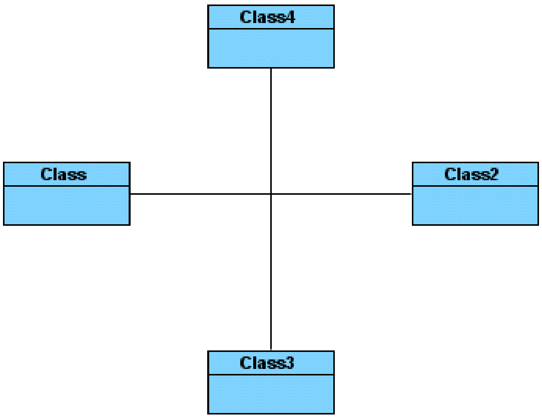
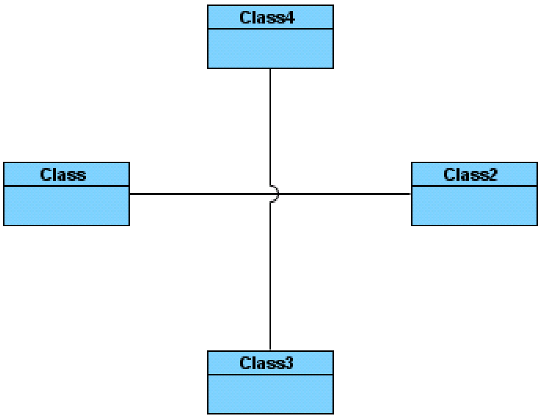


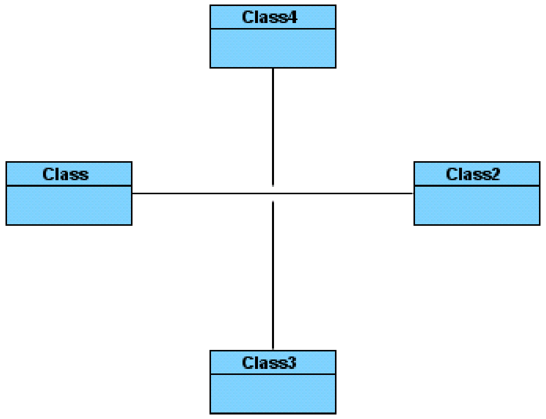
Figure 6-38 *arc sample*

NOTE: The line jumps option only appear if the connector was crossed by others.

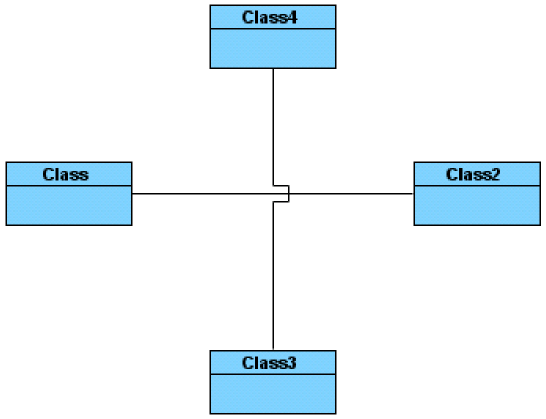
Line Jump Options

Name	Sample
Off	
Arc	

Gap



Square



Setting Diagram base Line Jumps Options

Beside the 4 options mentioned above, there also have **Follow Diagram** feature, you don't need to set it one by one if you want to change all connectors in the diagrams which defined as **Follow Diagram**.

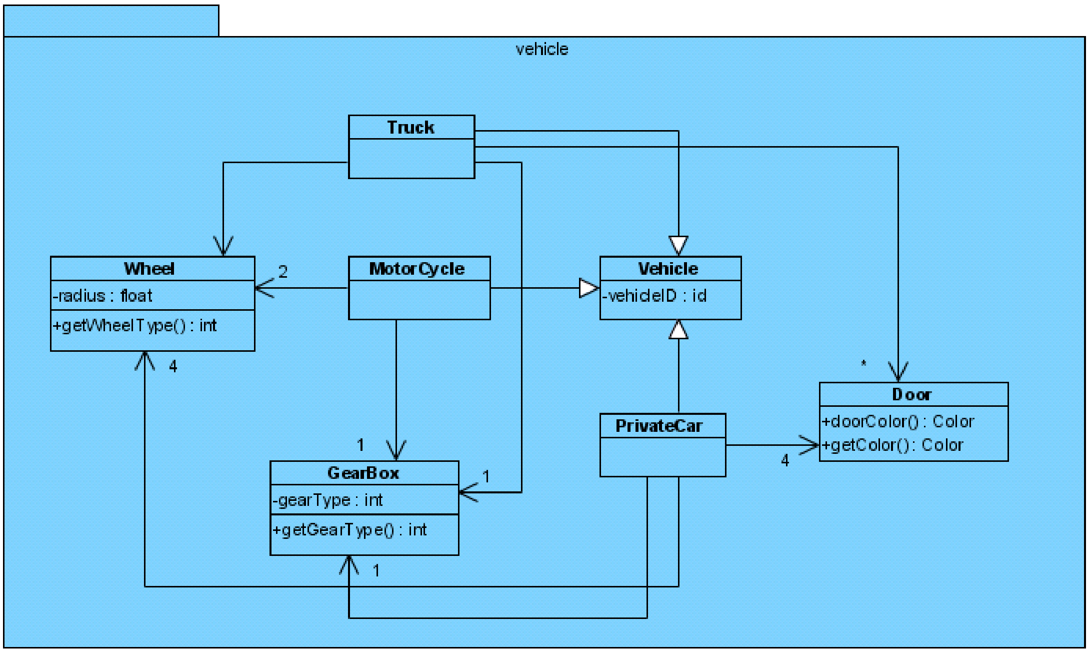


Figure 6-39 *Sample class diagram 2*

Click on the diagram, and select **Connectors > Gap** from the popup menu.

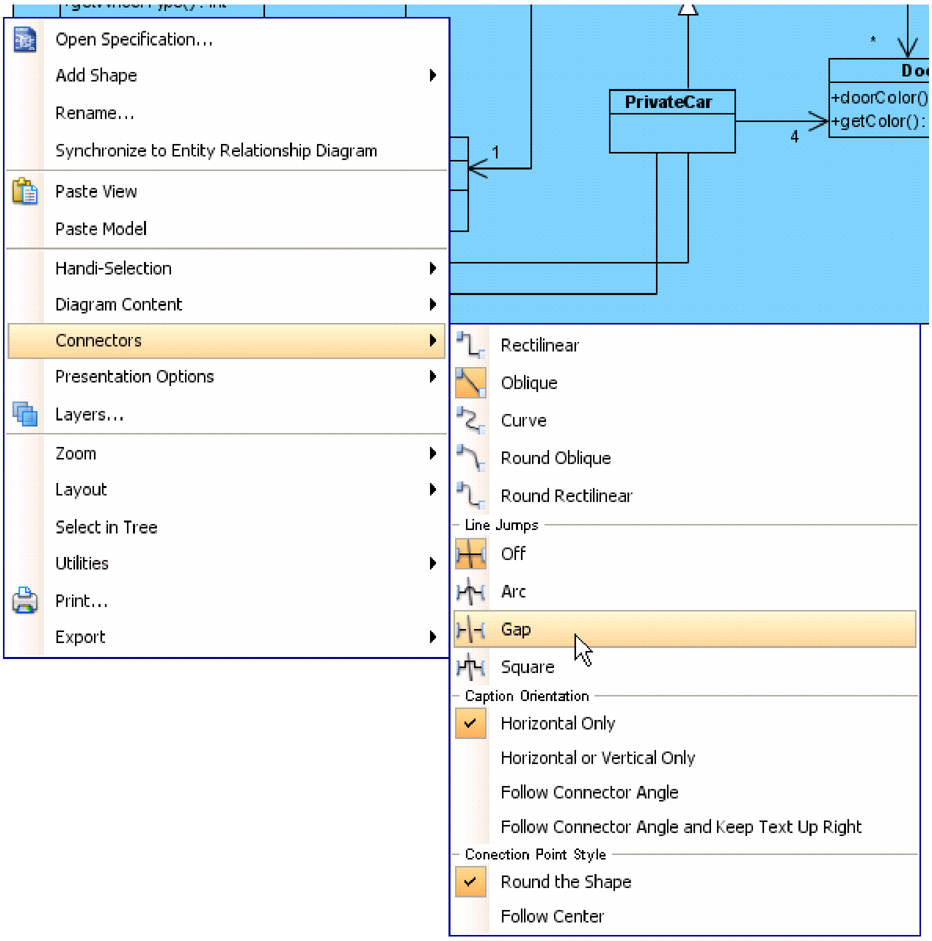


Figure 6-40 *Change gap line jumps*

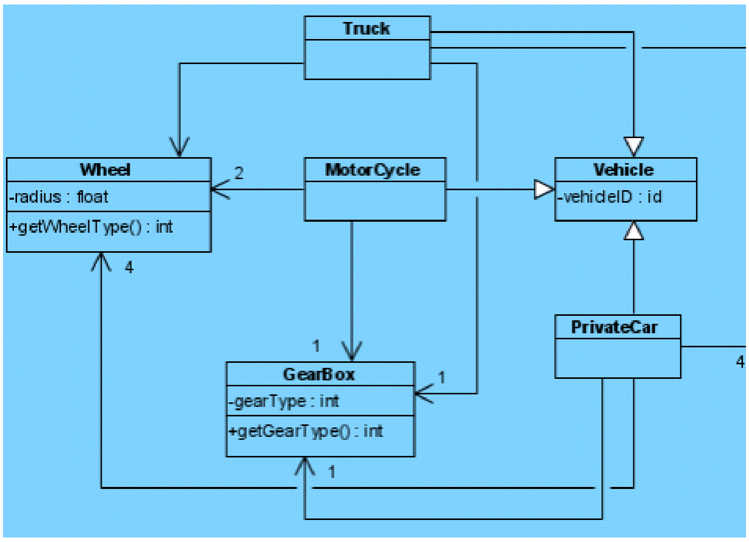


Figure 6-41 *gap sample*

Setting Connector Caption Orientation

By default, the caption of a connector is align horizontal only, but you also can customize it to **Follow Diagram**, **Horizontal Only**, **Horizontal or Verticle Only**, **Follow Connector Angle**, and **Follow Connector Angle and Keep Text Up Right**. You can either customize it one by one or change all connector in the diagram which defined **Follow Diagram**.

Setting Connector Caption Orientation

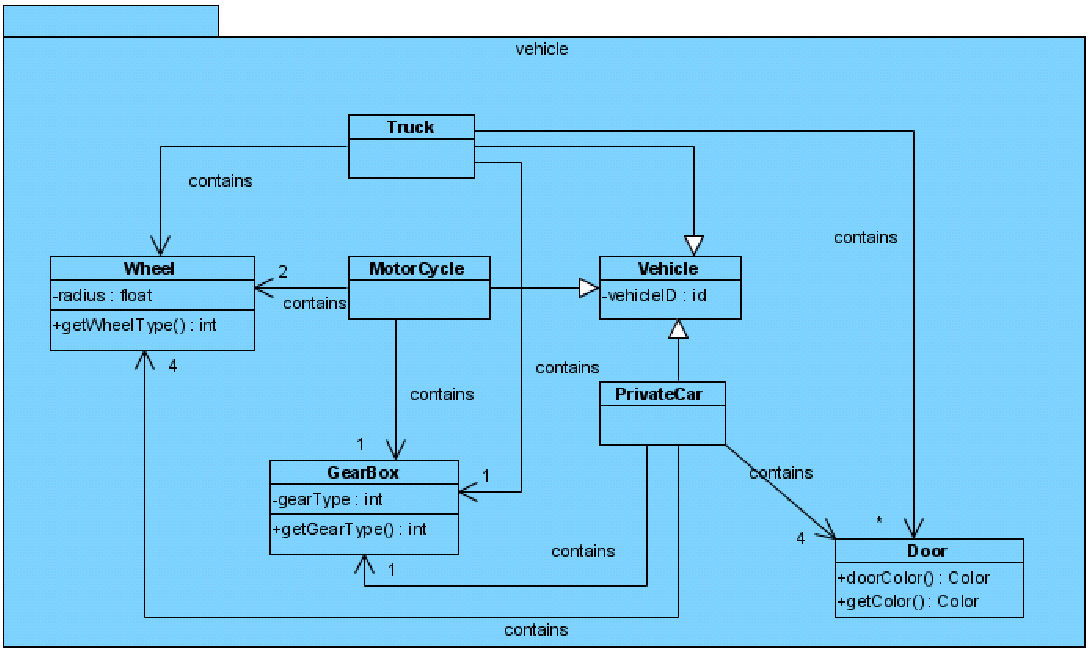


Figure 6-42 *Sample class diagram*

To customize the caption orientation option of a specify connector, select the connector, right click and select **Style and Formatting > Connector Style > Follow Connector Angle**.

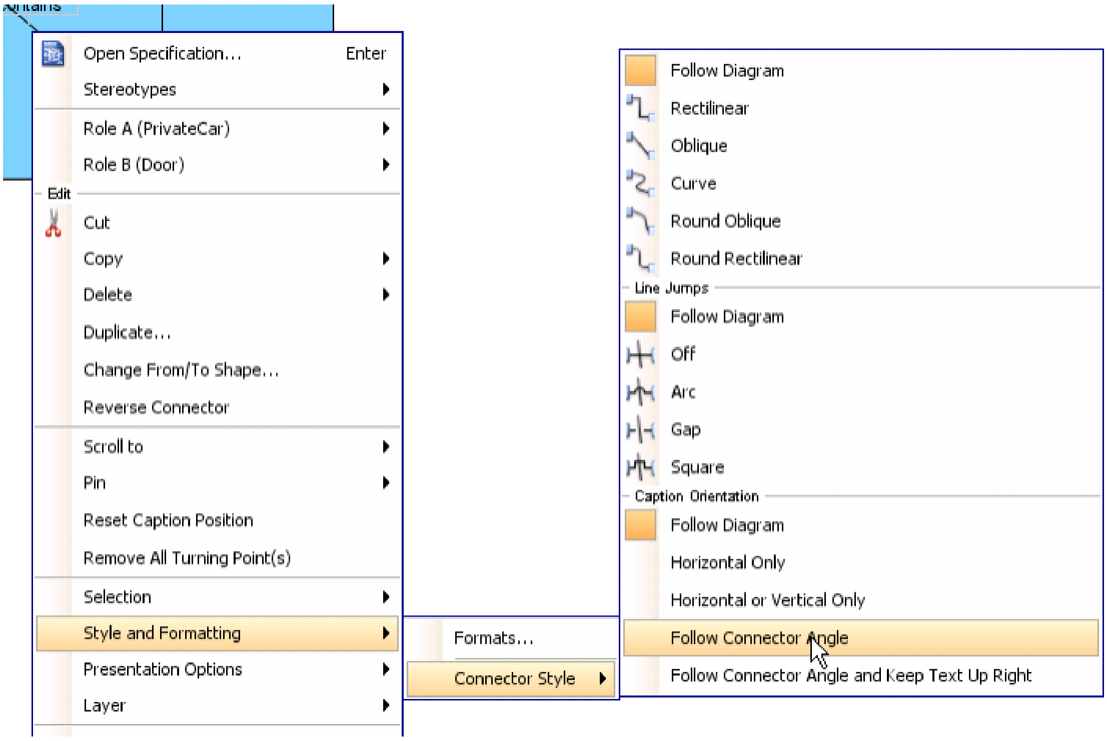


Figure 5-2 *change caption orientation to Follow Connector Angle*

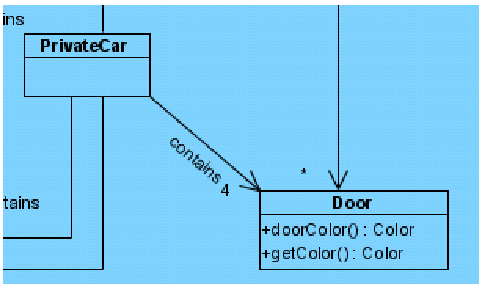
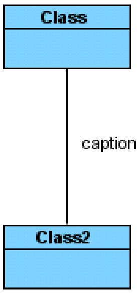
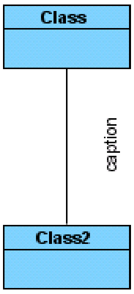
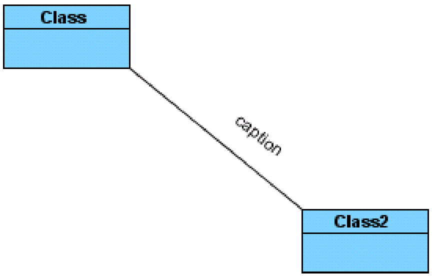
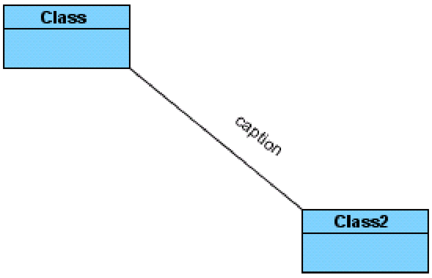


Figure 6-42 *Follow Connector Angle sample*

Caption Orientation Options

Name	Sample
Horizontal Only	
Horizontal or Verticle Only	
Follow Connector Angle	
Follow Connector Angle and Keep Text Up Right	

Setting Diagram base Connector Caption Direction

Beside the 4 options mentioned above, there also have **Follow Diagram** feature, you don't need to set it one by one if you want to change all connectors in the diagrams which defined as **Follow Diagram**.

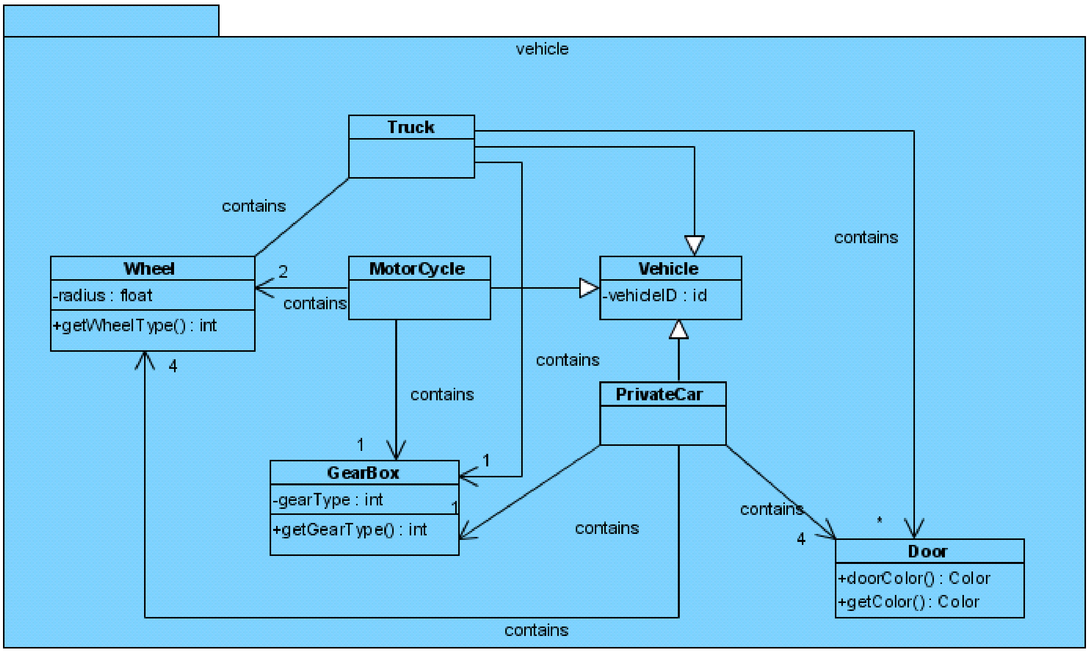


Figure 6-43 *Sample class diagram 2*

Click on the diagram, and select **Connectors > Follow Connector Angle and Keep Text Up Right** from the popup menu.

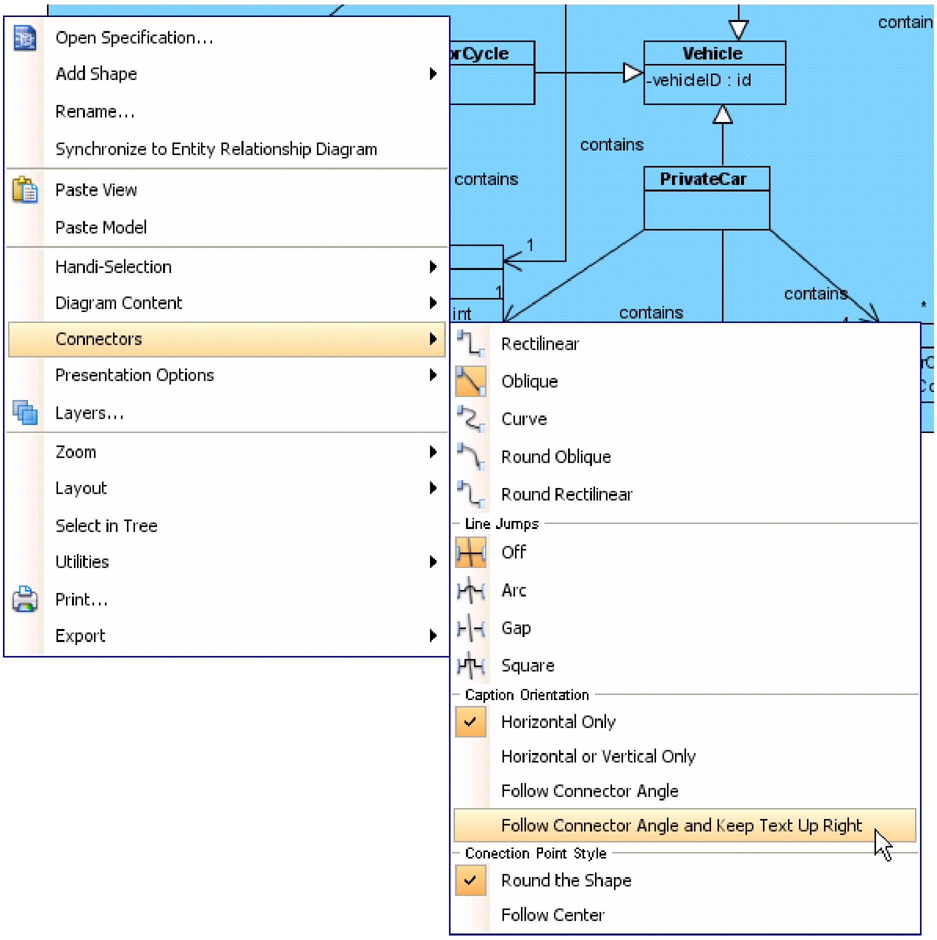


Figure 6-44 *Change caption orientation by diagram popup menu*

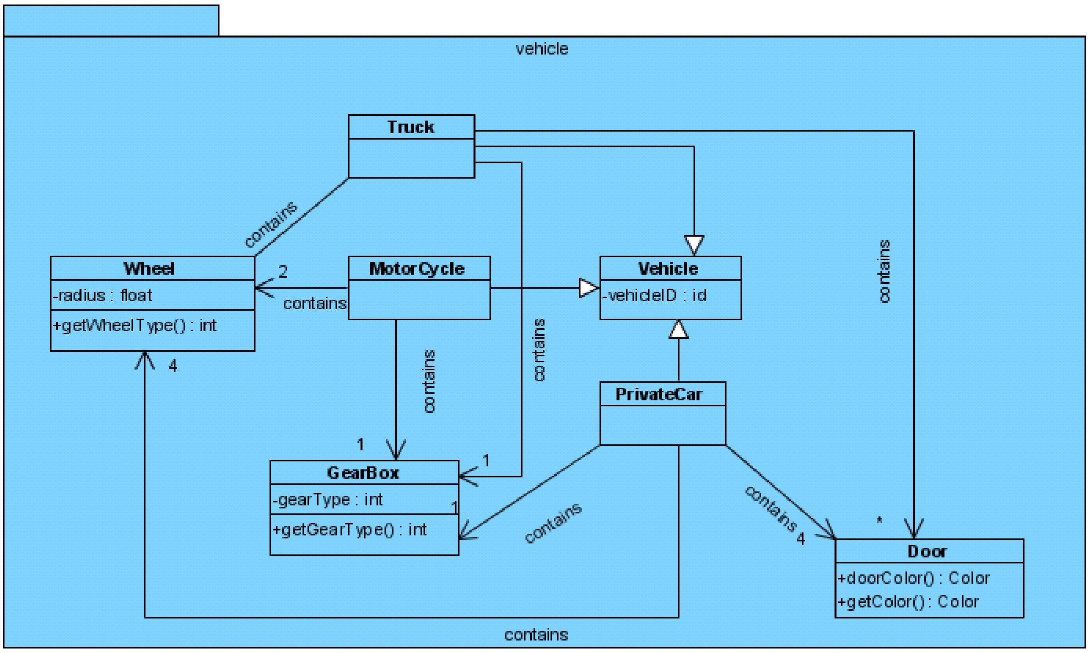


Figure 6-45 *Follow Connector Angle and Keep Text Up Right sample*

Creating User Interface Diagram

1. Select **File > New Diagram > Others > User Inteface** from the main menu.

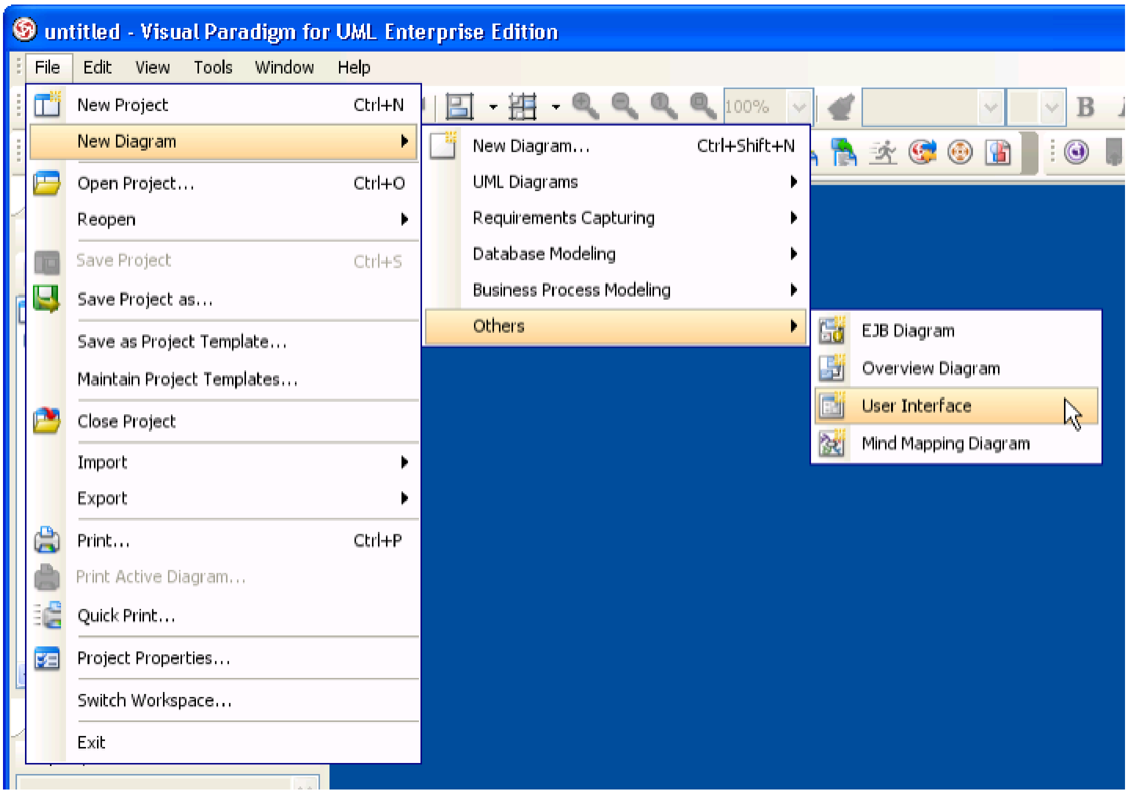


Figure 7-1 Create a User Interface diagram through the main menu

Alternatively, you can create a User Interface diagram with the steps below:
- Right click on **User Interface** in **Diagram Navigator** and select **New User Interface** from the popup menu.
- Click on the **New User Interface** button in toolbar
- Click on **New User Interface** in Start Pag

2. Enter the name of User Interface diagram.

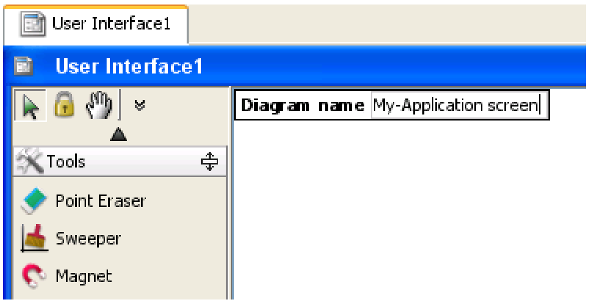


Diagram name My-Application screen

Figure 7-2 Enter the diagram name

3. Press the **Enter** key to confirm the diagram name. This creates a User Interface diagram.

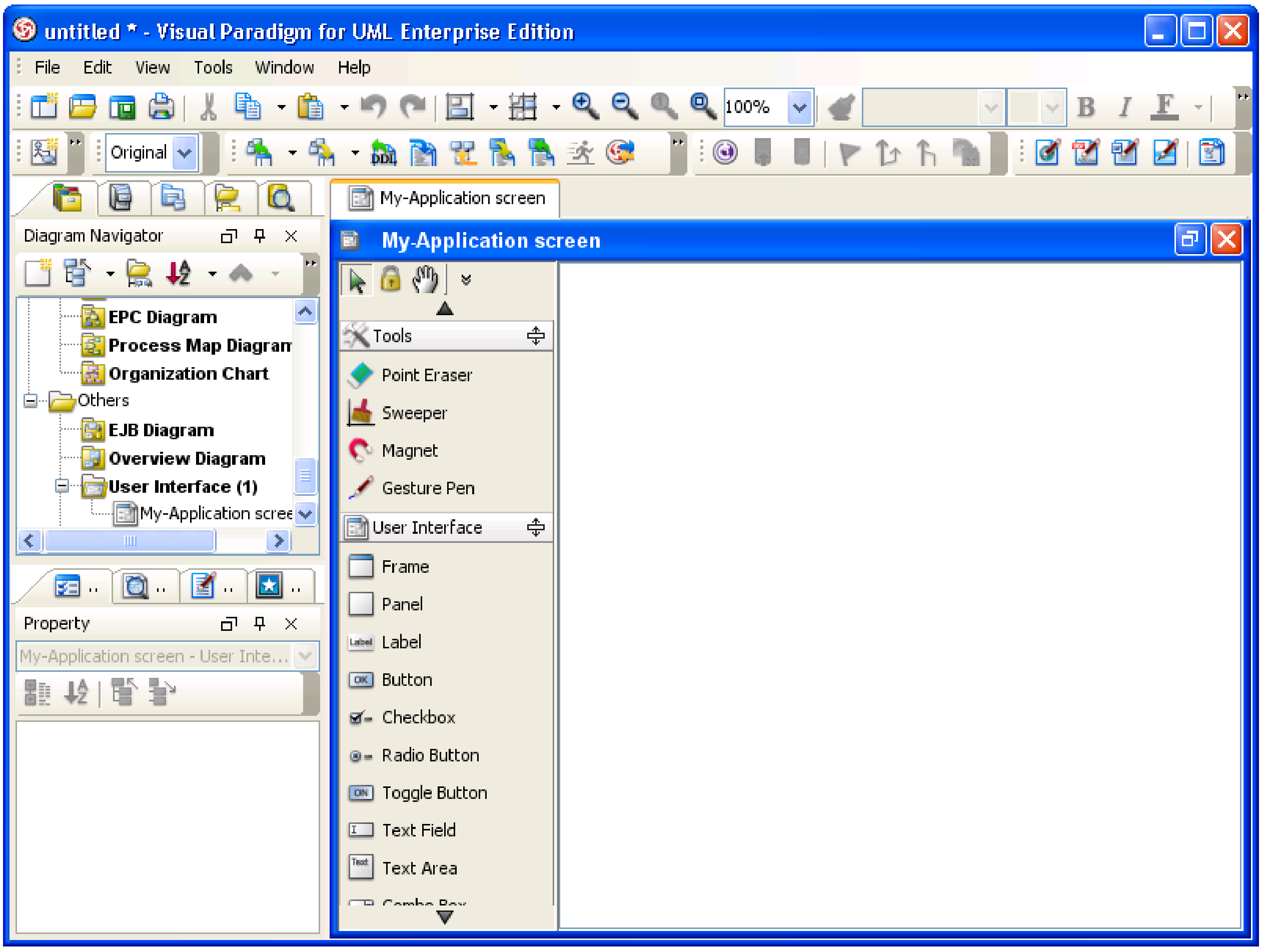


Figure 7-3 *A User Interface diagram is created*

Drawing Frame

A Frame is a top-level window with a title and a border. The size of the frame includes any area designated for the border. User can add components on a frame.

Creating a Frame

1. Select the **Frame** tool from the diagram toolbar.

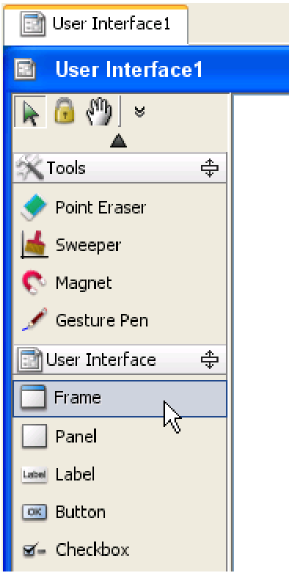


Figure 7-4 *Selecting Frame from diagram toolbar*

2. Click to set the starting point of the frame.
3. Drag diagonally from the starting point to expand the frame.

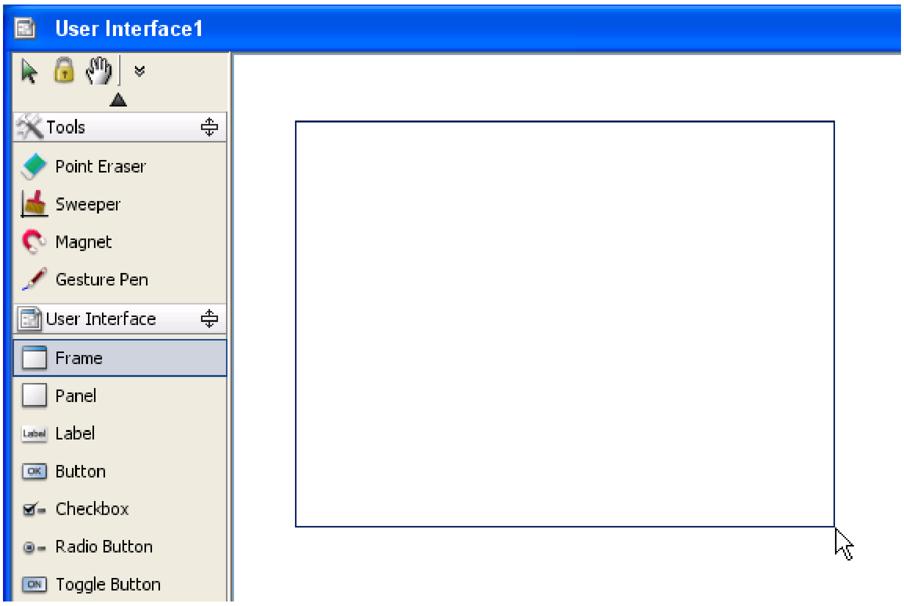
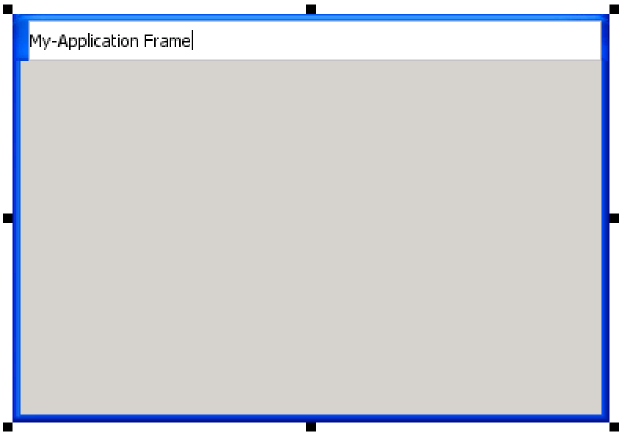


Figure 7-5 *Drag diagonally to expand the frame*

4. Release the mouse button to confirm the size of frame.

5. Enter the frame title.



My-Application Frame

Figure 7-6 Enter the frame title

Press the **Enter** key to confirm the title of frame. This creates the frame.

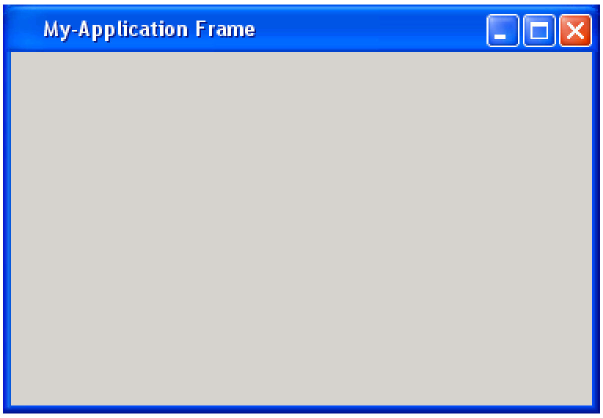


Figure 7-7 A frame is created

Frame Properties

There are several configurable properties for a frame. To configure a frame, right click on the Frame and select **Open Specification...** from the popup menu.

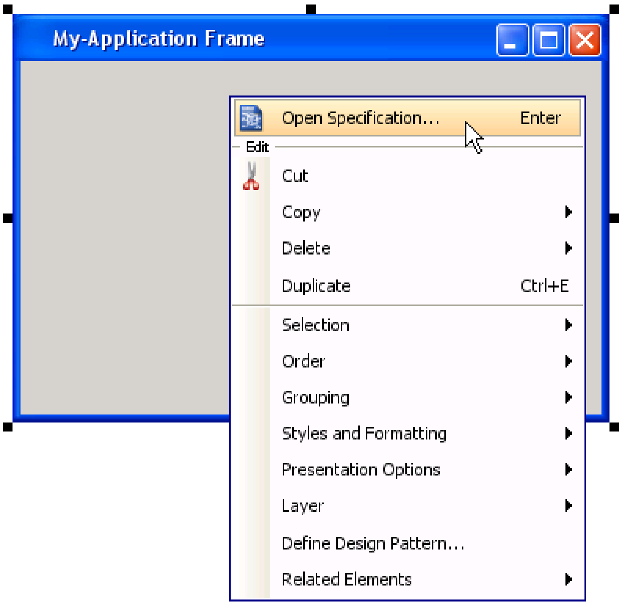
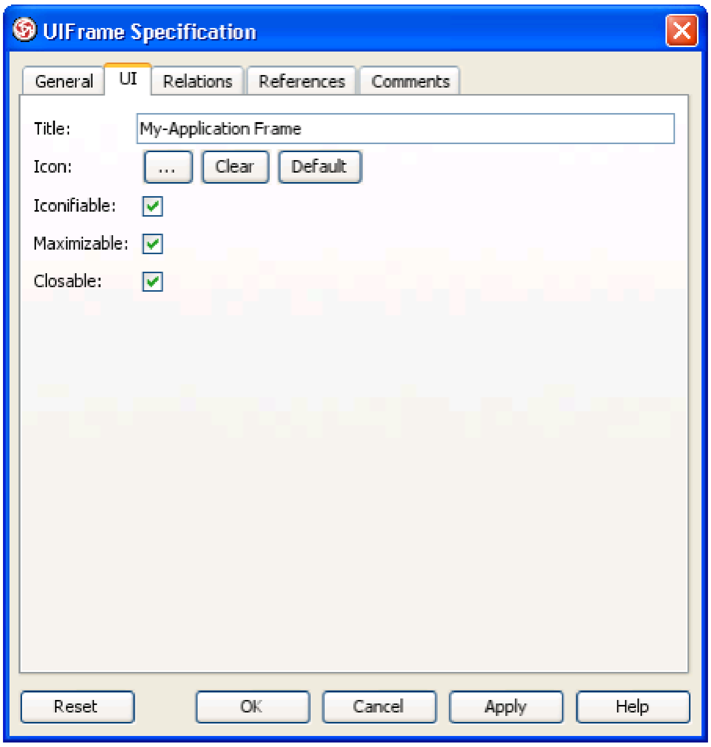


Figure 7-8 Opening frame specification

The **UI** tab is where user can configure a frame.



UIFrame Specification

Title: My-Application Frame

Icon: ... Clear Default

Iconifiable: ☑

Maximizable: ☑

Closable: ☑

Figure 7-9 *Frame specification's UI tab*

Title
Title refers to the text that appear at the top of frame, which use to be the caption of application or the function of opening frame.

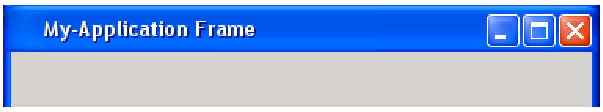


Figure 7-10 *A frame with title*

Icon
Icon is the tiny image that appear at the top left of frame. Users are required to provide a valid image file as icon.

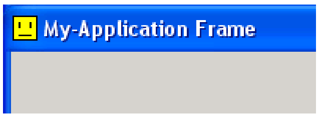


Figure 7-11 *A frame with icon*

Iconifiable
The Iconifiable state controls whether the minimize button is shown or not. When **Iconifiable** is set, the minimize button is shown, otherwise hidden.

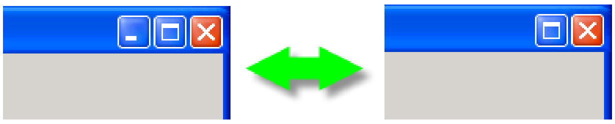


Figure 7-12 *Frames with and without minimize icon*

Maximizable
The Maximizable state controls whether the maximize button is shown or not. When **Maximizable** is set, the maximize button is shown, otherwise hidden.

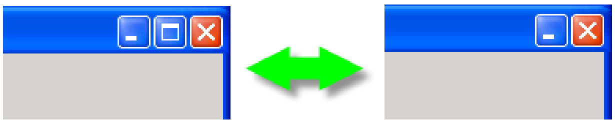


Figure 7-13 *Frames with and without maximize icon*

Closable

The Closable state controls whether the close button is shown or not. When **Closable** is set, the close button is shown, otherwise hidden.

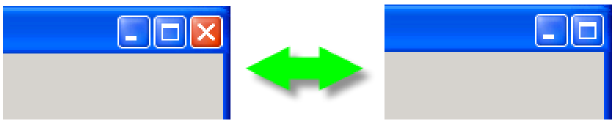


Figure 7-14 *Frames with and without close icon*

Drawing Label

A label is a text component that appear on a screen. The text "User", "Password", "Address" appear on a registration form are examples of labels.

Creating a Label

1. Select the **Label** tool from the diagram toolbar.

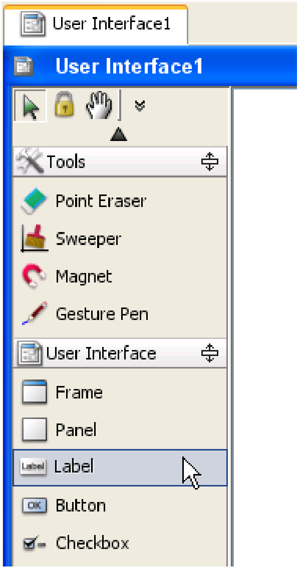


Figure 7-15 *Selecting Label from diagram toolbar*

2. Click on a container (e.g. a Frame) or diagram background to create a label.

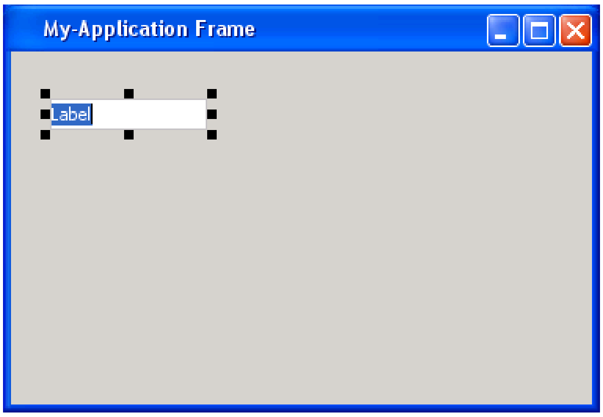


Figure 7-16 *Enter the label title*

3. Enter the label caption.

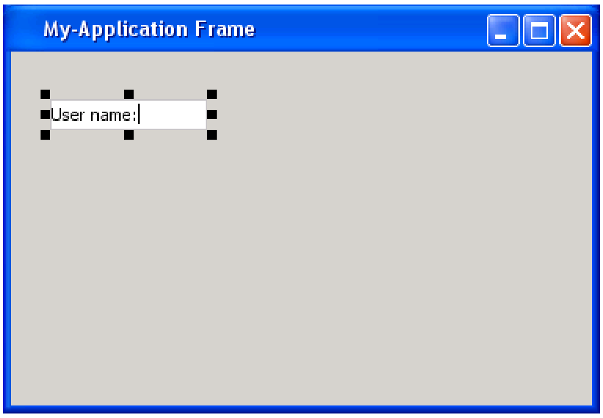


Figure 7-17 *Enter the label caption*

Press the **Enter** key to confirm the caption of label. This creates the label.

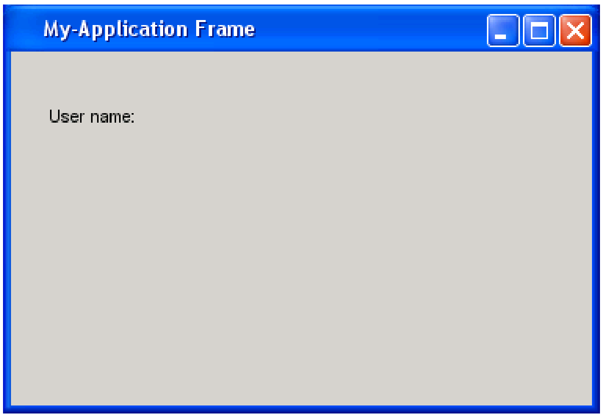


Figure 7-18 *A label is created*

Label Properties

There are several configurable properties for a label. To configure a label, right click on the label and select **Open Specification...** from the popup menu.

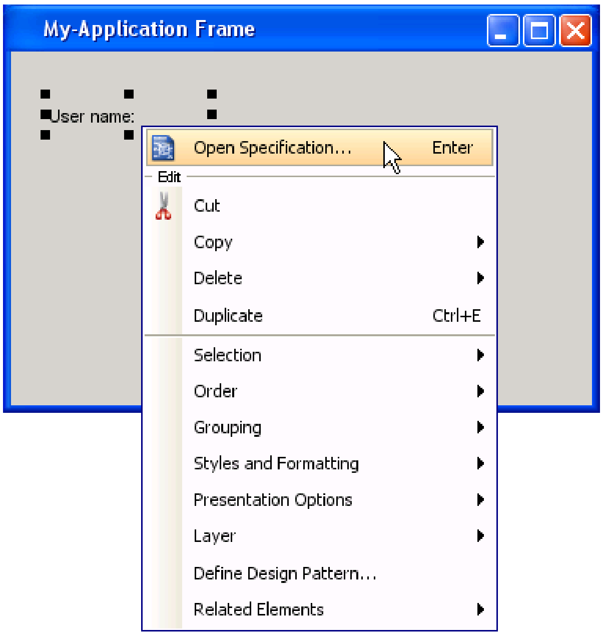
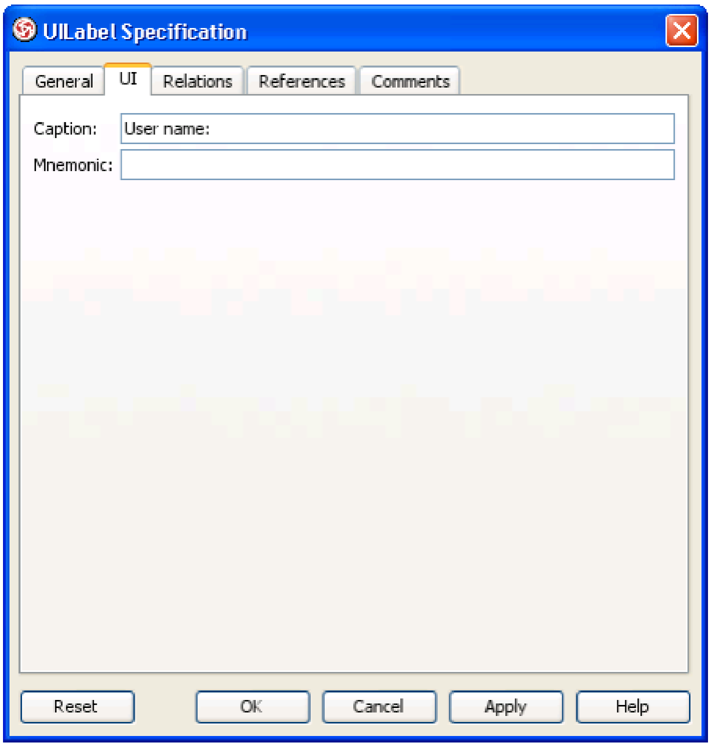


Figure 7-19 *Opening label specification*

The **UI** tab is where user can configure a label.



Caption: User name:

Mnemonic:

Figure 7-20 Label specification's UI tab

Caption

Caption is the text appear in a label.

User name:

Figure 7-21 A label with caption

Mnemonic

Mmnemonic is a key which enables users to select a label by simultaneously pressing the Alt key and the mmnemonic key on the keyboard.

Drawing Text Field

Text Field is a component that allows the editing of a single line of text. Fields in a registration form are typical examples of text fields.

Creating a Text Field

1. Select the **Text Field** tool from the diagram toolbar.

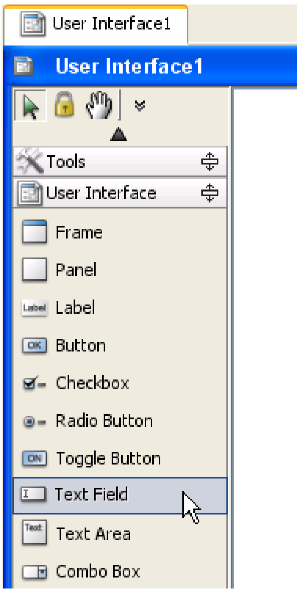


Figure 7-22 *Selecting Text Field from diagram toolbar*

2. Click to set the starting point of the text field.
3. Drag diagonally from the starting point to expand the text field.

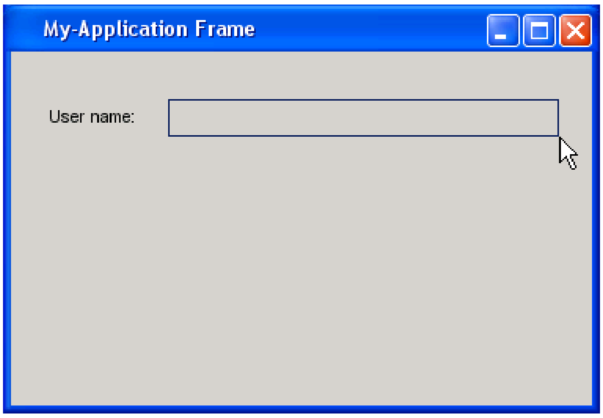


Figure 7-23 *Drag diagonally to expand the text field*

4. Release the mouse button to confirm the size of text field.

5. Enter the text field text.

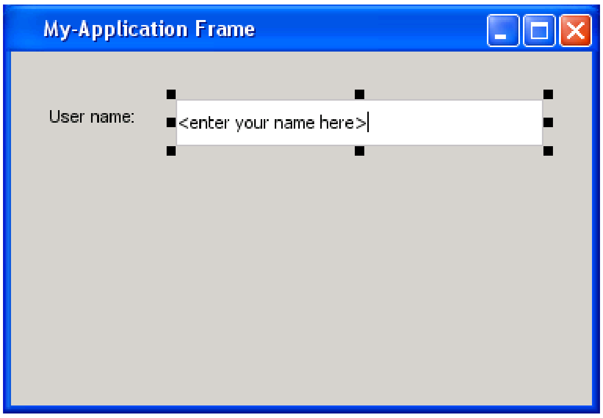


Figure 7-24 *Enter the text field title*

Press the **Enter** key to confirm the title of text field. This creates the text field.

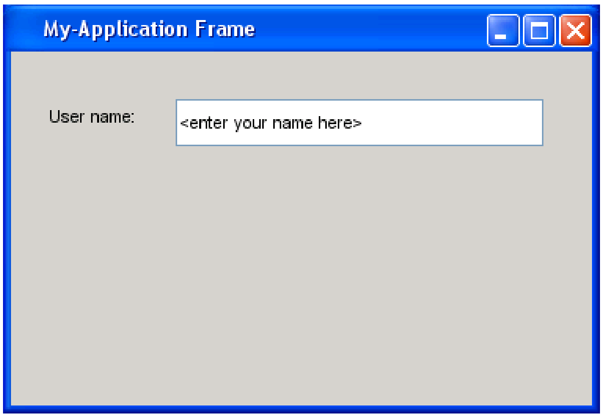


Figure 7-25 *A text field is created*

Text Field Properties

There are several configurable properties for a text field. To configure a text field, right click on the text field and select **Open Specification...** from the popup menu.

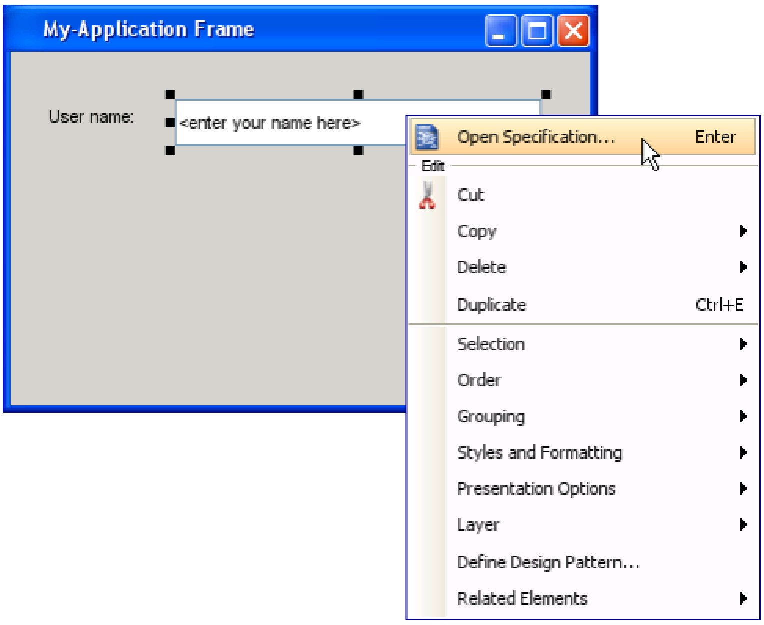
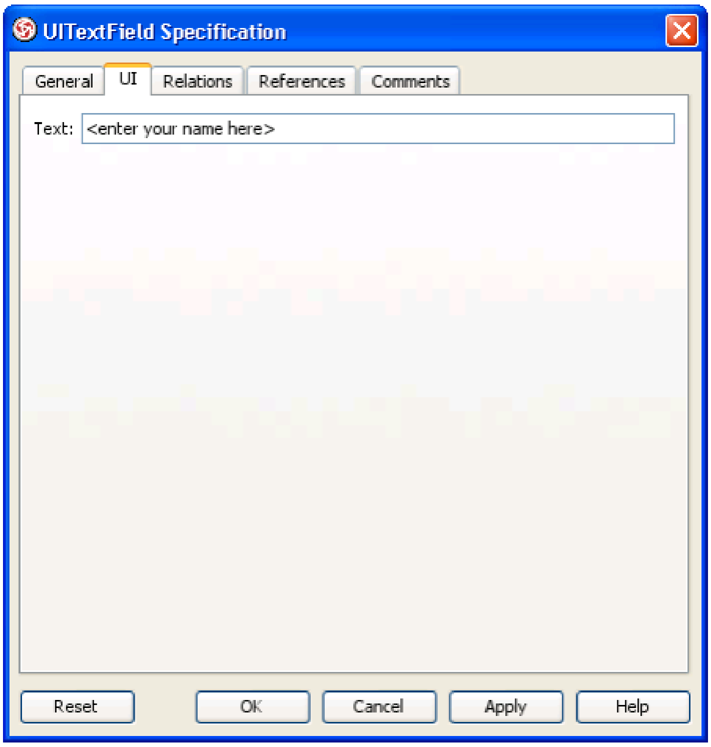


Figure 7-26 *Opening text field specification*

The **UI** tab is where user can configure a text field.



Text: <enter your name here>

Figure 7-27 *Text Field specification's UI tab*

Text
Text refers to the text that appear in the text field.

Drawing Scrollbar

A scrollbar is a bar with a kbow which appear at the bottom or right of a container, such as a frame. The user positions the knob in the scrollbar to determine the contents of the viewing area. The program typically adjusts the display so that the end of the scrollbar represents the end of the displayable contents, or 100% of the contents. The start of the scrollbar is the beginning of the displayable contents, or 0%. The position of the knob within those bounds then translates to the corresponding percentage of the displayable contents.

Creating a Scrollbar

1. Select the **Horizintal/Vertical Scrollbar** tool from the diagram toolbar.

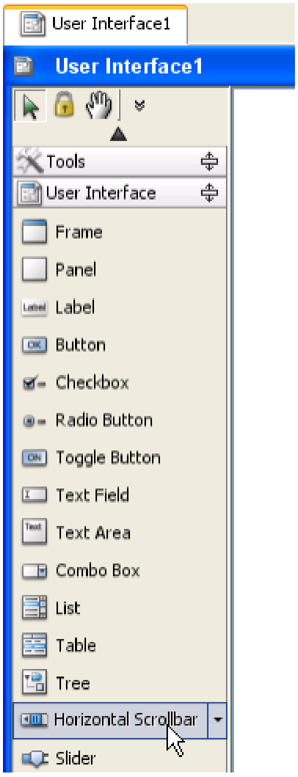


Figure 7-28 *Selecting Horizontal Scrollbar from diagram toolbar*

You can change between a horizontal or a vertical scrollbar tool by clicking on the tiny reverted triange, and selecting the correct type of scrollbar from the popup menu.

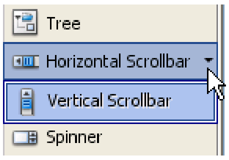


Figure 7-29 *Selecting horizontal or vertical scrollbar*

2. Click on a container (e.g. a Frame) or diagram background to create a scrollbar. If you create inside a container, and if it has no scrollbar with same orientation exists, the scrollbar will be docked to the bottom or the right of the container, depending on the scrollbar orientation.

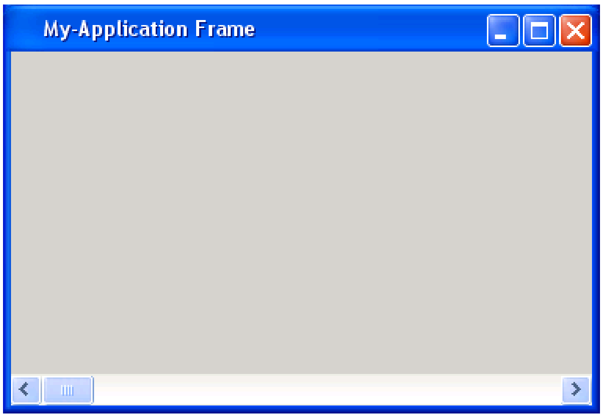


Figure 7-30 *Scrollbar is created*

Change Orientation with Resource-Centric Interface

A scrollbar can be oriented horizontally or vertically, which determines the contents of the viewing area when being scrolled. User can change the orientation of a scrollbar easily through the resource-centric interface. When a scrollbar is put inside a container, changing orientation will automatically dock to the bottom or the right of container. To change scrollbar orientation:

1. Move the mouse cursor over the scrollbar.

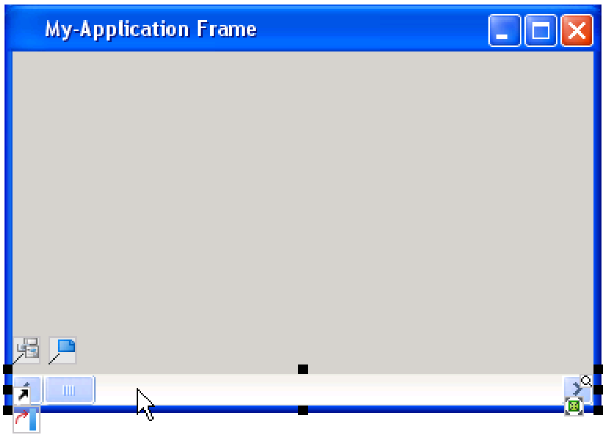


Figure 7-31 *Moving cursor over a scrollbar to display the resource icons*

2. Click on the resource icon **Switch Orientation**.

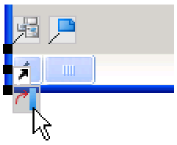


Figure 7-32 *Click on the resource icon for switching scrollbar orientation*

The orientation of scrollbar then switched.

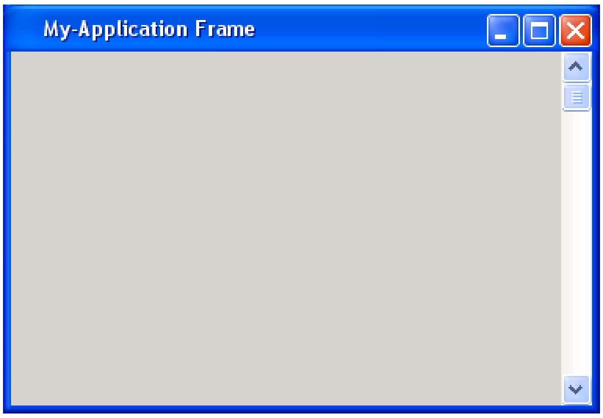


Figure 7-33*Scrollbar orientation switched from horizontal to vertical*

Automatic Sticking Scrollbar to Frame

In a normal User Interface design, scrollbars, no matter horizontal or vertical, are put at the bottom or the right of a container component. VP-UML follows this pratice. When creating a scrollbar in a container component, or when moving a scrollbar into a container component, the scrollbar will be docked to the border of component automatically.

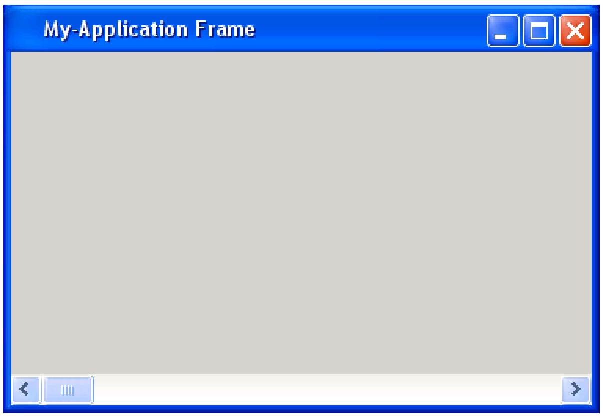


Figure 7-34 Scrollbar docked to the bottom of frame

Scrollbar Properties

There are several configurable properties for a scrollbar. To configure a scrollbar, right click on the label and select **Open Specification...** from the popup menu.

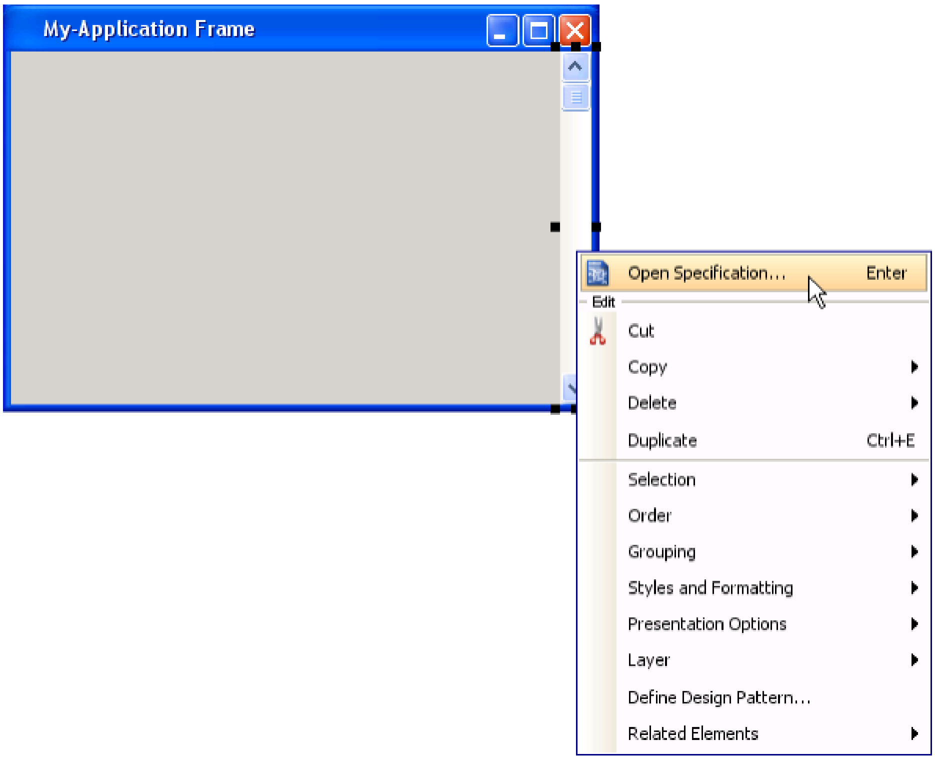
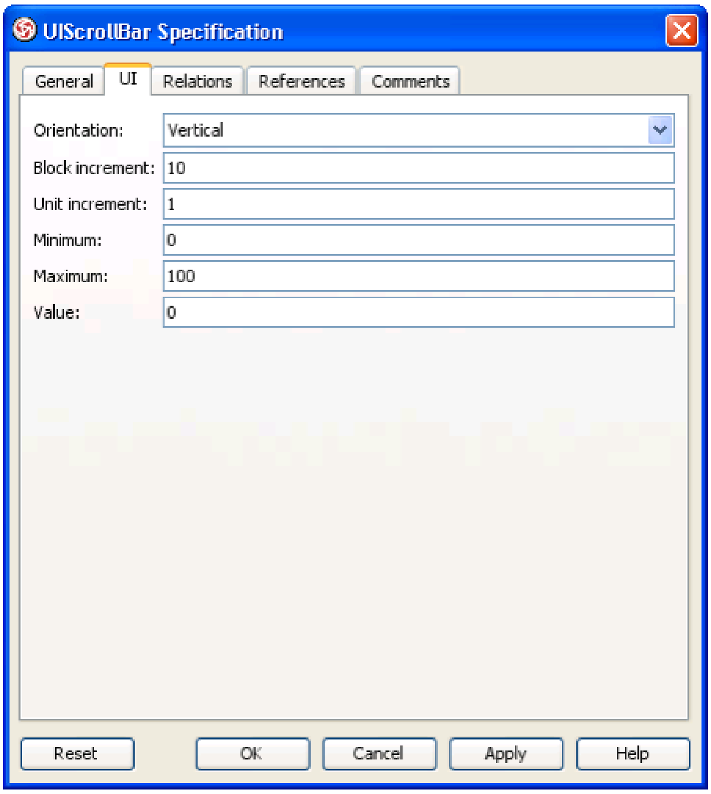


Figure 7-35 Opening scrollbar specification

The **UI** tab is where user can configure a scrollbar.



Orientation: Vertical

Block increment: 10

Unit increment: 1

Minimum: 0

Maximum: 100

Value: 0

Figure 7-36 Scrollbar specification's UI tab

Orientation
Indicates if the scroll bar is vertical or horizontal

Block increment
Amount the value changes when the scrollbar track is clicked on either side of the knob.

Unit increment
Amount the value changes when the end arrows of the scrollbar are clicked.

Minimum
The minimum value of scrollbar.

Maximum
The maximum value of scrollbar.

Value
Value which controls the location of the scroll bar knob.

Drawing Combo Box

Combo box is a component that combines a button and a drop-down list. The user can select a value from the drop-down list, which appears at the user's request.

Creating a Combo Box

1. Select the **Combo Box** tool from the diagram toolbar.

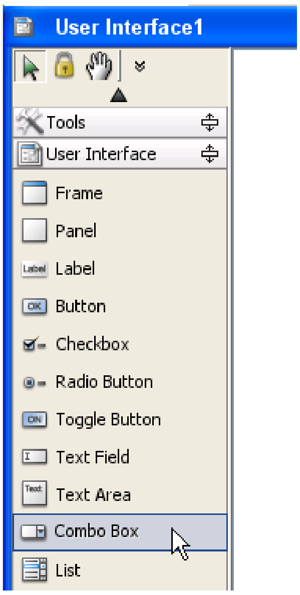


Figure 7-37 Selecting Combo Box from diagram toolbar

2. Click to set the starting point of the combo box.
3. Drag diagonally from the starting point to expand the combo box.

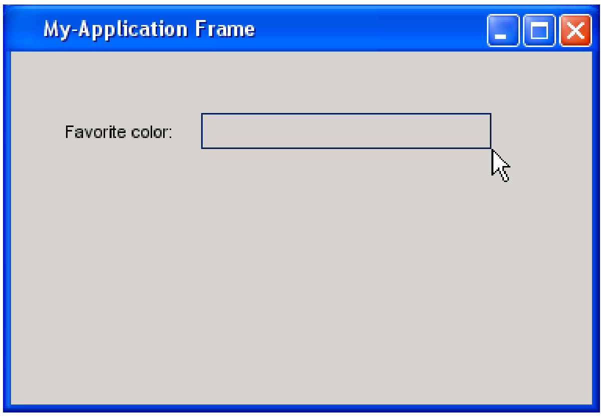


Figure 7-38 Drag diagonally to expand the Combo box

4. Release the mouse button to confirm the size of combo box. This creates the combo box.

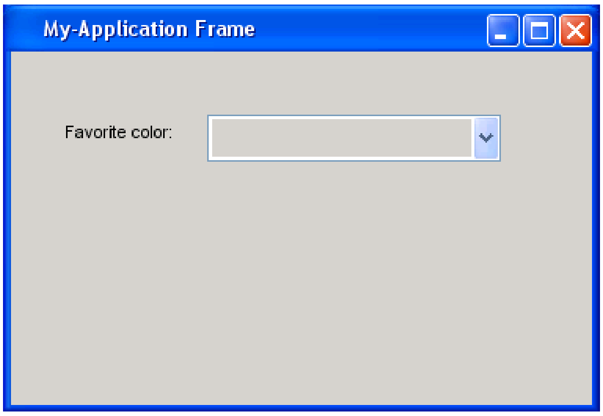


Figure 7-39 Combo box is created

Editing Combo Box Values

1. Rght click on the combo box and select **Open Specification...** from the popup menu.

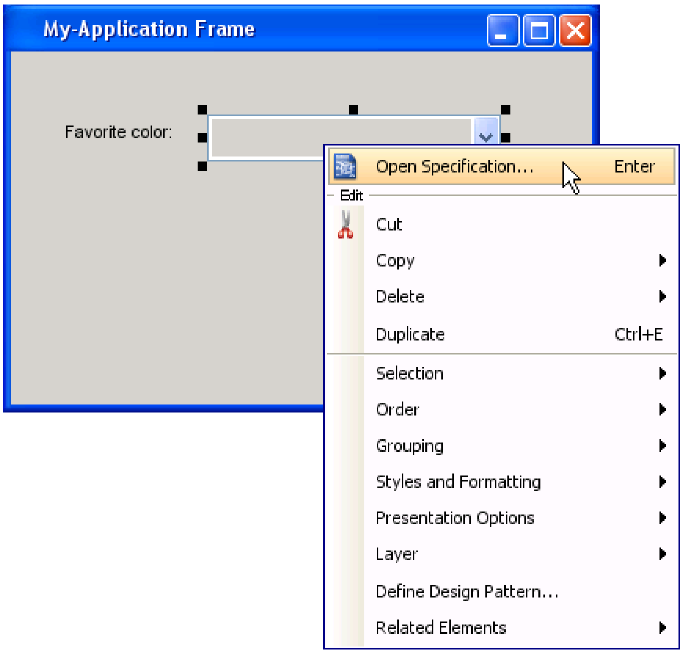
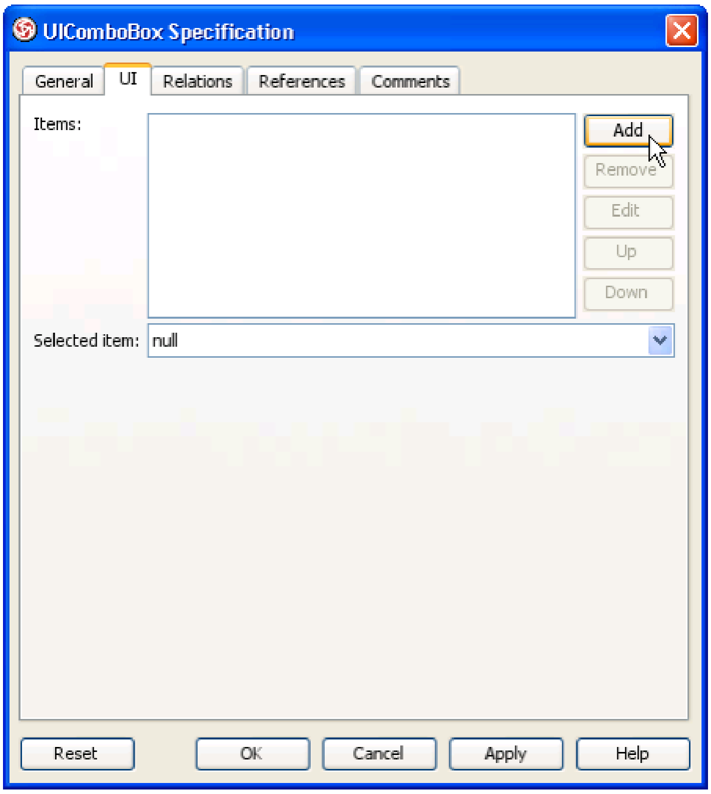


***Figure 7-40** Opening combo box specification*

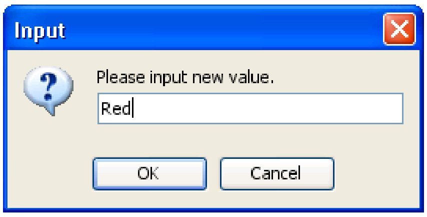
2. In the **UI** tab of the **Combo Box** specification dialog box, click **Add**.



Selected item: null

***Figure 7-41** Click on the Add button to add a combo box item*

3. Enter the value for of item and click **OK** to confirm.

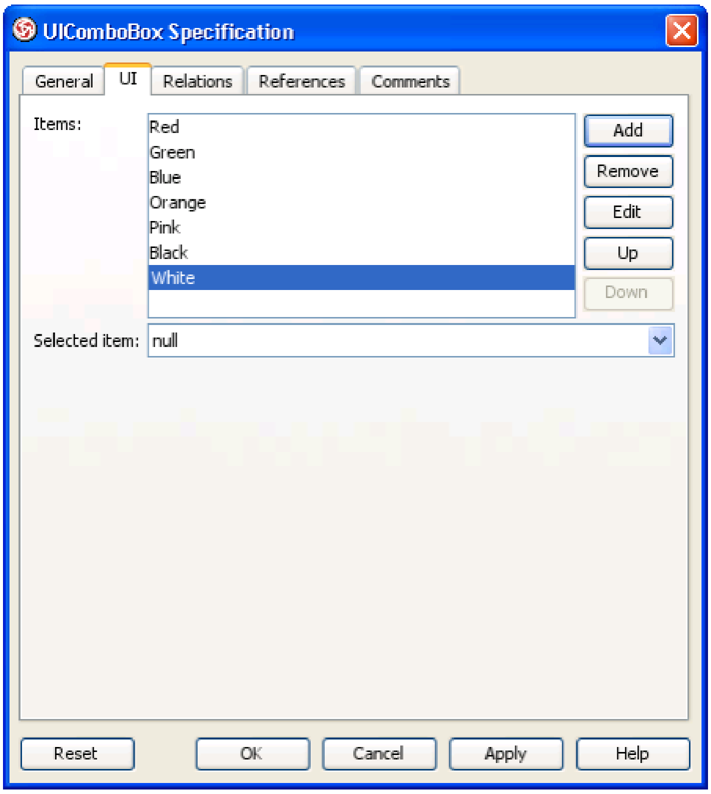


Please input new value.

Red

***Figure 7-42**Input dialog box for entering name of combo box item*

4. Repeat step 2 and 3 to create all the combo box items.



UIComboBox Specification

General | UI | Relations | References | Comments

Items:
Red
Green
Blue
Orange
Pink
Black
White

Selected item: null

*Figure 7-43*Combo box items are created

5. Click **OK** to confirm editing.

Combo Box Properties

There are several configurable properties for a combo box. To configure a combo box, right click on the combo box and select **Open Specification...** from the popup menu.

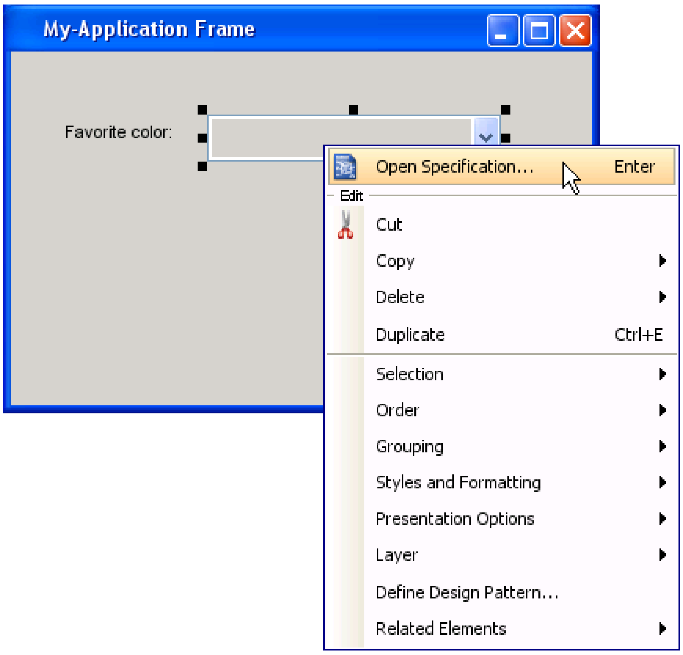
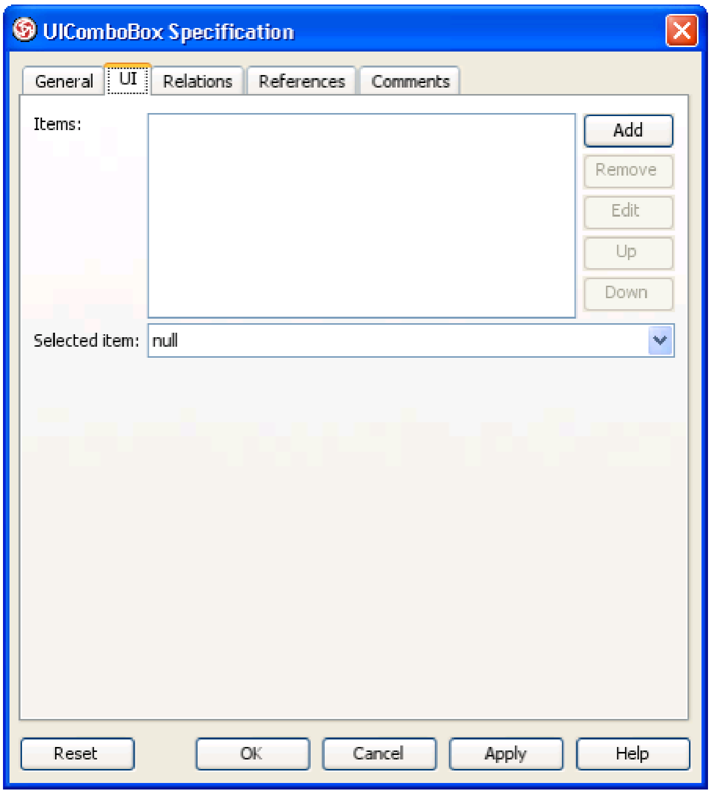


Figure 7-44 Opening scrollbar specification

The **UI** tab is where user can configure a combo box.



Selected item: null

Figure 7-45 *Combo box specification's UI tab*

Items
The items that can be selected by users.

Selected item
The default selected item.

Drawing List

Combo box is a component that combines a button and a drop-down list. The user can select a value from the drop-down list, which appears at the user's request.

Creating a List

1. Select the **List** tool from the diagram toolbar.

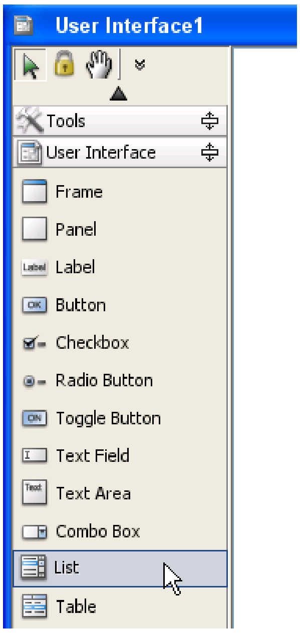


Figure 7-46 *Selecting List from diagram toolbar*

2. Click to set the starting point of the list.
3. Drag diagonally from the starting point to expand the list.

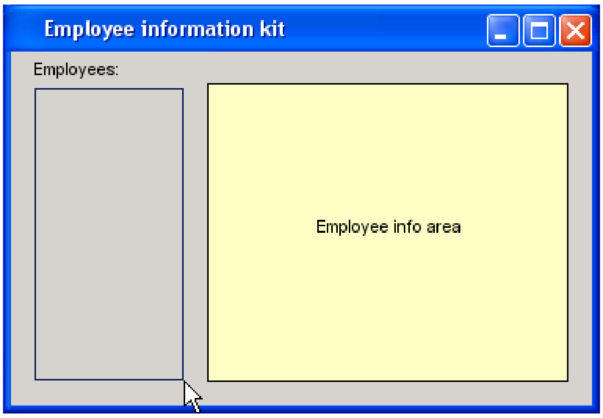


Figure 7-47 *Drag diagonally to expand the List*

4. Release the mouse button to confirm the size of list. This creates the list.

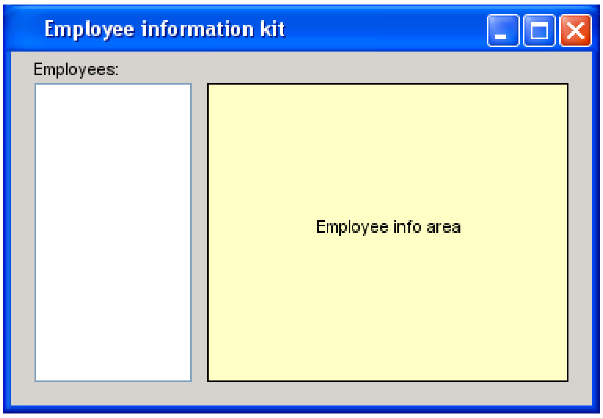


Figure 7-48 *List is created*

Editing List Values

1. Right click on the list and select **Open Specification...** from the popup menu.

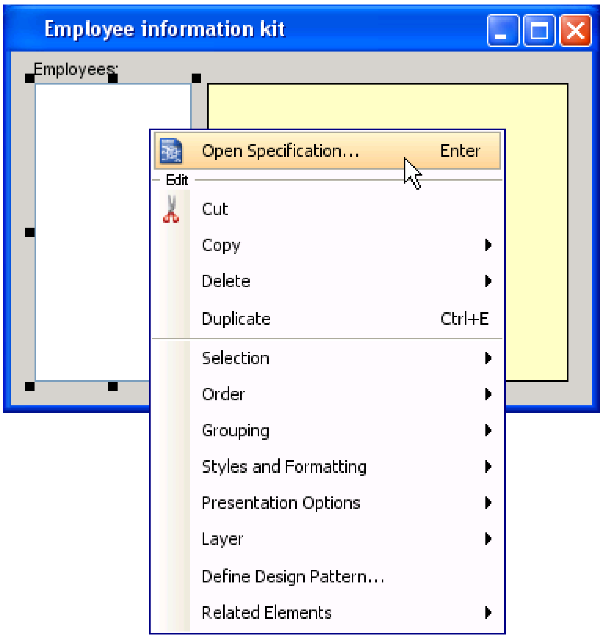
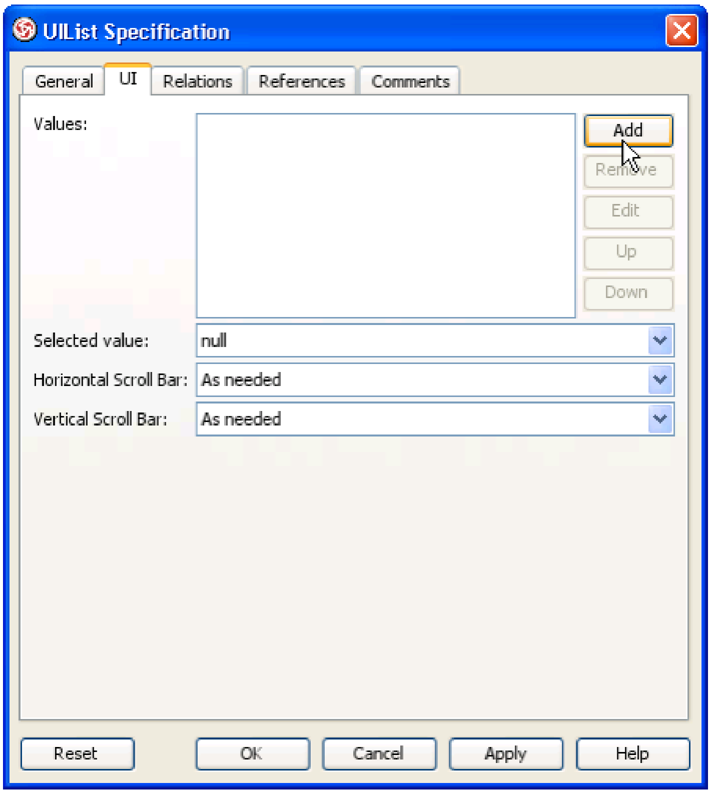


Figure 7-49 *Opening list specification*

2. In the **UI** tab of the **List** specification dialog box, click **Add**.



Values:

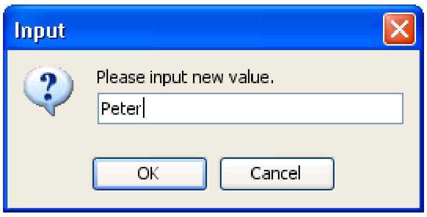
Selected value: null

Horizontal Scroll Bar: As needed

Vertical Scroll Bar: As needed

Figure 7-50 *Click on the Add button to add a list value*

3. Enter the value for of item and click **OK** to confirm.

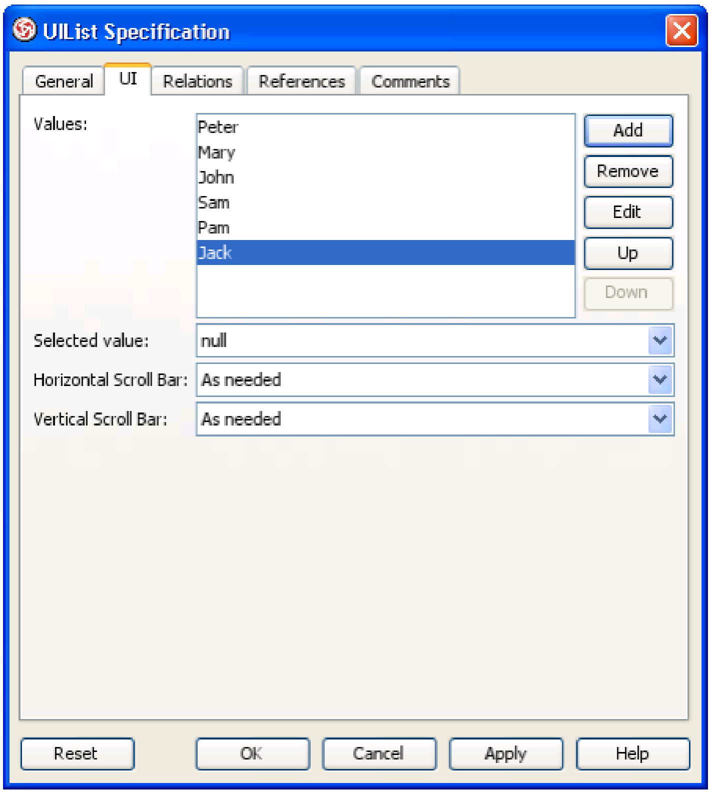


Please input new value.

Peter

*Figure 7-51*Input dialog box for entering name of list item

4. Repeat step 2 and 3 to create all the list items.



Values:
Peter
Mary
John
Sam
Pam
Jack

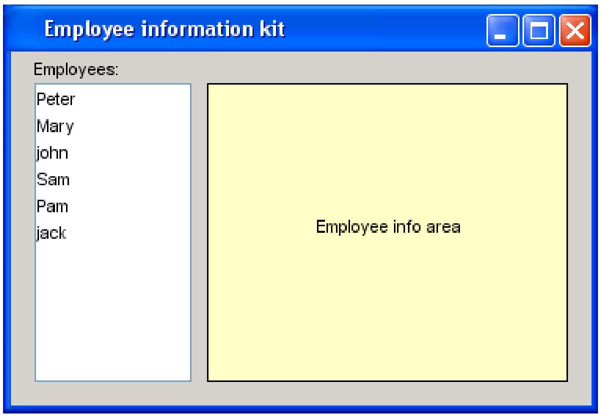
Selected value: null

Horizontal Scroll Bar: As needed

Vertical Scroll Bar: As needed

*Figure 7-52*List items are created

5. Click **OK** to confirm editing.



Employees:
Peter
Mary
john
Sam
Pam
jack

Employee info area

*Figure 7-53*List with values

List Properties

There are several configurable properties for a list. To configure a list, right click on the list and select **Open Specification...** from the popup menu.

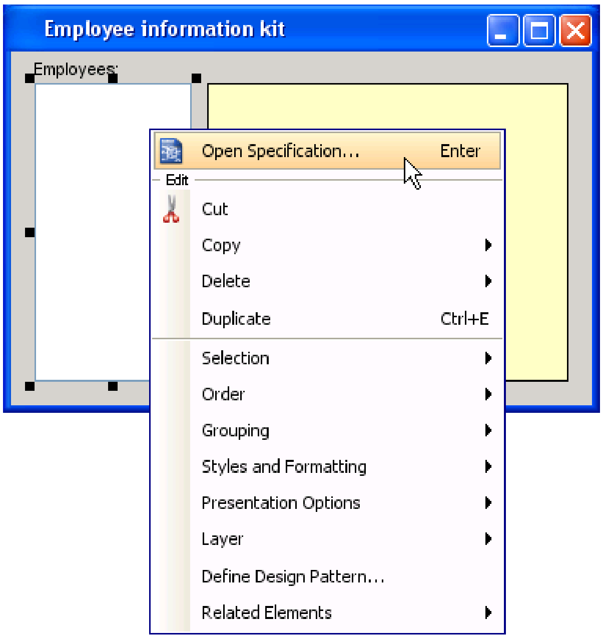
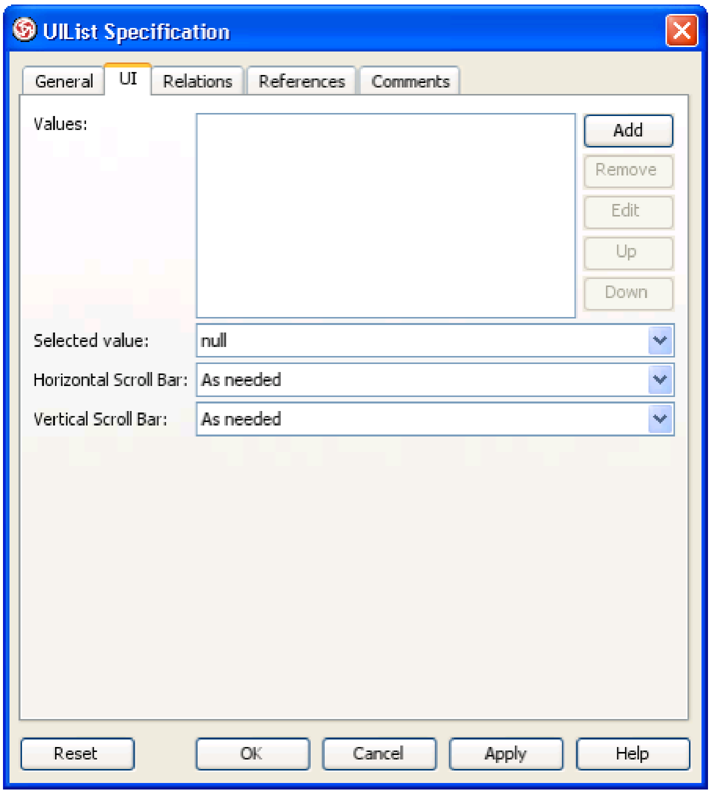


Figure 7-54 Opening list specification

The **UI** tab is where user can configure a list.



Values:

Selected value: null

Horizontal Scroll Bar: As needed

Vertical Scroll Bar: As needed

Figure 7-55 List specification's UI tab

Values

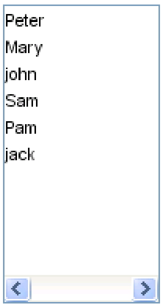
The items that can be selected by user.

Selected value

The default selected item in the list.

Horizontal Scroll Bar

Determines when a horizontal scrollbar will appear in a list.

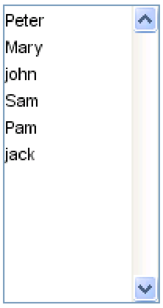


Peter
Mary
john
Sam
Pam
jack

Figure 7-56 A horizontal scrollbar

Vertical Scroll Bar

Determines when a vertical scrollbar will appear in a list.



Peter
Mary
john
Sam
Pam
jack

Figure 7-57 A vertical scrollbar

Drawing Table

Table is used to display and edit regular two-dimensional tables of cells.

Creating a Table

1. Select the **Table** tool from the diagram toolbar.

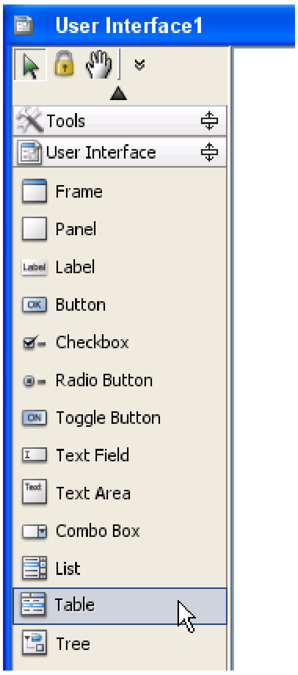


Figure 7-58 *Selecting Table from diagram toolbar*

2. Click to set the starting point of the table.
3. Drag diagonally from the starting point to expand the table.

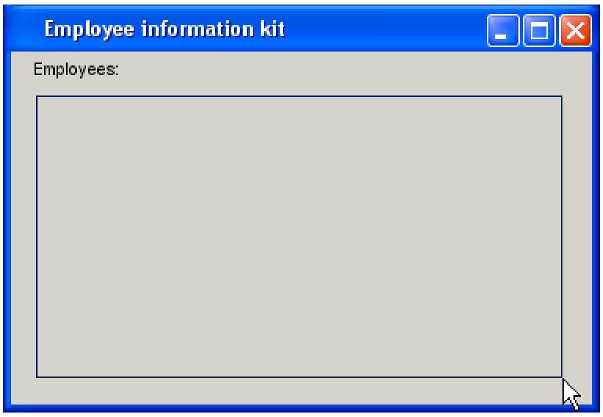


Figure 7-59 *Drag diagonally to expand the Table*

4. Release the mouse button to confirm the size of table. This creates the table.

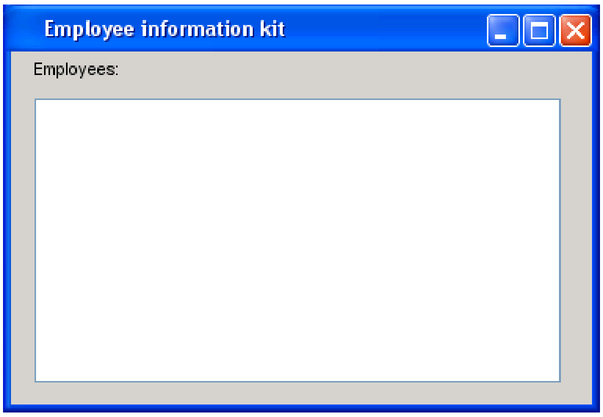


Figure 7-60 *Table is created*

Editing Table Contents

Basic Setup

To insert columns and rows into a table:

1. Right click on the table and select **Open Specification...** from the popup menu.

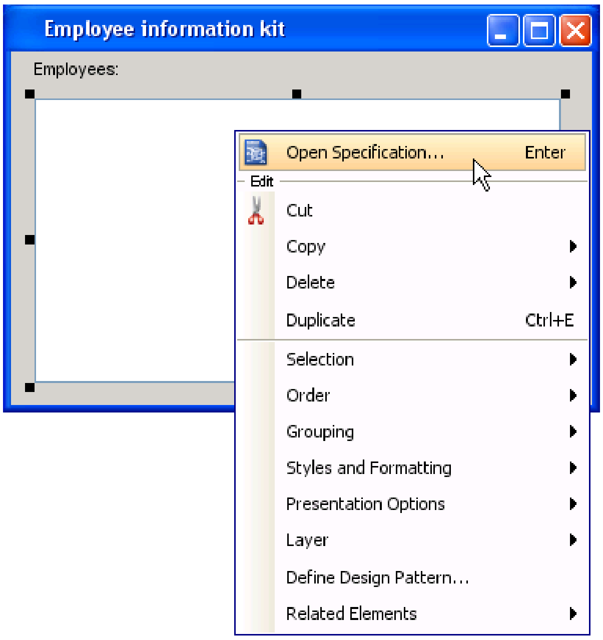
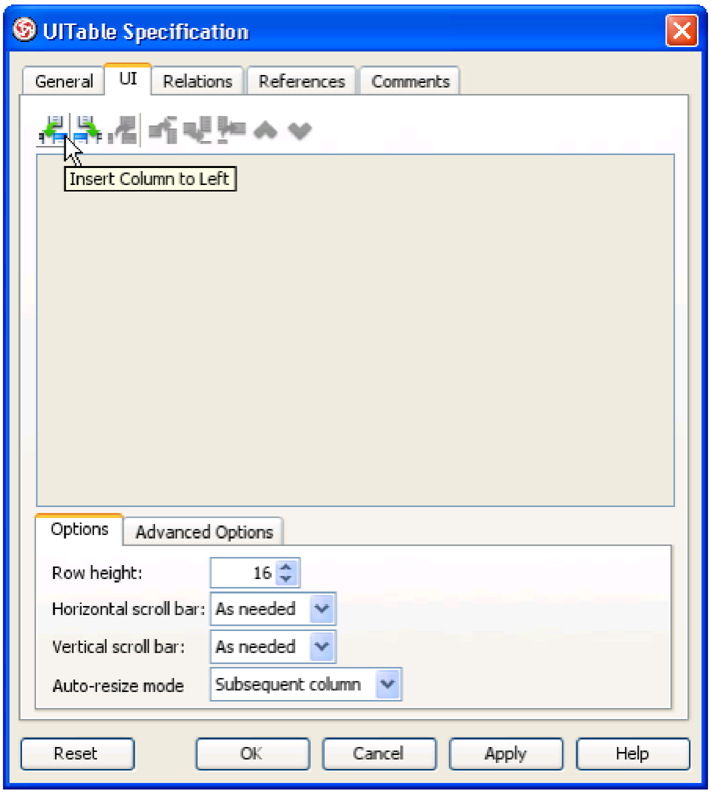


Figure 7-61 *Opening table specification*

2. In the **UI** tab of the **Table** specification dialog box, click either **Insert Column to Left** or **Insert Column to Right** to insert the first column.



Row height: 16

Horizontal scroll bar: As needed

Vertical scroll bar: As needed

Auto-resize mode: Subsequent column

Figure 7-62 *Insert a column into table*

3. Enter the column name and click **OK** to confirm.

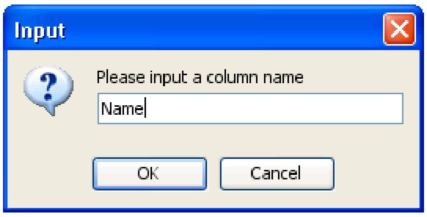
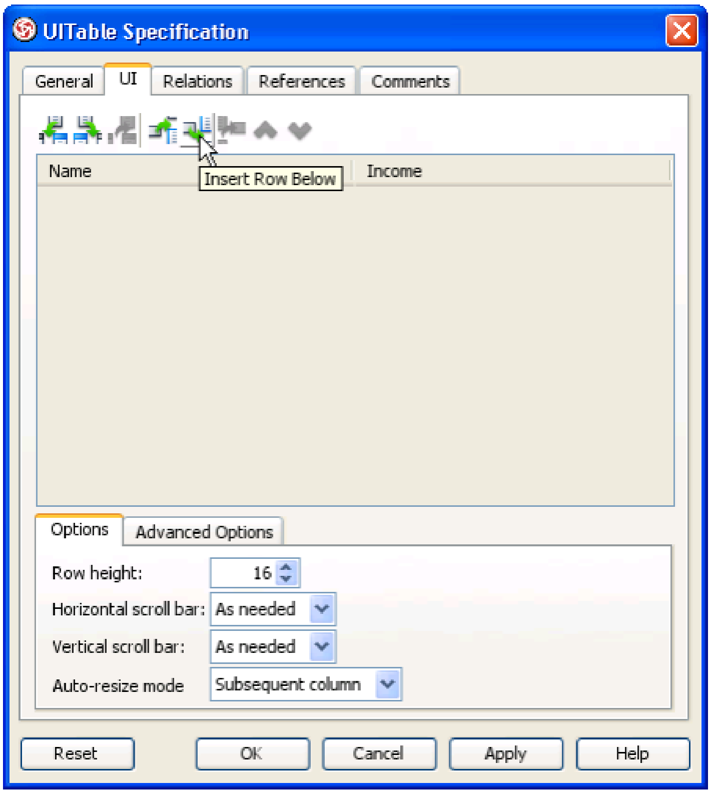


Figure 7-63Input dialog box for entering name of column

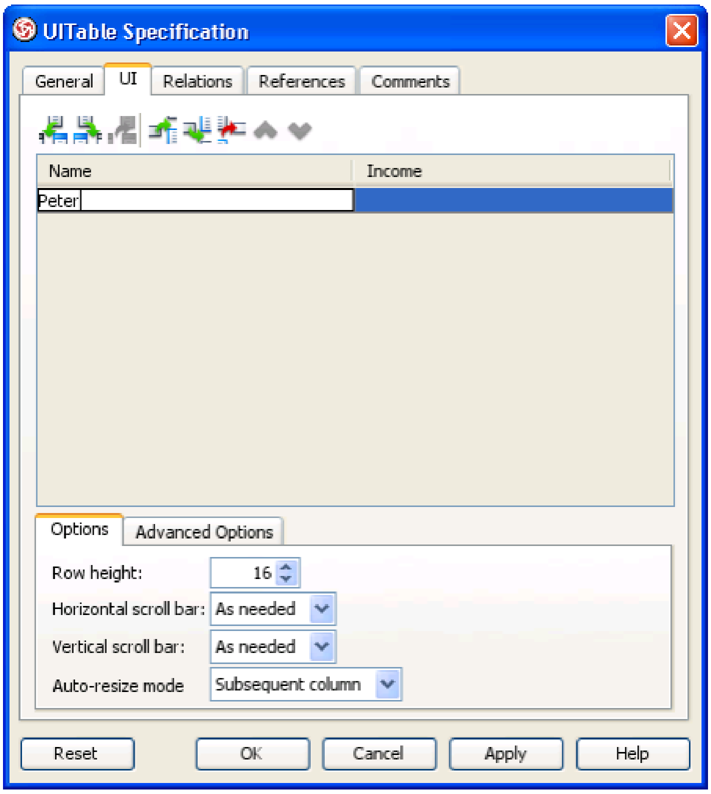
4. Repeat step 2 and 3 to create all columns.
5. Click **Insert Row Below** to insert a row. Note that a row can be inserted only when there is a column exist.



Row height: 16
Horizontal scroll bar: As needed
Vertical scroll bar: As needed
Auto-resize mode Subsequent column

Figure 7-64Insert a row into table

6. A row is created with empty cell(s). Fill in the cell(s) if necessasry. Cell can be edited by double clicking or by pressing the **F2** key.



Name	Income
Peter	

Row height: 16

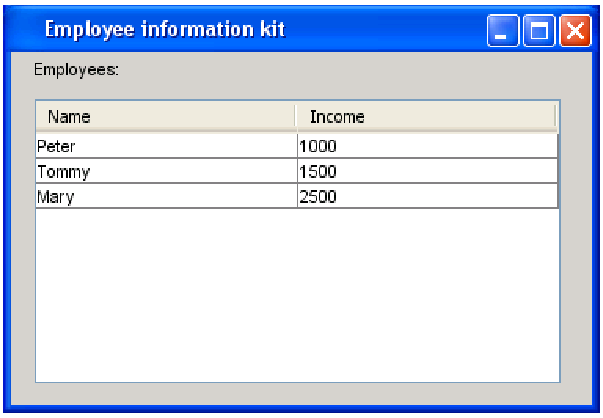
Horizontal scroll bar: As needed

Vertical scroll bar: As needed

Auto-resize mode: Subsequent column

*Figure 7-65*Entering cell content

7. Repeat step 5 and 6 to create all rows.
8. Click **OK** to confirm editing.



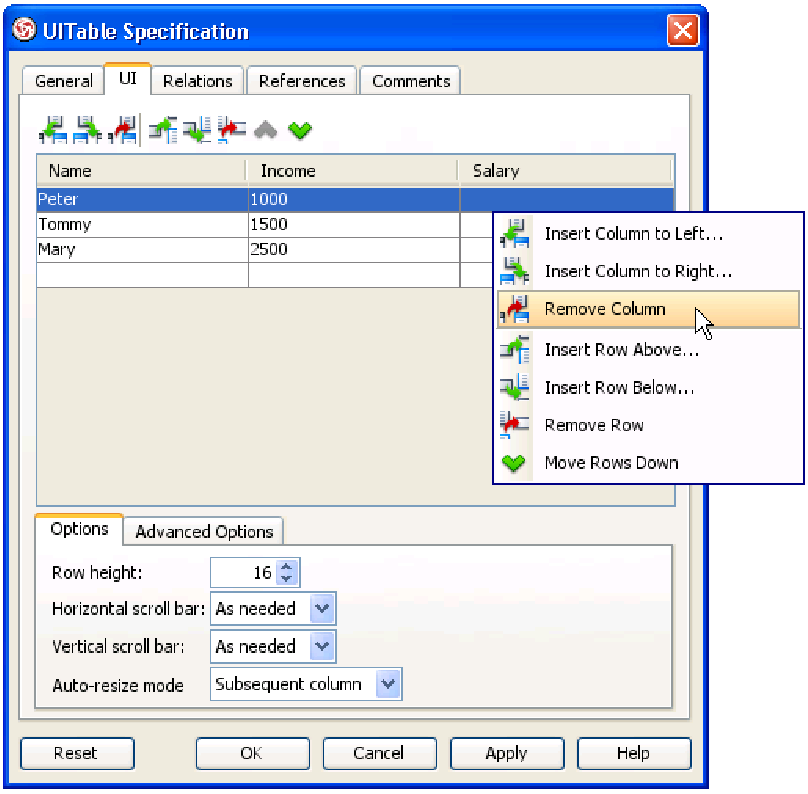
Employees:

Name	Income
Peter	1000
Tommy	1500
Mary	2500

*Figure 7-66*Table with data

Removing a Column

To remove a column in a table, open the specification dialog box of the table, open the UI tab, right click on a cell of a column and select Remove Column in the popup menu.



Name	Income	Salary
Peter	1000	
Tommy	1500	
Mary	2500	

Row height: 16

Horizontal scroll bar: As needed

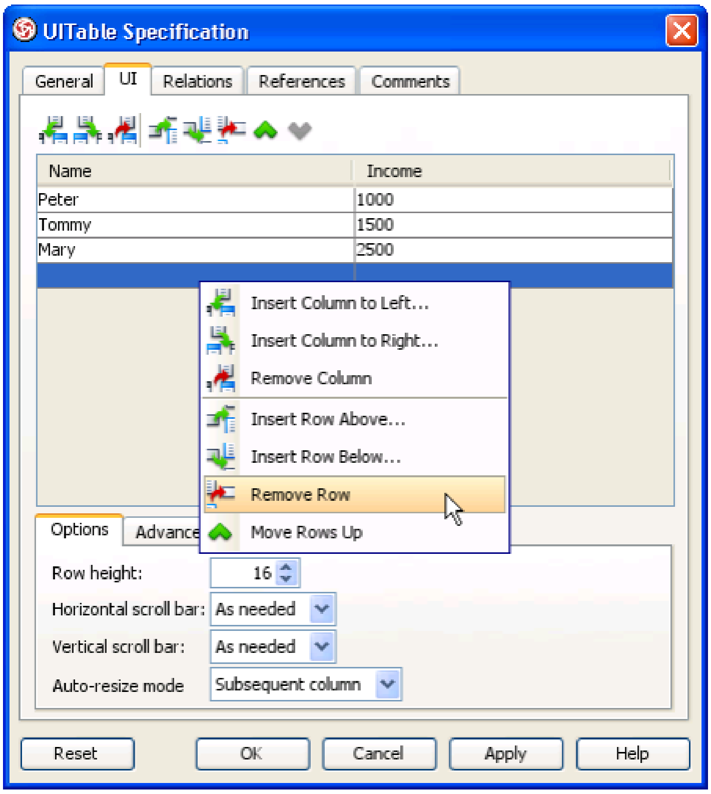
Vertical scroll bar: As needed

Auto-resize mode Subsequent column

__Figure 7-67__To remove a column in table

Removing a Row

To remove a row in a table, open the specification dialog box of the table, open the **UI** tab, right click on a row and select **Remove Row** in the popup menu.



Name	Income
Peter	1000
Tommy	1500
Mary	2500

Row height: 16

Horizontal scroll bar: As needed

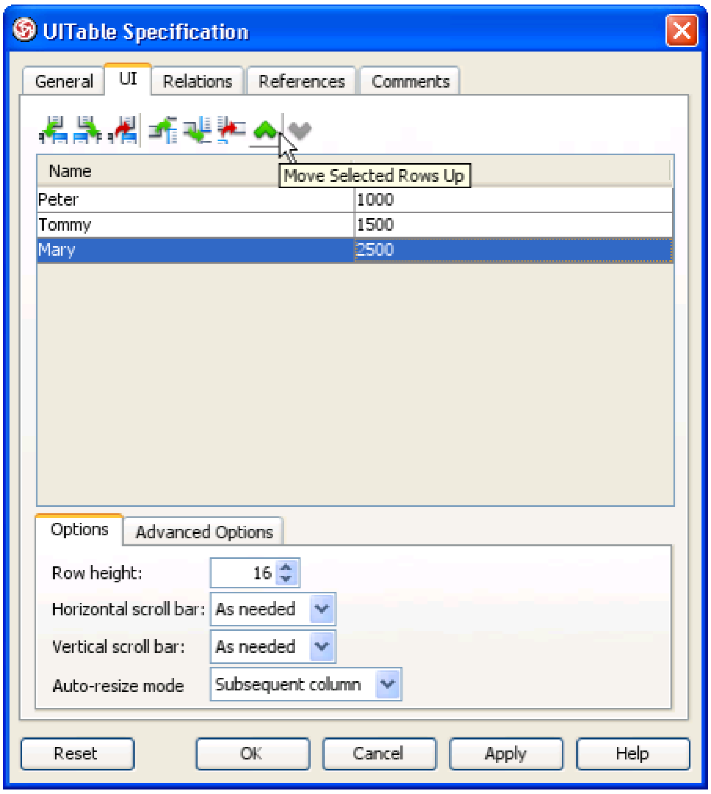
Vertical scroll bar: As needed

Auto-resize mode Subsequent column

__Figure 7-68__To remove a row in table

Reordering Rows

To reorder rows, open the specification dialog box of the table, open the **UI** tab,select the row(s) to reorder, and click on the **Move Selected Rows Up** or **Move Selected Rows Down** button.



Name	
Peter	1000
Tommy	1500
Mary	2500

Row height: 16

Horizontal scroll bar: As needed

Vertical scroll bar: As needed

Auto-resize mode Subsequent column

Figure 7-69Moving a row in table upwards

Table Properties

There are several configurable properties for a table. To configure a table, right click on the table and select **Open Specification...** from the popup menu.

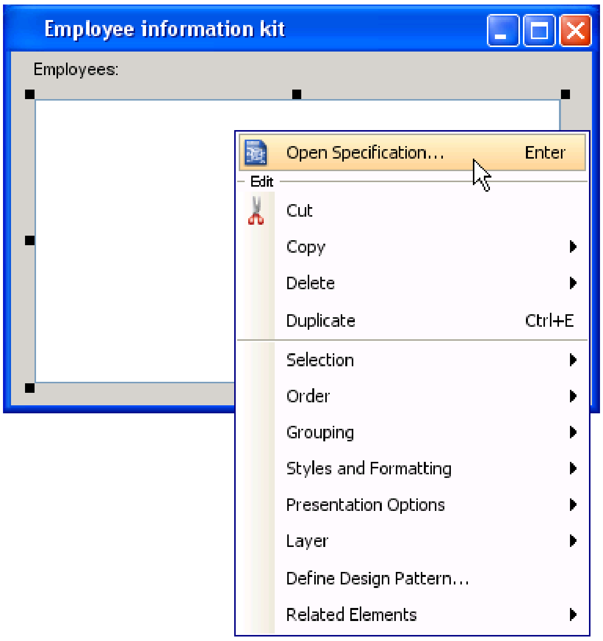
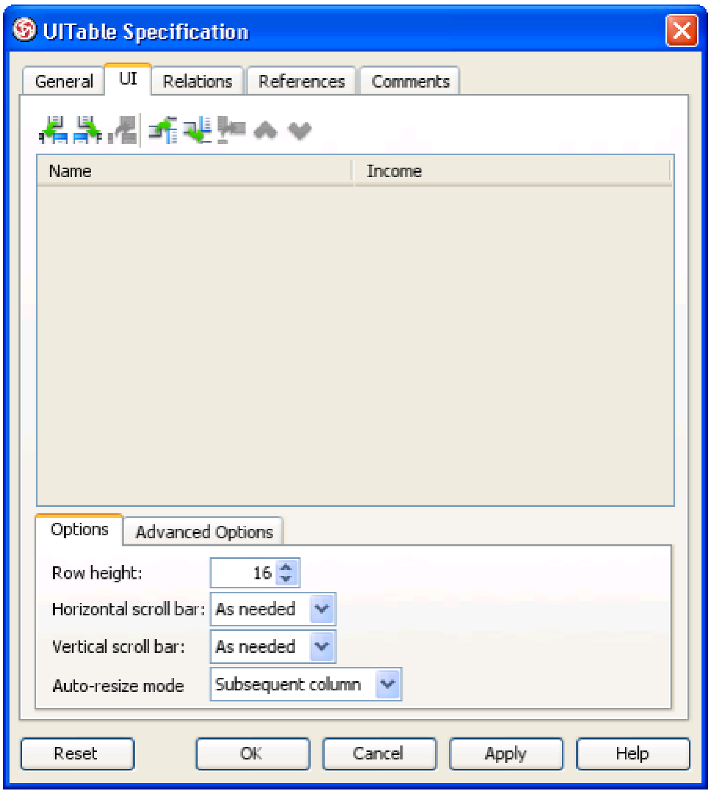


Figure 7-70 Opening table specification

The **UI** tab is where user can configure a table.



Row height: 16

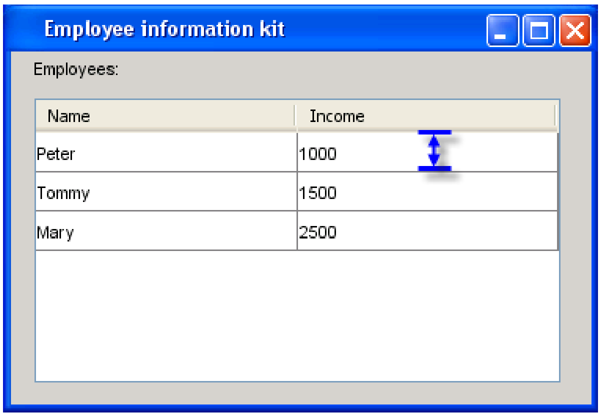
Horizontal scroll bar: As needed

Vertical scroll bar: As needed

Auto-resize mode Subsequent column

Figure 7-71 Table specification's UI tab

Row height
The height of rows in table.



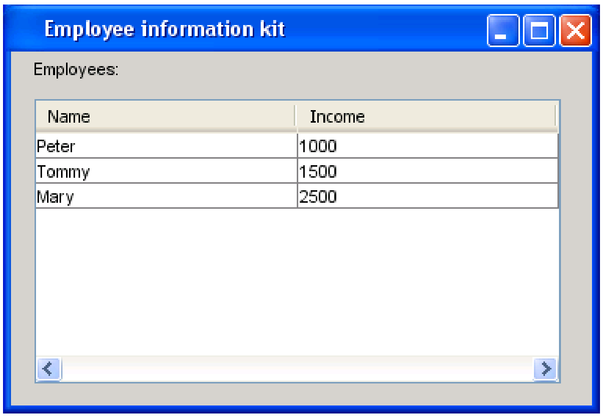
Employees:

Name	Income
Peter	1000
Tommy	1500
Mary	2500

Figure 7-72 Height of table

Horizontal scroll bar

Determines when a horizontal scrollbar will appear in a table.



Employees:

Name	Income
Peter	1000
Tommy	1500
Mary	2500

Figure 7-73 A horizontal scrollbar in table

Vertical scroll bar

Determines when a vertical scrollbar will appear in a table.

Auto-resize mode

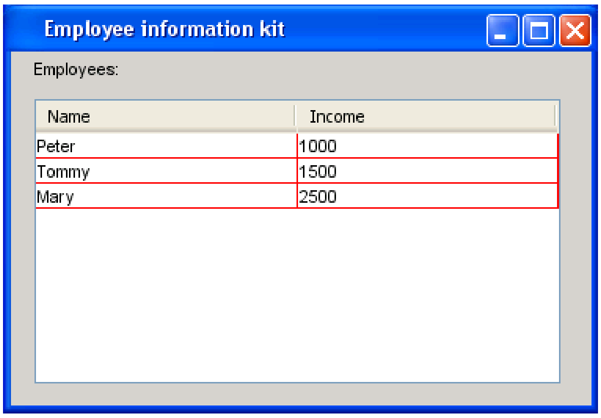
Determines how the widths of columns will be affected when other columns in the table are being resized.

Auto-resize mode	Description
Off	Disable auto resizing
Next column	When a column is being resized, all columns to the right and left of margin are updated
Subsequent columns	When a column is being resized, all columns to the right are resized at the same time
Last column	When a column is being resized, width of the right-most column is updated
All column	When a column is being resized, widths of all columnes are changed

Table 7-1 Description of table's auto resize mode

Grid color

Determins the color of grid in table.



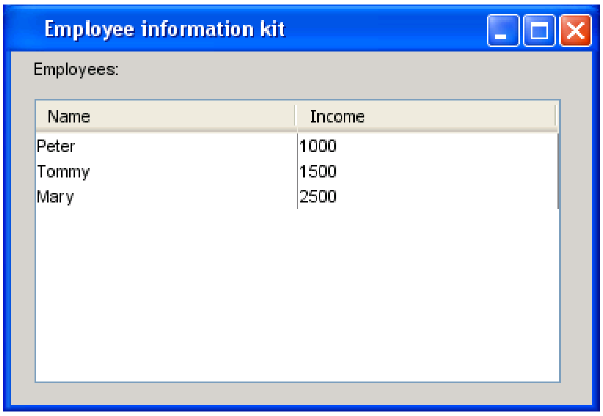
Employees:

Name	Income
Peter	1000
Tommy	1500
Mary	2500

Figure 7-74 Table with grid color in red

Show horizontal line

Determines the visibility of horizontal lines in table.



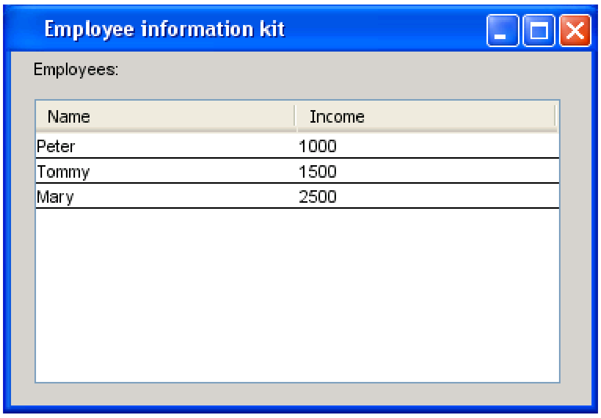
Employees:

Name	Income
Peter	1000
Tommy	1500
Mary	2500

Figure 7-75 *Table with horizontal lines*

Show vertical line

Determines the visibility of vertical lines in table.



Employees:

Name	Income
Peter	1000
Tommy	1500
Mary	2500

Figure 7-76 *Table without vertical lines*

Drawing Tree

A Tree is a component that displays a set of hierarchical data as an outline

Creating a Tree

1. Select the **Tree** tool from the diagram toolbar.

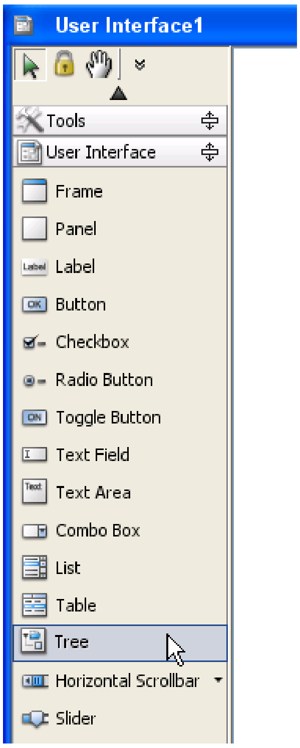


Figure 7-77 *Selecting Tree from diagram toolbar*

2. Click to set the starting point of the tree.
3. Drag diagonally from the starting point to expand the tree.

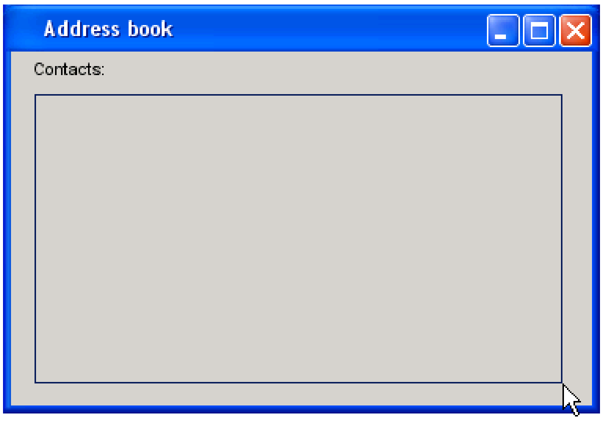


Figure 7-78 *Drag diagonally to expand the Tree*

4. Release the mouse button to confirm the size of tree. This creates the tree.

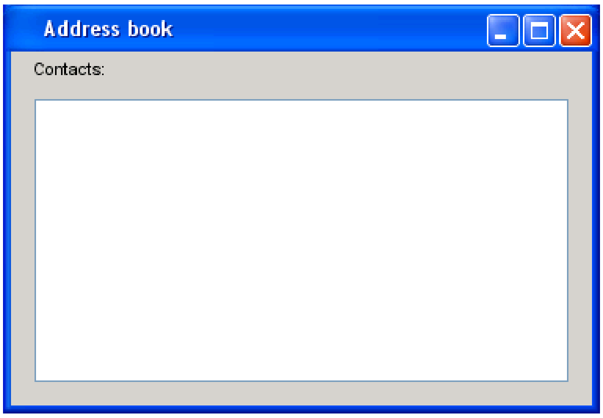


Figure 7-79 *Tree is created*

Editing Tree Contents

Basic Setup

To insert columns and rows into a tree:

1. Right click on the tree and select **Open Specification...** from the popup menu.

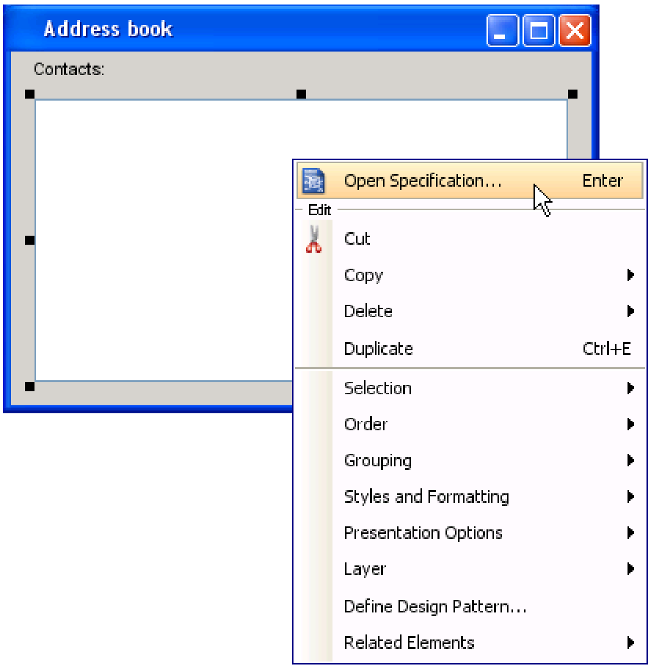
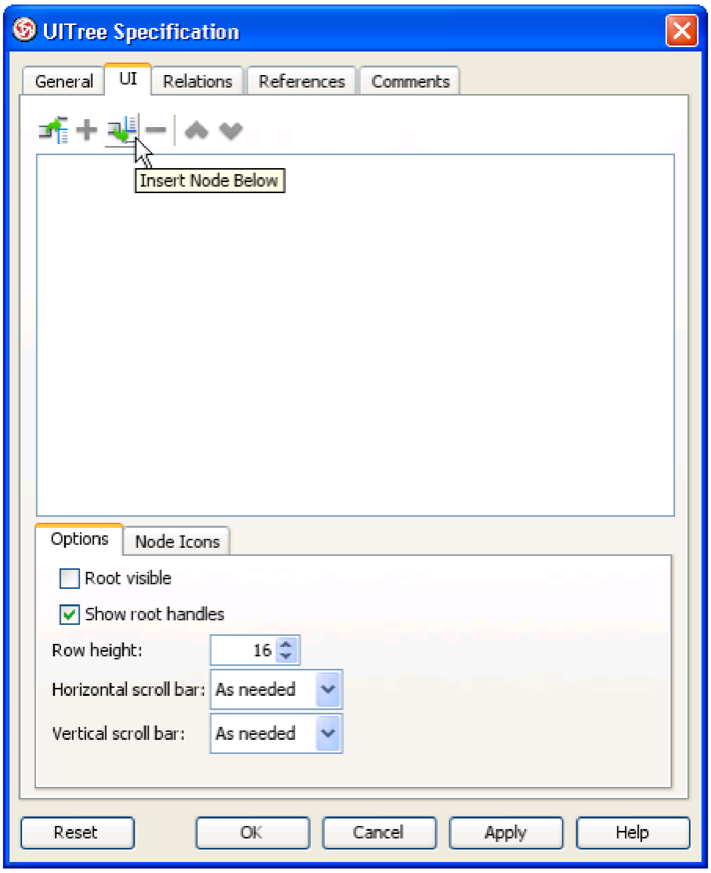


Figure 7-80 *Opening tree specification*

2. In the **UI** tab of the **Tree** specification dialog box, click either **Insert Node Below** or **Insert Node Above** to insert the first column. Note that **Insert Node Below/Above** means to create an adjacent node.



☐ Root visible

☑ Show root handles

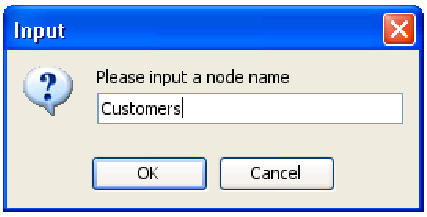
Row height: 16

Horizontal scroll bar: As needed

Vertical scroll bar: As needed

Figure 7-81 *Insert a node into tree*

3. Enter the node name and click **OK** to confirm.



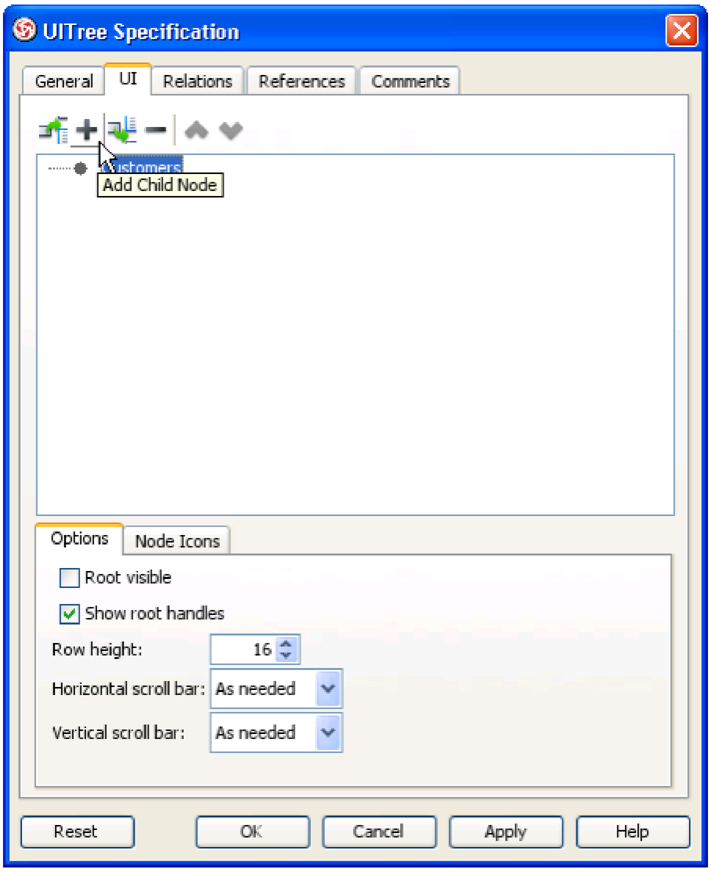
Please input a node name

Customers

Figure 7-82*Input dialog box for entering name of node*

4. To add a child node, select an existing node.

5. Click **Add Child Node**.



☐ Root visible

☑ Show root handles

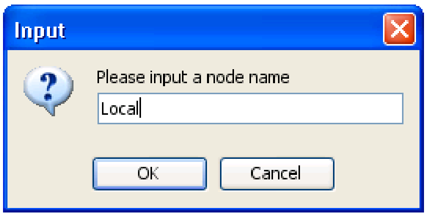
Row height: 16

Horizontal scroll bar: As needed

Vertical scroll bar: As needed

Figure 7-83*Insert a child node*

6. Enter the node name and click **OK** to confirm.

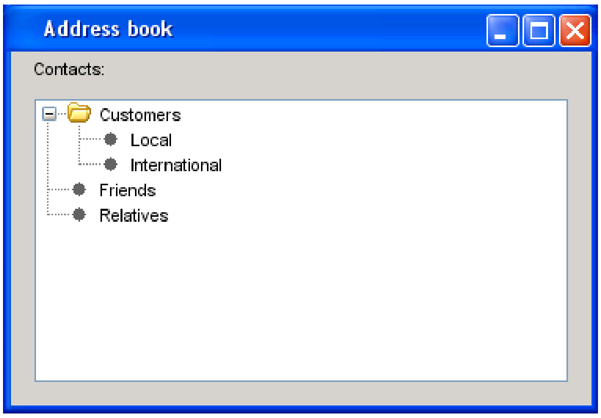


Please input a node name

Local

Figure 7-84*Entering node name*

7. Repeat step 2 to 6 to create all nodes.
8. Click **OK** to confirm editing.



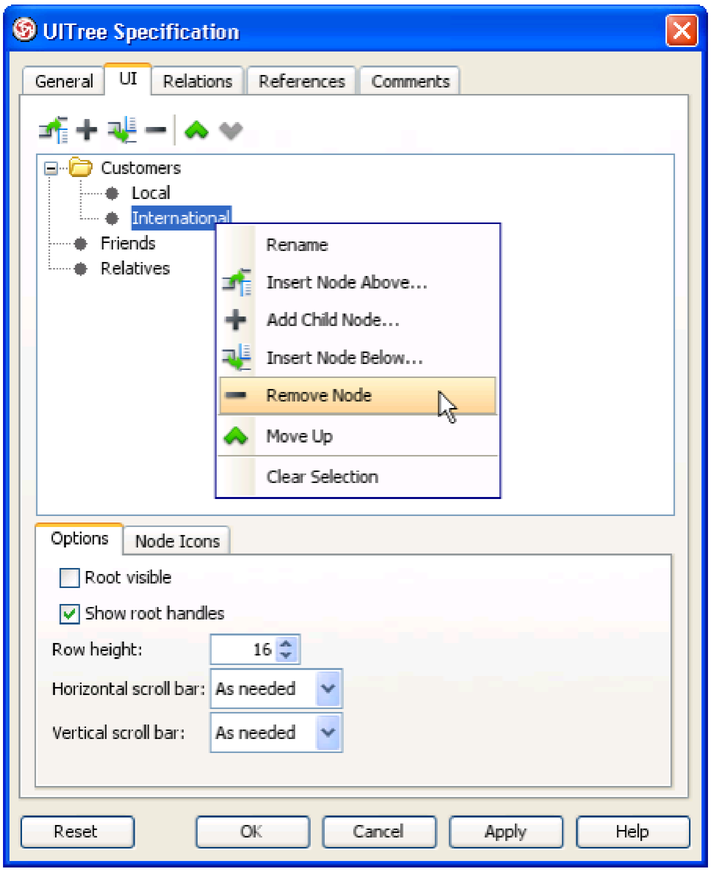
Contacts:

Customers
Local
International
Friends
Relatives

Figure 7-85*Tree with data*

Removing a Node

To remove a node in a tree, open the specification dialog box of the tree, open the **UI** tab, right click on the node(s) to remove and select **Remove Node** in the popup menu.



□ Root visible

☑ Show root handles

Row height: 16

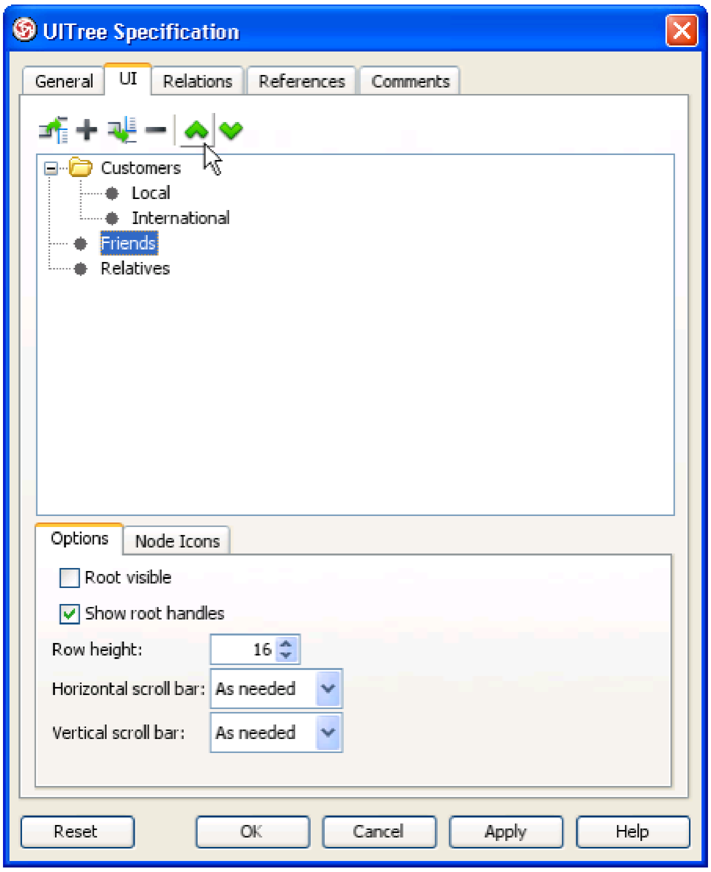
Horizontal scroll bar: As needed

Vertical scroll bar: As needed

Figure 7-86*To remove a node in tree*

Reordering Nodes

To reorder nodes, open the specification dialog box of the tree, open the **UI** tab,select the node(s) to reorder, and click on the **Move Up** or **Move Down** button.



- Customers
  - Local
  - International
- Friends
- Relatives

☐ Root visible

☑ Show root handles

Row height: 16

Horizontal scroll bar: As needed

Vertical scroll bar: As needed

*Figure 7-87*Moving a node in tree upwards

Tree Properties

There are several configurable properties for a tree. To configure a tree, right click on the tree and select **Open Specification...** from the popup menu.

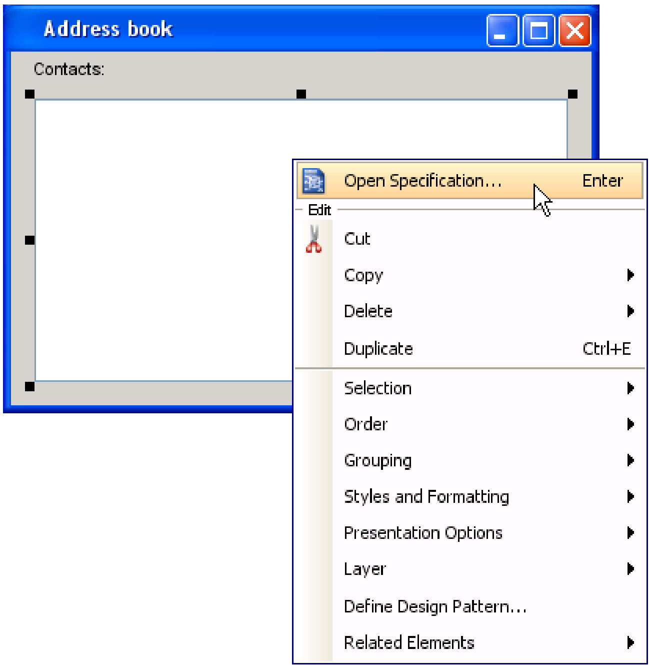
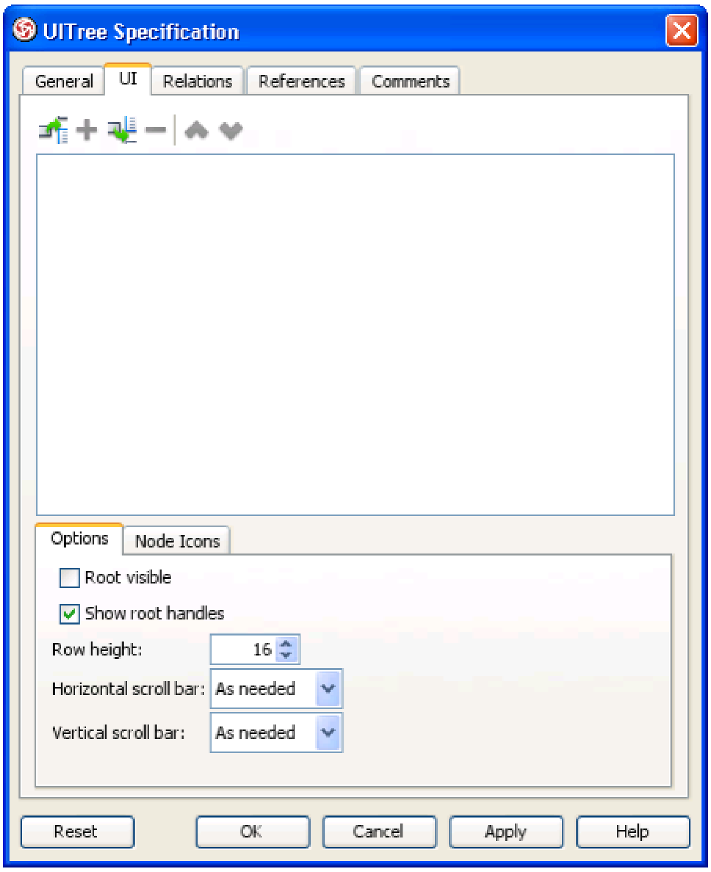


Figure 7-88 Opening tree specification

The **UI** tab is where user can configure a tree.



□ Root visible

☑ Show root handles

Row height: 16

Horizontal scroll bar: As needed

Vertical scroll bar: As needed

Figure 7-89 *Tree specification's UI tab*

Root visible

Determines whether the root of tree will appear.

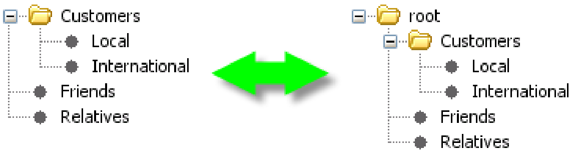


Figure 7-90 *Making the root of tree visible*

Show root handles

Determines whether the expand/collapse button for root will appear.

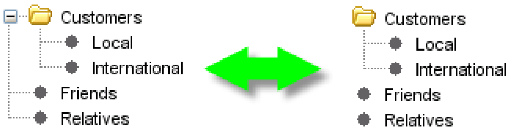


Figure 7-91 *Making the handle of root node visible*

Row height
The height of nodes in tree.

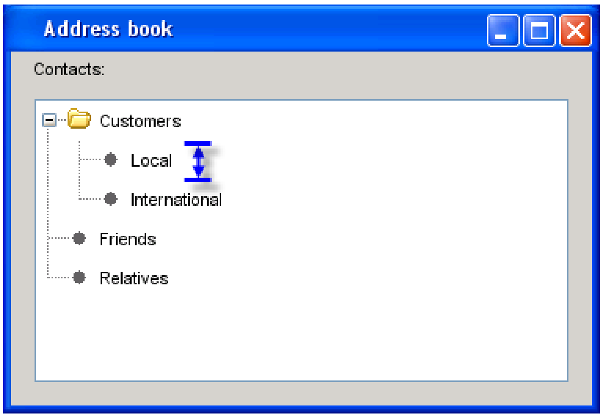


Figure 7-92 Height of node

Horizontal scroll bar
Determines when a horizontal scrollbar will appear in a tree.

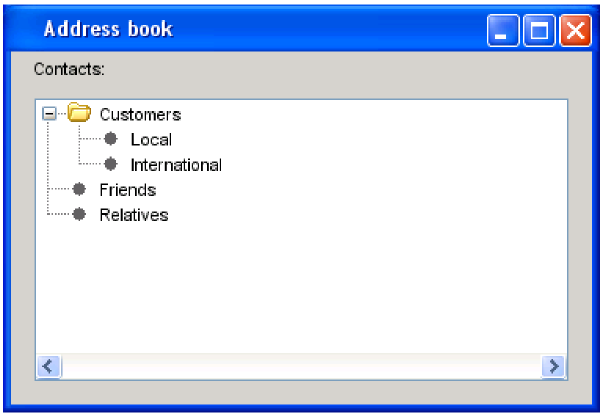


Figure 7-93 A horizontal scrollbar in tree

Vertical scroll bar
Determines when a vertical scrollbar will appear in a tree.

Node Icons
Set the image icon to appear for a node when at different state.

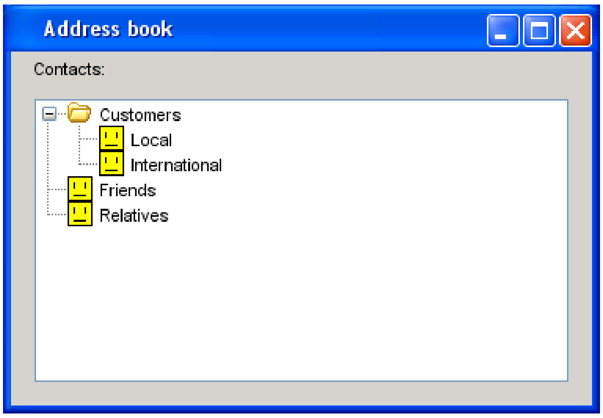


Figure 7-94 Nodes with icons assigned

Options	Description
Default node icon	Icon for all nodes when default icon for collapsed and expanded nodes are not set.
Default collapsed icon	Icon for collapsed nodes. It overwrites the default node icon setting.

Default expanded icon	Icon for expanded nodes. It overwrites the default node icon setting.

Table 7-2 *Description of tree's node icons*

Annotating the UI Design with Callout Shape

Annotation is an extra information, such as comments, notes, explanation, or other type of external mark that describes a model. It is an effective way to edit and review work in a work group environment. Designers often need to be able to jot down information which should not be part of the model itself. In this situation, annotation helps. For instance, annotation can be added to a User Interface design to specify validation checking.

To annotate a User Interface design:

1. Select the **Callout** tool from the diagram toolbar. Note that the **Callout** tool is under the **Common** category.

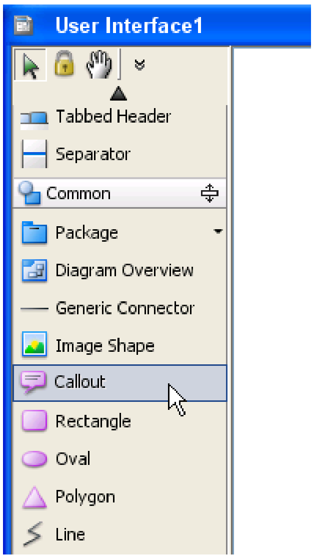


Figure 7-95 Selecting Callout from diagram toolbar

2. Click on the diagram to create a Callout shape.

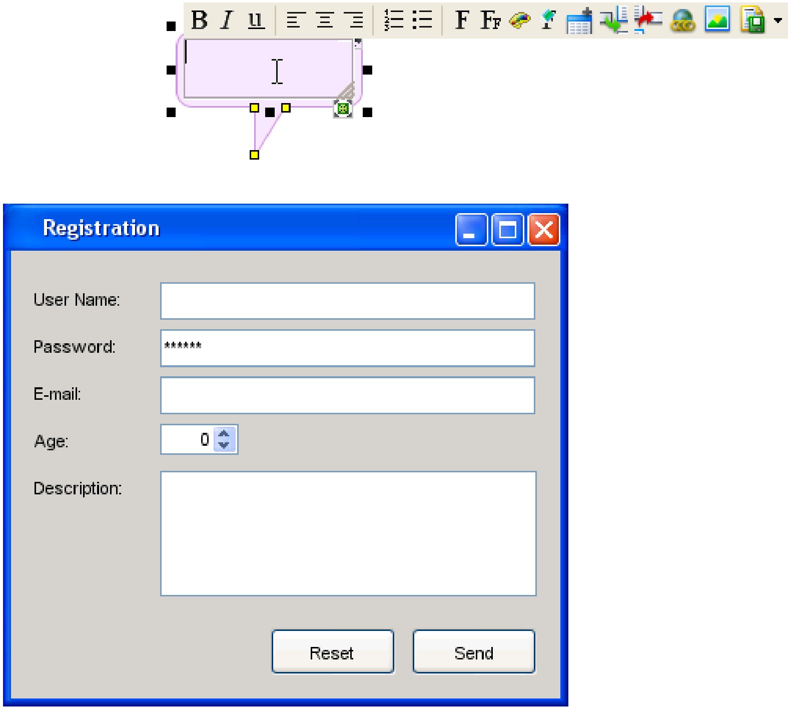
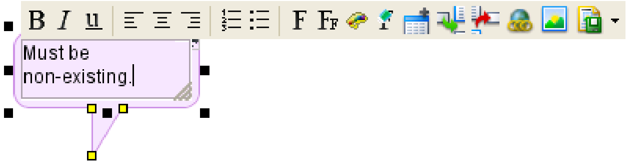


Figure 7-96 Callout shape created

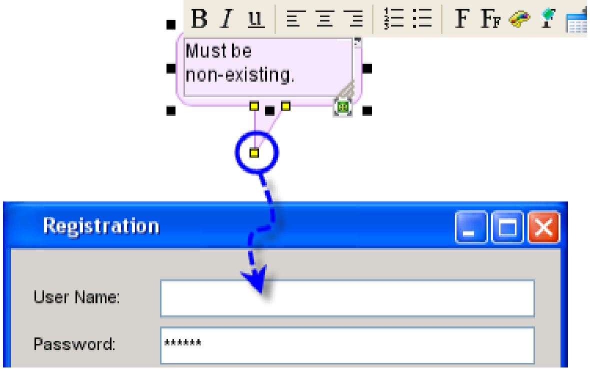
3. Enter the content of annotation. Formattings can be applied through the buttons at the bar on top of the callout shape.



Must be
non-existing.

Figure 7-97 Enter the content of Callout shape

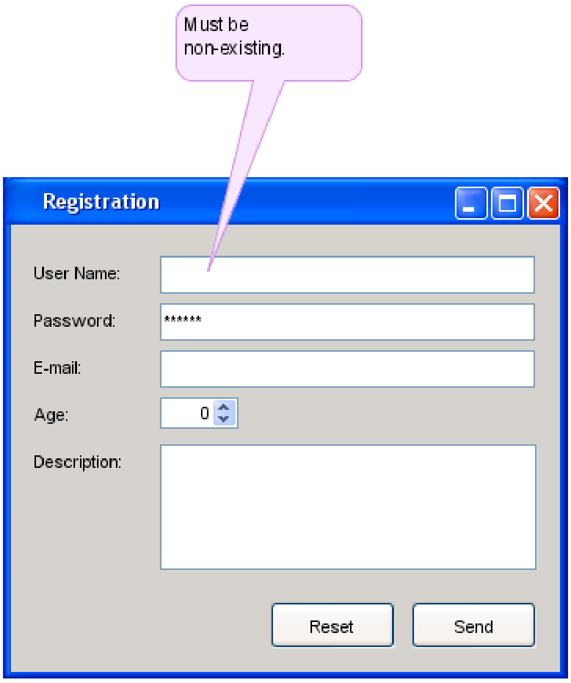
4. We need to attach the pointer of Callout shape to the User Interface component that we want to annotate. Drag on the bottom-most yellow box of the Callout shape towards the User Interface component.



Must be
non-existing.

Figure 7-98 Reposition Callout pointer

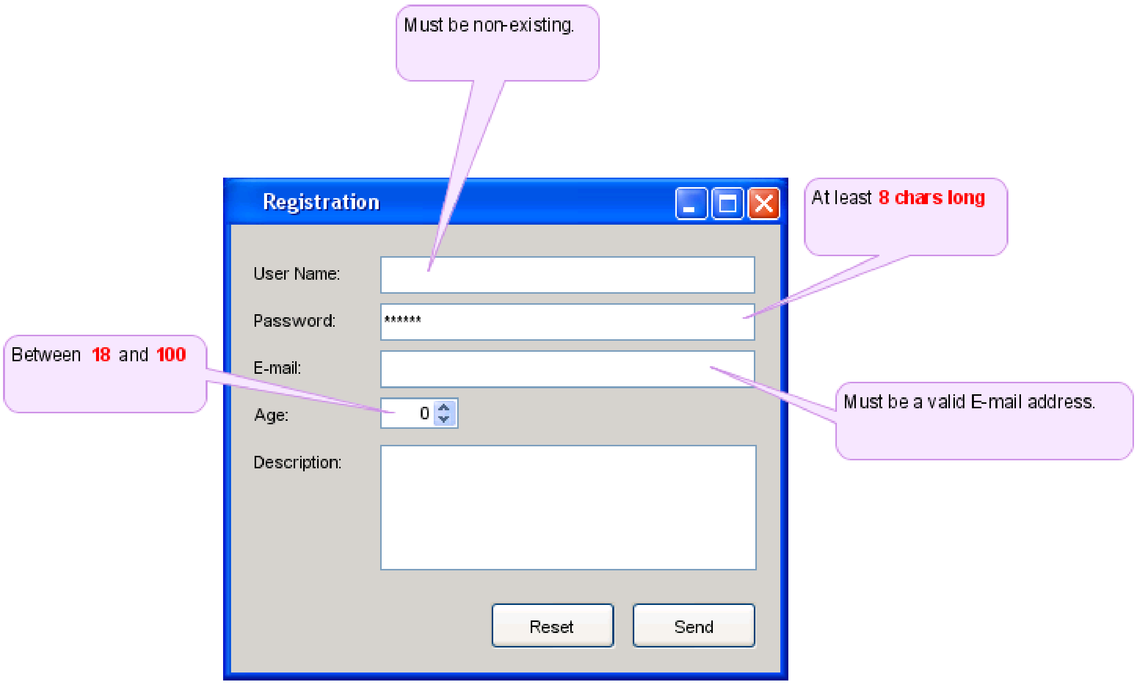
5. Release the mouse button to confirm the pointer position when reached the component we need to annotate.
6. Click on the diagram background to confirm editing.



Must be
non-existing.

Figure 7-99 Using Callout shape to annotate a text field

7. Repeat the above steps to annotate the whole design with Callout shapes.



Registration

User Name:

Password: ******

E-mail:

Age: 0

Description:

Must be non-existing.

At least 8 chars long

Between 18 and 100

Must be a valid E-mail address.

Figure 7-100 *User Interface diagram with annotations*

Drawing Mind Mapping Diagram

Creating Mind Mapping Diagram (Tree)

Right click on the diagram type node in Diagram Navigator and select New **Mind Mapping Diagram** in popup memu.

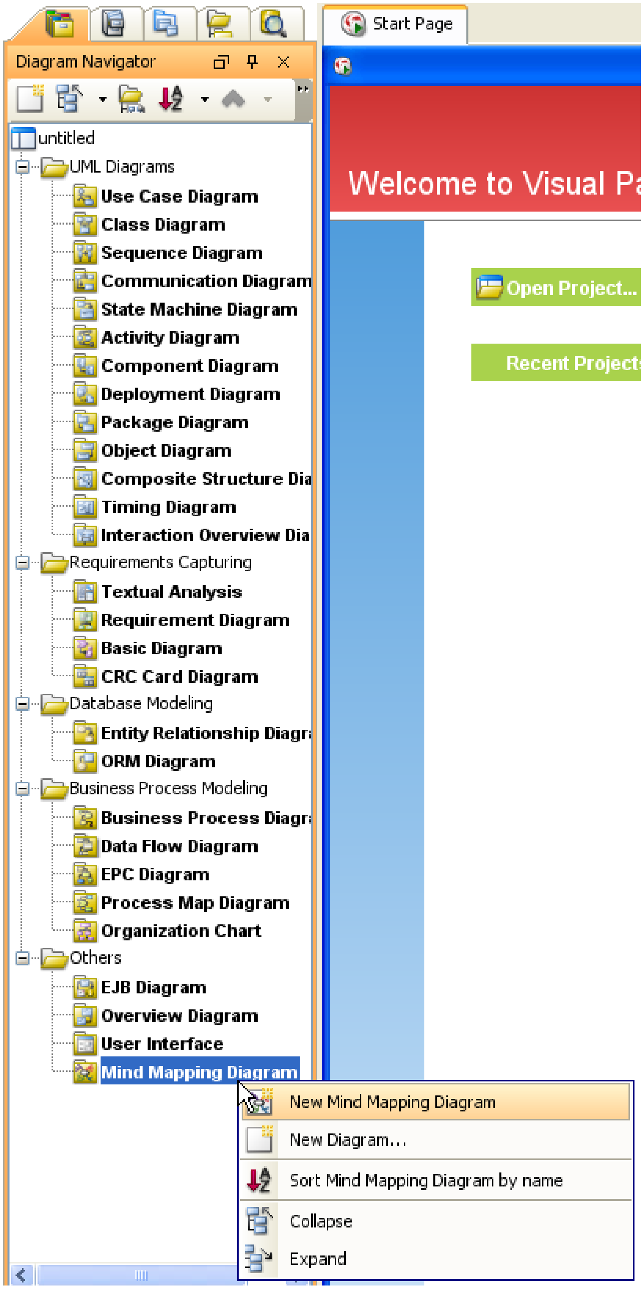


Figure 8-1 *Select New Mind Mapping Diagram from popup menu of Diagram Navigator*

Naming Root Node

New diagram created a node and started editing by default. Let's type *Root* as name .

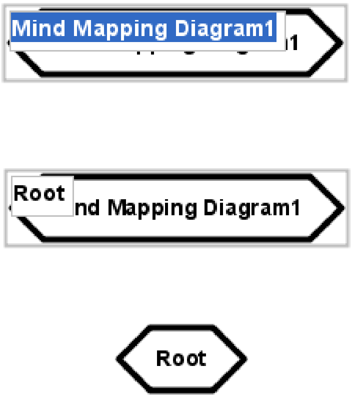


Figure 8-2 *Renaming node*

Creating Branch with Resource Centric Interface

1. Click on the node to show resource centric interface.

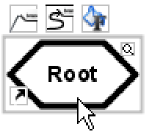


Figure 8-3 *Showing resource centric interface*

2. mouse over **Branch -> Node** resource.

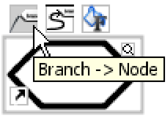


Figure 8-4 *Mouse over resource*

3. Drag resource to empty space on diagram pane.

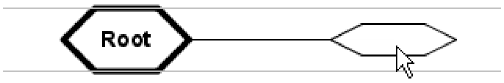


Figure 8-5 *Dragging resource*

4. Release the mouse, new branch and node are created.

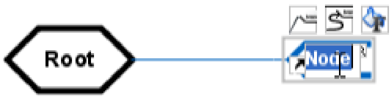


Figure 8-6 *Branch andn Node are created*

Applying Color and Icon to Mind Mapping Diagram

Changing Line Style of Node

1. Select node to show resource centric interface. Click on **Format** and select **Line...** from popup menu.

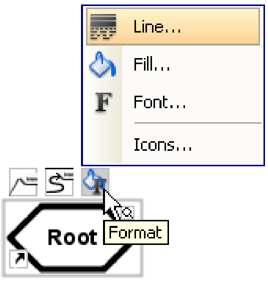
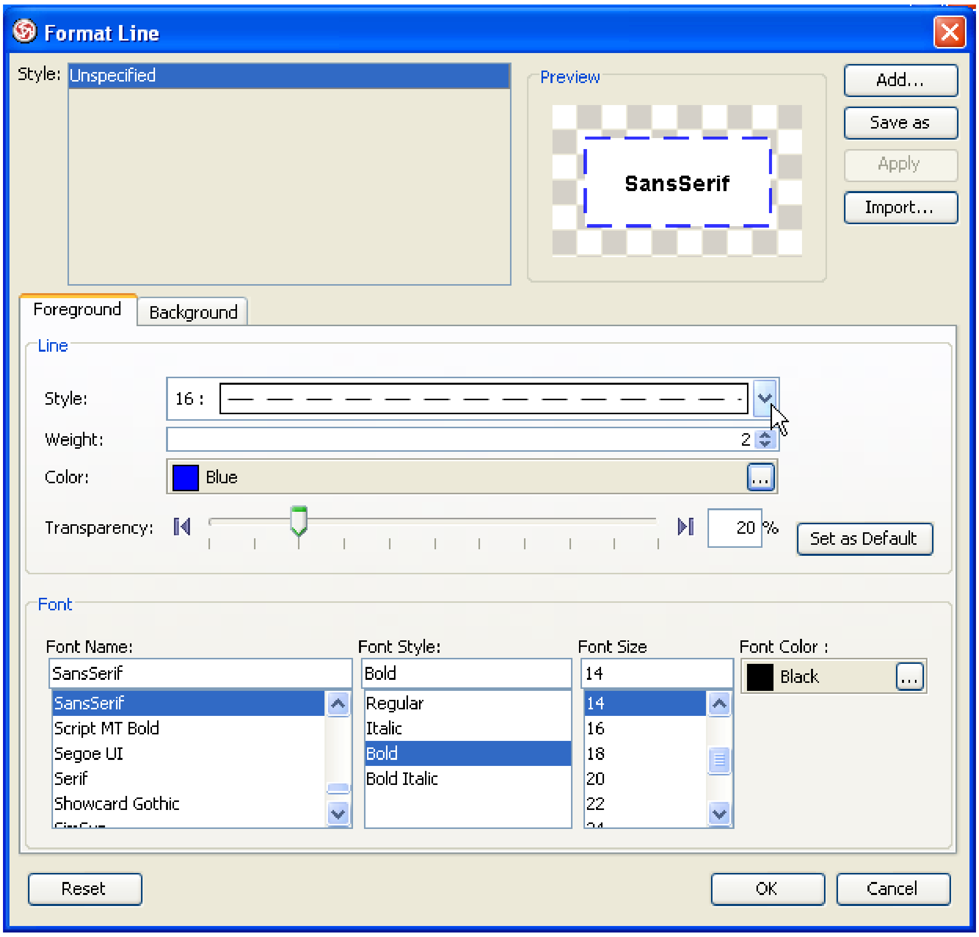


Figure 8-7 *Resource centic interface*



Style: Unspecified

Line

Style: 16 :

Weight: 2

Color: Blue

Transparency: 20 %

Font

Font Name: SansSerif (SansSerif, Script MT Bold, Segoe UI, Serif, Showcard Gothic)

Font Style: Bold (Regular, Italic, Bold, Bold Italic)

Font Size 14 (14, 16, 18, 20, 22)

Font Color : Black

Figure 8-8 *Formatting Dialog*

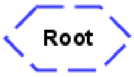


Figure 8-9 *Line formant Modified*

Changing Fill Color of Node

1. Select node to show resource centric interface. Click on **Format** and select **Fill...** from popup menu.

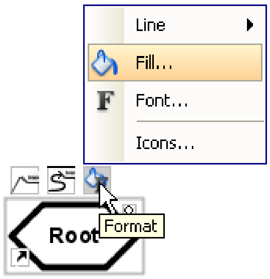
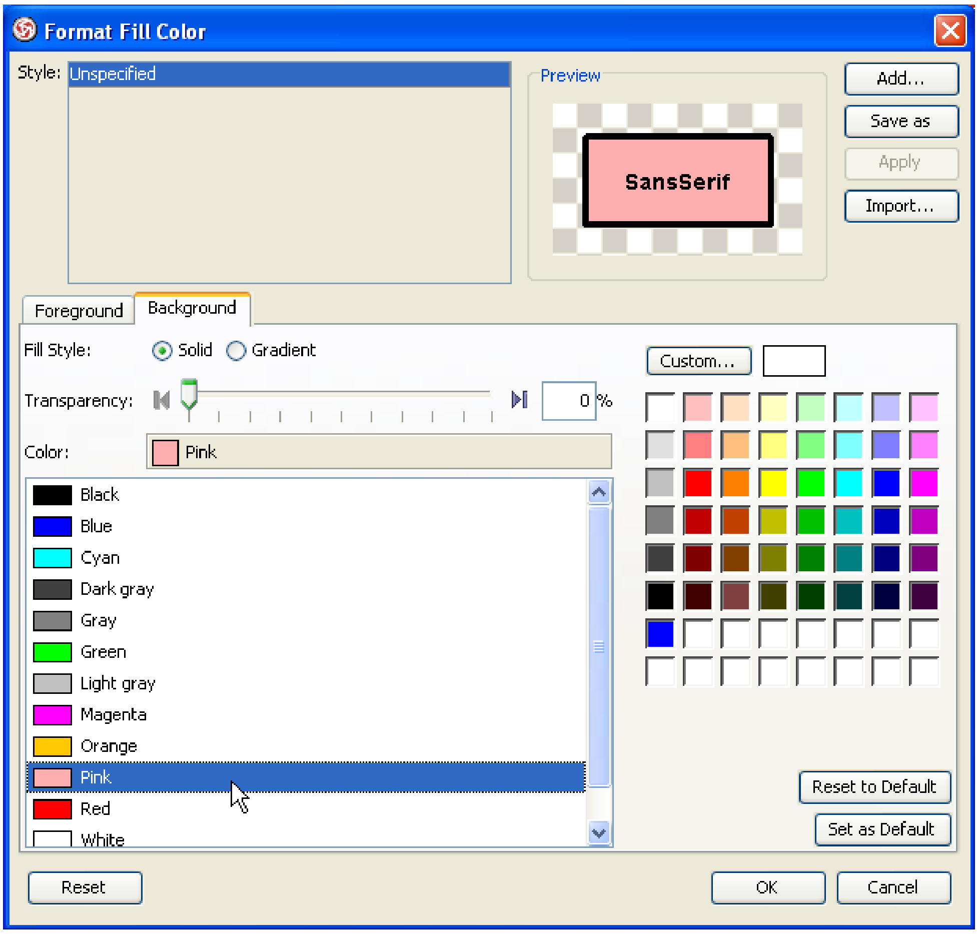


Figure 8-10 Resource centic interface

2. Format fill dialog show, modify TODO.



Style: Unspecified

Fill Style: ● Solid ○ Gradient

Transparency: 0 %

Color: Pink

- Black
- Blue
- Cyan
- Dark gray
- Gray
- Green
- Light gray
- Magenta
- Orange
- Pink
- Red
- White

Figure 8-11 Formatting Dialog

3. Here is the result.

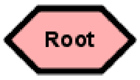


Figure 8-12 Fill formant Modified

Changing Font of Node

1. Select node to show resource centric interface. Click on **Format** and select **Font...** from popup menu.

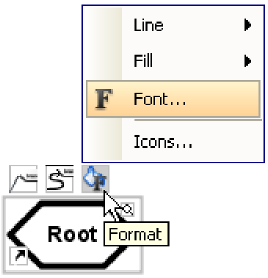
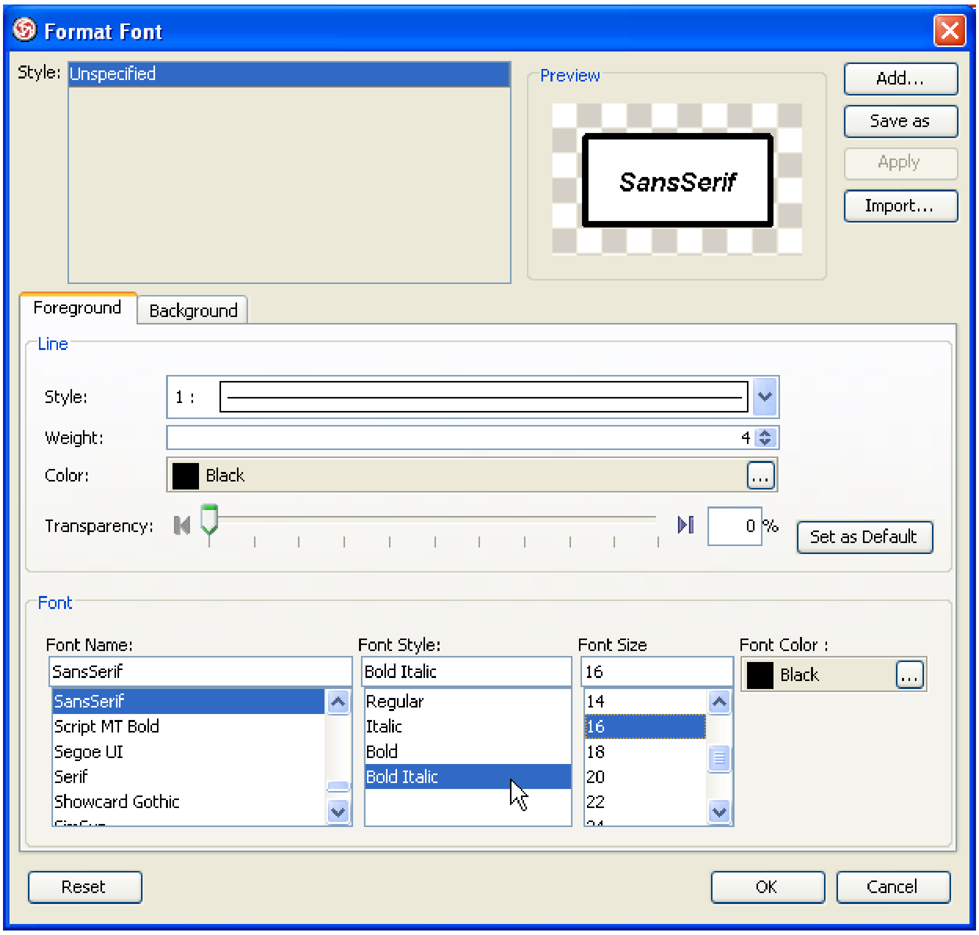


Figure 8-13 *Resource centic interface*



Style: Unspecified

Line

Style: 1 :

Weight: 4

Color: Black

Transparency: 0 %

Font

Font Name: SansSerif

Font Style: Bold Italic

Font Size 16

Font Color : Black

Figure 8-14 *Formatting Dialog*

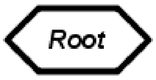


Figure 8-15 *Font formant Modified*

Changing Icon of Node

1. Select node to show resource centric interface. Click on **Format** and select **Icons...** from popup menu.

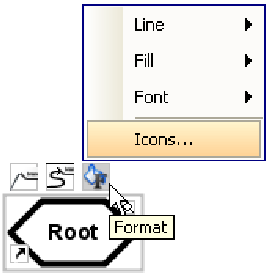


Figure 8-16 *Resource centic interface*

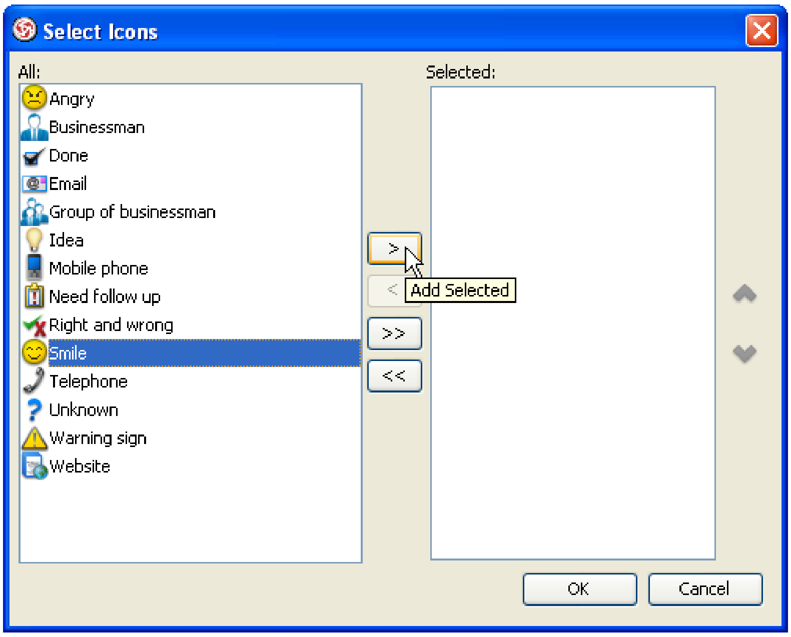


Figure 8-17 *Select icons Dialog*

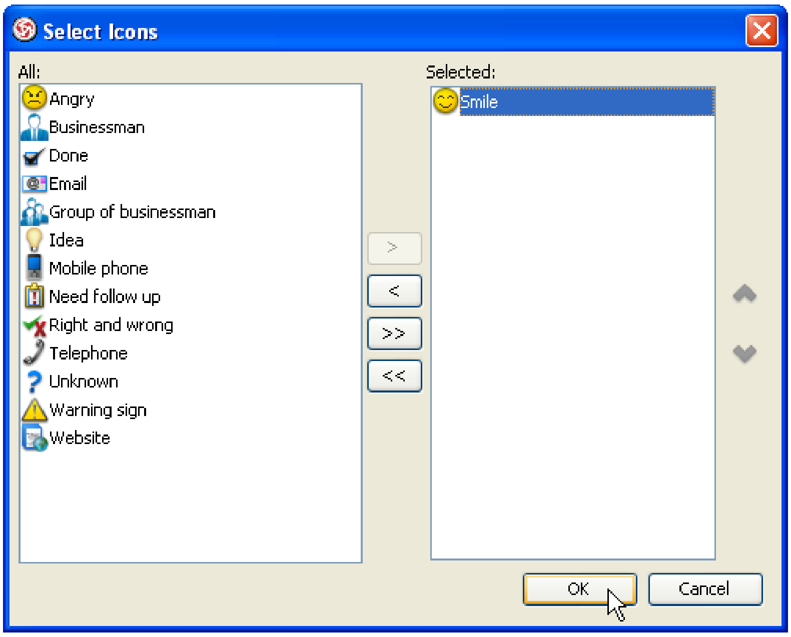


Figure 8-18 *Select Icons Dialog with selected icon*

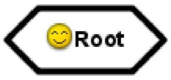


__Figure 8-19__ Icon added

Manage and Apply Styles

VP-UML allows you to define you own style and apply to other shape by simple steps.

Adding Style

To open the Style dialog, Select **View > Styles...** from the main menu.

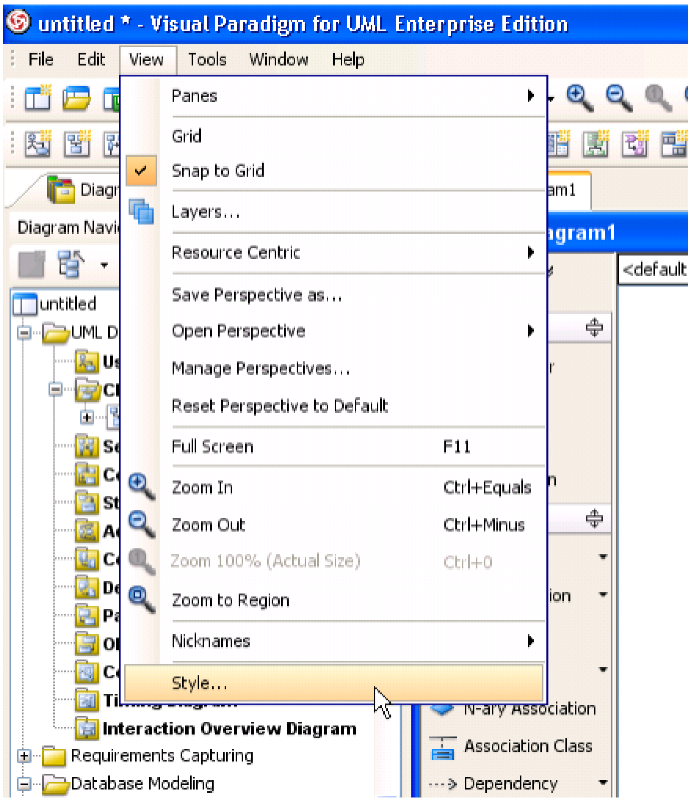


Figure 8-20 *Open Style dialog*

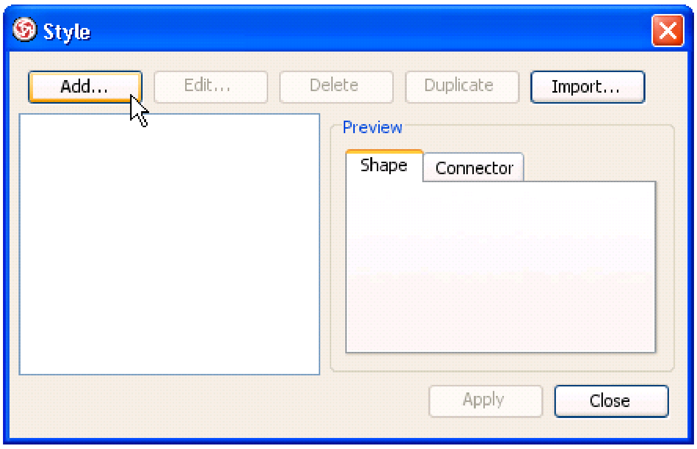
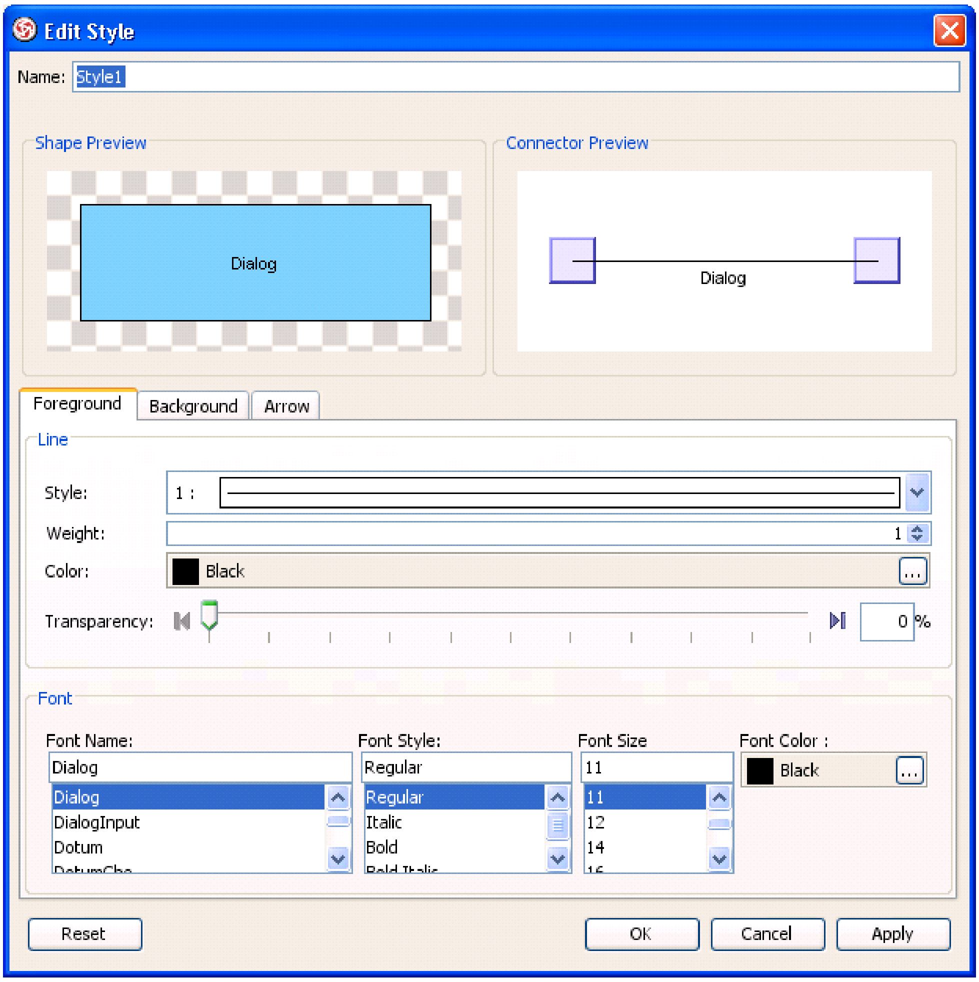


Figure 8-21 *Style dialog*

In the Style dialog, click **Add...** to create and edit a new style.



Name: Style1

Style: 1 :

Weight: 1

Color: Black

Transparency: 0 %

Font Name: Dialog

Dialog
DialogInput
Dotum
DotumChe

Font Style: Regular

Regular
Italic
Bold
Bold Italic

Font Size 11

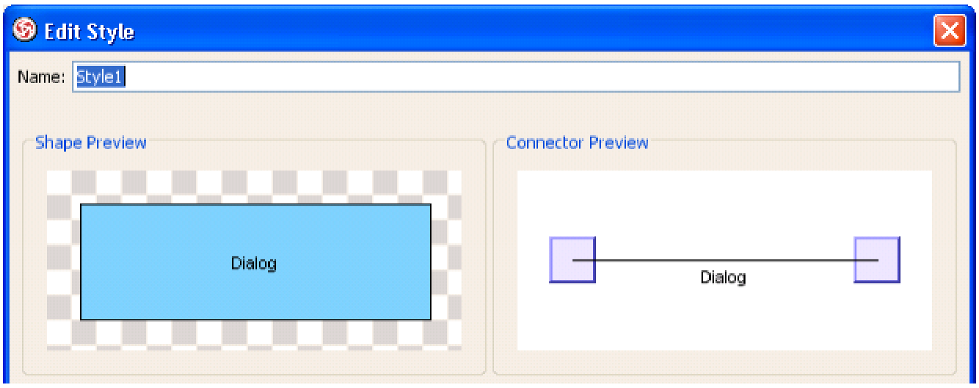
11
12
14
16

Font Color : Black

Figure 8-22 Edit Style dialog

In the **Edit Style** dialog, you can change the followings setting of the style:

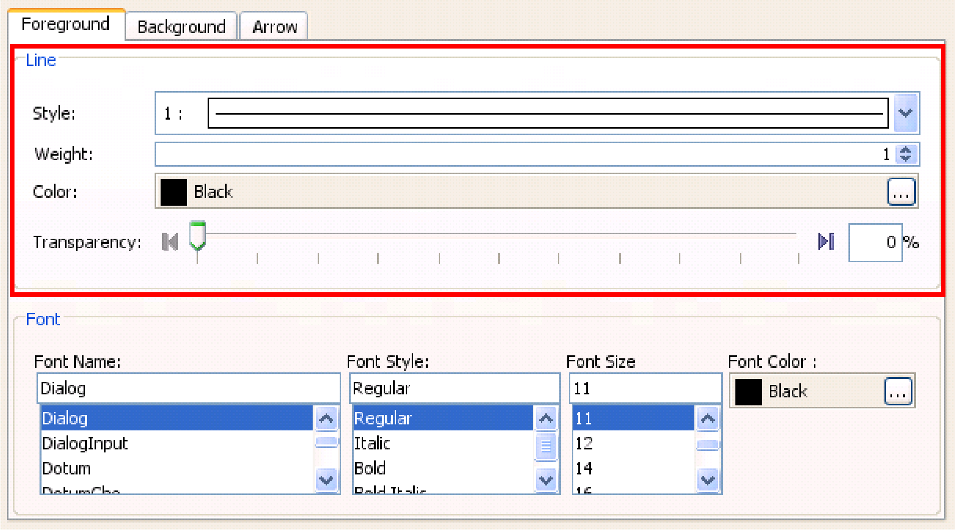
- Name



Name: Style1

Figure 8-23 edit name

- Foreground Line Style

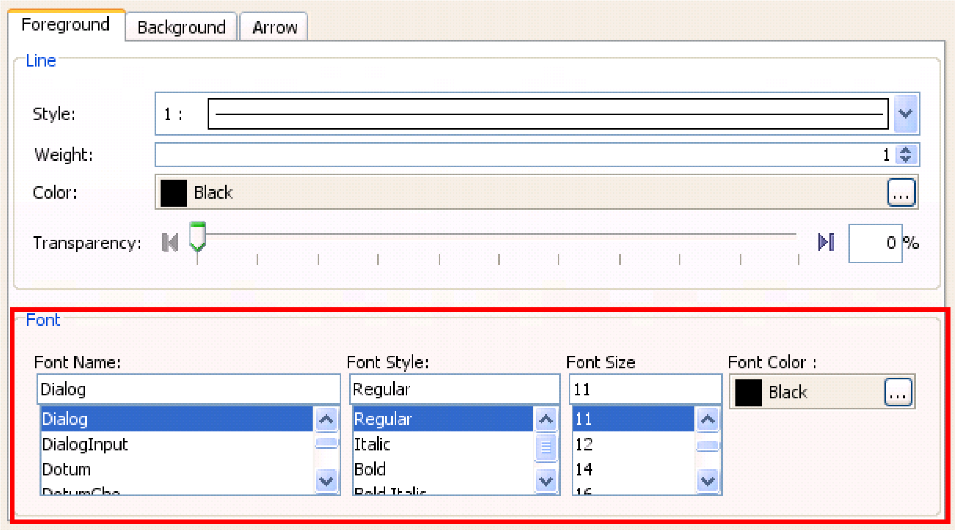


Style: 1 :
Weight: 1
Color: Black
Transparency: 0 %

Font Name: Dialog
Font Style: Regular
Font Size 11
Font Color : Black

Figure 8-24 edit line style

- Font Style

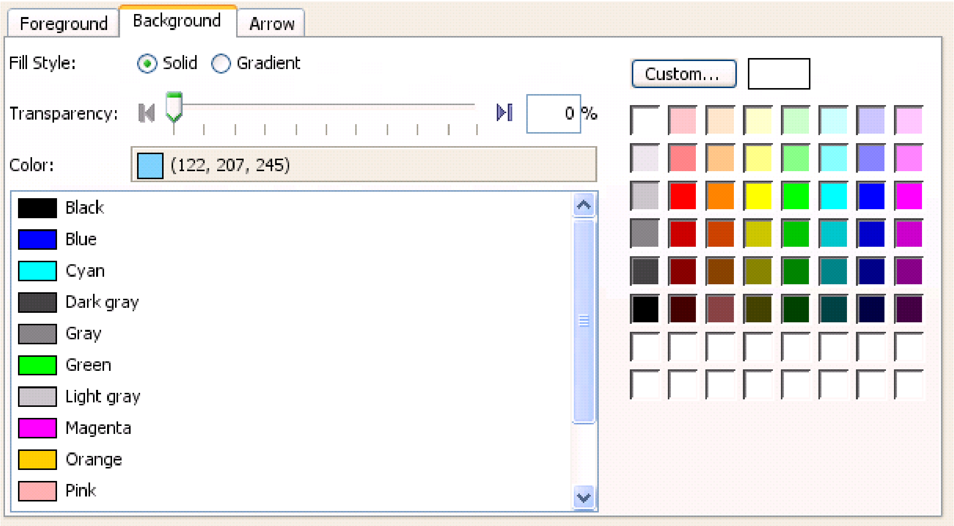


Style: 1 :
Weight: 1
Color: Black
Transparency: 0 %

Font Name: Dialog
Font Style: Regular
Font Size 11
Font Color : Black

Figure 8-25 edit font style

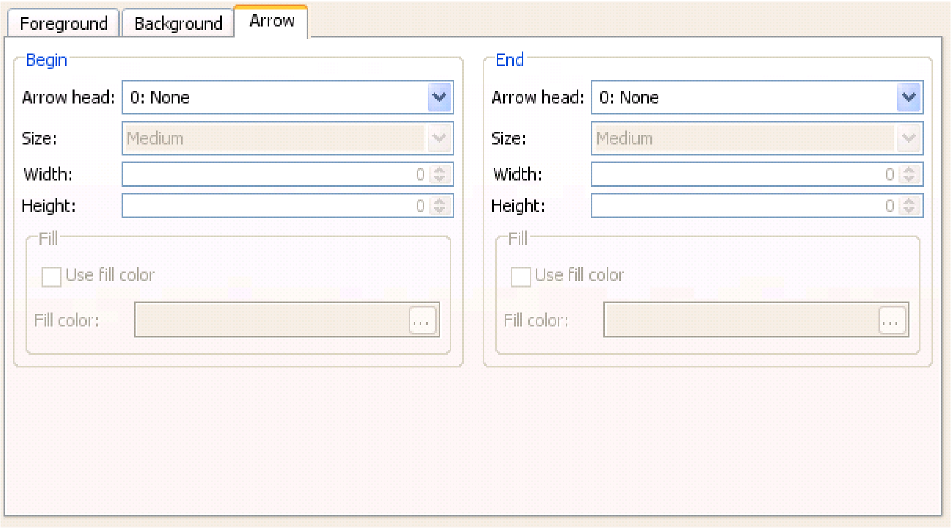
- Background Style



Fill Style: ● Solid ○ Gradient
Transparency: 0 %
Color: (122, 207, 245)

Figure 8-26 edit background style

- Arrow Style

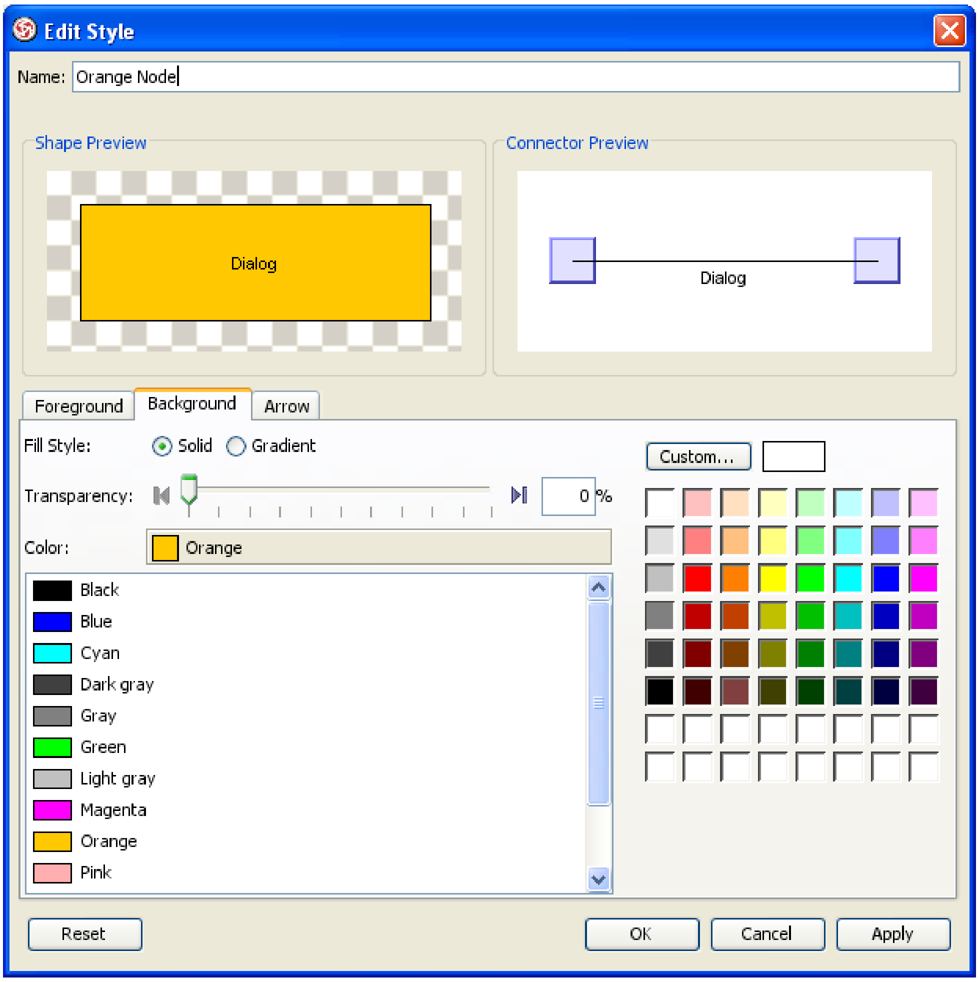


Begin
Arrow head: 0: None
Size: Medium
Width: 0
Height: 0
Fill
□ Use fill color
Fill color: ...

End
Arrow head: 0: None
Size: Medium
Width: 0
Height: 0
Fill
□ Use fill color
Fill color: ...

Figure 8-27 edit arrow style

After change the settings, select **OK** to add the style.



Name: Orange Node

Fill Style: ● Solid ○ Gradient

Transparency: 0 %

Color: Orange

Figure 8-28 style settings

The style was added to the project.

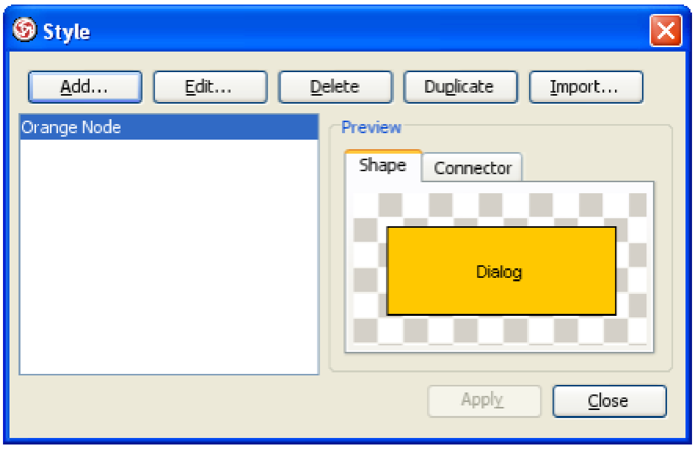


Figure 8-29 style added

Applying Style

Upon keeping the Style dialog open, create a new use case diagram together with a new use case.

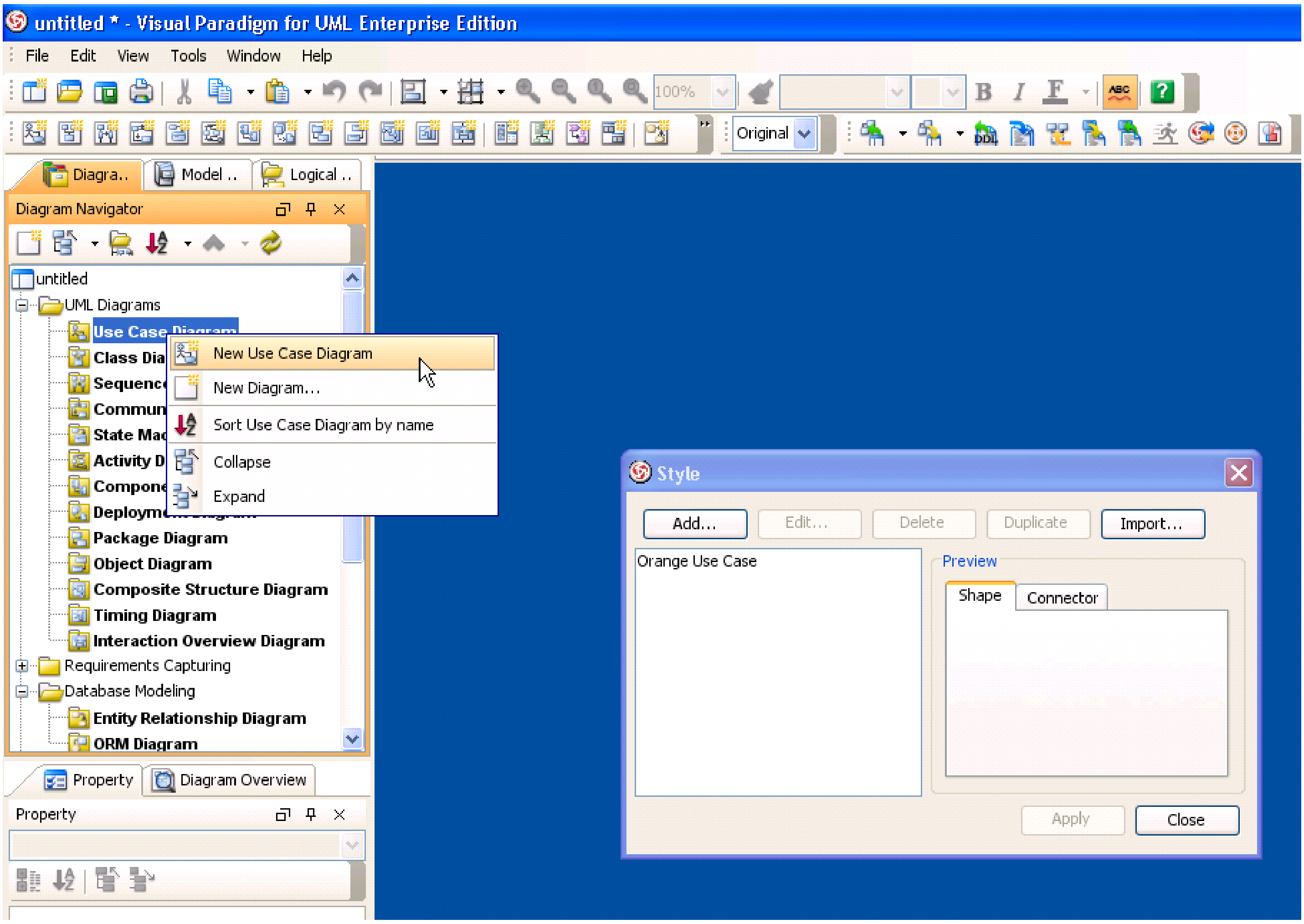


Figure 8-30 New use case diagram

Select the shape, and click **Apply** in the **Style** dialog.

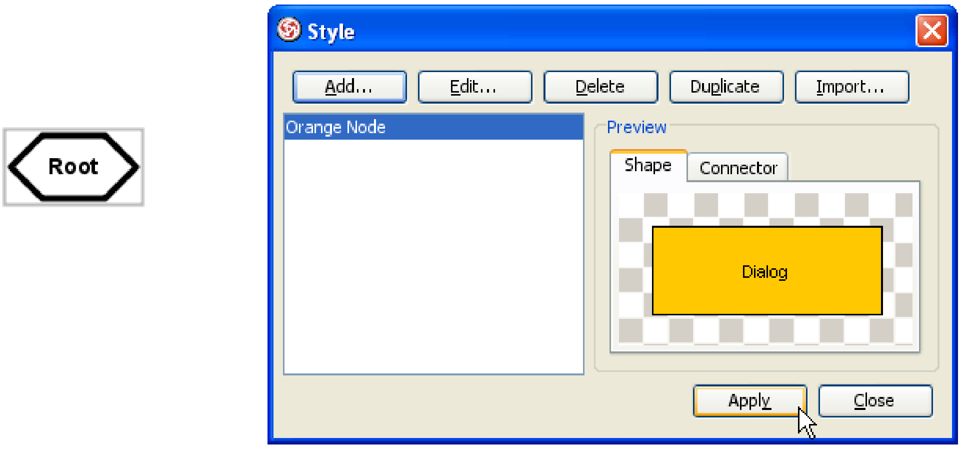


Figure 8-31 *Apply style to shape*

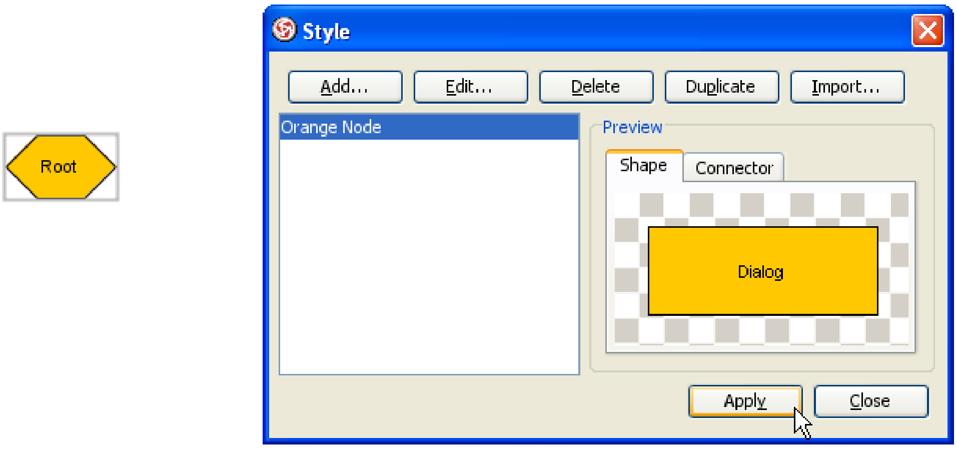


Figure 8-32 *Apply style result*

Relating Mind Mapping Node with Link

Making Link between Nodes

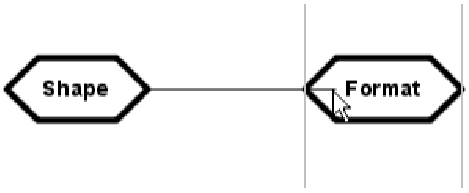


Figure 8-34 *Dragging*

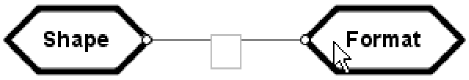


Figure 8-35 *Link created*

Type of Link

Reference to External Documents

Reference to Files

1. Mouse over shape and click on References resource.

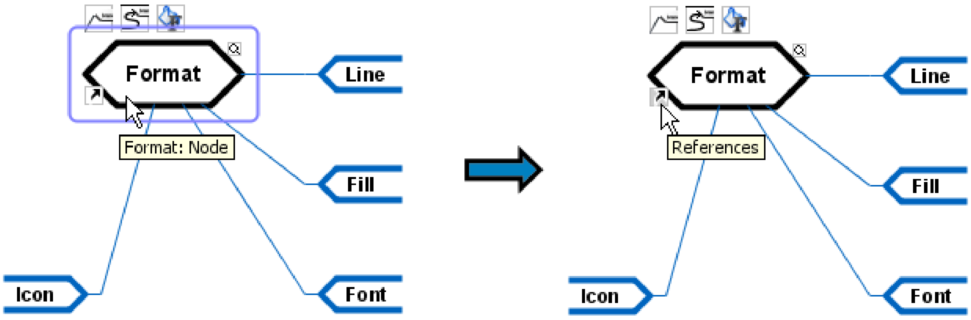


Figure 8-36 *Mouse over references resource*

2. Click on Edit References... menu item.

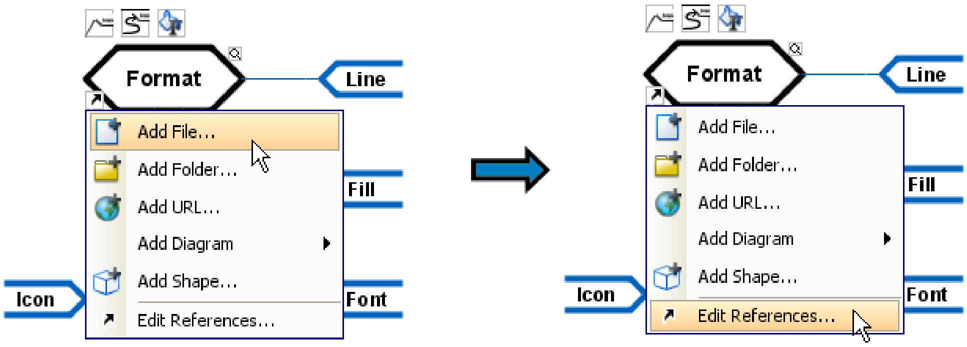


Figure 8-37 *Edit references from resource*

3. Press Add File... button.

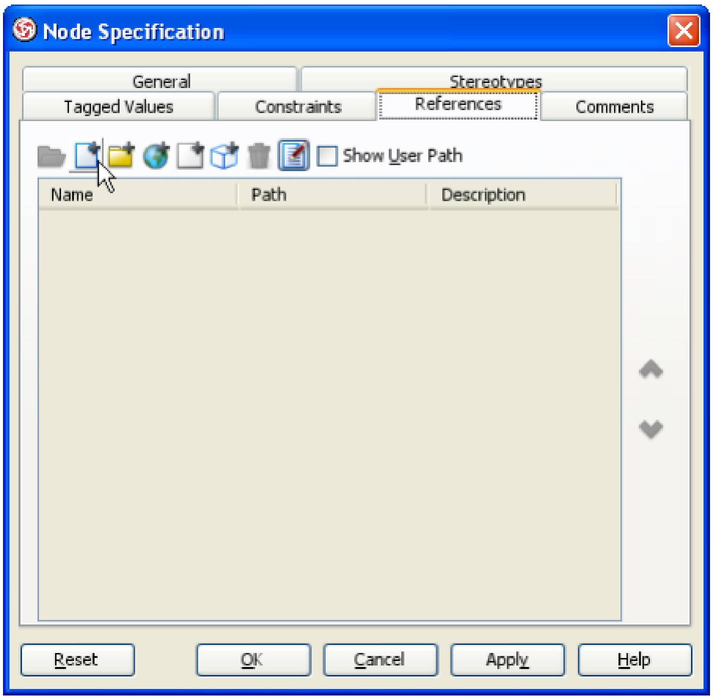
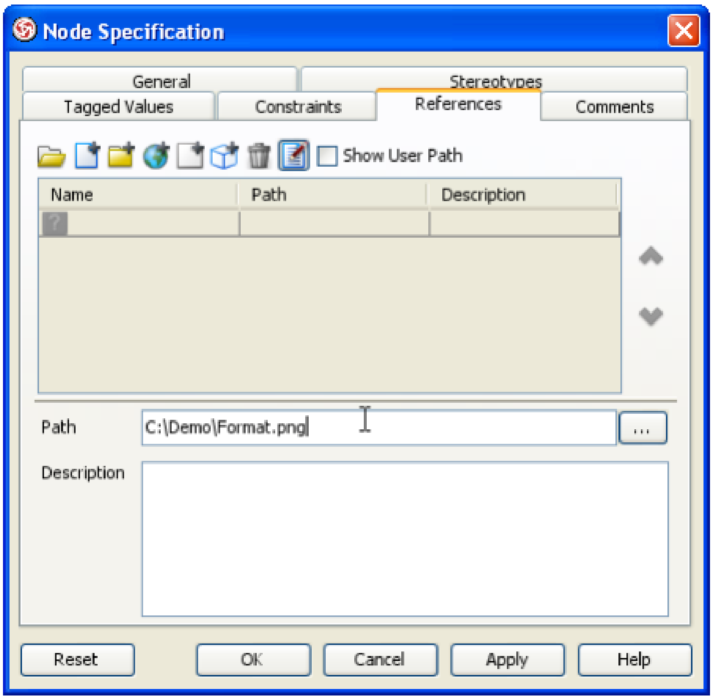


Figure 8-38 *Adding file reference*

4. Input path of reference file.

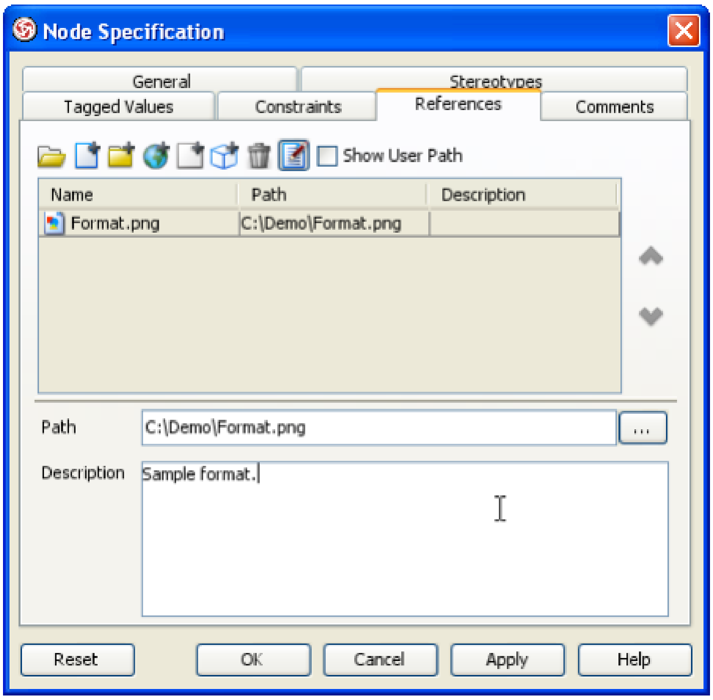


Path C:\Demo\Format.png

Description

Figure 8-39 *Inputting file reference path*

5. Input description of the file.



☐ Show User Path

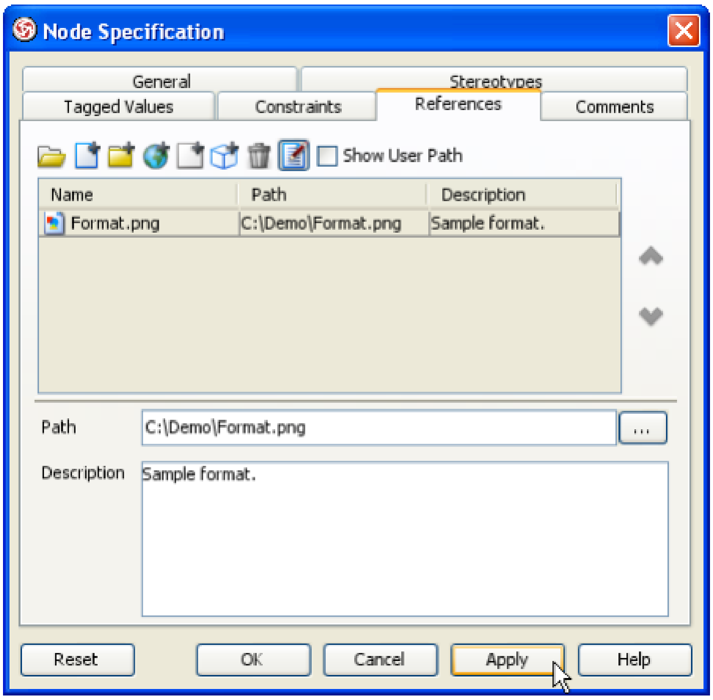
Name	Path	Description
Format.png	C:\Demo\Format.png	

Path: C:\Demo\Format.png

Description: Sample format.

Figure 8-40 *Inputting file reference description*

6. Press Apply button to confirm the reference creation.



☐ Show User Path

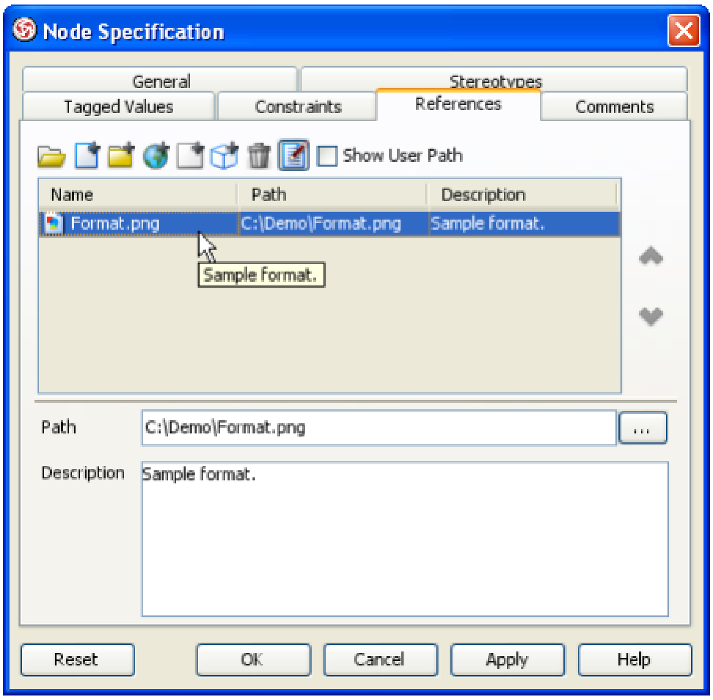
Name	Path	Description
Format.png	C:\Demo\Format.png	Sample format.

Path: C:\Demo\Format.png

Description: Sample format.

Figure 8-41 *Apply file reference*

7. Select the file reference to be opened.



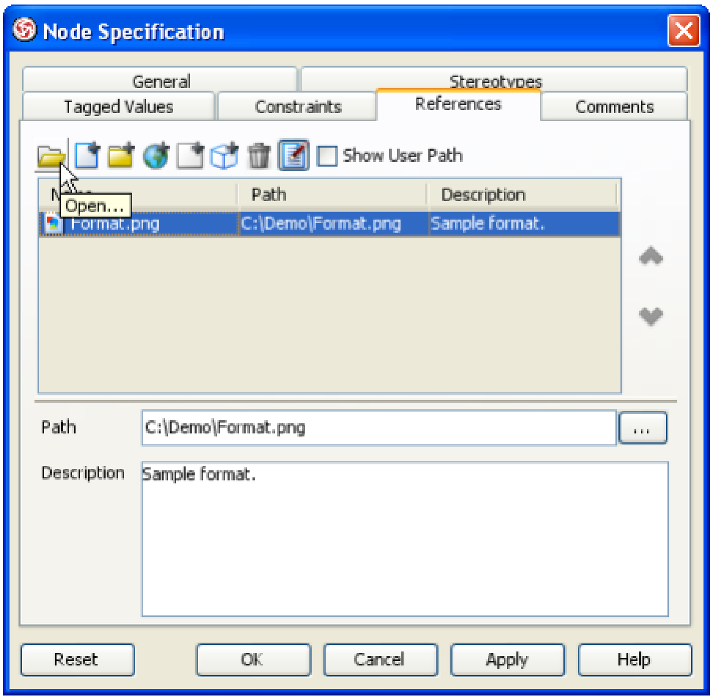
Name	Path	Description
Format.png	C:\Demo\Format.png	Sample format.

Path: C:\Demo\Format.png

Description: Sample format.

Figure 8-42 Select file reference

8. Press Open... button.



Name	Path	Description
Format.png	C:\Demo\Format.png	Sample format.

Path: C:\Demo\Format.png

Description: Sample format.

Figure 8-43 Open file reference

Reference to Folder

1. Mouse over shape and click on References resource.

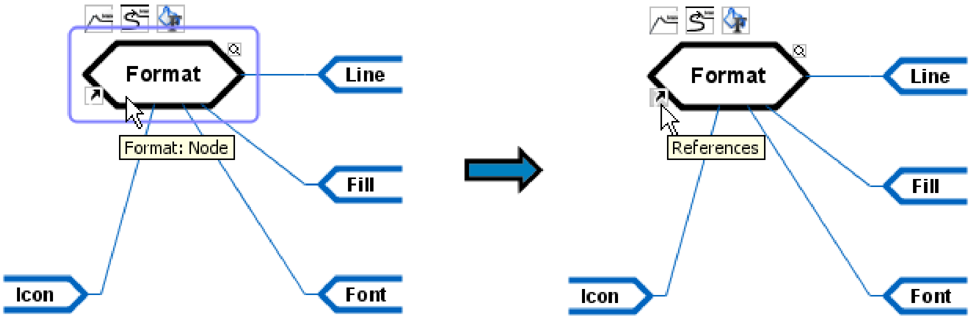


Figure 8-44 Mouse over references resource

2. Click on Add Folder... menu item.

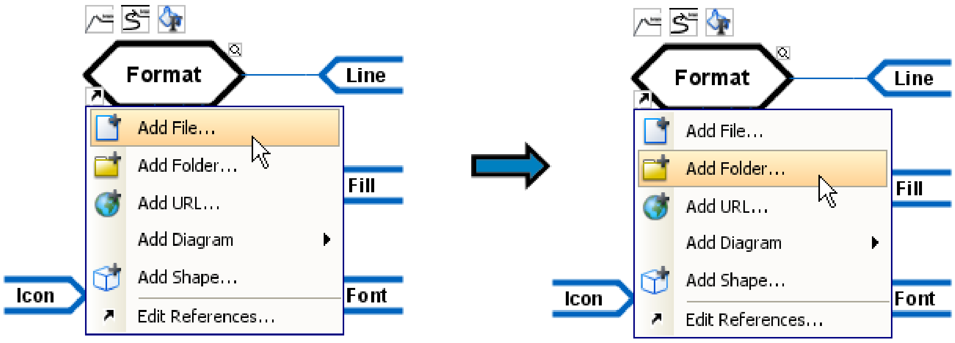


Figure 8-45 Add folder from resource

3. Press ... button.

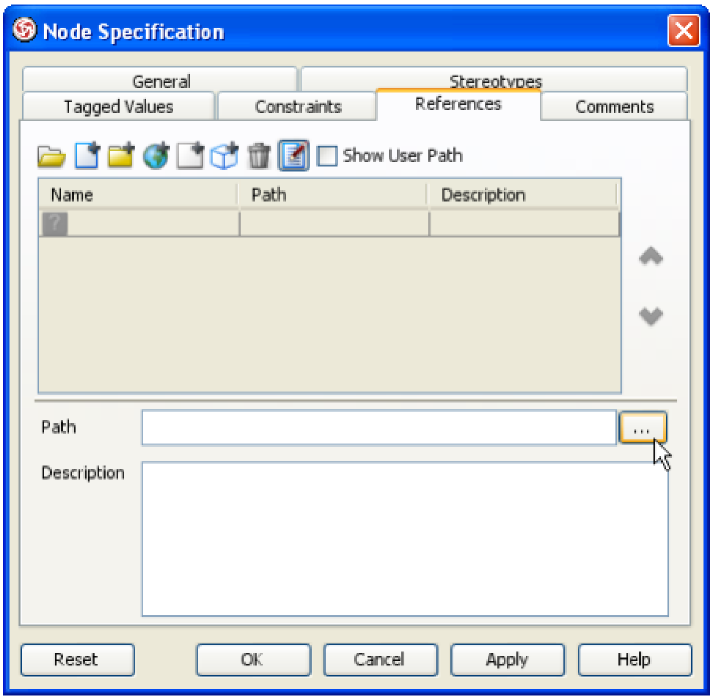
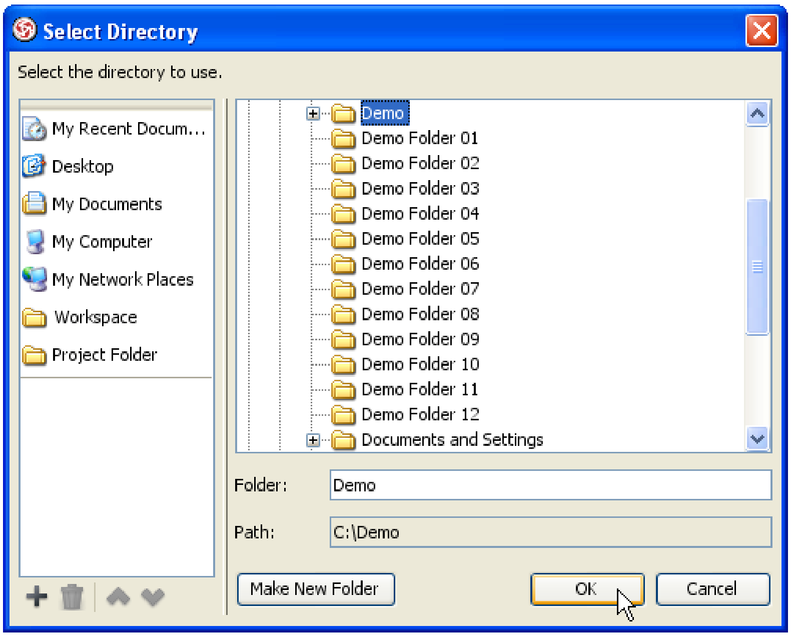


Figure 8-46 Browse folder

4. Select folder to be referenced.



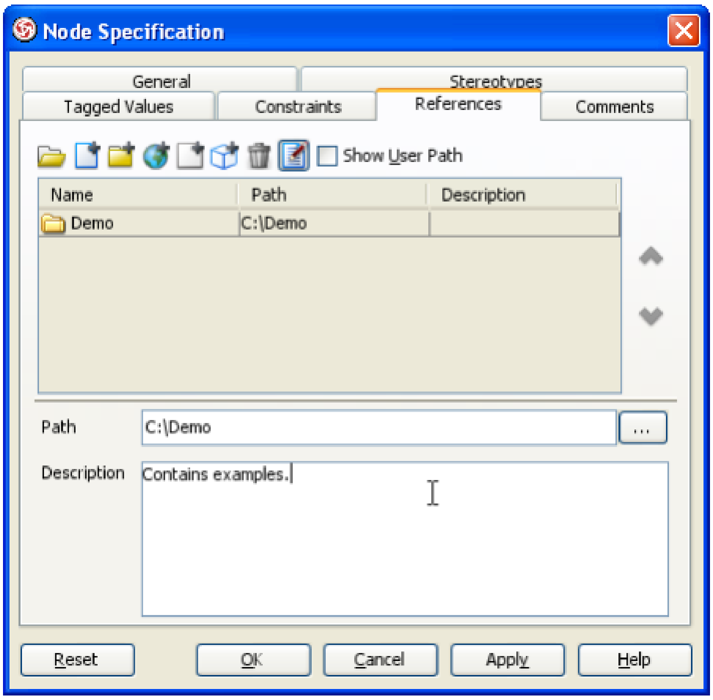
Select the directory to use.

Folder: Demo

Path: C:\Demo

__Figure 8-47__ Select reference folder

5. Input description of the folder.



□ Show User Path

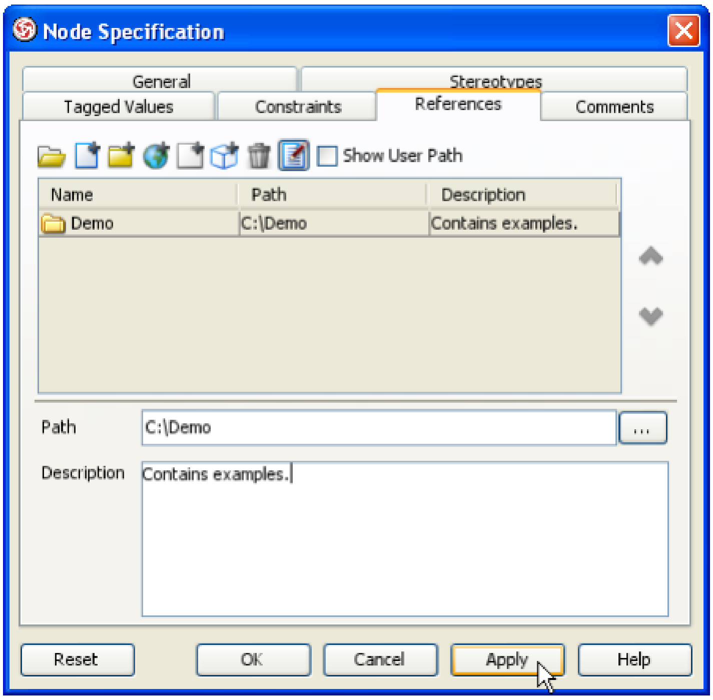
Name	Path	Description
Demo	C:\Demo	

Path: C:\Demo

Description: Contains examples.

__Figure 8-48__ Inputting folder reference description

6. Press Apply button to confirm the reference creation.



Node Specification

General | Stereotypes | Tagged Values | Constraints | References | Comments

☐ Show User Path

Name	Path	Description
Demo	C:\Demo	Contains examples.

Path: C:\Demo

Description: Contains examples.

Figure 8-49 Apply folder reference

7. Mouse over shape and click on References resource.

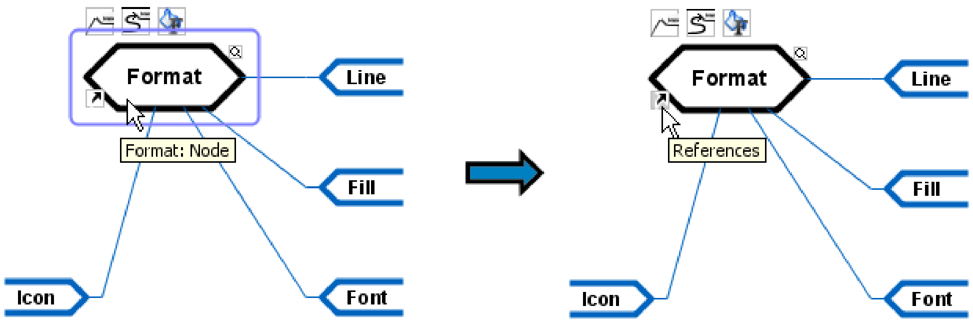


Figure 8-50 Mouse over references resource

8. Click on folder resource menu item.

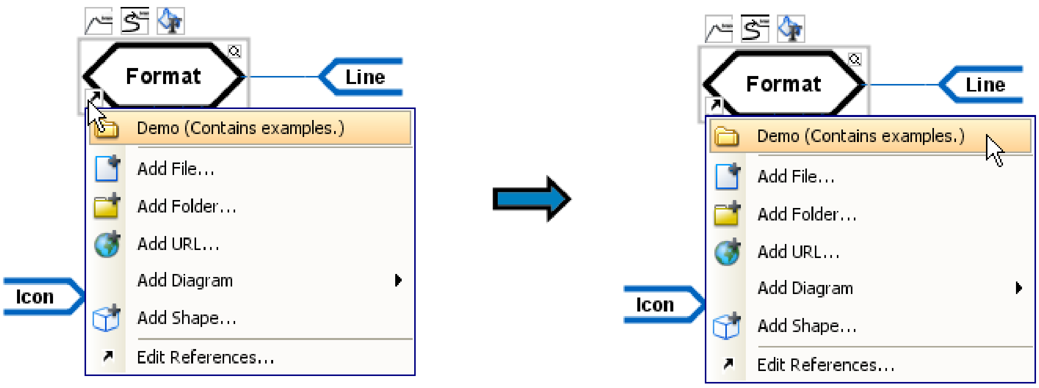


Figure 8-51 Open folder reference

Reference to URL

1. Right click on shape and click on Open Specification... menu item.

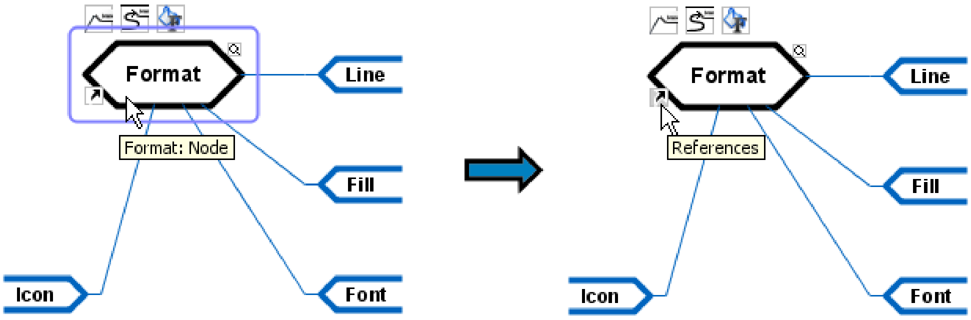
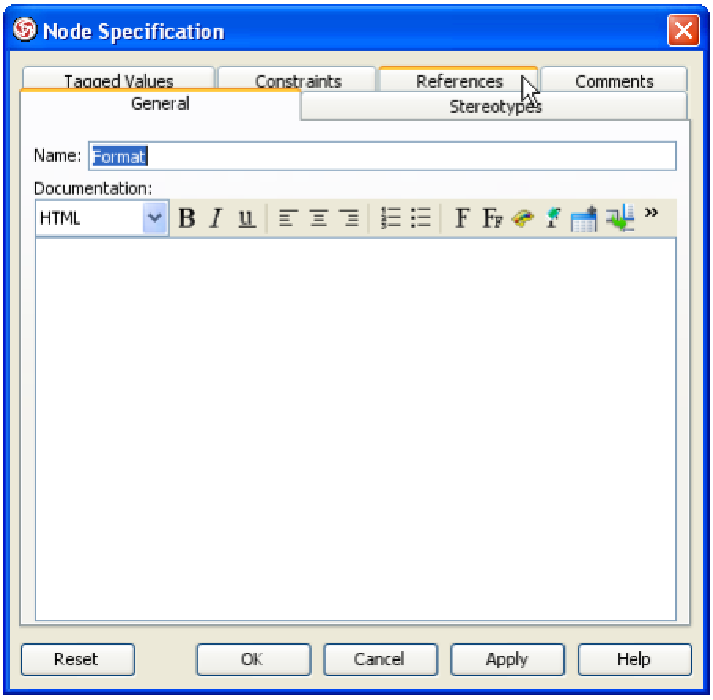


Figure 8-52 *Show popup of shape*

2. Switch to References tab.



Name: Format

Figure 8-53 *Switch to references tab*

3. Press Add URL... button.

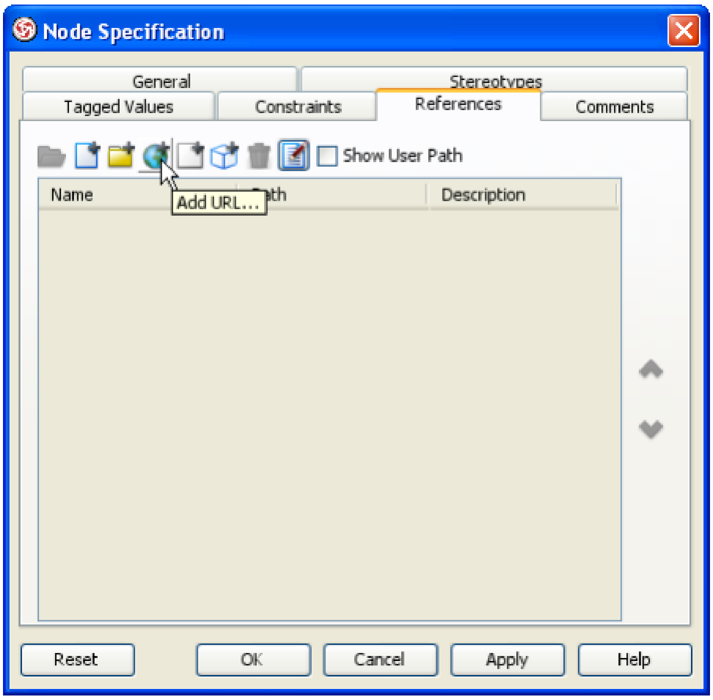
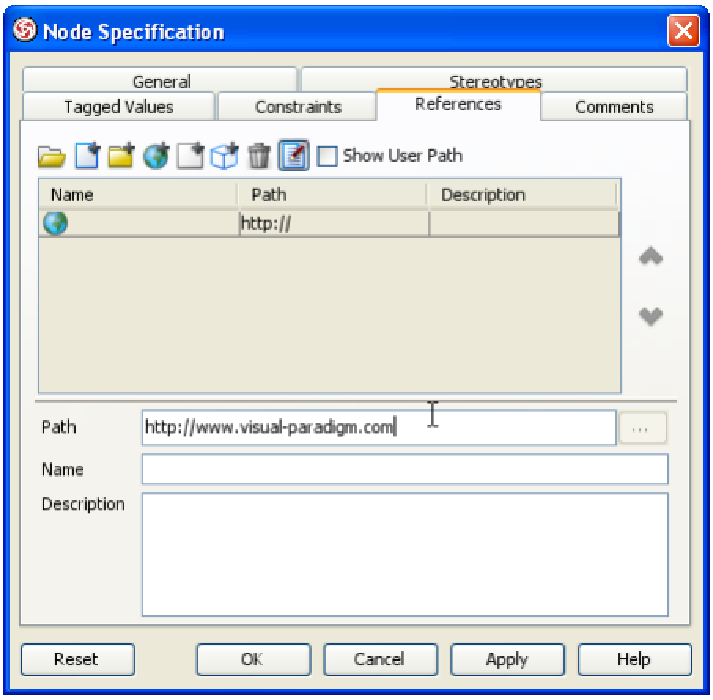


Figure 8-54 Adding URL reference

4. Input path of URL.



Name	Path	Description
	http://	

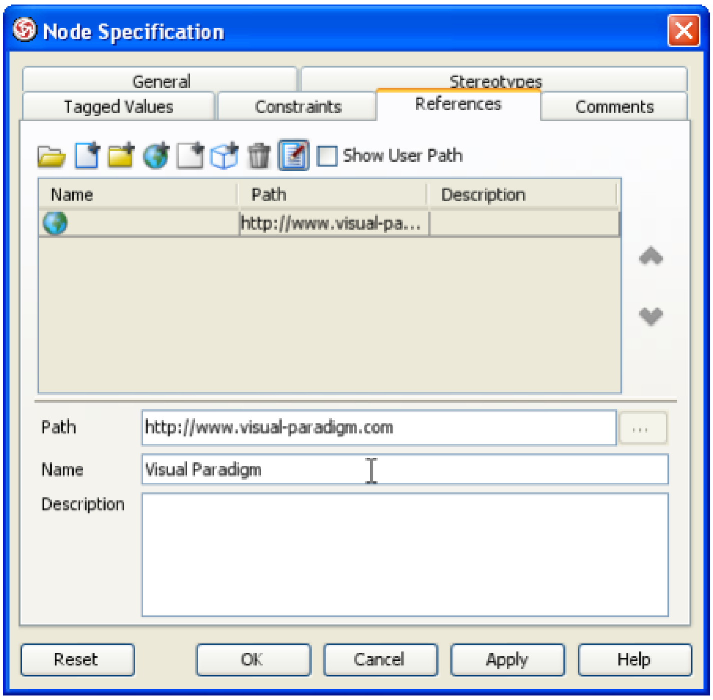
Path: http://www.visual-paradigm.com

Name:

Description:

Figure 8-55 Inputting URL reference path

5. Input URL name.



Name	Path	Description
	http://www.visual-pa...	

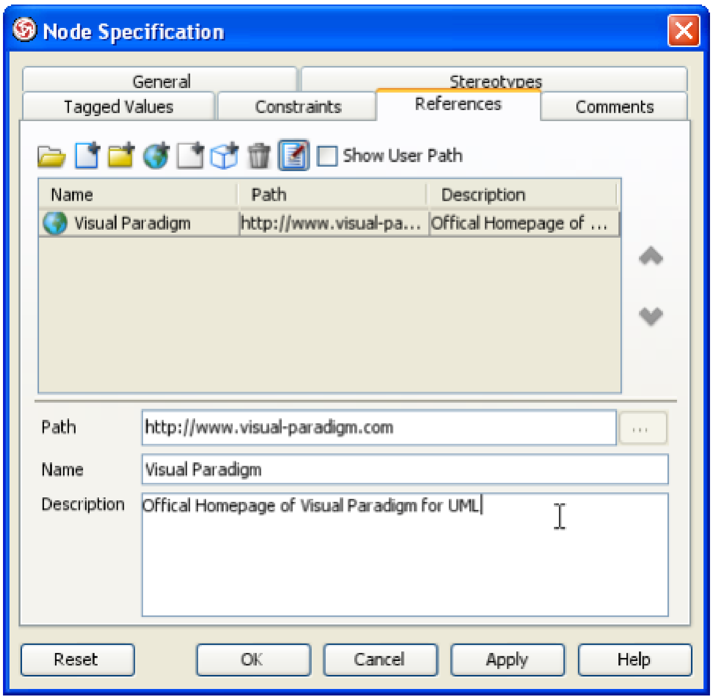
Path: http://www.visual-paradigm.com

Name: Visual Paradigm

Description:

Figure 8-56 *Inputting URL reference name*

6. Input description of the URL.



Name	Path	Description
Visual Paradigm	http://www.visual-pa...	Offical Homepage of ...

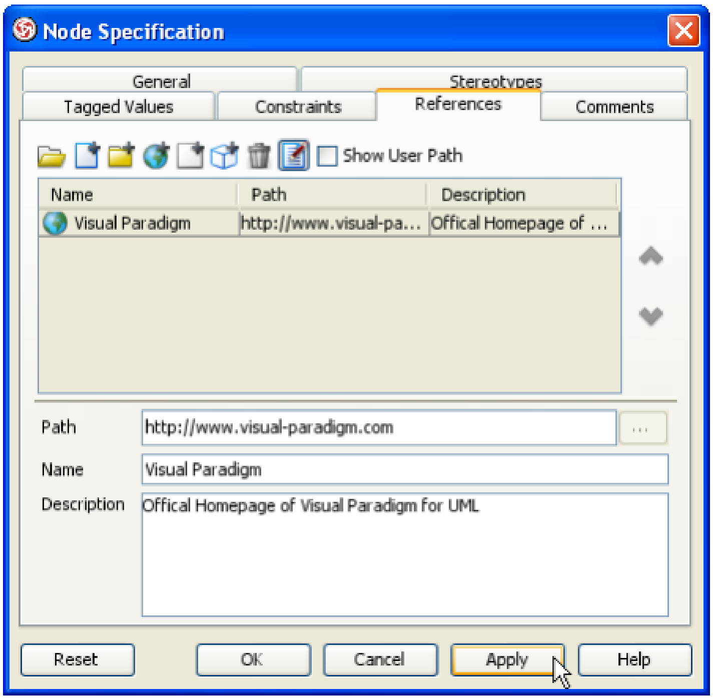
Path: http://www.visual-paradigm.com

Name: Visual Paradigm

Description: Offical Homepage of Visual Paradigm for UML

Figure 8-57 *Inputting URL reference description*

7. Press Apply button to confirm the reference creation.



Name	Path	Description
Visual Paradigm	http://www.visual-pa...	Offical Homepage of ...

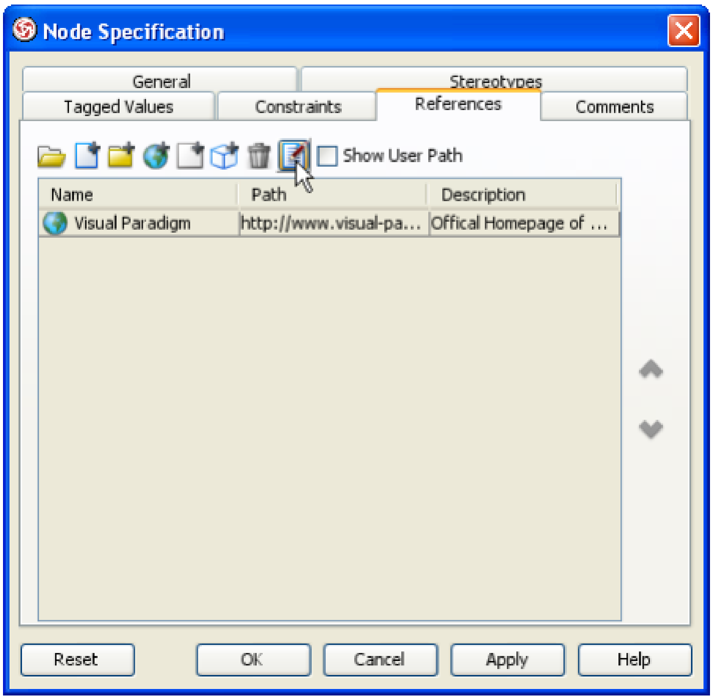
Path: http://www.visual-paradigm.com

Name: Visual Paradigm

Description: Offical Homepage of Visual Paradigm for UML

Figure 8-58 *Apply URL reference*

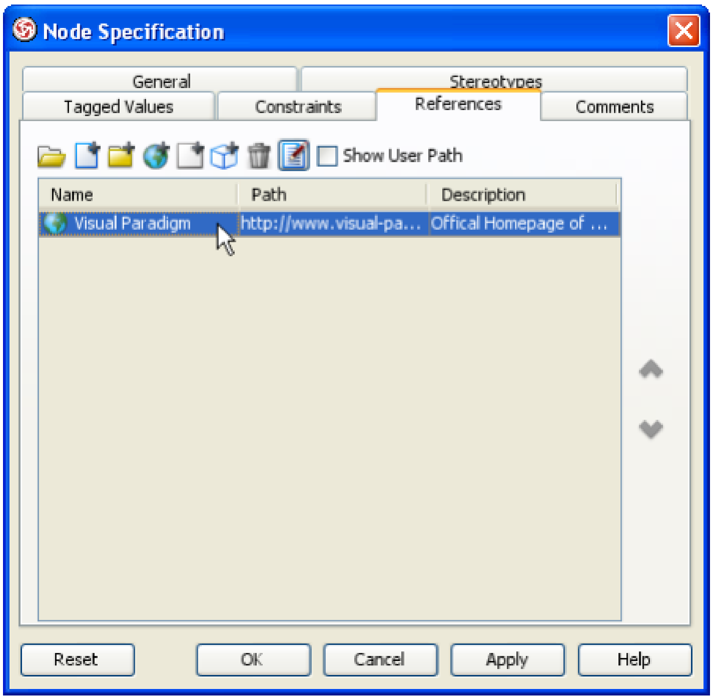
8. Press Details button.



Name	Path	Description
Visual Paradigm	http://www.visual-pa...	Offical Homepage of ...

Figure 8-59 *Hide reference details*

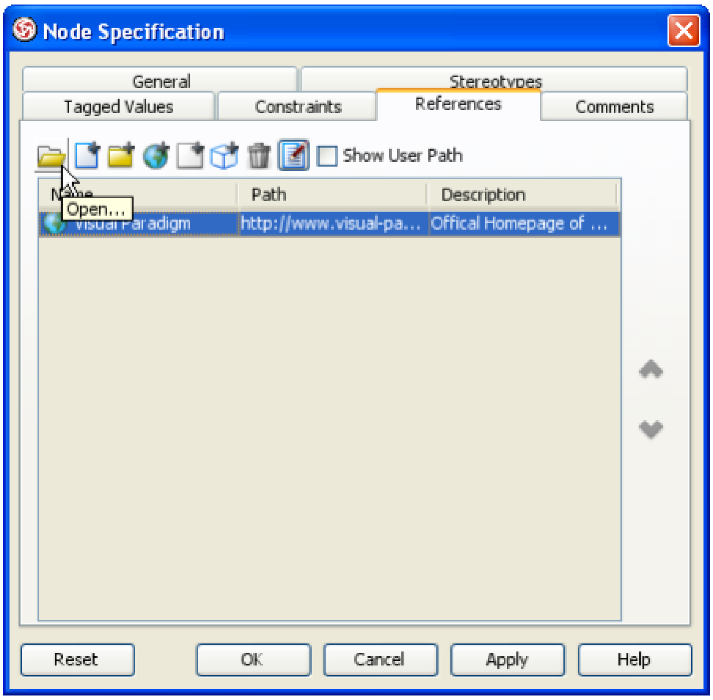
9. Select the URL reference to be opened.



Name	Path	Description
Visual Paradigm	http://www.visual-pa...	Offical Homepage of ...

Figure 8-60 Select URL reference

10. Press Open... button.



Name	Path	Description
Visual Paradigm	http://www.visual-pa...	Offical Homepage of ...

Figure 8-61 Open URL reference

Reference to other Diagram Elements

Reference to Diagrams

1. Mouse over shape and click on References resource.

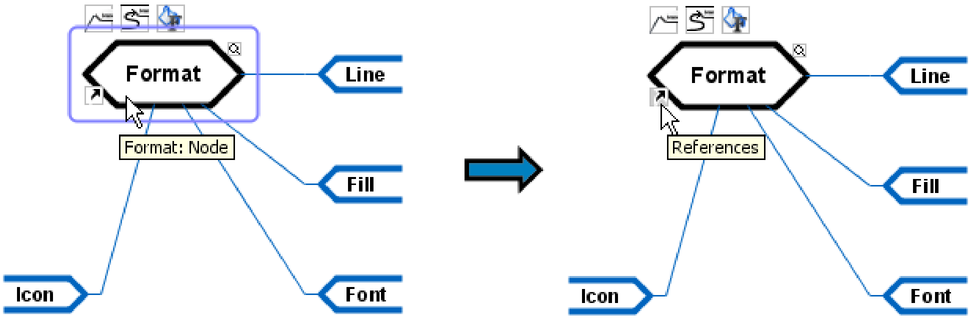


Figure 8-62 Mouse over references resource

2. Click on Edit References... menu item.

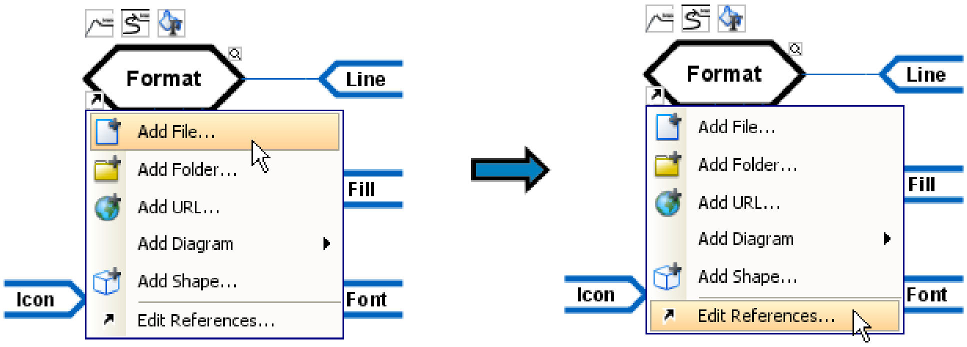


Figure 8-63 Edit references from resource

3. Press Add Diagram... button.

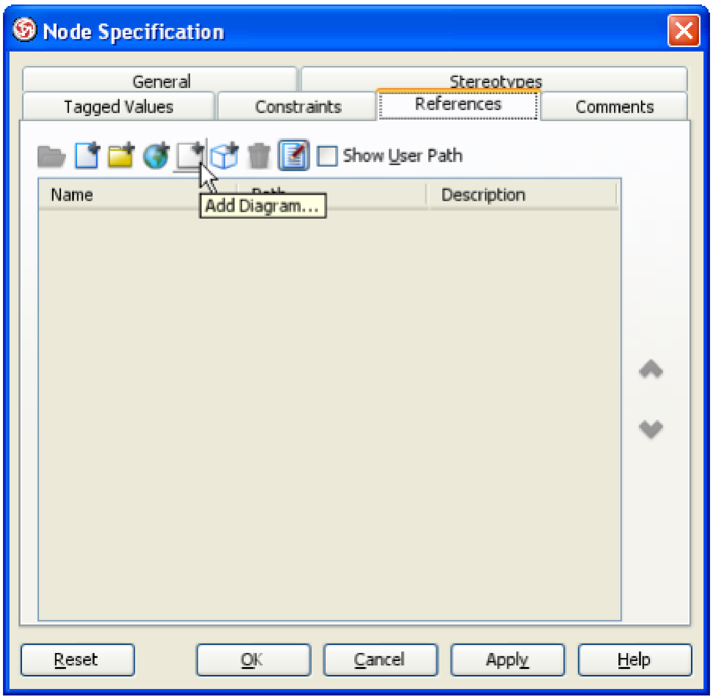


Figure 8-64 Adding diagram reference

4. Select reference diagram.

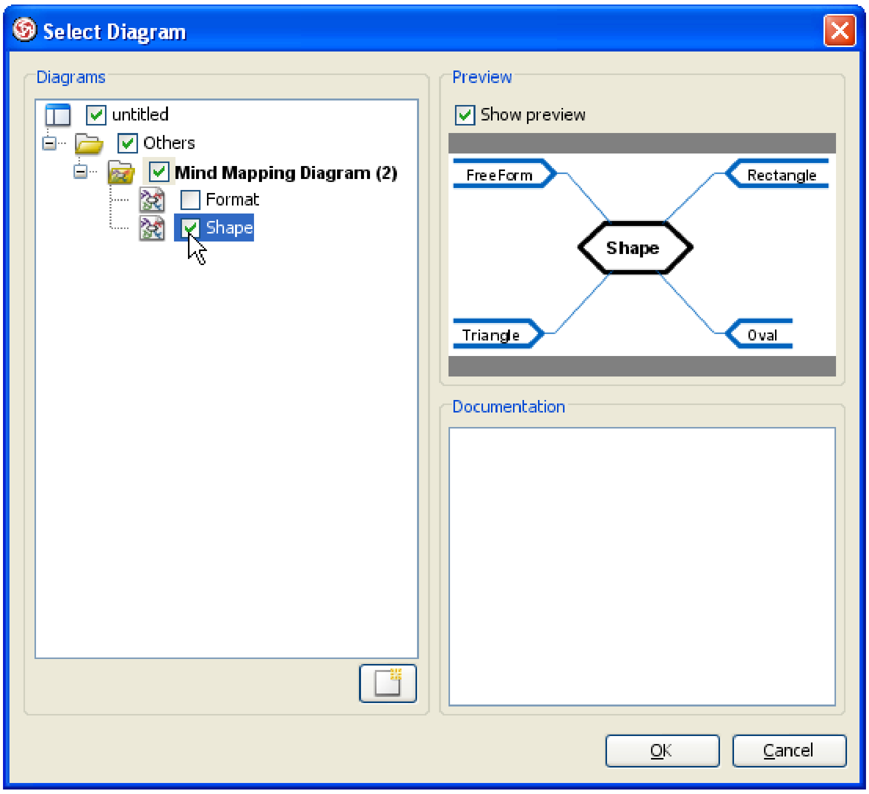
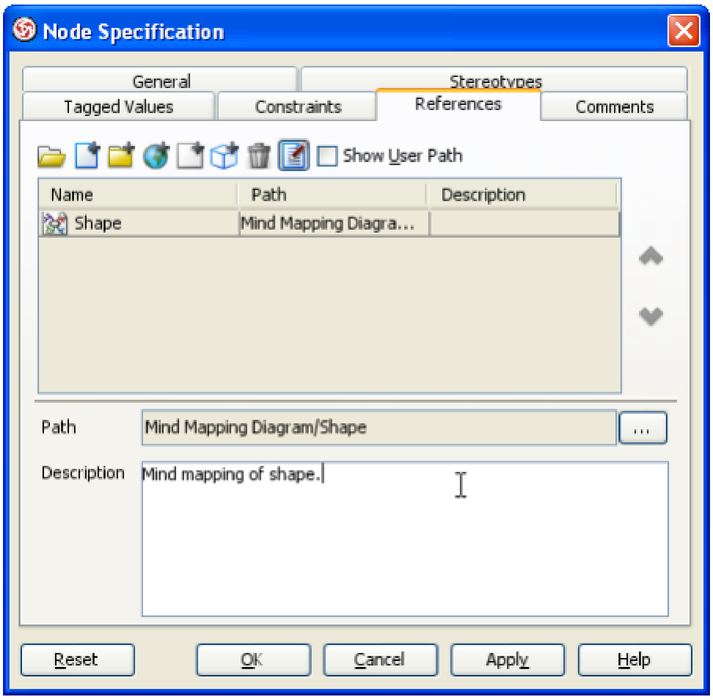


Figure 8-65 Selecting reference diagram

5. Input description of the diagram.



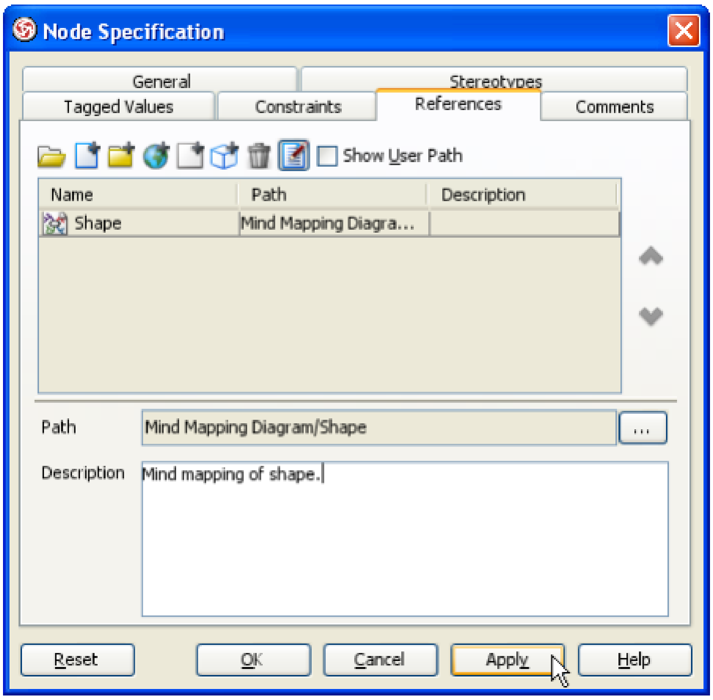
Name	Path	Description
Shape	Mind Mapping Diagra...	

Path: Mind Mapping Diagram/Shape

Description: Mind mapping of shape.

Figure 8-66 *Inputting diagram reference description*

6. Press Apply button to confirm the reference creation.



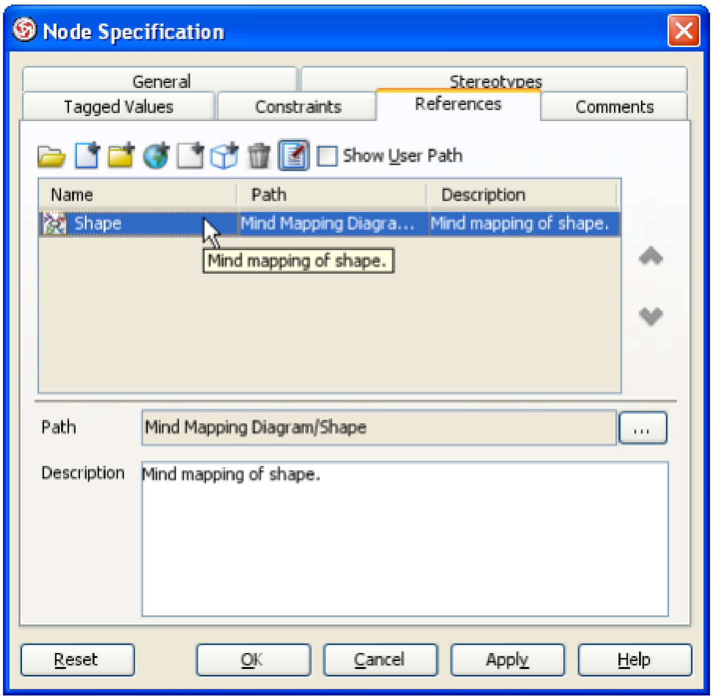
Name	Path	Description
Shape	Mind Mapping Diagra...	

Path: Mind Mapping Diagram/Shape

Description: Mind mapping of shape.

Figure 8-67 *Apply diagram reference*

7. Select the diagram reference to be opened.



Name	Path	Description
Shape	Mind Mapping Diagra...	Mind mapping of shape.

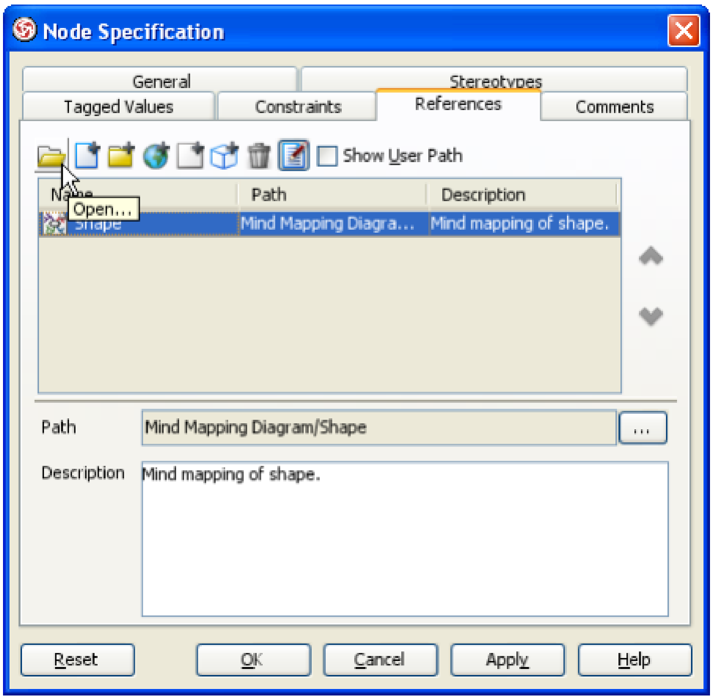
Mind mapping of shape.

Path: Mind Mapping Diagram/Shape

Description: Mind mapping of shape.

Figure 8-68 *Select diagram reference*

8. Press Open... button.



Name	Path	Description
Shape	Mind Mapping Diagra...	Mind mapping of shape.

Path: Mind Mapping Diagram/Shape

Description: Mind mapping of shape.

Figure 8-69 *Open diagram reference*

Reference to Shapes

1. Mouse over shape and click on References resource.

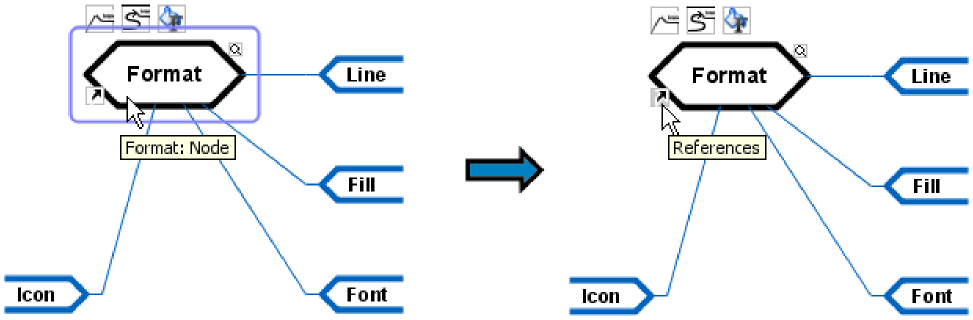


Figure 8-70 Mouse over references resource

2. Click on Add Shape... menu item.

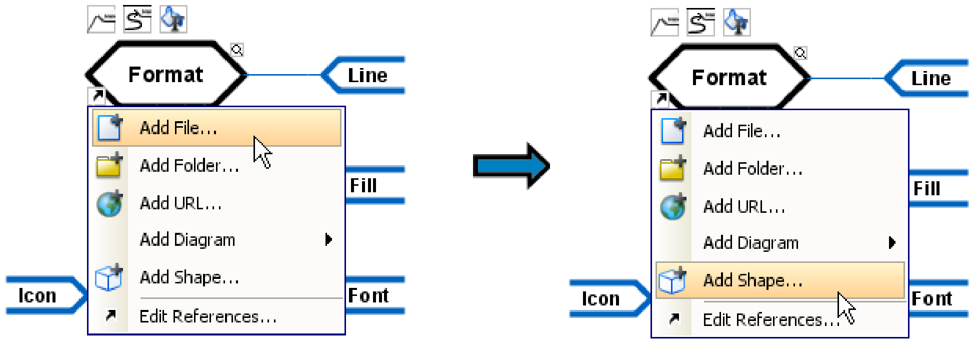


Figure 8-71 Add shape from resource

3. Select shape to be referenced.

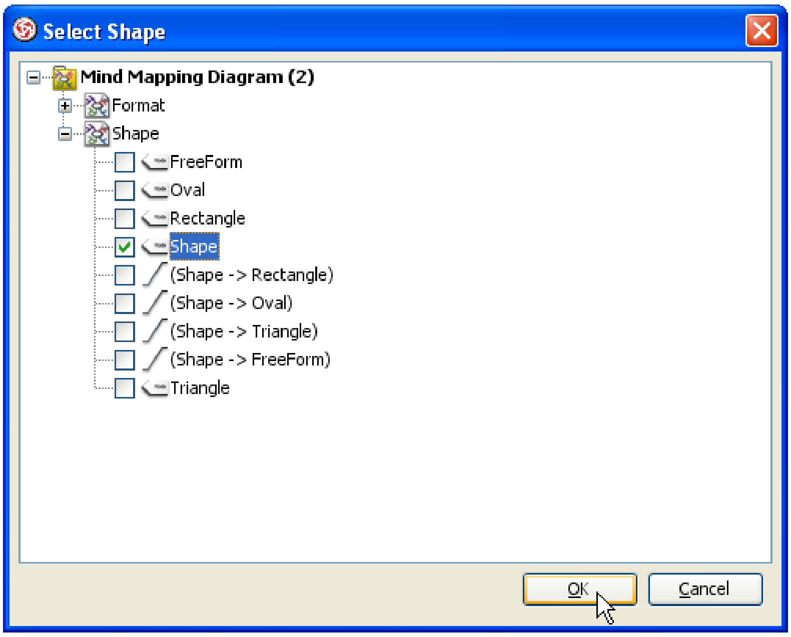
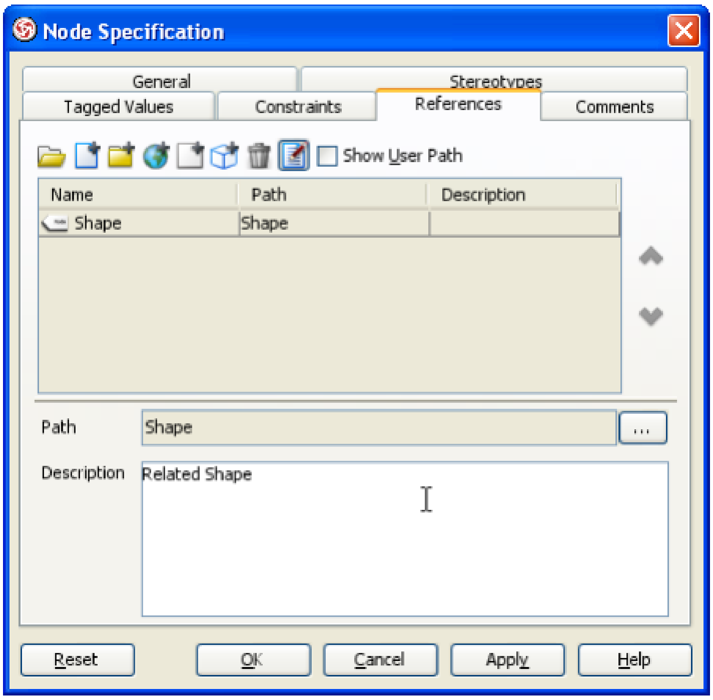


Figure 8-72 Select reference shape

4. Input description of the shape.



□ Show User Path

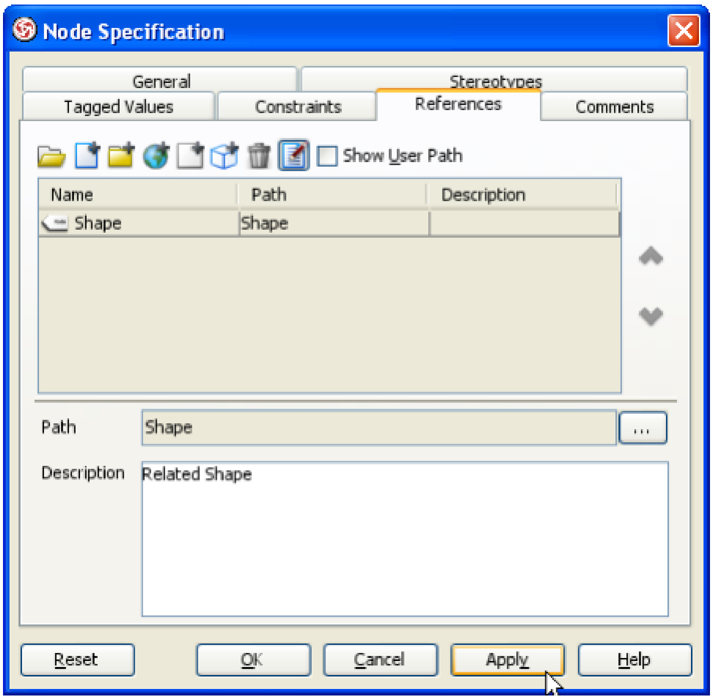
Name	Path	Description
Shape	Shape	

Path: Shape

Description: Related Shape

Figure 8-73 *Inputting shape reference description*

5. Press Apply button to confirm the reference creation.



□ Show User Path

Name	Path	Description
Shape	Shape	

Path: Shape

Description: Related Shape

Figure 8-74 *Apply folder reference*

6. Mouse over shape and click on References resource.

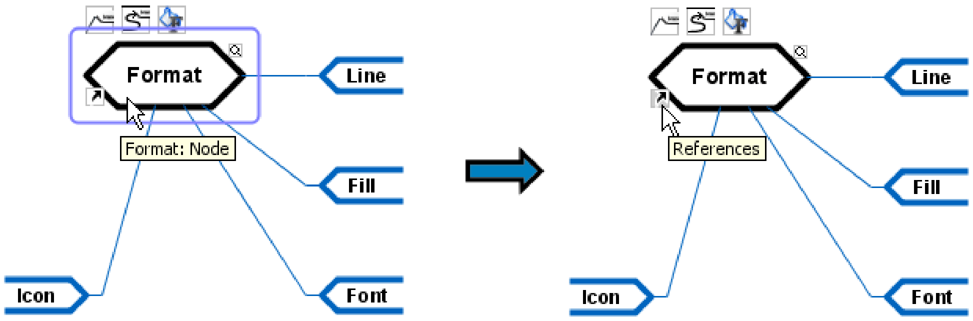


Figure 8-75 Mouse over references resource

7. Click on shape reference menu item.

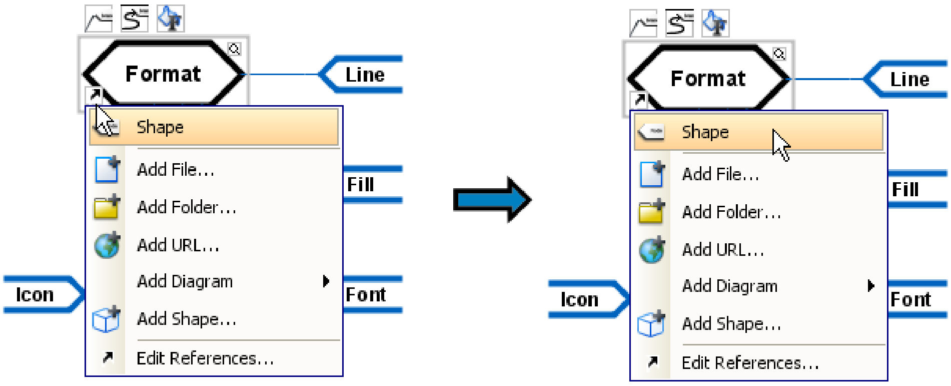


Figure 8-76 Open shape reference

Re-connecting Child Nodes

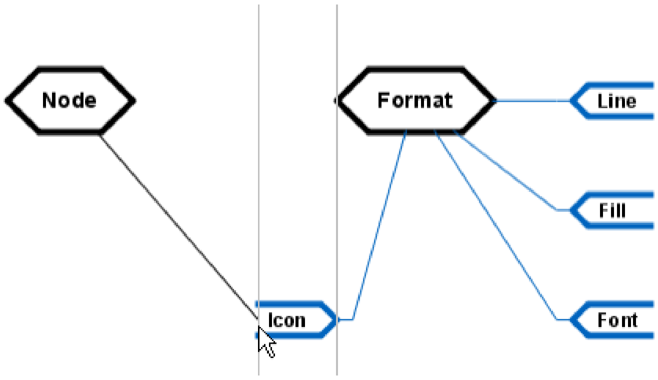


Figure 8-78 *Dragging*

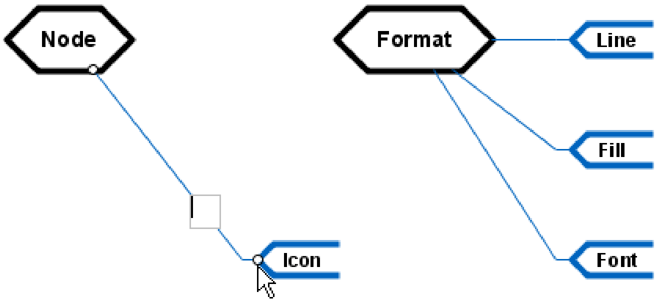


Figure 8-79 *Parent changed*

Defining Design Pattern

Design pattern feature helps you define and reuse design pattern in your project, across projects, or share with your team member. This feature is available in all diagram types, the following example show you how to define the Factory Pattern in **Class Diagram**.

Draw Factory Pattern

1. Create a class diagram.
2. Draw the factory pattern. When apply pattern, the name of classes and operations here are used as default name and they are expected to change. The names will also help you to identify the element in the pattern.

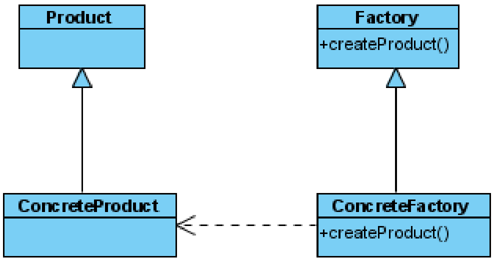


Figure 9-1 *Factory pattern*

Define Design Pattern

1. Select All by **Ctrl-A**, or select **Edit > Select All** from the main menu. Or you can select any shapes you want to include in the pattern.
2. Right click on any selected shapes, select **Define Design Pattern...** from the popup menu.

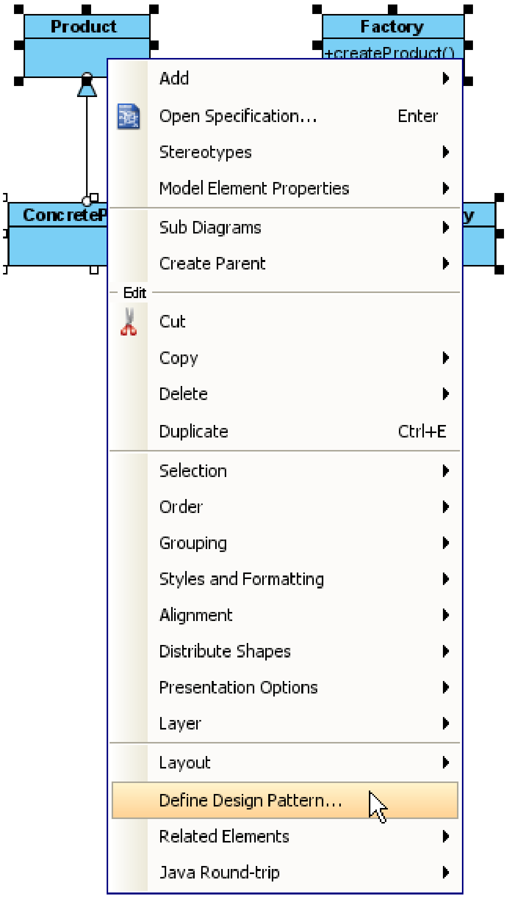
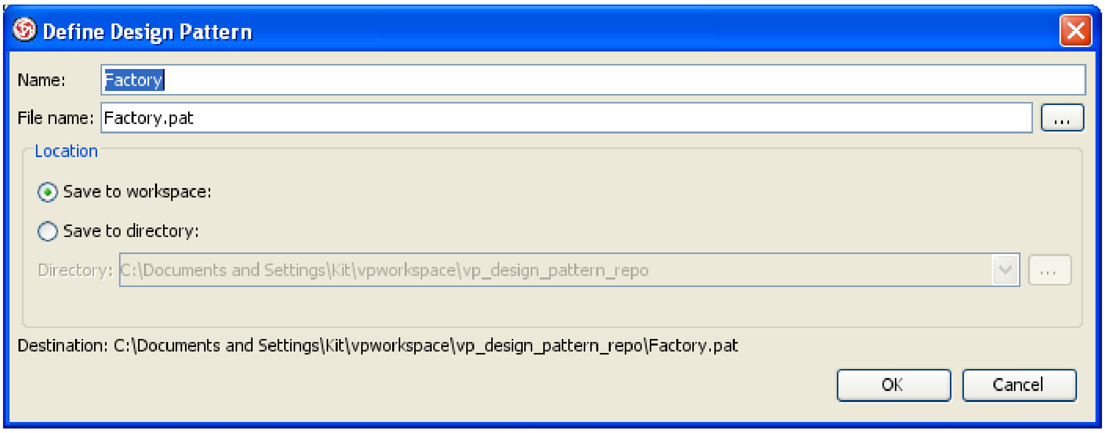


Figure 9-2 *Define Design Pattern*

3. In **Define Design Pattern** dialog, specify the name and file name for the pattern, the extension is **.pat**. By default, patterns will be saved to workspace and it can synchronize to teamwork server. Or you can save to other directory and share the **.pat** file with your team member for reuse. Click **OK** button to finish define design pattern.



Define Design Pattern

Name: Factory

File name: Factory.pat

Location

● Save to workspace:

○ Save to directory:

Directory: C:\Documents and Settings\Kit\vpworkspace\vp_design_pattern_repo

Destination: C:\Documents and Settings\Kit\vpworkspace\vp_design_pattern_repo\Factory.pat

Figure 9-3 *Design pattern name*

Applying Design Pattern

1. Create a class diagram
2. Right click on diagram, select **Utilities > Apply Design Pattern...** from the popup menu.

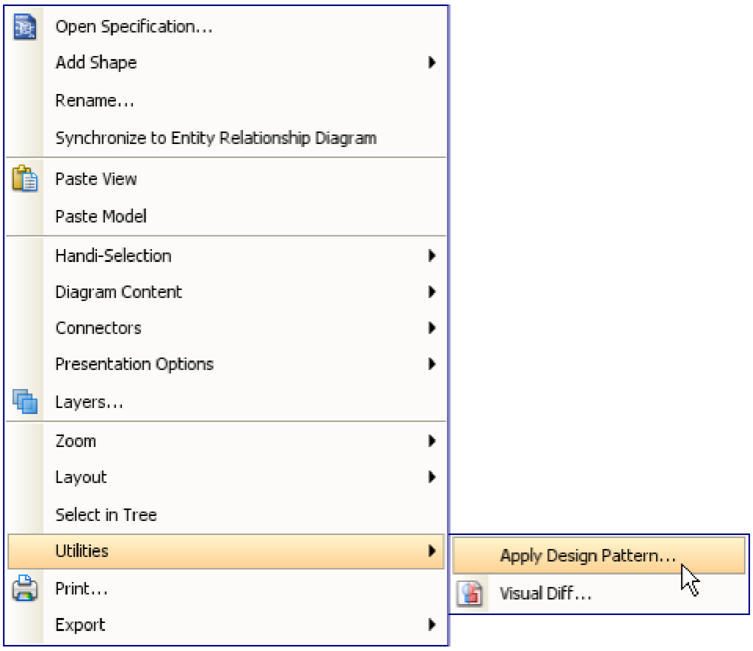


Figure 9-4 *Apply design pattern*

3. In **Design Pattern** dialog, you can see a list of defined patterns, select *Factory* from the list.

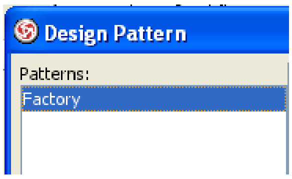
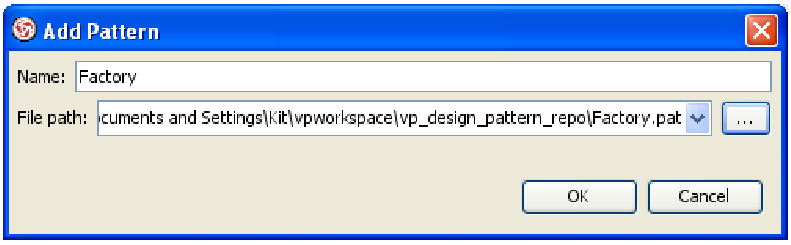


Figure 9-5 *Select pattern*

If you have a **.pat** file, click **Add** button to import into the list.



Name: Factory

File path: cuments and Settings\Kit\vpworkspace\vp_design_pattern_repo\Factory.pat

Figure 9-6 *Add pattern*

4. On the right hand side of the dialog, you can see the image of the pattern. Fill in the name of classes and operations on the bottom of dialog.

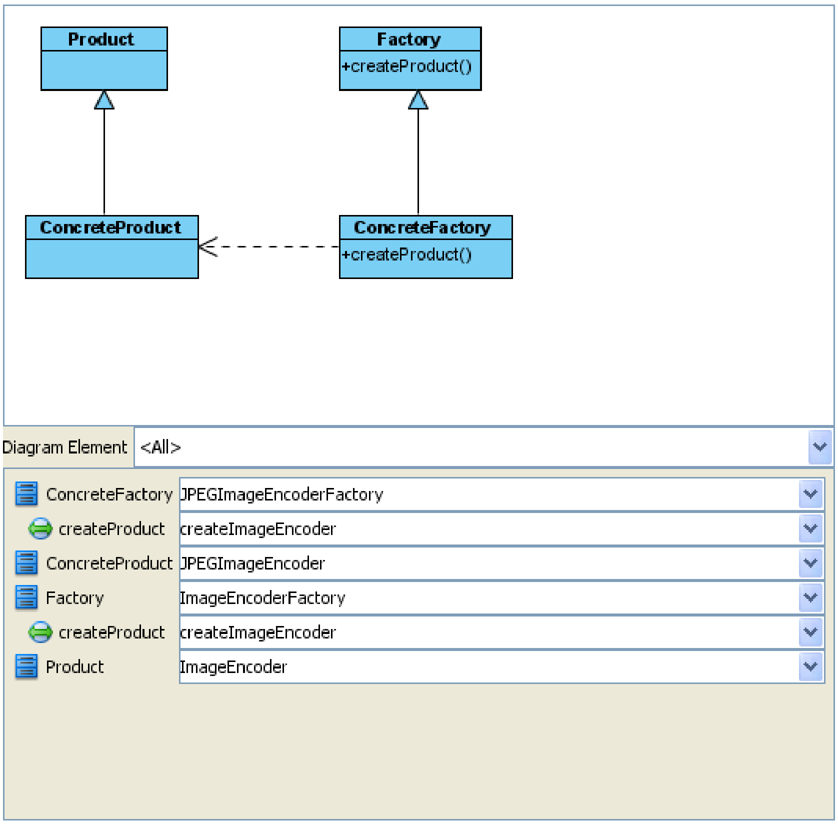


Diagram Element <All>

ConcreteFactory	JPEGImageEncoderFactory
createProduct	createImageEncoder
ConcreteProduct	JPEGImageEncoder
Factory	ImageEncoderFactory
createProduct	createImageEncoder
Product	ImageEncoder

Figure 9-7 *Fill in values*

You can also click on the shape or select a diagram element from the **Diagram Element** combo box to filter the list.

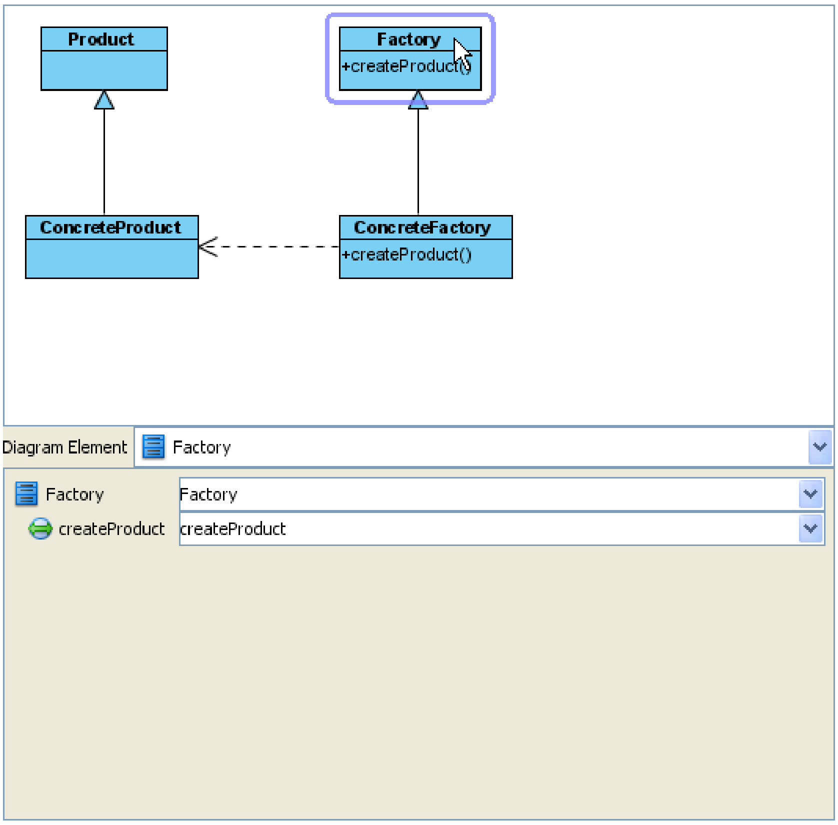


Diagram Element Factory

Factory	Factory
createProduct	createProduct

Figure 9-8 *Filter pattern element*

5. Finally, click **OK**. The pattern will be applied to the diagram.

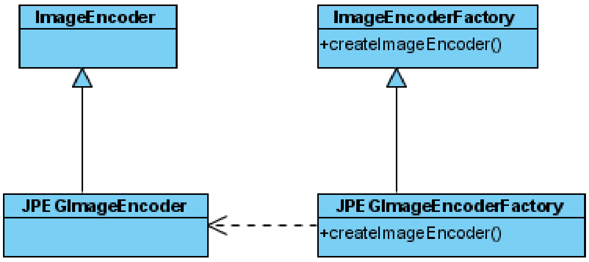


Figure 9-9 *Applied pattern*

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.

Synchronize local design pattern to server

1. Open **Teamwork Client** from **Toolbar**, or **Tools > Teamwork > Open Teamwork Client...** from the main menu.
2. Login to the teamwork server.
3. From the **Teamwork Client** dialog, select **Repository > Synchronize Design Pattern to Server** from the menu.

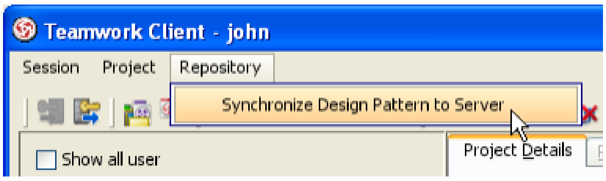
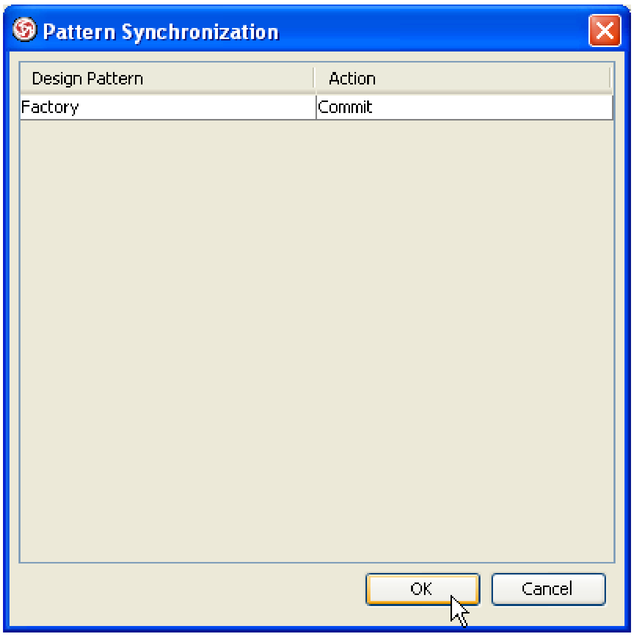


Figure 9-10 Synchronize Design Pattern to Server

4. You'll see **Pattern Synchronization** dialog, verify the design patterns and actions and click **OK** to continue. The pattern will be committed to teamwork server.



Design Pattern	Action
Factory	Commit

Figure 9-11 Commit pattern

Synchronize design pattern from server

1. Start VP-UML with another user, using a different workspace.
2. Create a class diagram.
3. Right click on diagram, select **Utilities > Apply Design Pattern...** from the popup menu.
4. From **Design Pattern** dialog, you'll find the patterns list is empty because design patterns didn't exists in workspace.

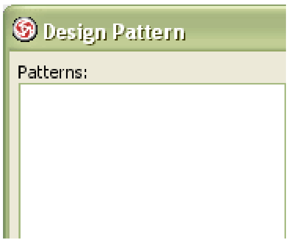


Figure 9-12 Apply design pattern

5. Open **Teamwork Client** and select **Repository > Synchronize Design Pattern to Server**.

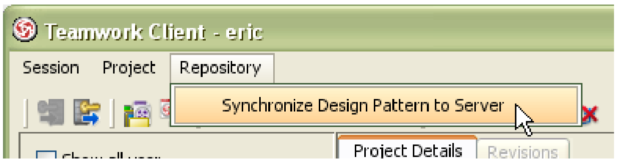
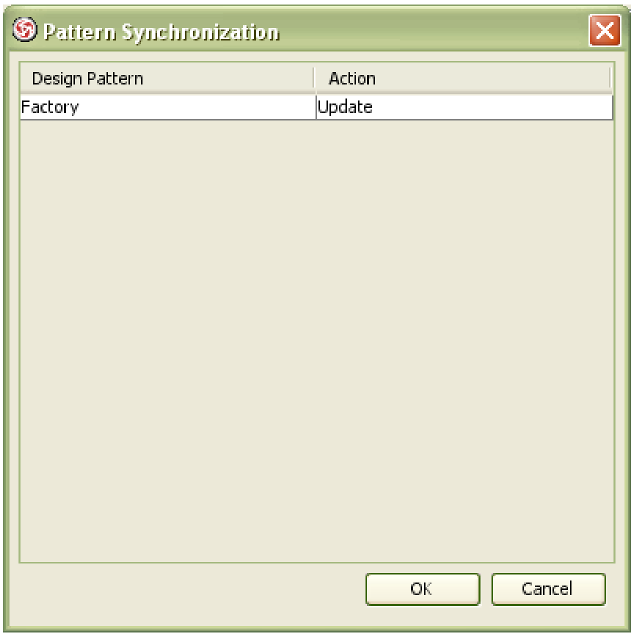


Figure 9-13 Synchronize Design Pattern to Server

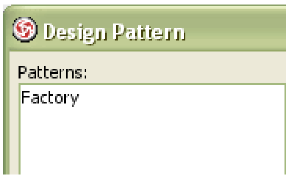
6. You'll see a **Pattern Synchronization** dialog showing that patterns are available for update from teamwork server. Click **OK** and the pattern will be updated from teamwork server to workspace.



Design Pattern	Action
Factory	Update

Figure 9-14 *Update pattern*

7. Apply design pattern again, the design pattern is available in the list. You can now select the pattern and apply to your project.



Patterns:
Factory

Figure 9-15 *Apply design pattern again*

What is Nickname?

One Model Element can have multiple names and documentations

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 **Business Process Diagram** in original name:

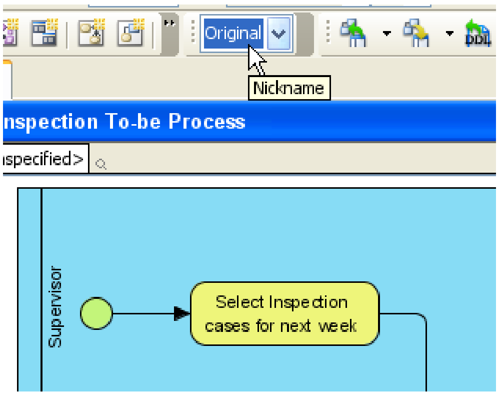


Figure 10-1 *Original name*

After you define a nickname and rename model element, switching nickname will refresh the diagram with the selected nickname.

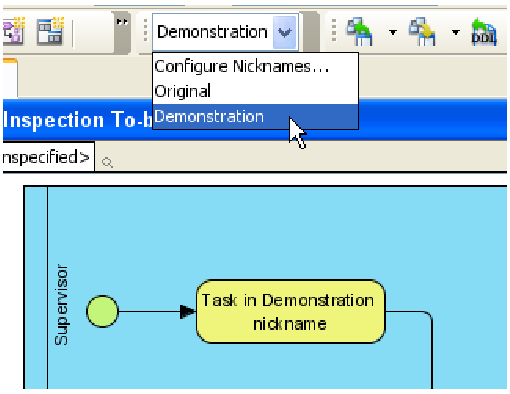


Figure 10-2 *Demonstration nickname*

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 choose their own language for modeling or view diagrams. The following example demonstrate the **Business Process Diagram** in English and French:

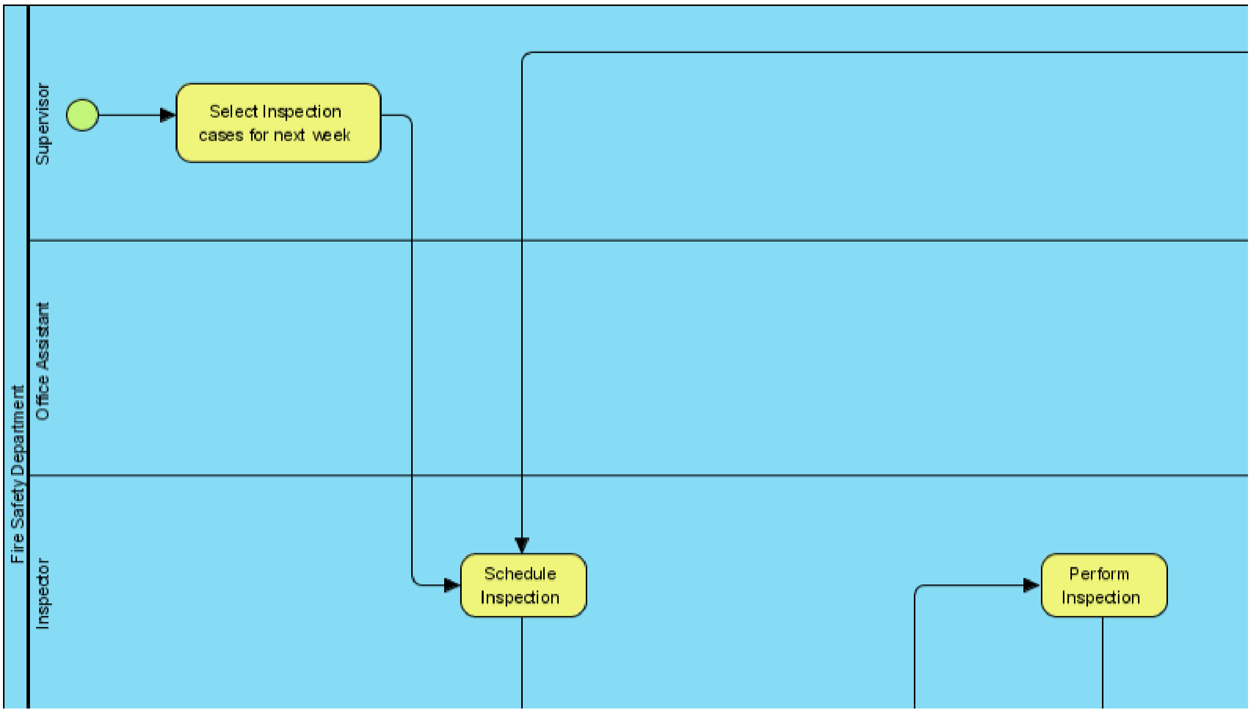


Figure 10-3 *BPD Original name*

You can create a *French* nickname and rename the model elements:

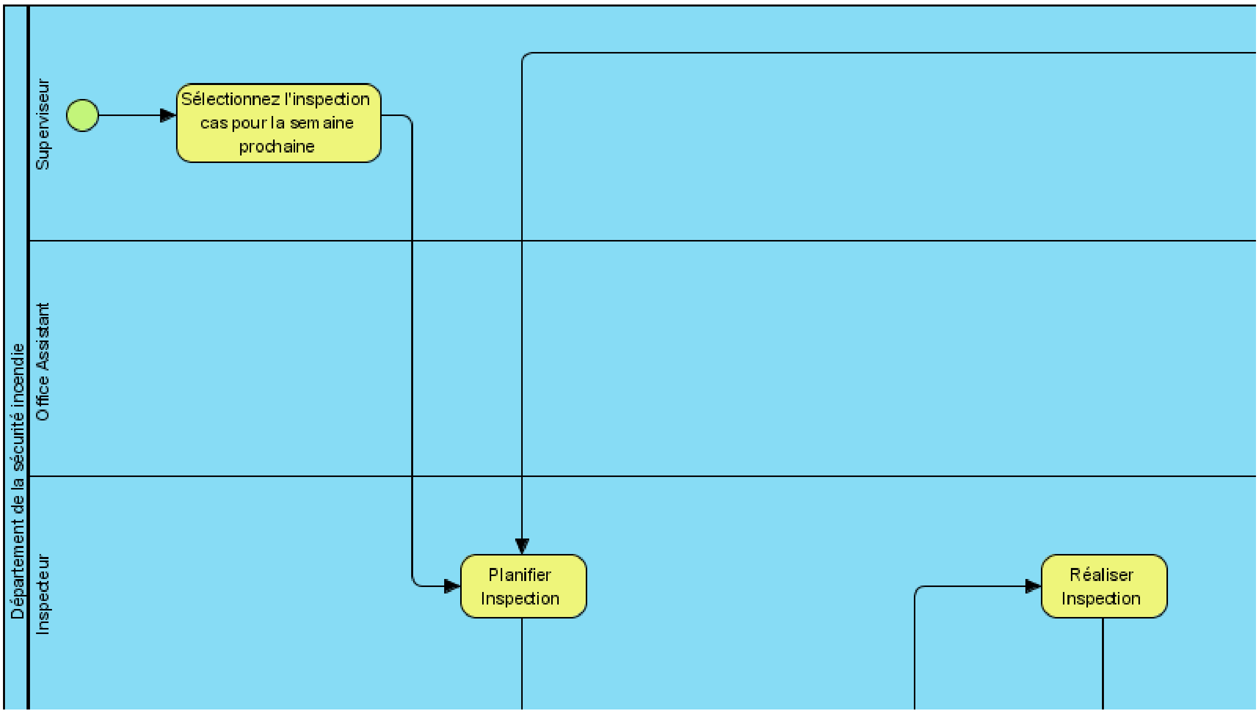
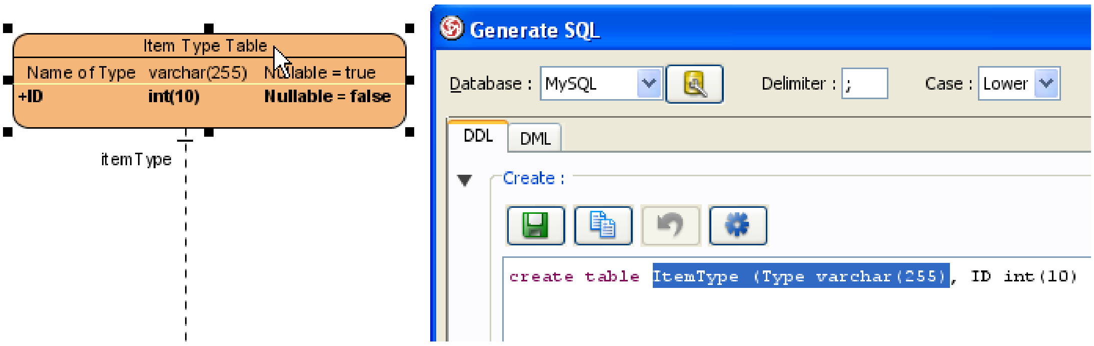

Département de la sécurité incendie
Superviseur
Office Assistant
Inspecteur
Sélectionnez l'inspection cas pour la sem aine prochaine
Planifier Inspection
Réaliser Inspection

Figure 10-4 *BPD French name*

Now you can switch between English (Original) and French anytime, or even creating more nicknames.

Increase readability of Entity Relationship Diagram

The name of **Entity** will be used to generate SQL, but Database Management System (DBMS) has many constraints on the name of Entity, Column, etc, and each DBMS are different . These constraints include the length of the name, reserved keywords, special characters, etc. They restricted the database designer to create an **Entity Relationship Diagram** (ERD) with meaningful names. With nickname, you can freely change any names to create a high readability ERD without affecting the generated SQL. The following diagram display **ERD** in nickname but generate SQL in original name:



Item Type Table
Name of Type varchar(255) Nullable = true
+ID int(10) Nullable = false
itemType
Generate SQL
DDL
DML
Create :

Database : MySQL

Delimiter : ;

Case : Lower

```
create table ItemType (Type varchar(255), ID int(10)
```

__Figure 10-5__ Generate SQL in nickname

Configure Nickname

Adding Nickname

1. First, ensure the **Nickname Toolbar** is showing. Right click on **Toolbar**, select **Nickname** if it is not selected already.

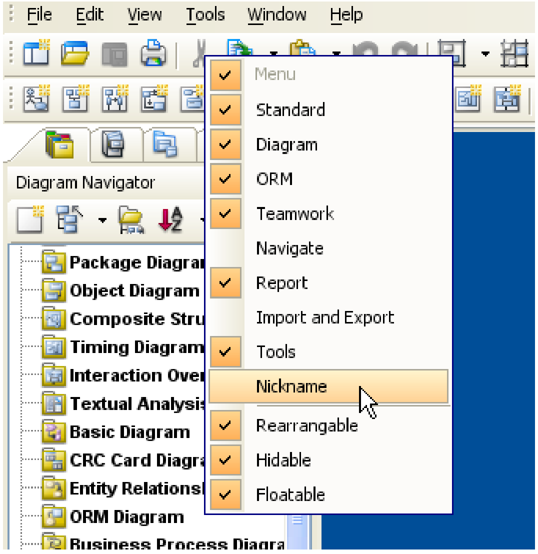


***Figure 10-6** Show Nickname Toolbar*

2. Then you'll see the **Nickname combo box** on the toolbar.

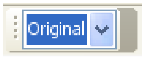


***Figure 10-7** Nickname Toolbar*

3. Click on **Nickname combo box**, select **Configure Nicknames...** from the popup.

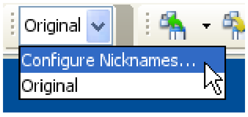
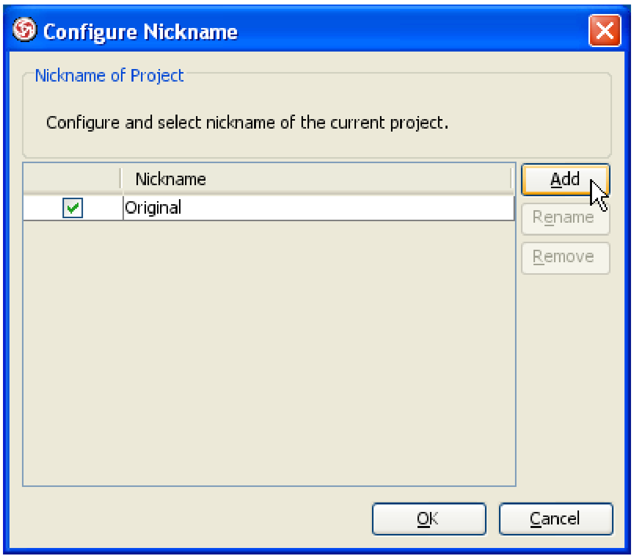


***Figure 10-8** Configure Nicknames*

4. In the **Configure Nickname** dialog, click **Add** button to create a new nickname.

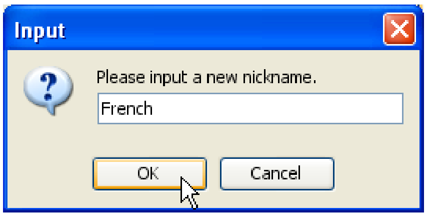


Configure and select nickname of the current project.

	Nickname
☑	Original

***Figure 10-9** Add nickname*

5. Input the name of nickname, and click **OK** button to confirm.

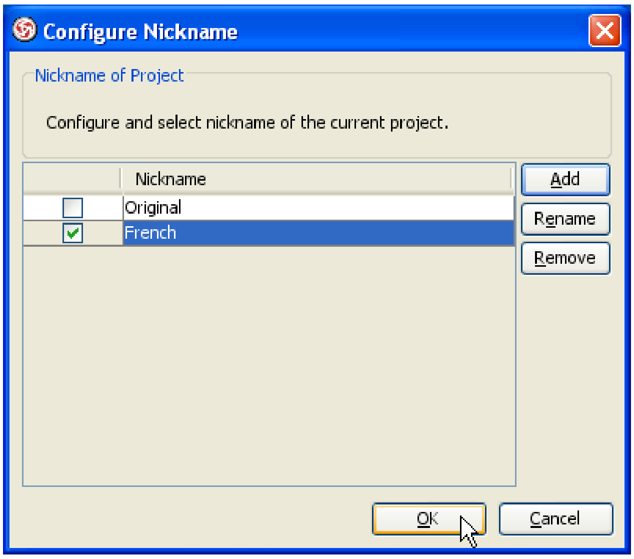


Please input a new nickname.

French

Figure 10-10 *Input name of nickname*

6. The check box of newly created nickname is automatically selected, click **OK** button to switch from **Original** nickname to the new nickname.



Configure and select nickname of the current project.

	Nickname
☐	Original
☑	French

Figure 10-11 *Select newly created nickname*

7. You'll see the **Nickname Toolbar** showing the newly created nickname. If you rename or set documentation for model element, it'll be applied to the selected nickname.

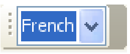
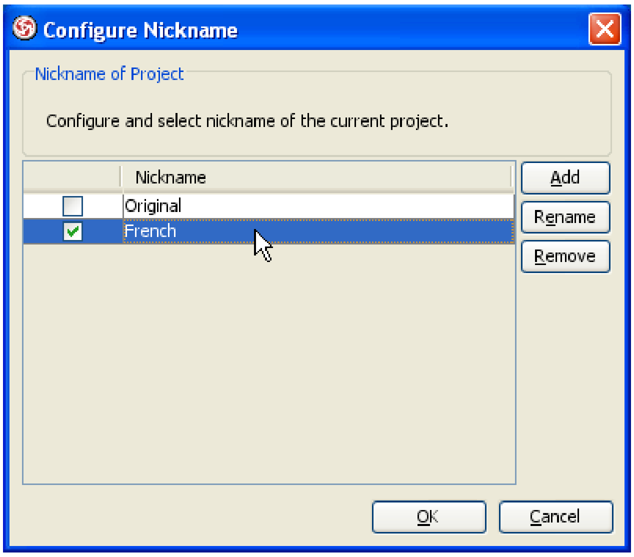


Figure 10-12 *Selected new nickname*

Rename Nickname

1. Open **Configure Nickname** dialog.
2. Select the row of nickname for rename.

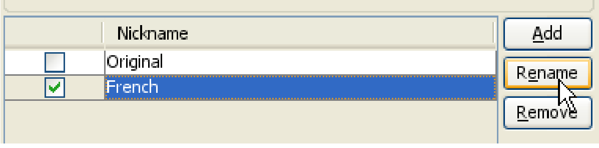


Configure and select nickname of the current project.

	Nickname
☐	Original
☑	French

Figure 10-13 *Configure Nickname*

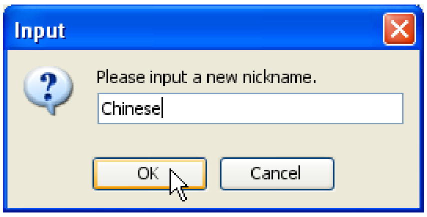
3. Click the **Rename** button, or press **F2** key.



	Nickname
☐	Original
☑	French

Figure 10-14 *Rename nickname*

4. Input the new name of nickname, it cannot duplicate with existing nicknames. Click **OK** button to confirm.



Input

Please input a new nickname.

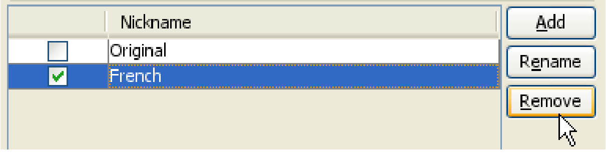
Chinese

Figure 10-15 *Input name of nickname*

5. Click **OK** button to close **Configure Nickname** dialog.

Remove Nickname

1. Open **Configure Nickname** dialog.
2. Select the row of nickname for remove.
3. Click the **Remove** button.



	Nickname
☐	Original
☑	French

Figure 10-15 *Remove nickname*

4. Click **OK** button to close **Configure Nickname** dialog.

Using Nickname

Input Name and Documentation for new Nickname

1. Select the nickname from **Nickname combo box**.

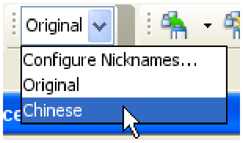


Figure 10-16 Select nickname

2. Select the shape you want to specify name for nickname.

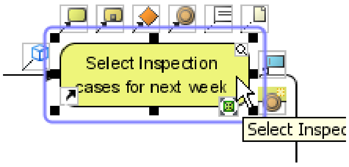
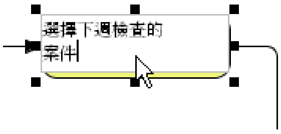


Figure 10-17 Select shape

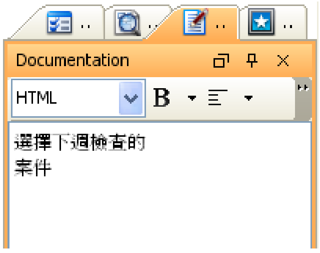
3. Double click on the selected shape or press **F2** key to rename.
4. Input the model element name for nickname.



選擇下週檢查的
案件

Figure 10-18 Input nickname

5. Select **Documentation Pane** and input documentation for the nickname.



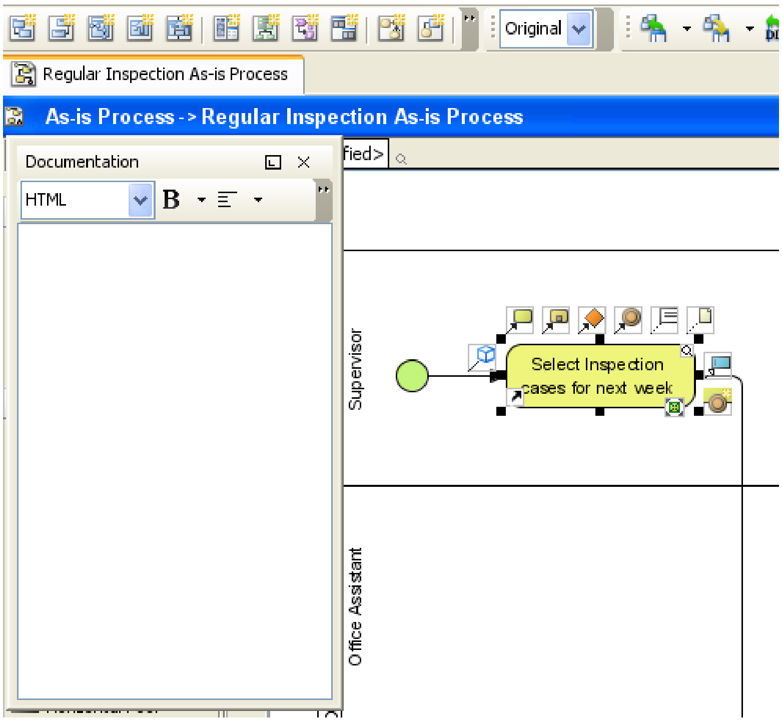
選擇下週檢查的
案件

Figure 10-19 Input documentation

6. Click on the diagram to apply the documentation.

Showing different Nickname

1. The diagram show the original name before switching nickname.



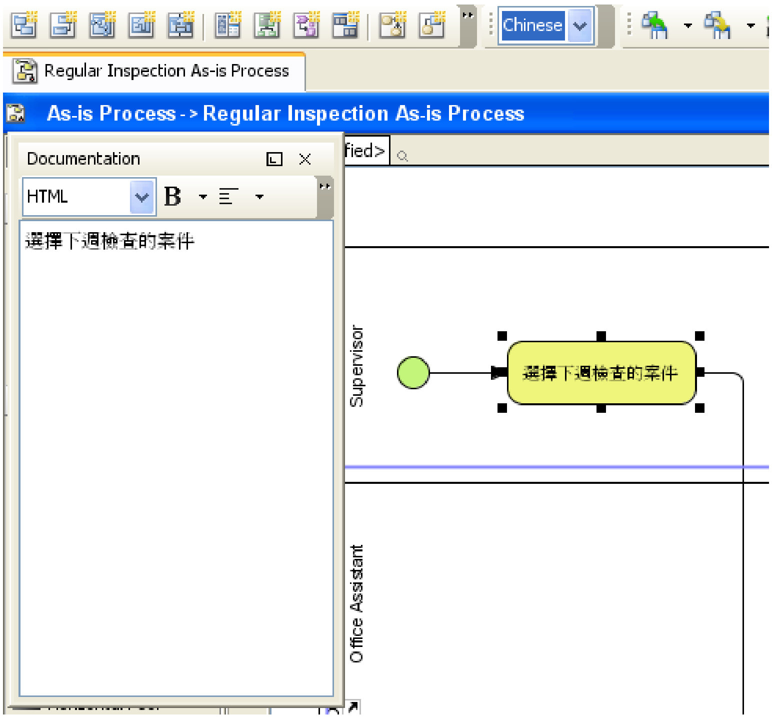
Original

Regular Inspection As-is Process

As-is Process -> Regular Inspection As-is Process

Figure 10-20 Diagram with original name

2. Select other nickname from **Nickname combo box**.
3. The diagram show the shapes with nickname, and the documentation also changed for nickname.



Chinese

Regular Inspection As-is Process

As-is Process -> Regular Inspection As-is Process

選擇下週檢查的案件

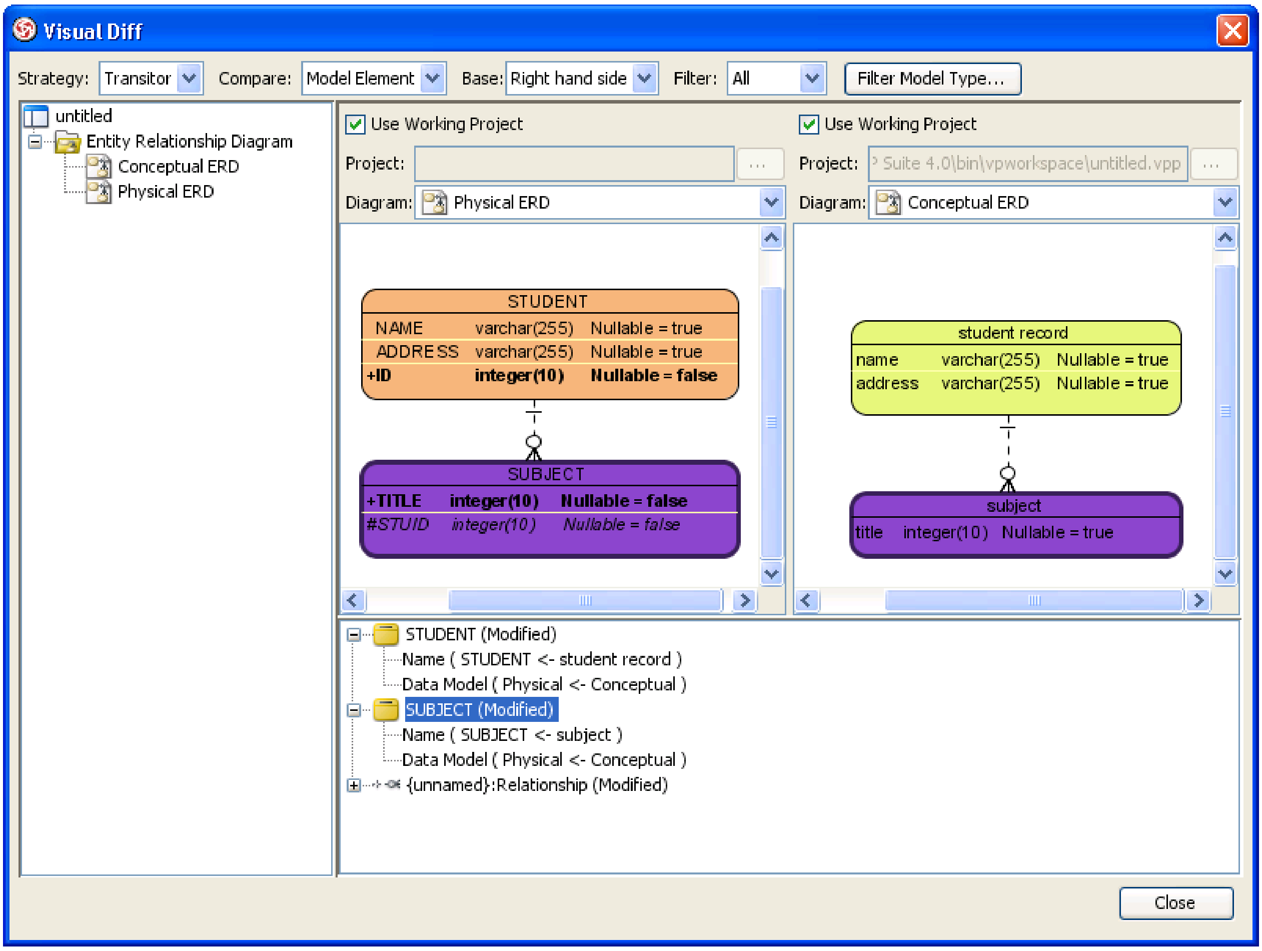
Figure 10-21 Diagram with nickname

What is Visual Diff?

There are times that we want to compare two diagrams. For example, to compare an ERD of conceptual model with an ERD of physical model, to compare a domain class diagram with a class diagram ready for implementation. VP-UML lets you compare differences between diagrams to trace the changes between them.

Diagram Comparison

By using the Visual Diff tool, users can compare two diagrams, know their differences by reading the result of comparison. Changes such as modification of properties (e.g. name) and addition/removal of containing models, etc can be found easily.



Visual Diff

Strategy: Transitor Compare: Model Element Base: Right hand side Filter: All

- untitled
  - Entity Relationship Diagram
    - Conceptual ERD
    - Physical ERD

☑ Use Working Project
Project:
Diagram: Physical ERD

☑ Use Working Project
Project: > Suite 4.0\bin\vpworkspace\untitled.vpp
Diagram: Conceptual ERD

STUDENT

NAME	varchar(255)	Nullable = true
ADDRESS	varchar(255)	Nullable = true
+ID	**integer(10)**	**Nullable = false**

SUBJECT

+TITLE	**integer(10)**	**Nullable = false**
#STUID	*integer(10)*	*Nullable = false*

student record

name	varchar(255)	Nullable = true
address	varchar(255)	Nullable = true

subject

title	integer(10)	Nullable = true

- STUDENT (Modified)
  - Name (STUDENT <- student record)
  - Data Model (Physical <- Conceptual)
- SUBJECT (Modified)
  - Name (SUBJECT <- subject)
  - Data Model (Physical <- Conceptual)
- {unnamed}:Relationship (Modified)

***Figure 11-1** The Visual Diff tool*

Various Comparison Strategies

A comparison strategy determines how two diagrams will be compared. Each strategy work best for a special purpose. You can select the appropriate strategy that suits your need. Here are the description of strategies.

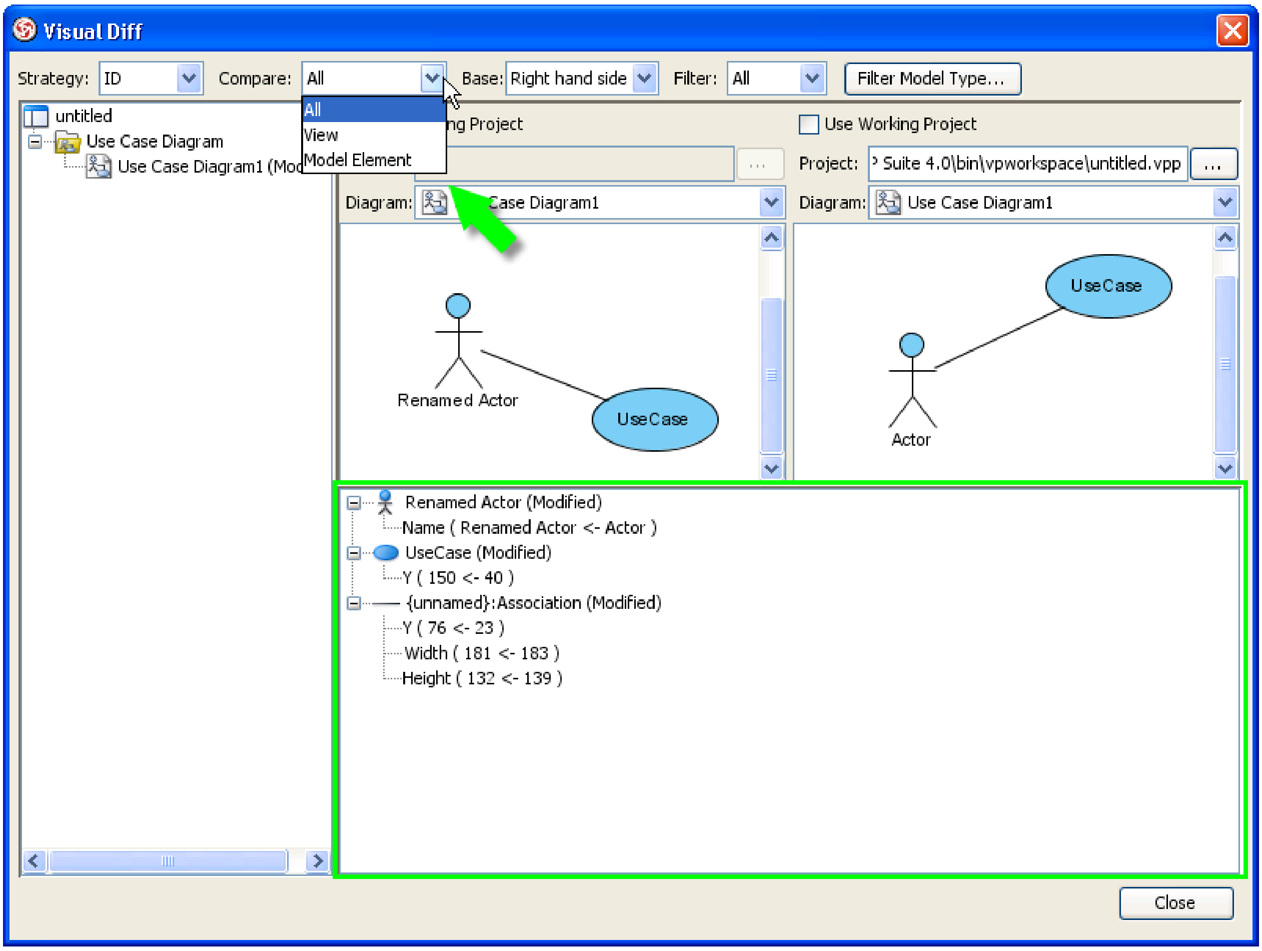
Strategy	Description
ID	Shapes will be matched base on their internal ID. Differences between shapes that have same ID will be displayed in the result pane. This strategy is useful when visualizing the changes of same shapes in two projects.
Name	Shapes will be matched base on their names. This strategy is useful when visualizing differences for external works. Typical examples are to compare databases and class models.
Transitor	Shapes will be matched base on their transition established by Model Transitor. This way of comparison is useful when visualizing differences between Models.

***Table 11-1** Description of comparison strategies*

Compare View Only, Model Element Only, or Both

Comparison can cover view, model element, or both of them, which affects the result to be displayed. By comparing view, differences in view settings such as the coordinate of shapes will be considered as changes. By comparing model element, differences in model element level like their names are considered as changes.

The following screenshot displays both view and model element differences.



Strategy: ID Compare: All Base: Right hand side Filter: All

☐ Use Working Project

Project: Suite 4.0\bin\vpworkspace\untitled.vpp

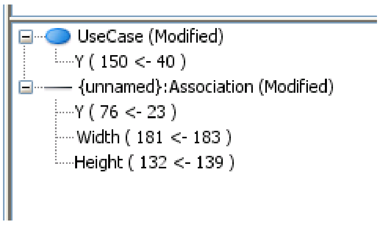
Diagram: Case Diagram1

Diagram: Use Case Diagram1

```
⊟ Renamed Actor (Modified)
    Name ( Renamed Actor <- Actor )
⊟ UseCase (Modified)
    Y ( 150 <- 40 )
⊟ {unnamed}:Association (Modified)
    Y ( 76 <- 23 )
    Width ( 181 <- 183 )
    Height ( 132 <- 139 )
```

Figure 11-2 *Compare both View and Model Element*

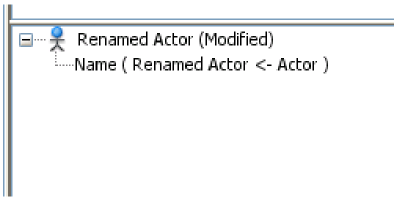
Below is the result when selected only to compare **View**. Differences in coordinates and shapes' width and height are therefore considered as changed.



```
⊟ UseCase (Modified)
    Y ( 150 <- 40 )
⊟ {unnamed}:Association (Modified)
    Y ( 76 <- 23 )
    Width ( 181 <- 183 )
    Height ( 132 <- 139 )
```

Figure 11-3 *Compare View*

Below is the result when selected only to compare **Model Element**. Rename of model elmeent is therefore considered as a change.

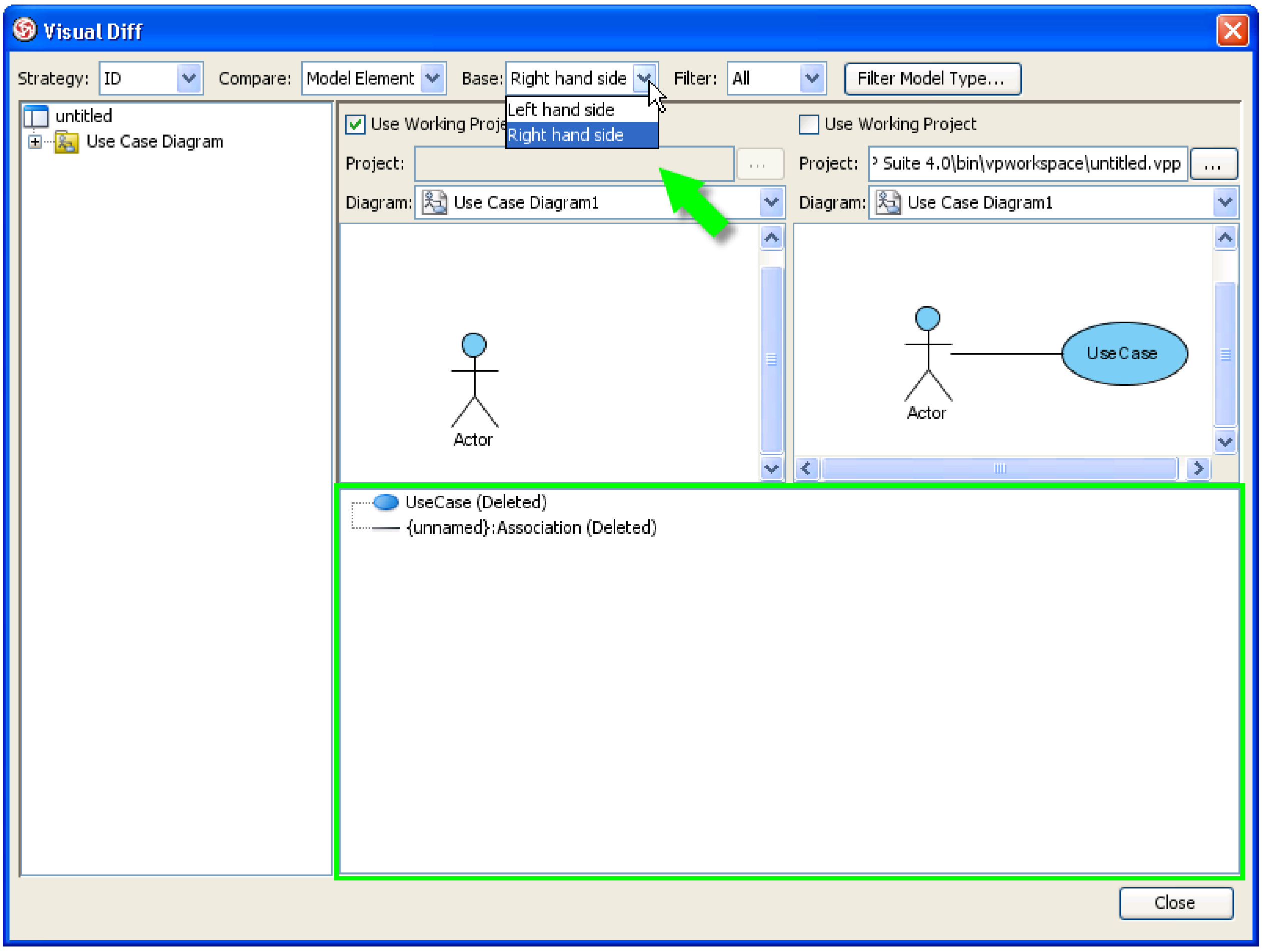


```
⊟ Renamed Actor (Modified)
    Name ( Renamed Actor <- Actor )
```

Figure 11-4 *Compare Model Element*

Compare Left to Right, or the Other Way Round

Comparisons are made between two diagrams, which are put at the left and right hand side in the **Visual Diff** dialog box respectively. By default, comparison is based on left hand side, which means that, if a shape does not exist on the left hand side but exist on the right hand side, the result pane will show that the shape is newly added. However, it is unknown to VP-UML whether the shape should be said as created or removed. Therefore, user can switch the base from right to left. By doing so, the absent of shape on the left hand side will result in a report of deleted shape instead of an addition of shape.



Visual Diff

Strategy: ID   Compare: Model Element   Base: Right hand side   Filter: All

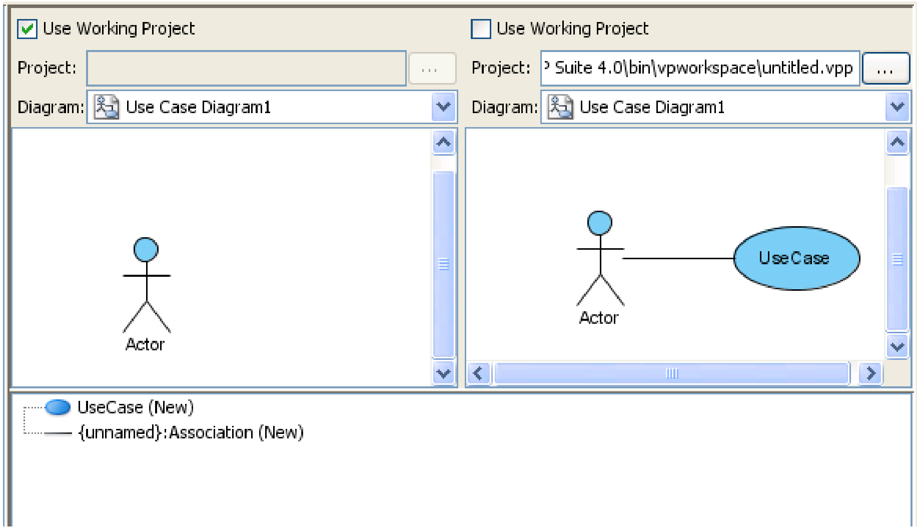
☑ Use Working Project
Project:
Diagram: Use Case Diagram1

□ Use Working Project
Project: › Suite 4.0\bin\vpworkspace\untitled.vpp
Diagram: Use Case Diagram1

UseCase (Deleted)
{unnamed}:Association (Deleted)

Figure 11-5 Comparing diagrams with right hand side as base

Below is the result when the base is switched from right hand side to left hand side. Deletion of model is said to be addition (see the text **New**).



☑ Use Working Project
Project:
Diagram: Use Case Diagram1

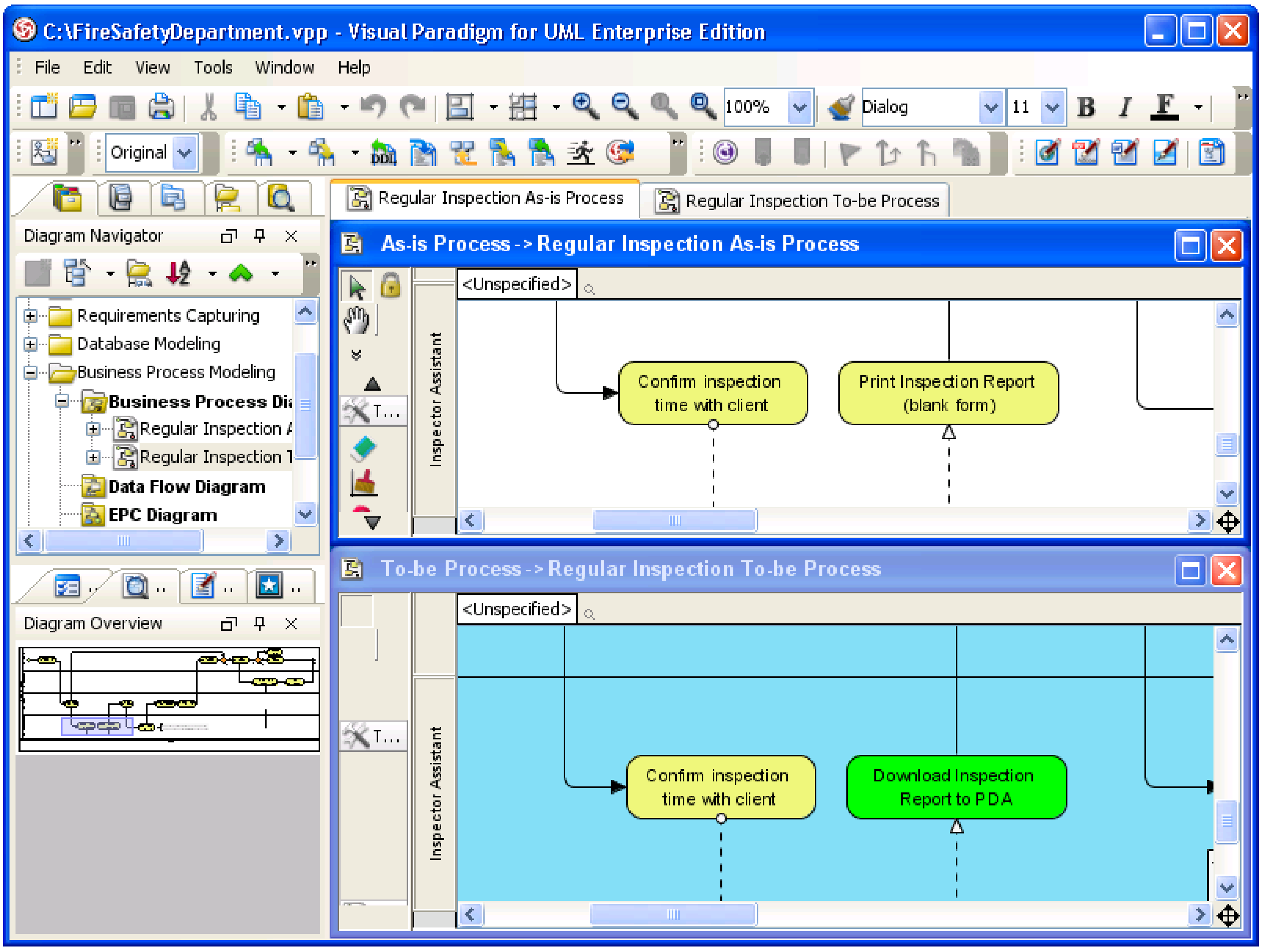
□ Use Working Project
Project: › Suite 4.0\bin\vpworkspace\untitled.vpp
Diagram: Use Case Diagram1

UseCase (New)
{unnamed}:Association (New)

Figure 11-6 Comparing diagrams with left hand side as base

Comparing As-is and To-be Business Process Diagram

Here are two Business Process Diagrams, one for modeling the As-is Process, and the other one is for modeling the To-be Process. We will make use of **Visual Diff** to find the differences between them.



C:\FireSafetyDepartment.vpp - Visual Paradigm for UML Enterprise Edition
File
Edit
View
Tools
Window
Help
100%
Dialog
11
Original
Regular Inspection As-is Process
Regular Inspection To-be Process
Diagram Navigator
Requirements Capturing
Database Modeling
Business Process Modeling
Business Process Dia
Regular Inspection A
Regular Inspection T
Data Flow Diagram
EPC Diagram
Diagram Overview
As-is Process -> Regular Inspection As-is Process
<Unspecified>
Inspector Assistant
Confirm inspection time with client
Print Inspection Report (blank form)
To-be Process -> Regular Inspection To-be Process
<Unspecified>
Inspector Assistant
Confirm inspection time with client
Download Inspection Report to PDA

Figure 11-7 *BPDs for As-Is Process and To-Be Process*

1. From the diagram of As-Is Process, select **Tools > Visual Diff...** from the main menu.

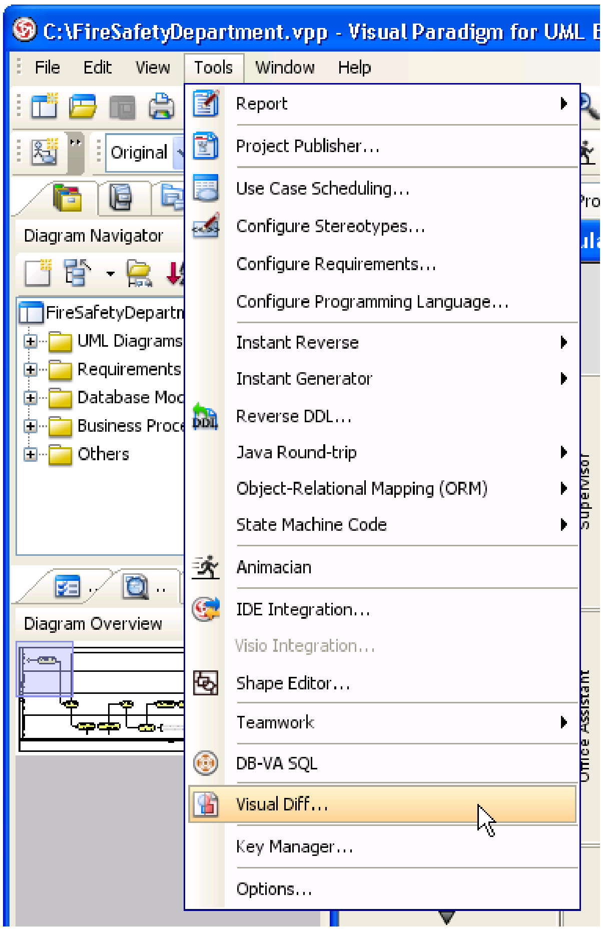
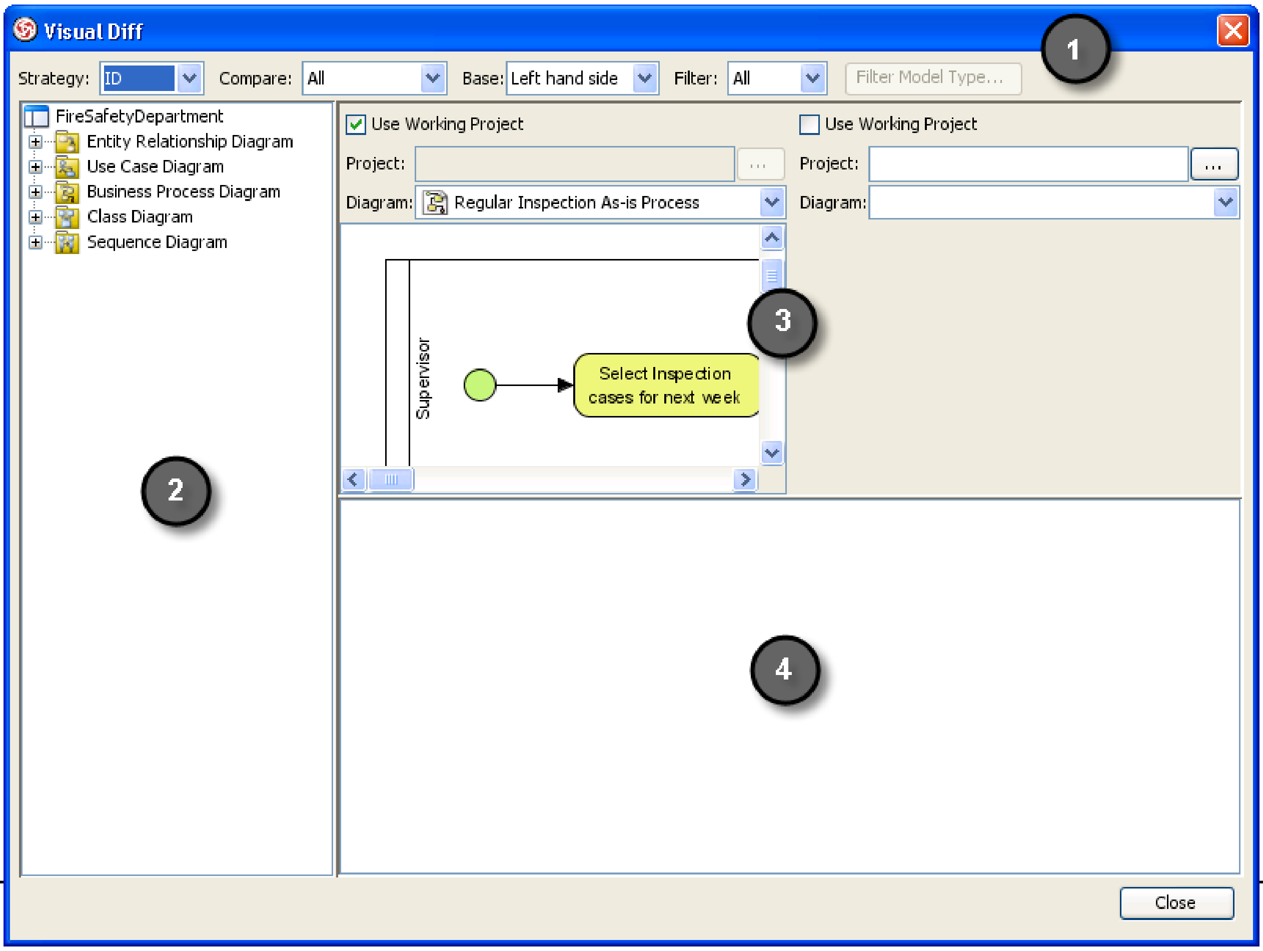


Figure 11-8 *Launching Visual Diff through the main menu*

NOTE: Besides starting through the main menu, you can start Visual Diff through the ways below:

- Right-click on diagram background and selecting **Utilities > Visual Diff...** from the popup menu.
- Click on the **Visual Diff** button on the **Tools** toolbar.

This starts **Visual Diff**.



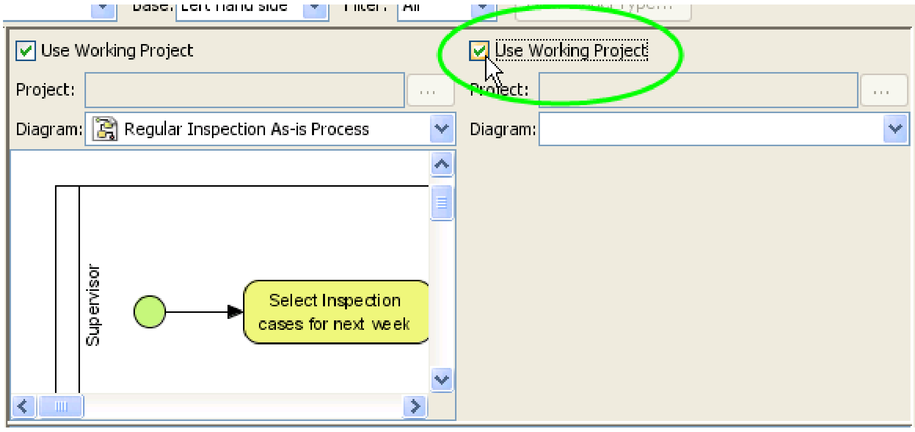
Strategy: ID Compare: All Base: Left hand side Filter: All

☑ Use Working Project
Project:
Diagram: Regular Inspection As-is Process

☐ Use Working Project
Project:
Diagram:

Figure 11-9 *The Visual Diff dialog box*

2. The left hand side is showing the currently opening diagram. Let's keep it unchanged. Now, select **Use Working Project** for the right hand side so that we can select a diagram in the same project to compare with.



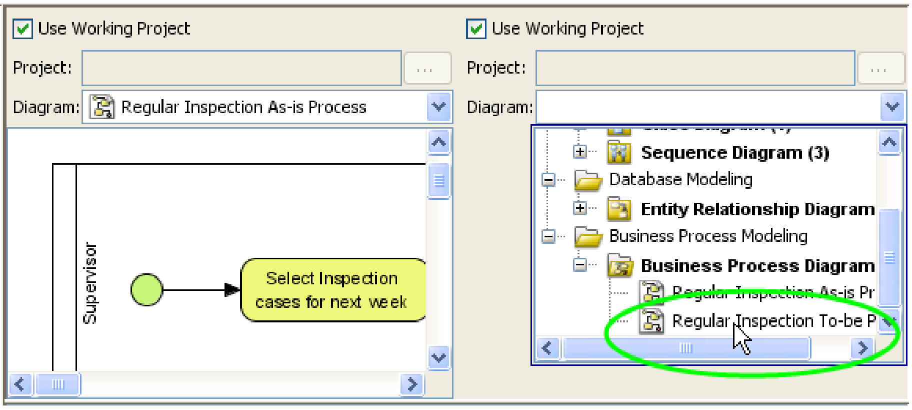
☑ Use Working Project
Project:
Diagram: Regular Inspection As-is Process

☑ Use Working Project
Project:
Diagram:

Figure 11-10 *Select Use Working Project*

NOTE: To compare with diagram in another project, uncheck **Use Working Project** and select the project file in the Project field.

3. Select the *To-be Process* to compare with.



☑ Use Working Project

Project:

Diagram: Regular Inspection As-is Process

☑ Use Working Project

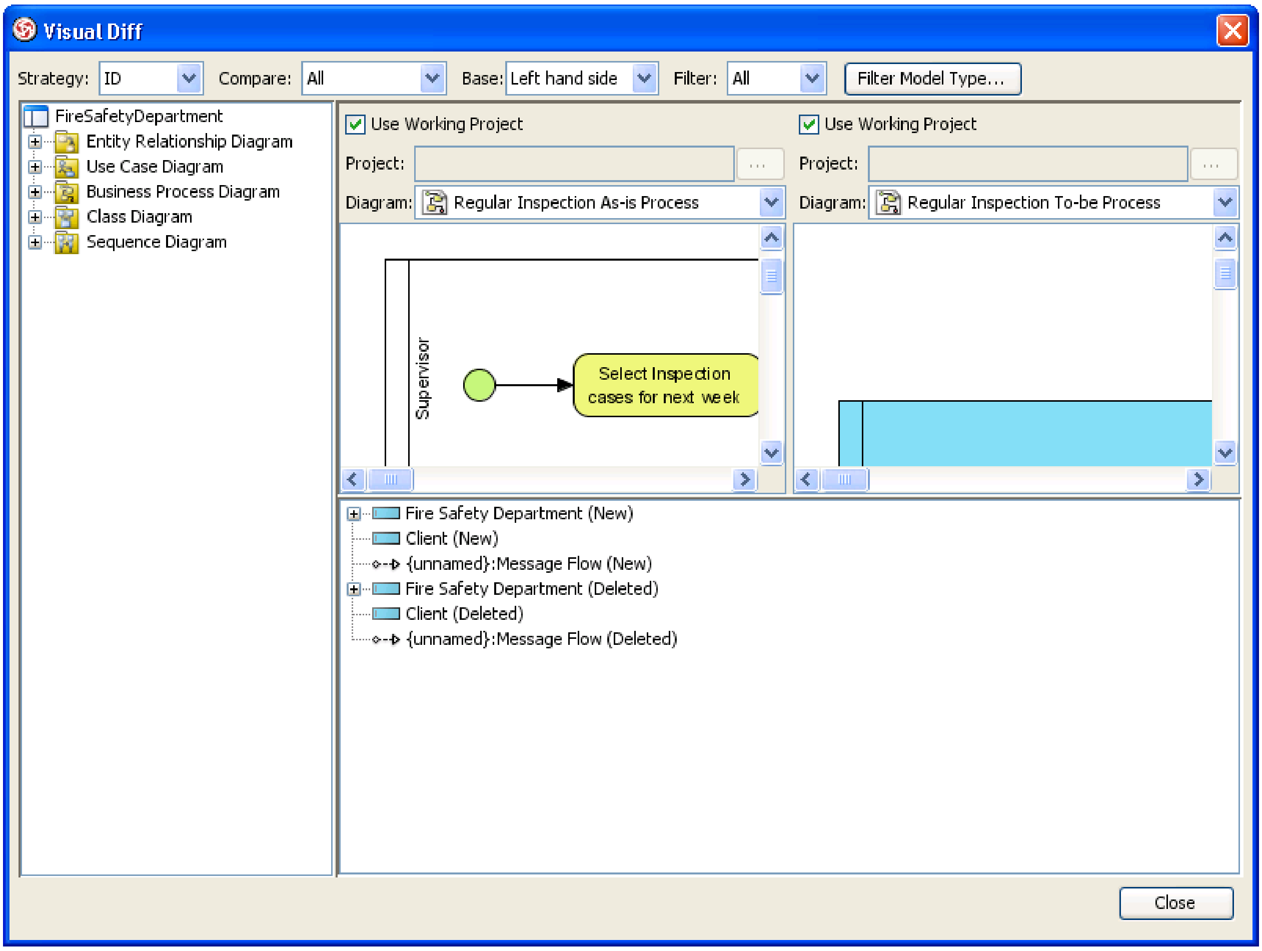
Project:

Diagram:

- Sequence Diagram (3)
- Database Modeling
  - Entity Relationship Diagram
- Business Process Modeling
  - Business Process Diagram
    - Regular Inspection As-is Pr
    - Regular Inspection To-be P

Figure 11-11 *Select diagram to compare*

Visual Diff is updated to show the two diagrams side by side. The result pane at the bottom is updated, too. However, we need to configure **Visual Diff** in order to compare in the way we want.



Visual Diff

Strategy: ID Compare: All Base: Left hand side Filter: All Filter Model Type...

- FireSafetyDepartment
  - Entity Relationship Diagram
  - Use Case Diagram
  - Business Process Diagram
  - Class Diagram
  - Sequence Diagram

☑ Use Working Project

Project:

Diagram: Regular Inspection As-is Process

☑ Use Working Project

Project:

Diagram: Regular Inspection To-be Process

- Fire Safety Department (New)
- Client (New)
- {unnamed}:Message Flow (New)
- Fire Safety Department (Deleted)
- Client (Deleted)
- {unnamed}:Message Flow (Deleted)

Figure 11-12 *Diagrams selected*

4. Select *Name* as **Strategy**.

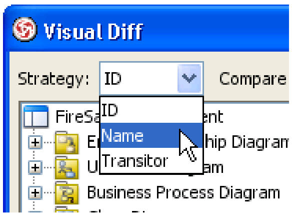


Figure 11-13 *Select a comparison strategy*

Below is a description of available **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 useful when visualizing 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. Typical examples are to compare databases and class models.
Transitor	Shapes will be matched base on their transition established by Model Transitor. This way of comparison is useful when visualizing differences between Models.

Table 11-3 *Description of comparison strategies*

5. Select *Model Element* as the scope of comparison.

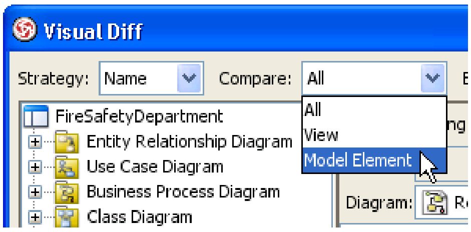


Figure 11-14 *Select the item to compare*

Below is a description of available **Compare** options.

Option	Description
All	Both view and model element are displayed.
View	Differences such as coordinates, width, height and color of shapes are displayed.
Model Element	Differences such as model name is displayed.

Table 11-4 *Description of compare options*

6. Select *Left hand side* as **Base**.

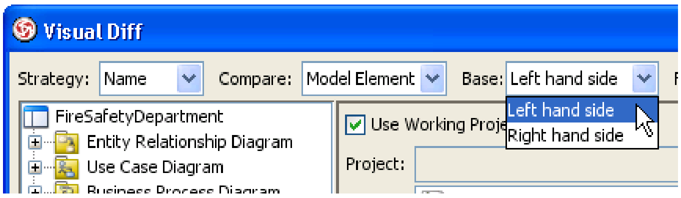


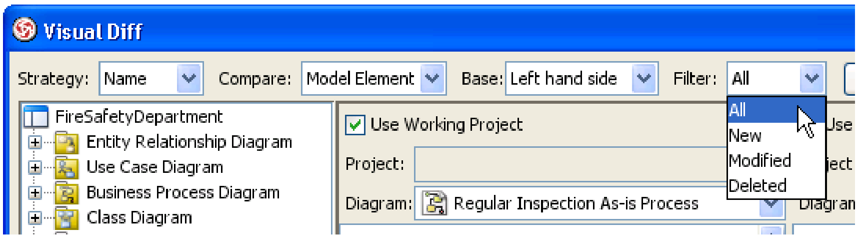
Figure 11-15 *Select a base*

Below is a description of available **Bases**.

Base	Description
Left hand side	The default base selection which cause comparison to be made base on the diagram on the left 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	Cause comparison to be made base 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.

Table 11-5 *Description of bases*

7. Select *All* as the **Filter**.



Visual Diff

Strategy: Name Compare: Model Element Base: Left hand side Filter: All

- FireSafetyDepartment
  - Entity Relationship Diagram
  - Use Case Diagram
  - Business Process Diagram
  - Class Diagram

☑ Use Working Project

Project:

Diagram: Regular Inspection As-is Process

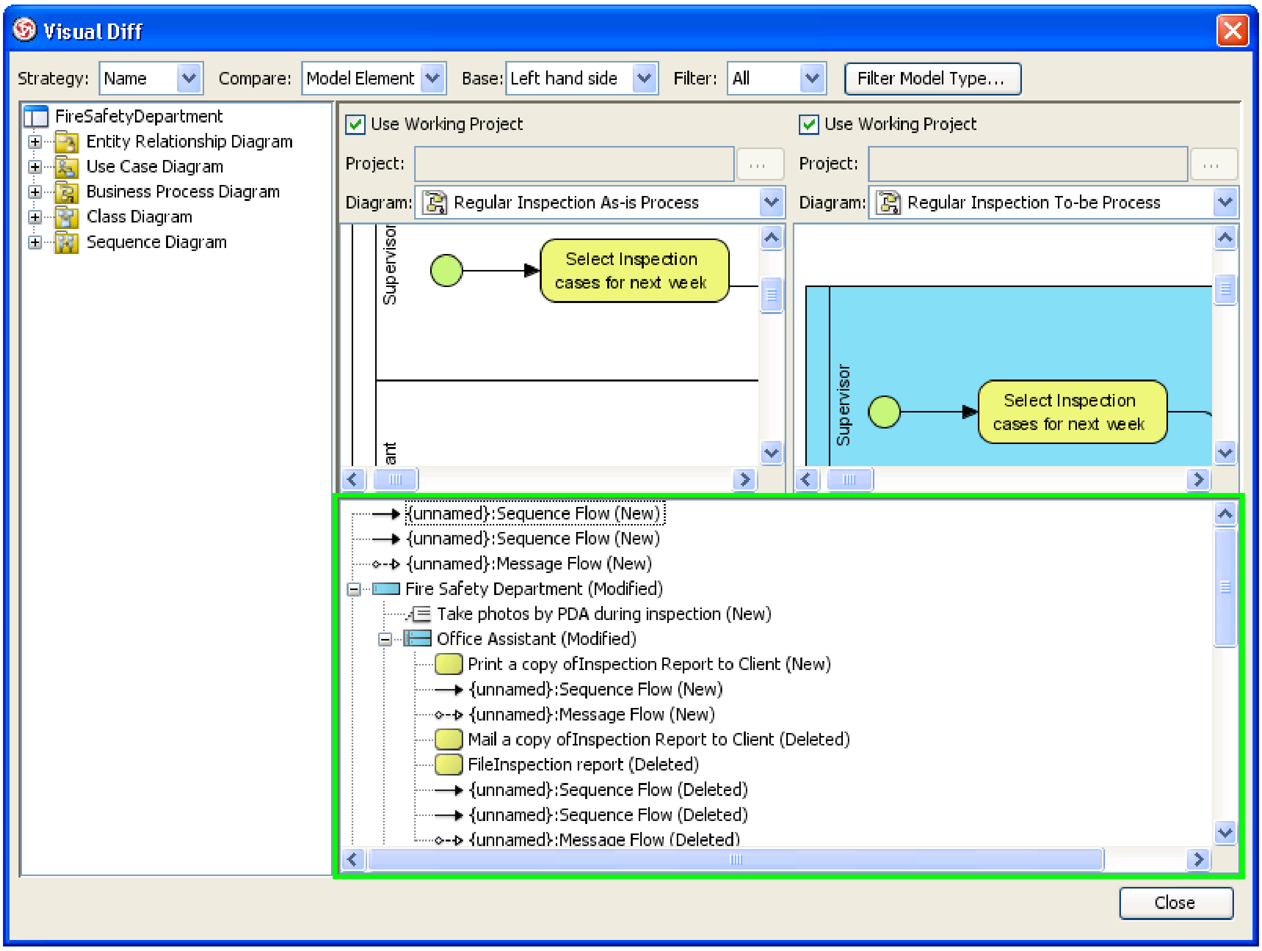
Figure 11-16 *Select a filter*

Below is a description of available **Filters**.

Filter	Description
All	Display all kinds of differences which includes the addition, modification and removal of shapes.
New	Display only results about the addition of shape, and hide the rest.
Modified	Display only results about the modification of shapes, and hide the rest.
Deleted	Display only results about the removal of shapes, and hide the rest.

Table 11-6 *Description of filters*

8. Once everything is set, we can see the differences of the two diagrams from the result pane.



Visual Diff

Strategy: Name Compare: Model Element Base: Left hand side Filter: All

- FireSafetyDepartment
  - Entity Relationship Diagram
  - Use Case Diagram
  - Business Process Diagram
  - Class Diagram
  - Sequence Diagram

☑ Use Working Project

Project:

Diagram: Regular Inspection As-is Process

☑ Use Working Project

Project:

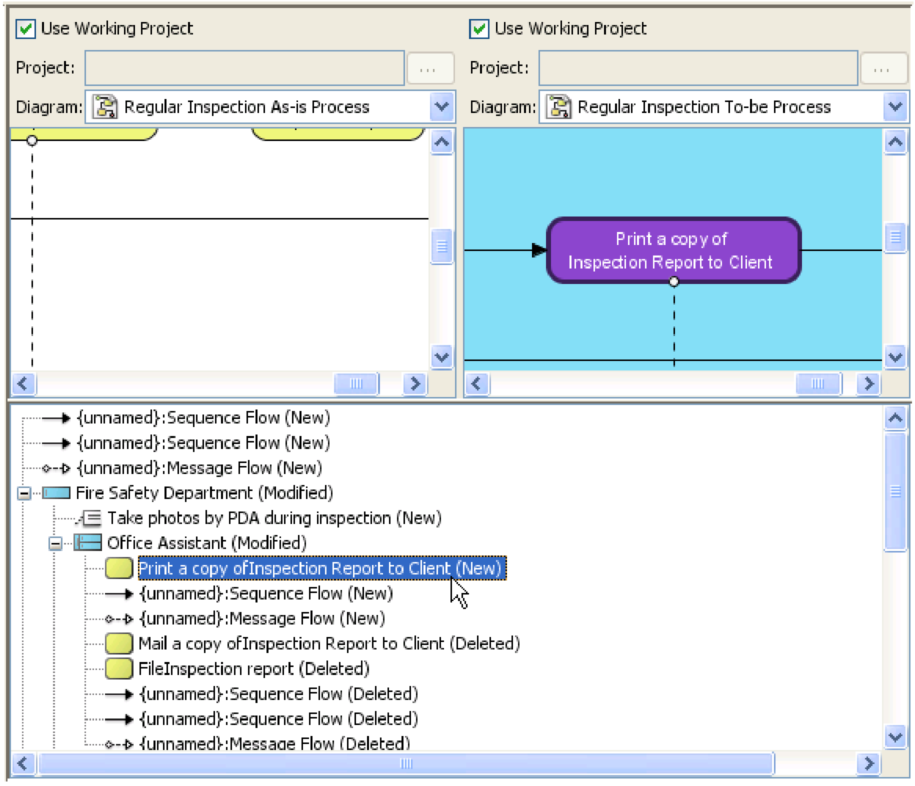
Diagram: Regular Inspection To-be Process

Supervisor
Select Inspection cases for next week
ant
Supervisor
Select Inspection cases for next week

- {unnamed}:Sequence Flow (New)
- {unnamed}:Sequence Flow (New)
- {unnamed}:Message Flow (New)
- Fire Safety Department (Modified)
  - Take photos by PDA during inspection (New)
  - Office Assistant (Modified)
    - Print a copy ofInspection Report to Client (New)
    - {unnamed}:Sequence Flow (New)
    - {unnamed}:Message Flow (New)
    - Mail a copy ofInspection Report to Client (Deleted)
    - FileInspection report (Deleted)
    - {unnamed}:Sequence Flow (Deleted)
    - {unnamed}:Sequence Flow (Deleted)
    - {unnamed}:Message Flow (Deleted)

Figure 11-17 *Result of comparison is obtained*

9. It is possible to click on a node to select the shape in diagrams. Selected shape is painted in dark purple.



☑ Use Working Project

Project:

Diagram: Regular Inspection As-is Process

☑ Use Working Project

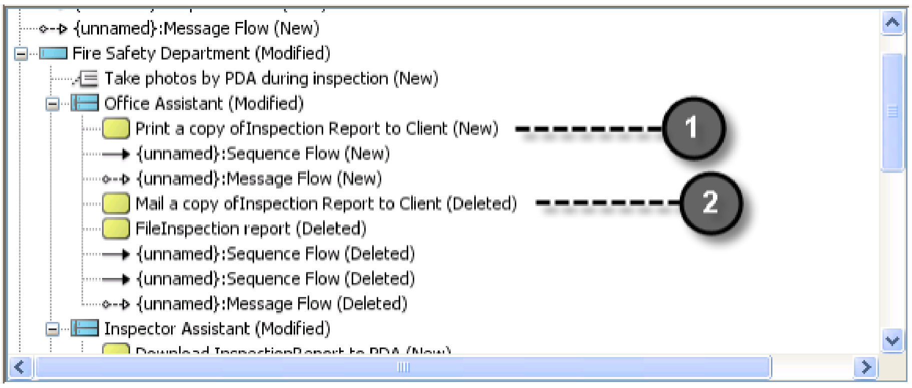
Project:

Diagram: Regular Inspection To-be Process

```
{unnamed}:Sequence Flow (New)
{unnamed}:Sequence Flow (New)
{unnamed}:Message Flow (New)
Fire Safety Department (Modified)
    Take photos by PDA during inspection (New)
    Office Assistant (Modified)
        Print a copy ofInspection Report to Client (New)
        {unnamed}:Sequence Flow (New)
        {unnamed}:Message Flow (New)
        Mail a copy ofInspection Report to Client (Deleted)
        FileInspection report (Deleted)
        {unnamed}:Sequence Flow (Deleted)
        {unnamed}:Sequence Flow (Deleted)
        {unnamed}:Message Flow (Deleted)
```

Figure 11-18 *Select a node to cause the shape to be selected in diagram*

Below is a description of results.



```
{unnamed}:Message Flow (New)
Fire Safety Department (Modified)
    Take photos by PDA during inspection (New)
    Office Assistant (Modified)
        Print a copy ofInspection Report to Client (New)
        {unnamed}:Sequence Flow (New)
        {unnamed}:Message Flow (New)
        Mail a copy ofInspection Report to Client (Deleted)
        FileInspection report (Deleted)
        {unnamed}:Sequence Flow (Deleted)
        {unnamed}:Sequence Flow (Deleted)
        {unnamed}:Message Flow (Deleted)
    Inspector Assistant (Modified)
        Download InspectionReport to PDA (New)
```

Figure 11-19 *Result of comparison*

Result	Description
1	The Task is newly added.
2	The Task is deleted.

Table 11-7 *Description of result*

Comparing Logical and Physical ERD

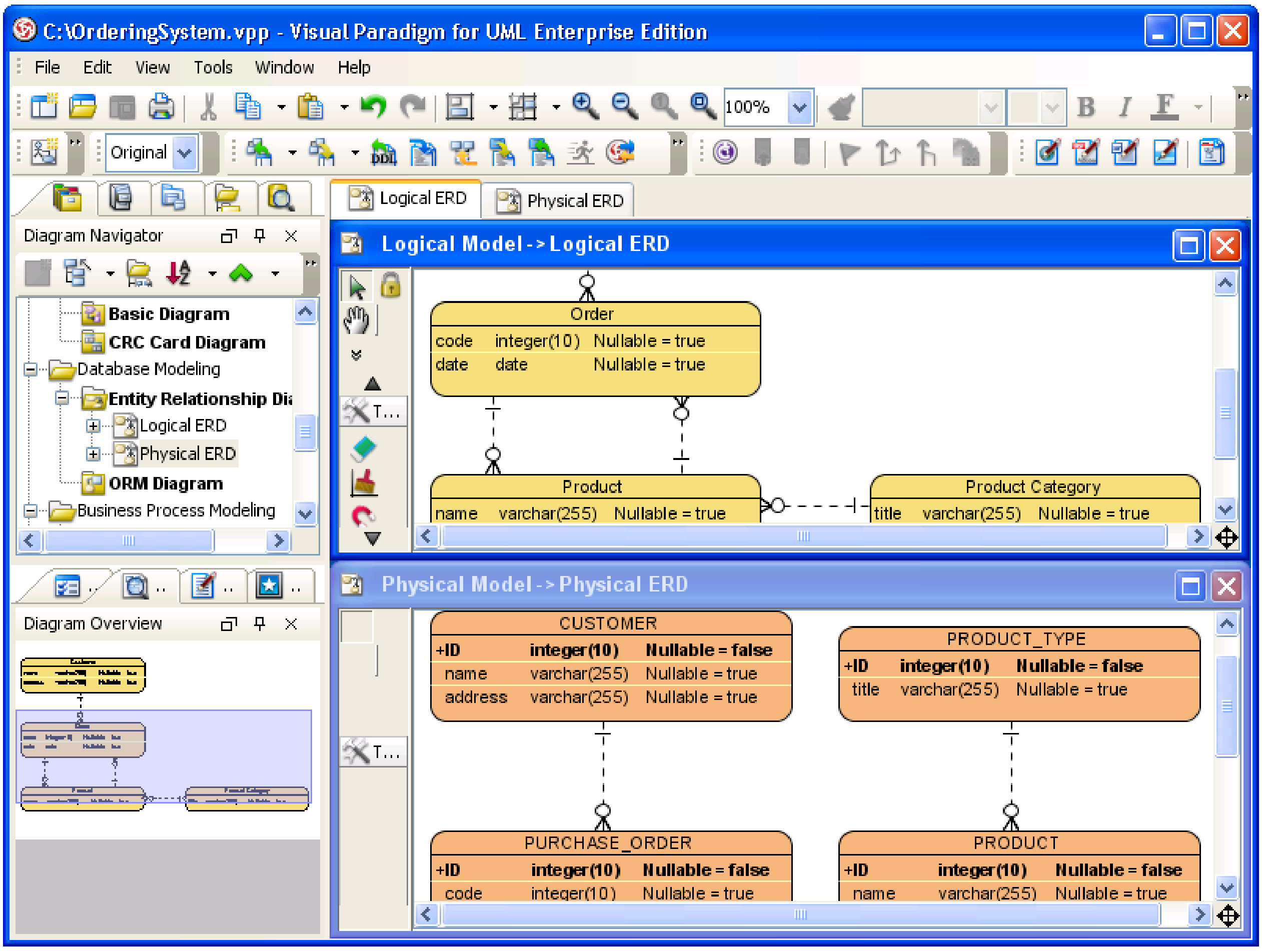
Entity relationship diagram (ERD) represents a detailed picture of the entities needed for a business. In forward engineering, ERD will be transformed into a relational database eventually. There are at least two types of ERD &ndash; Logical and Physical. They are used in different stages of development, 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, thus confirming the database is exactly following the initial business requirements regardless of the changes.

Here are two ERDs, one for modeling the Logical Model, and the other one is for modeling the Physical Model. We will make use of **Visual Diff** to find the differences between Logical and Physical ERD.



Order		
code	integer(10)	Nullable = true
date	date	Nullable = true

Product		
name	varchar(255)	Nullable = true

Product Category		
title	varchar(255)	Nullable = true

CUSTOMER		
+ID	integer(10)	Nullable = false
name	varchar(255)	Nullable = true
address	varchar(255)	Nullable = true

PRODUCT_TYPE		
+ID	integer(10)	Nullable = false
title	varchar(255)	Nullable = true

PURCHASE_ORDER		
+ID	integer(10)	Nullable = false
code	integer(10)	Nullable = true

PRODUCT		
+ID	integer(10)	Nullable = false
name	varchar(255)	Nullable = true

__Figure 11-20__ ERDs for Logical and Physical Model

1. From the diagram of *Logical ERD*, select **Tools > Visual Diff...** from the main menu.

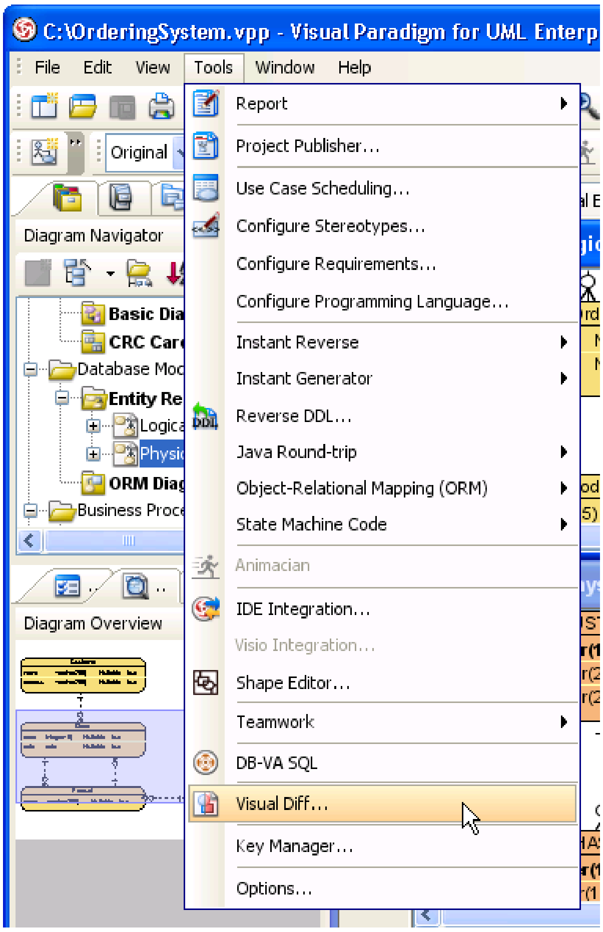
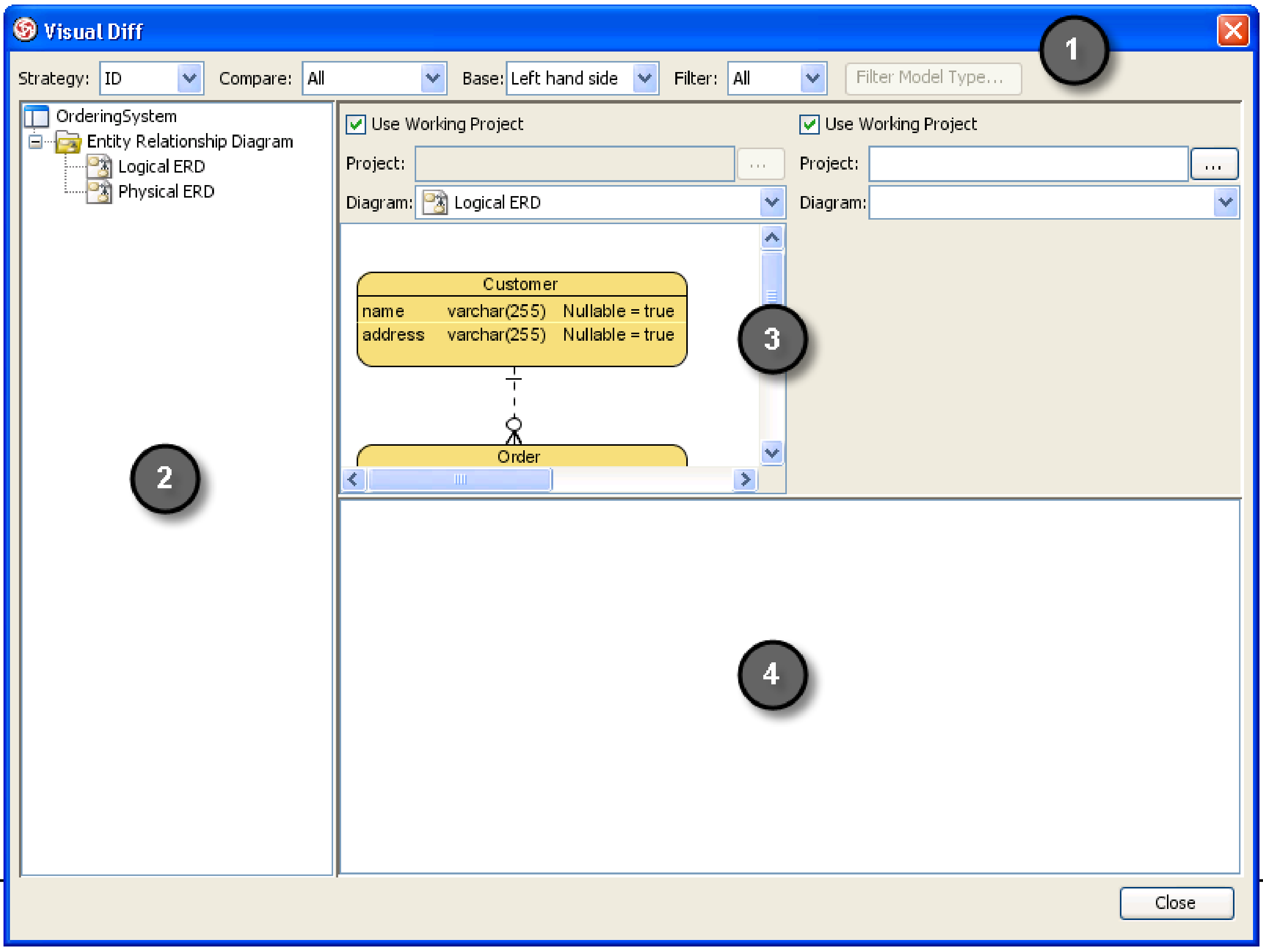


Figure 11-21 *Launching Visual Diff through the main menu*

NOTE: Besides starting through the main menu, you can start Visual Diff through the ways below:

- Right-click on diagram background and selecting **Utilities > Visual Diff...** from the popup menu.
- Click on the **Visual Diff** button on the **Tools** toolbar.

This starts **Visual Diff**.


Visual Diff

Strategy: ID Compare: All Base: Left hand side Filter: All Filter Model Type...

OrderingSystem
Entity Relationship Diagram
Logical ERD
Physical ERD

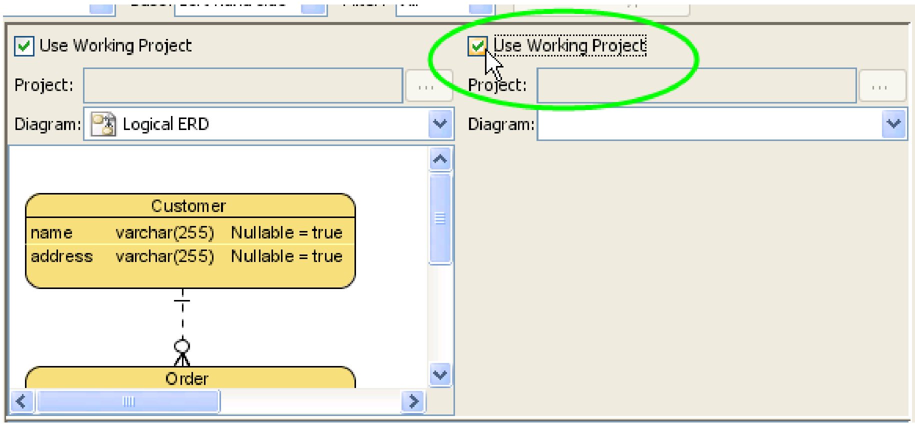
☑ Use Working Project
Project:
Diagram: Logical ERD

☑ Use Working Project
Project:
Diagram:

1
2
3
4
Customer
name varchar(255) Nullable = true
address varchar(255) Nullable = true
Order
Close

Figure 11-22 *The Visual Diff dialog box*

2. The left hand side is showing the currently opening diagram. Let's keep it unchanged. Now, select **Use Working Project** for the right hand side so that we can select a diagram in the same project to compare with.



☑ Use Working Project
Project:
Diagram: Logical ERD

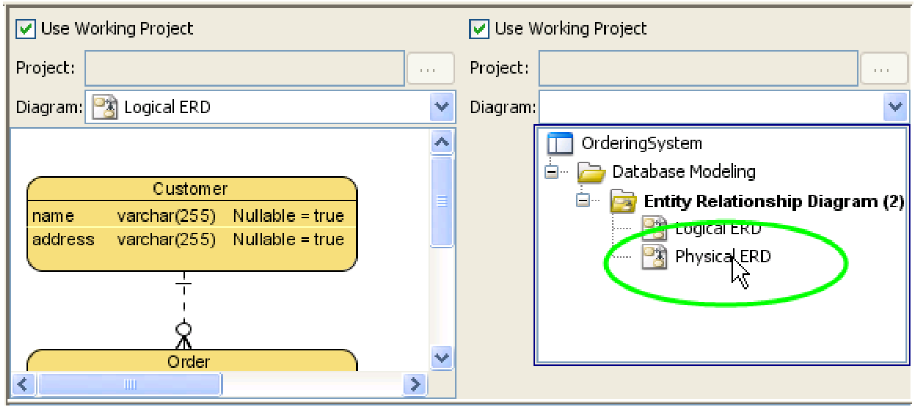
☑ Use Working Project
Project:
Diagram:

Customer		
name	varchar(255)	Nullable = true
address	varchar(255)	Nullable = true

Figure 11-23 *Select Use Working Project*

NOTE: To compare with diagram in another project, uncheck **Use Working Project** and select the project file in the Project field.

3. Select the *Physical ERD* to compare with.



☑ Use Working Project | ☑ Use Working Project

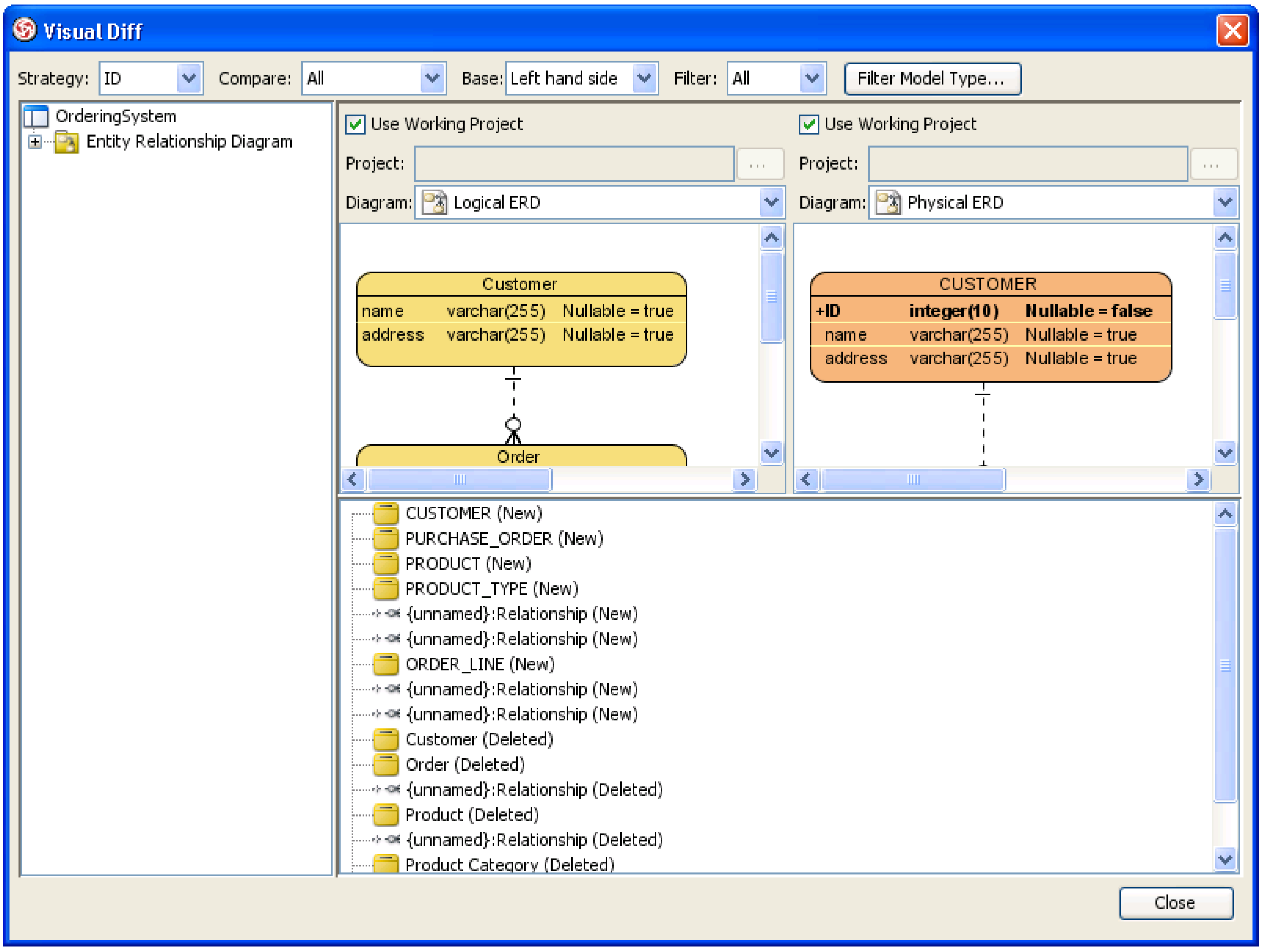
Project: | Project:

Diagram: Logical ERD | Diagram:

Customer
name varchar(255) Nullable = true
address varchar(255) Nullable = true
Order
OrderingSystem
Database Modeling
Entity Relationship Diagram (2)
Logical ERD
Physical ERD

Figure 11-24 Select diagram to compare

Visual Diff is updated to show the two diagrams side by side. The result pane at the bottom is updated, too. However, we need to configure **Visual Diff** in order to compare in the way we want.



Visual Diff

Strategy: ID Compare: All Base: Left hand side Filter: All

☑ Use Working Project | ☑ Use Working Project

Project: | Project:

Diagram: Logical ERD | Diagram: Physical ERD

- CUSTOMER (New)
- PURCHASE_ORDER (New)
- PRODUCT (New)
- PRODUCT_TYPE (New)
- {unnamed}:Relationship (New)
- {unnamed}:Relationship (New)
- ORDER_LINE (New)
- {unnamed}:Relationship (New)
- {unnamed}:Relationship (New)
- Customer (Deleted)
- Order (Deleted)
- {unnamed}:Relationship (Deleted)
- Product (Deleted)
- {unnamed}:Relationship (Deleted)
- Product Category (Deleted)

Filter Model Type...
OrderingSystem
Entity Relationship Diagram
Customer
name varchar(255) Nullable = true
address varchar(255) Nullable = true
Order
CUSTOMER
+ID integer(10) Nullable = false
name varchar(255) Nullable = true
address varchar(255) Nullable = true
Close

Figure 11-25 Diagrams selected

4. Select *Transitor* as **Strategy**.

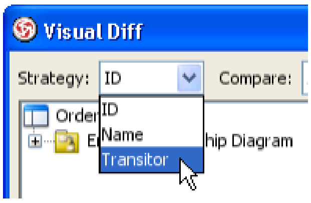


Figure 11-26 *Select a comparison strategy*

Below is a description of available **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 useful when visualizing 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. Typical examples are to compare databases and class models.
Transitor	Shapes will be matched base on their transition established by Model Transitor. This way of comparison is useful when visualizing differences between Models.

Table 11-9 *Description of comparison strategies*

5. Select *Model Element* as the scope of comparison.

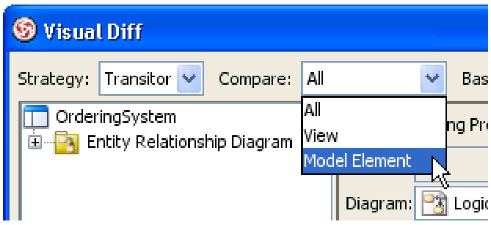


Figure 11-27 *Select the item to compare*

Below is a description of available **Compare** options.

Option	Description
All	Both view and model element are displayed.
View	Differences such as coordinates, width, height and color of shapes are displayed.
Model Element	Differences such as model name is displayed.

Table 11-10 *Description of compare options*

6. Select *Left hand side* as **Base**.

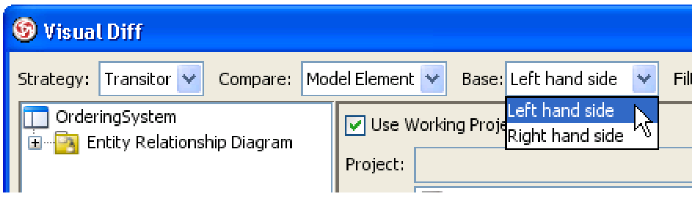


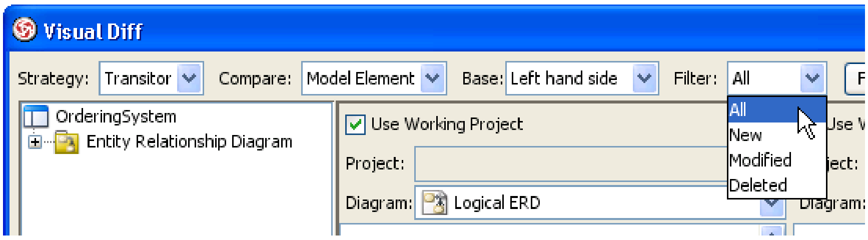
Figure 11-28 *Select a base*

Below is a description of available **Bases**.

Base	Description
Left hand side	The default base selection which cause comparison to be made base on the diagram on the left 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	Cause comparison to be made base 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.

Table 11-11 *Description of bases*

7. Select *All* as the **Filter**.



Strategy: Transitor Compare: Model Element Base: Left hand side Filter: All

OrderingSystem
Entity Relationship Diagram

☑ Use Working Project
Project:
Diagram: Logical ERD

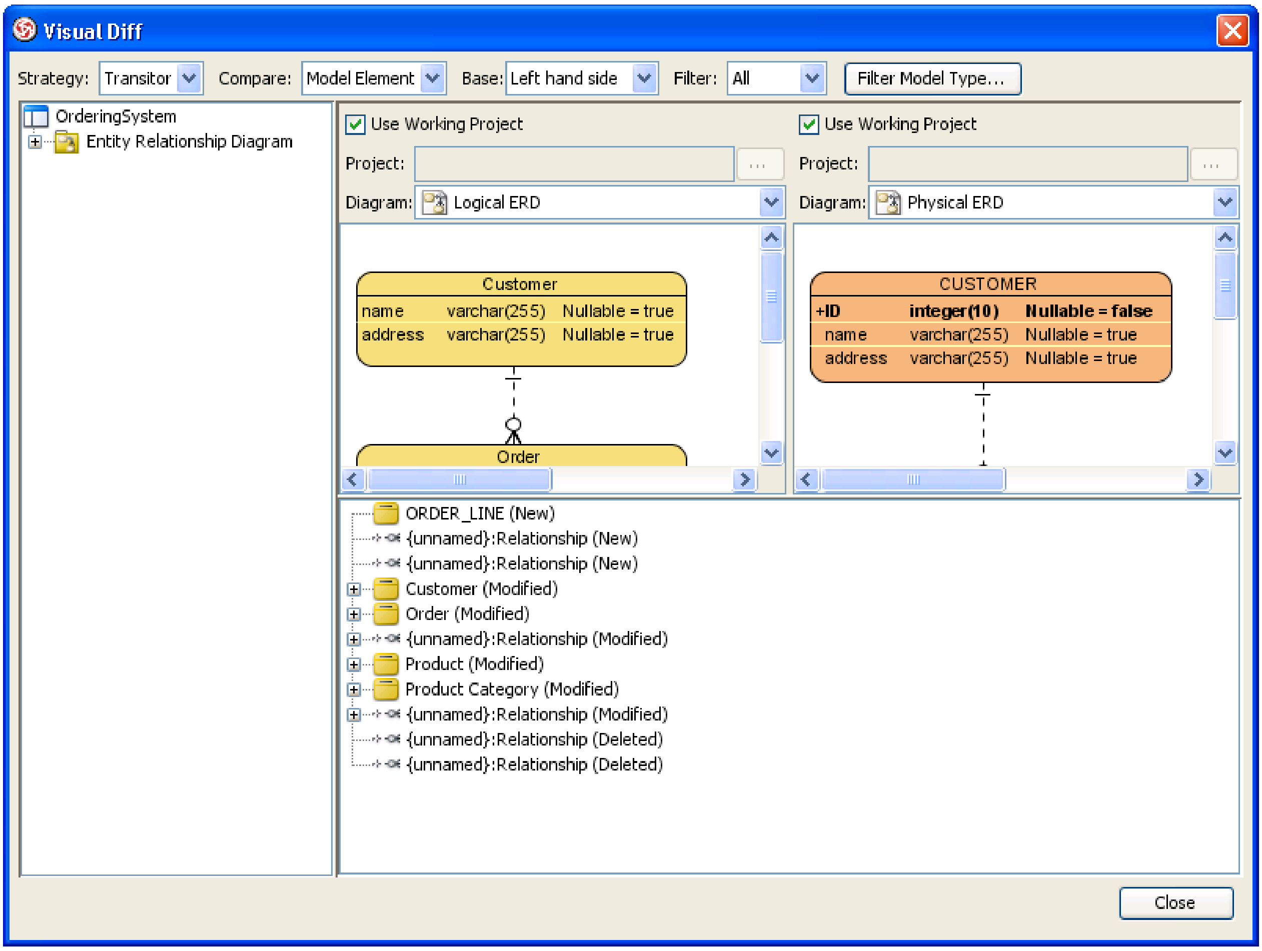
Figure 11-29 *Select a filter*

Below is a description of available **Filters**.

Filter	Description
All	Display all kinds of differences which includes the addition, modification and removal of shapes.
New	Display only results about the addition of shape, and hide the rest.
Modified	Display only results about the modification of shapes, and hide the rest.
Deleted	Display only results about the removal of shapes, and hide the rest.

Table 11-12 *Description of filters*

8. Once everything is set, we can see the differences of the two diagrams from the result pane.



Strategy: Transitor Compare: Model Element Base: Left hand side Filter: All

OrderingSystem
Entity Relationship Diagram

☑ Use Working Project
Project:
Diagram: Logical ERD

☑ Use Working Project
Project:
Diagram: Physical ERD

Customer		
name	varchar(255)	Nullable = true
address	varchar(255)	Nullable = true

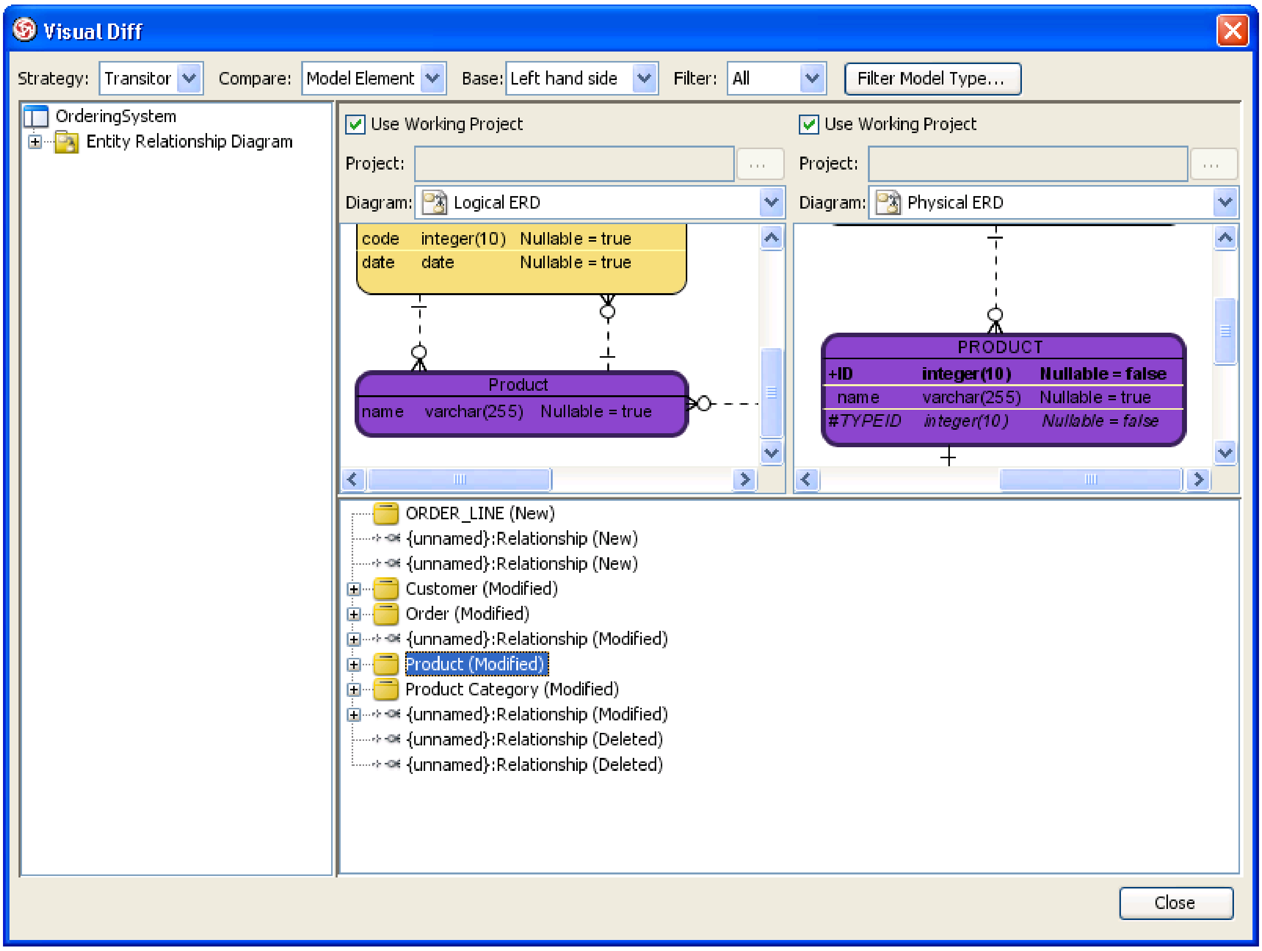
Order

CUSTOMER		
+ID	integer(10)	Nullable = false
name	varchar(255)	Nullable = true
address	varchar(255)	Nullable = true

ORDER_LINE (New)
{unnamed}:Relationship (New)
{unnamed}:Relationship (New)
Customer (Modified)
Order (Modified)
{unnamed}:Relationship (Modified)
Product (Modified)
Product Category (Modified)
{unnamed}:Relationship (Modified)
{unnamed}:Relationship (Deleted)
{unnamed}:Relationship (Deleted)

Figure 11-30 *Result of comparison is obtained*

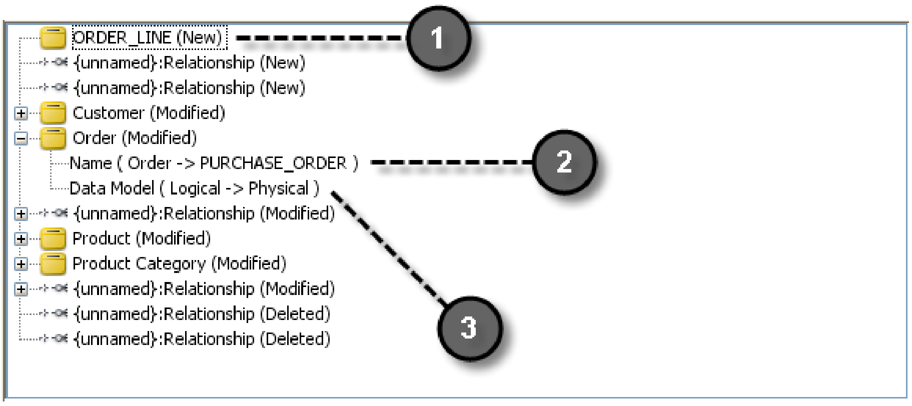
9. It is possible to click on a node to select the shape in diagrams. Selected shape is painted in dark purple.



ORDER_LINE (New)
{unnamed}:Relationship (New)
{unnamed}:Relationship (New)
Customer (Modified)
Order (Modified)
{unnamed}:Relationship (Modified)
Product (Modified)
Product Category (Modified)
{unnamed}:Relationship (Modified)
{unnamed}:Relationship (Deleted)
{unnamed}:Relationship (Deleted)

Figure 11-31 *Select a node to cause the shape to be selected in diagram*

Below is a description of results.



ORDER_LINE (New)
{unnamed}:Relationship (New)
{unnamed}:Relationship (New)
Customer (Modified)
Order (Modified)
Name (Order -> PURCHASE_ORDER)
Data Model (Logical -> Physical)
{unnamed}:Relationship (Modified)
Product (Modified)
Product Category (Modified)
{unnamed}:Relationship (Modified)
{unnamed}:Relationship (Deleted)
{unnamed}:Relationship (Deleted)

Figure 11-32 *Result of comparison*

Result	Description
1	The Entity is newly added.
2	The Entity is renamed.
3	The Data Model is changed from Logical to Physical.

Table 11-13 *Description of result*

What is Animacian?

Animacian is a tool that helps you makes possible paths in a diagram active 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 begin, 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.

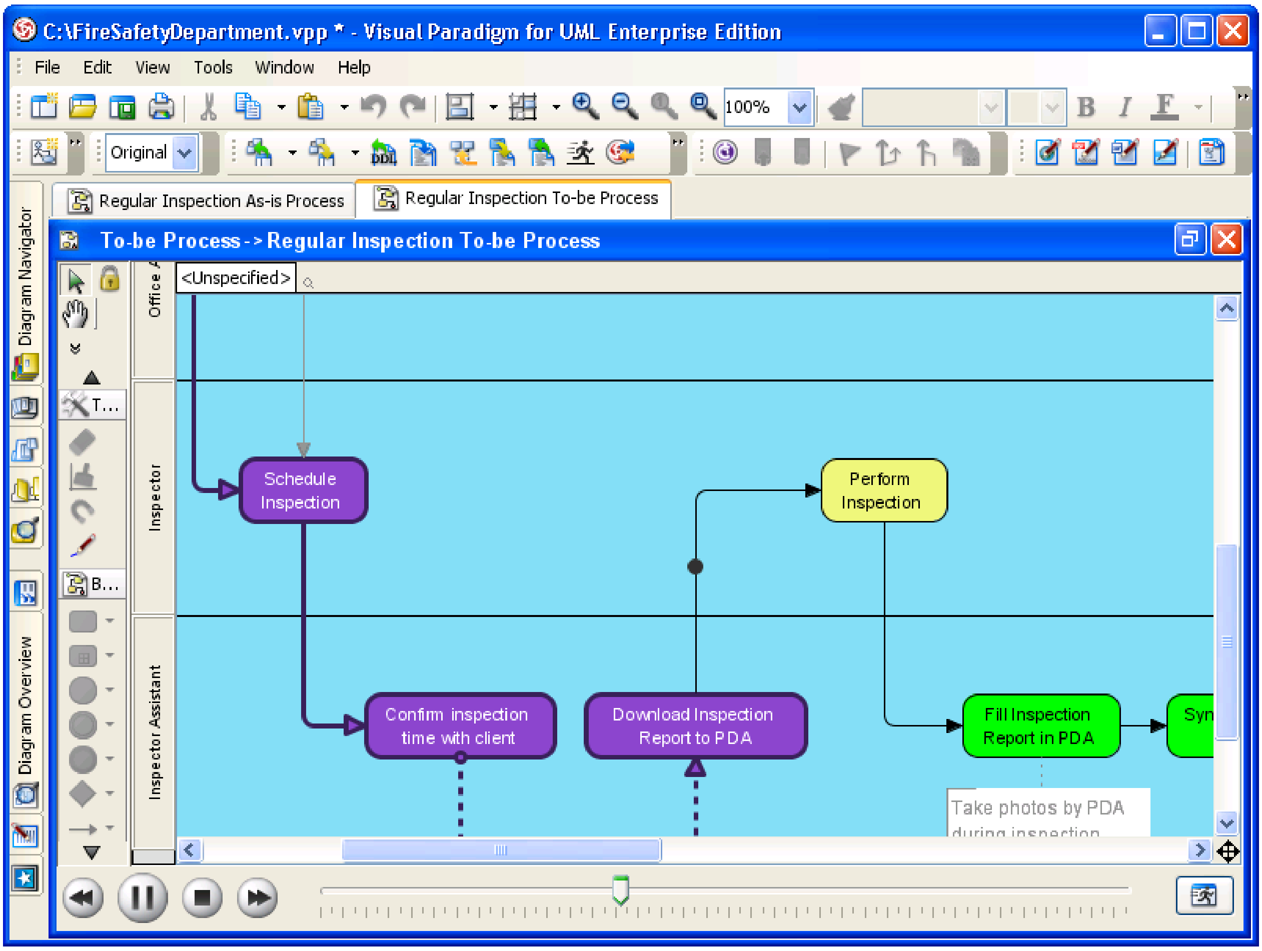
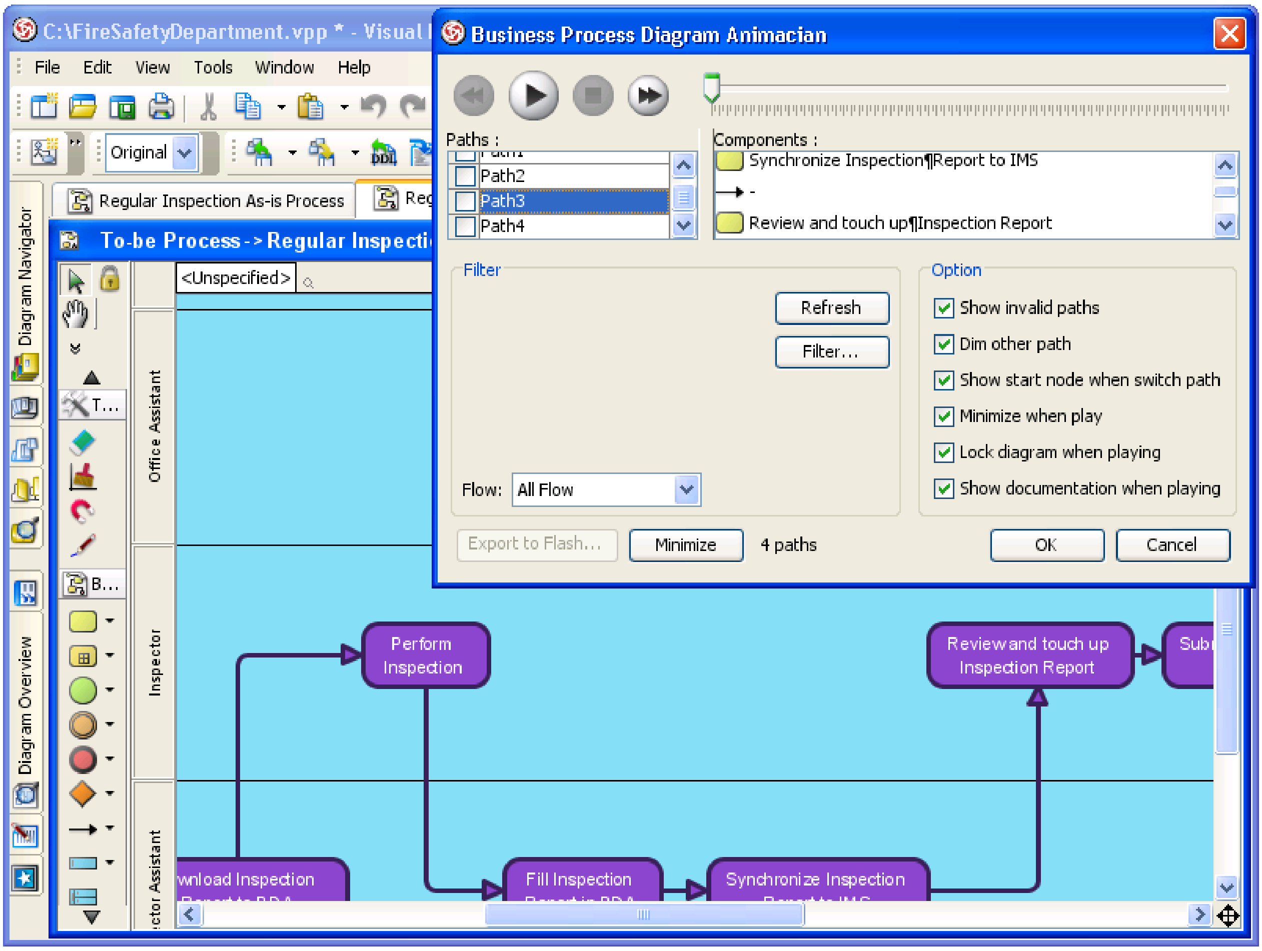

C:\FireSafetyDepartment.vpp * - Visual Paradigm for UML Enterprise Edition
File
Edit
View
Tools
Window
Help
100%
Original
Regular Inspection As-is Process
Regular Inspection To-be Process
To-be Process -> Regular Inspection To-be Process
<Unspecified>
Office A
Inspector
Inspector Assistant
Schedule Inspection
Perform Inspection
Confirm inspection time with client
Download Inspection Report to PDA
Fill Inspection Report in PDA
Syn
Take photos by PDA during inspection
Diagram Navigator
Diagram Overview
T...
B...

Figure 12-1 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.



Business Process Diagram Animacian

Paths :
- □ Path1
- □ Path2
- □ Path3
- □ Path4

Components :
- Synchronize Inspection¶Report to IMS
- -
- Review and touch up¶Inspection Report

Filter

Flow: All Flow

Option
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

4 paths

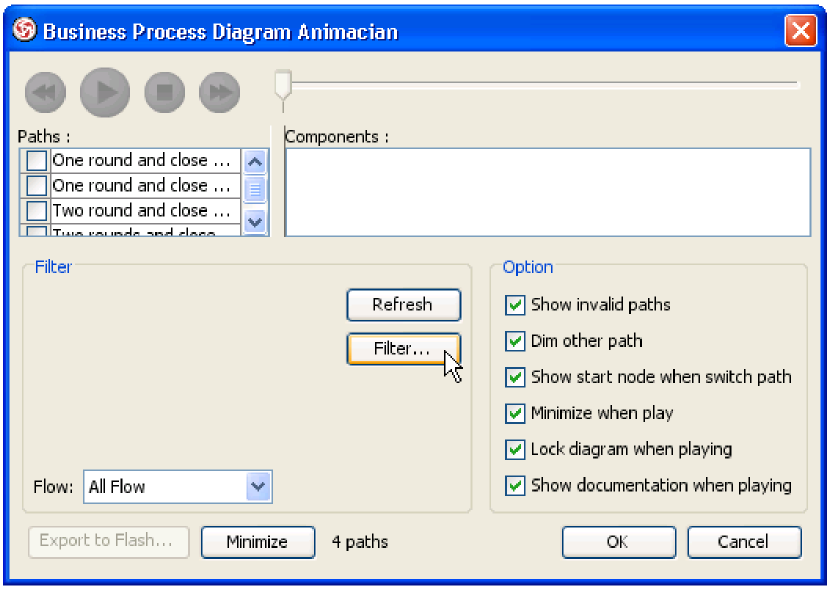
C:\FireSafetyDepartment.vpp * - Visual
File
Edit
View
Tools
Window
Help
Original
Regular Inspection As-is Process
Reg
To-be Process -> Regular Inspecti
<Unspecified>
Diagram Navigator
Diagram Overview
Office Assistant
Inspector
ctor Assistant
T...
B...
Refresh
Filter...
Export to Flash...
Minimize
OK
Cancel
Perform Inspection
Review and touch up Inspection Report
Sub
wnload Inspection Report to PDA
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS

Figure 12-2 Paths are identified from a diagram

Filter Business Paths Base on Conditions

The Animacian dialog box is where you can configure animation and select path to animate. You can apply filter on path identification, which clears undesired path that does not match certain condition. To configure filter:

1. Click **Filter...** in the Animacian dialog box.



Business Process Diagram Animacian

Paths :

- □ One round and close ...
- □ One round and close ...
- □ Two round and close ...
- □ Two rounds and close

Components :

Option

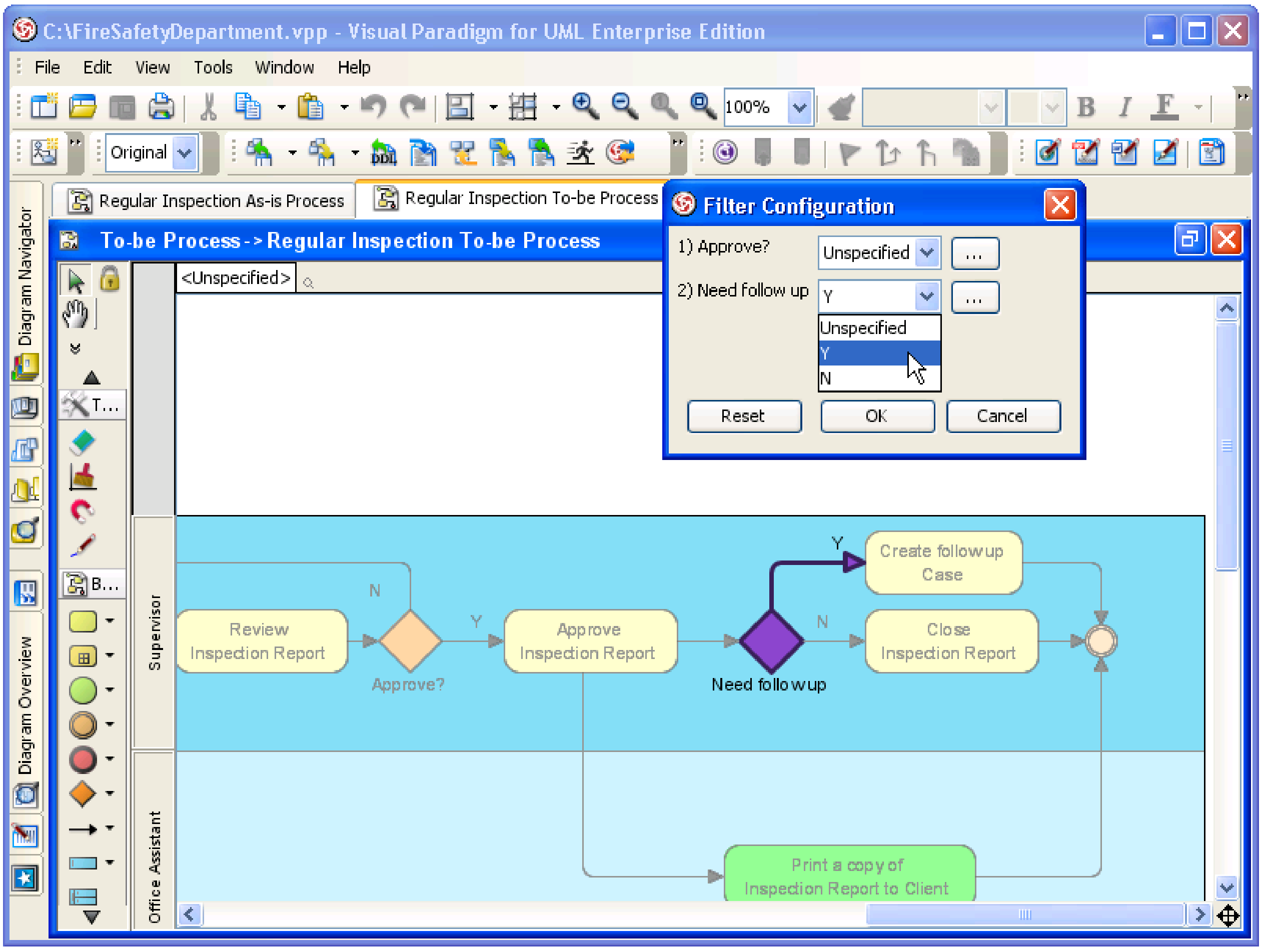
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

Flow: All Flow

4 paths

Figure 12-3 Clicking on the Filter button in the Animacian dialog box

2. In the **Filter Configuration** dialog box, conditional flows, if any, are listed. Click on a drop down menu to select the expected outgoing flow of a condition. By doing so, only paths that cover the selected flow are identified, the rest will be ignored.



C:\FireSafetyDepartment.vpp - Visual Paradigm for UML Enterprise Edition

Filter Configuration

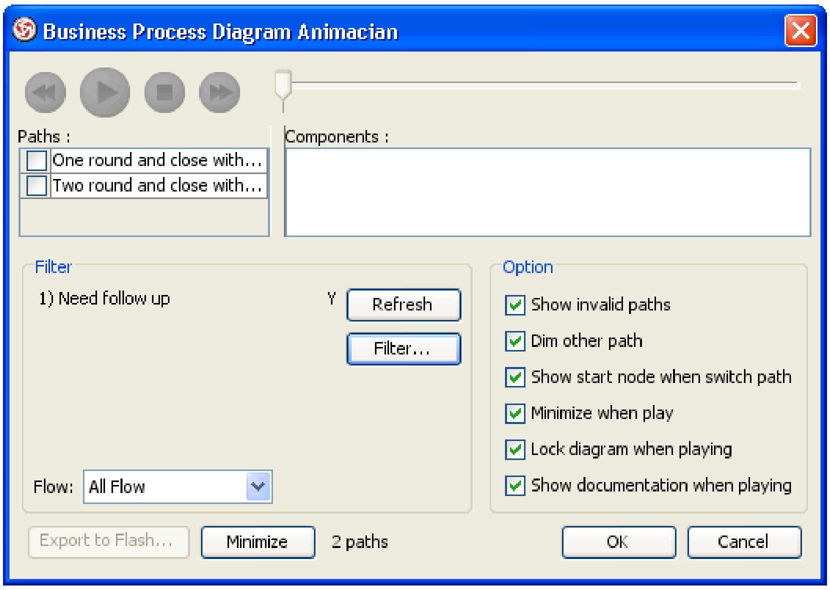
1) Approve? Unspecified

2) Need follow up Y

Figure 12-4 Configuring filter

NOTE: A condition can have more than two outgoing flows. You can allow multiple outgoing flows by pressing the **...** button next to the drop down menu, and selecting the flows in the **Select Multiple Values** dialog box.

3. Click **OK** to confirm. The identified paths will be reduced accordingly.



Paths :

□ One round and close with...

□ Two round and close with...

Components :

Filter

1) Need follow up Y

Flow: All Flow

Option

☑ Show invalid paths

☑ Dim other path

☑ Show start node when switch path

☑ Minimize when play

☑ Lock diagram when playing

☑ Show documentation when playing

2 paths

Figure 12-5 *Paths are reduced as a result of applying filter*

Walking through a Path Step-by-Step

Instead of letting the animation to run itself, you can control it yourself. The horizontal bar that appear at the bottom of VP-UML when animating lets you control the animation. Besides pausing, playing and stopping the animation, you can also move a shape backward or forward by pressing the ⏪ and ⏩ button. By making use with the forward and backward buttons, you can walk through a path shape by shape.

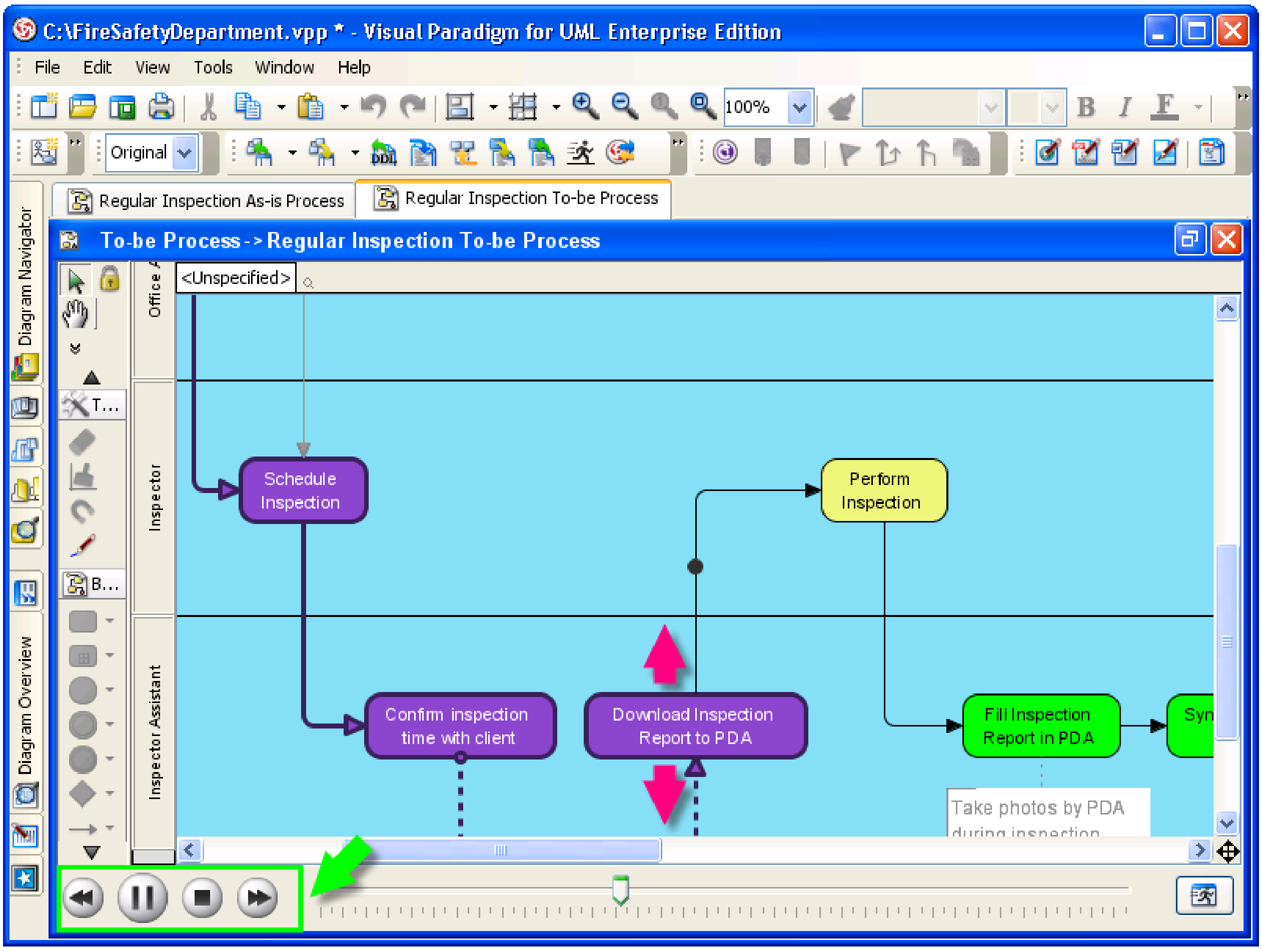
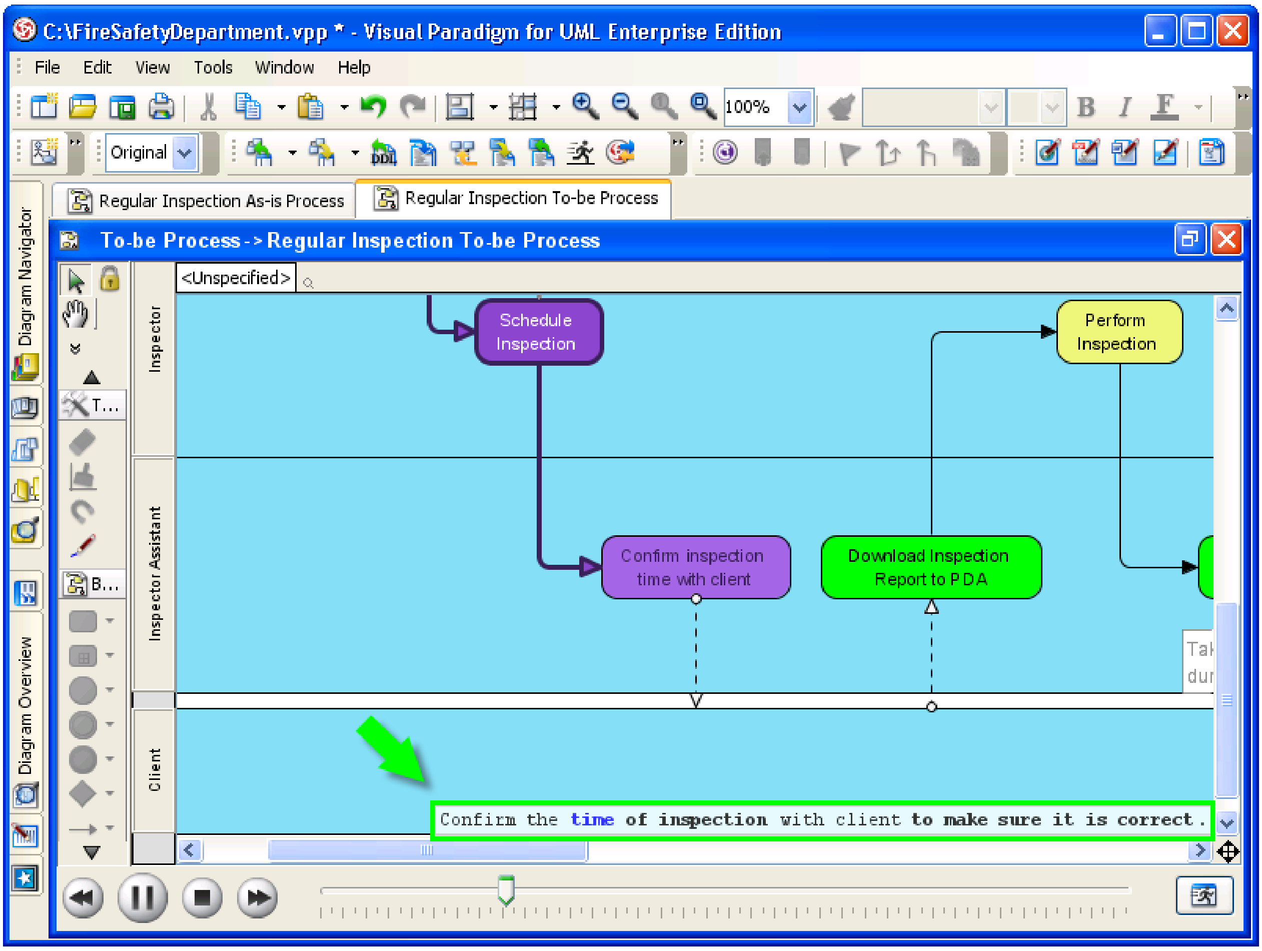

C:\FireSafetyDepartment.vpp * - Visual Paradigm for UML Enterprise Edition
File
Edit
View
Tools
Window
Help
100%
Original
Regular Inspection As-is Process
Regular Inspection To-be Process
To-be Process -> Regular Inspection To-be Process
<Unspecified>
Office A
Inspector
Inspector Assistant
Diagram Navigator
Diagram Overview
T...
B...
Schedule Inspection
Perform Inspection
Confirm inspection time with client
Download Inspection Report to PDA
Fill Inspection Report in PDA
Syn
Take photos by PDA during inspection

Figure 12-6 *Walking though a path step by step*

Showing Documentation of Current Step When Playing Animation

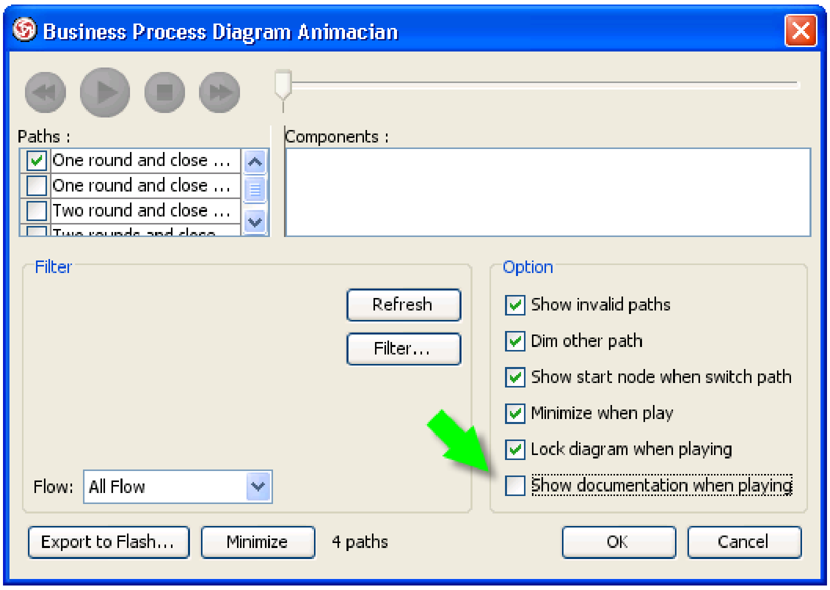
When walking through a path, the documentation of the visiting shape, if written, will appear instantly at the bottom right corner of VP-UML.


C:\FireSafetyDepartment.vpp * - Visual Paradigm for UML Enterprise Edition
File
Edit
View
Tools
Window
Help
100%
Original
Regular Inspection As-is Process
Regular Inspection To-be Process
To-be Process -> Regular Inspection To-be Process
<Unspecified>
Diagram Navigator
Diagram Overview
Inspector
Inspector Assistant
Client
Schedule Inspection
Perform Inspection
Confirm inspection time with client
Download Inspection Report to PDA
Tal
dur
T...
B...

Confirm the time of inspection with client to make sure it is correct.

Figure 12-7 Documentation of shape appear when animating

To turn this function on or off, open the **Animacian** dialog box, then check or uncheck **Show documentation when playing**.


Business Process Diagram Animacian

Paths :
- ☑ One round and close ...
- □ One round and close ...
- □ Two round and close ...
- □ Two rounds and close

Components :

Filter

Refresh

Filter...

Flow: All Flow

Option
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- □ Show documentation when playing

Export to Flash... | Minimize | 4 paths | OK | Cancel

Figure 12-8 Option for showing documentation when playing

1. To launch **Animacian**, select **Tools > Animacian** from the main menu.

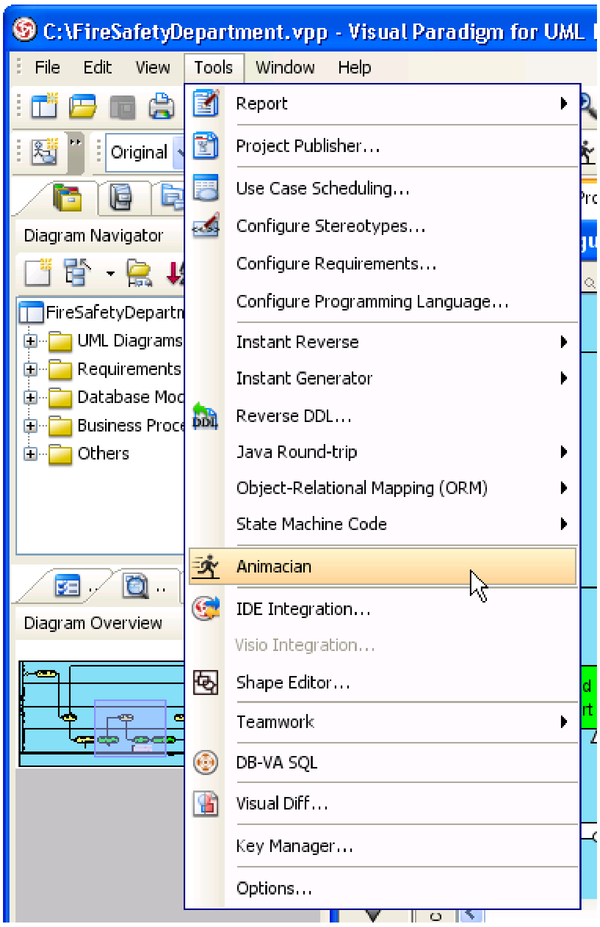
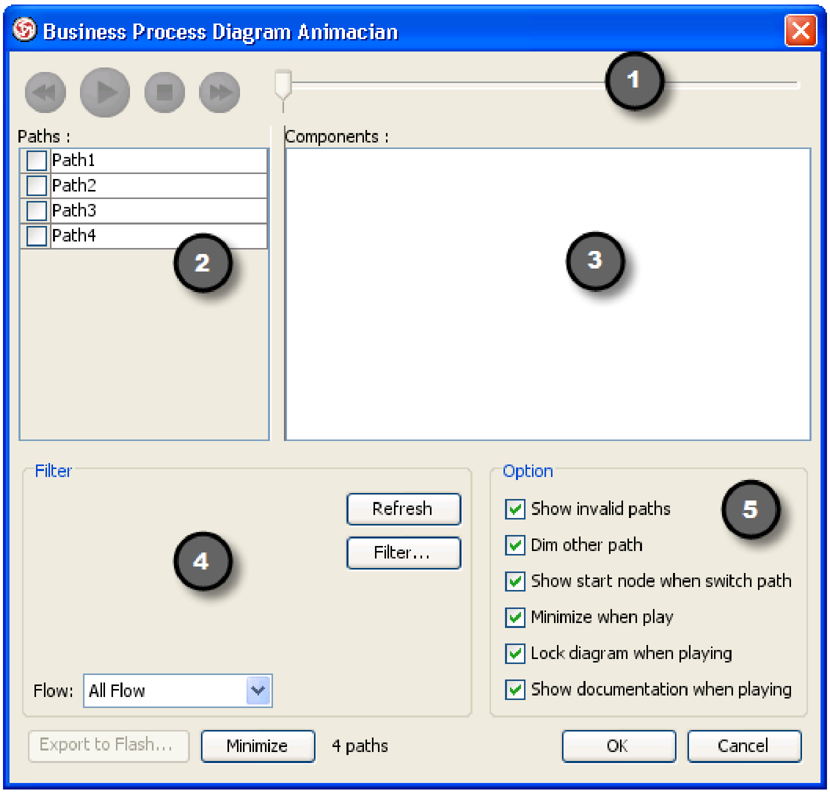


Figure 12-9 Opening Animacian through the main menu

NOTE: Animacian can also be started by any of the ways below:

- Right-click on the diagram background and select **Utilities > Animacian...** from the popup menu.
- Click on the Animacian button in the toolbar.

This starts Animacian. The **Animacian** dialog box is where you can select an execution path to play an animation.



□ Path1
□ Path2
□ Path3
□ Path4

Flow: All Flow

☑ Show invalid paths
☑ Dim other path
☑ Show start node when switch path
☑ Minimize when play
☑ Lock diagram when playing
☑ Show documentation when playing

4 paths

Figure 12-10 The Animacian dialog box

Below is a description of different parts of dialog box.

Region	Description
1	The buttons and slider are for controlling the flow of animation. Once the Play button ▶ is pressed, the Animacian dialog box will

2. The **Paths** list lists the available animation paths of the diagram. Each path represents a possible way to go walk through the diagram. By default, paths are named as Path1, Path2, and so forth. It is recommended to give meaningful name to the paths for better clarification. To rename a path, move the mouse cursor over a path in the list.

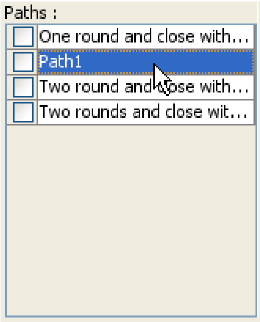

Paths :
□ One round and close with...
□ Path1
□ Two round and close with...
□ Two rounds and close wit...

Figure 12-11 *Going to rename a path*

3. Double-click on the path to enter editing mode.

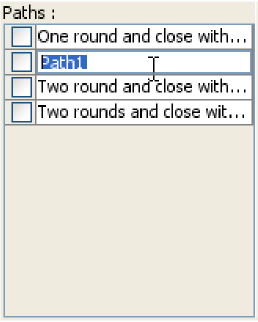

Paths :
□ One round and close with...
□ Path1
□ Two round and close with...
□ Two rounds and close wit...

Figure 12-12 *Entered editing mode of path name*

4. Enter the name of path.

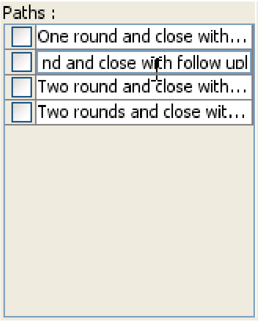

Paths :
□ One round and close with...
□ nd and close with follow upl
□ Two round and close with...
□ Two rounds and close wit...

Figure 12-13 *Enter name of path*

5. Press the **Enter** key to confirm editing.

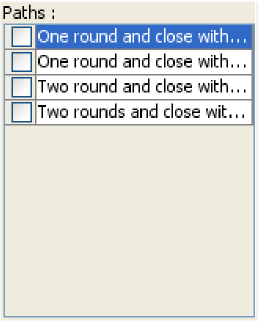
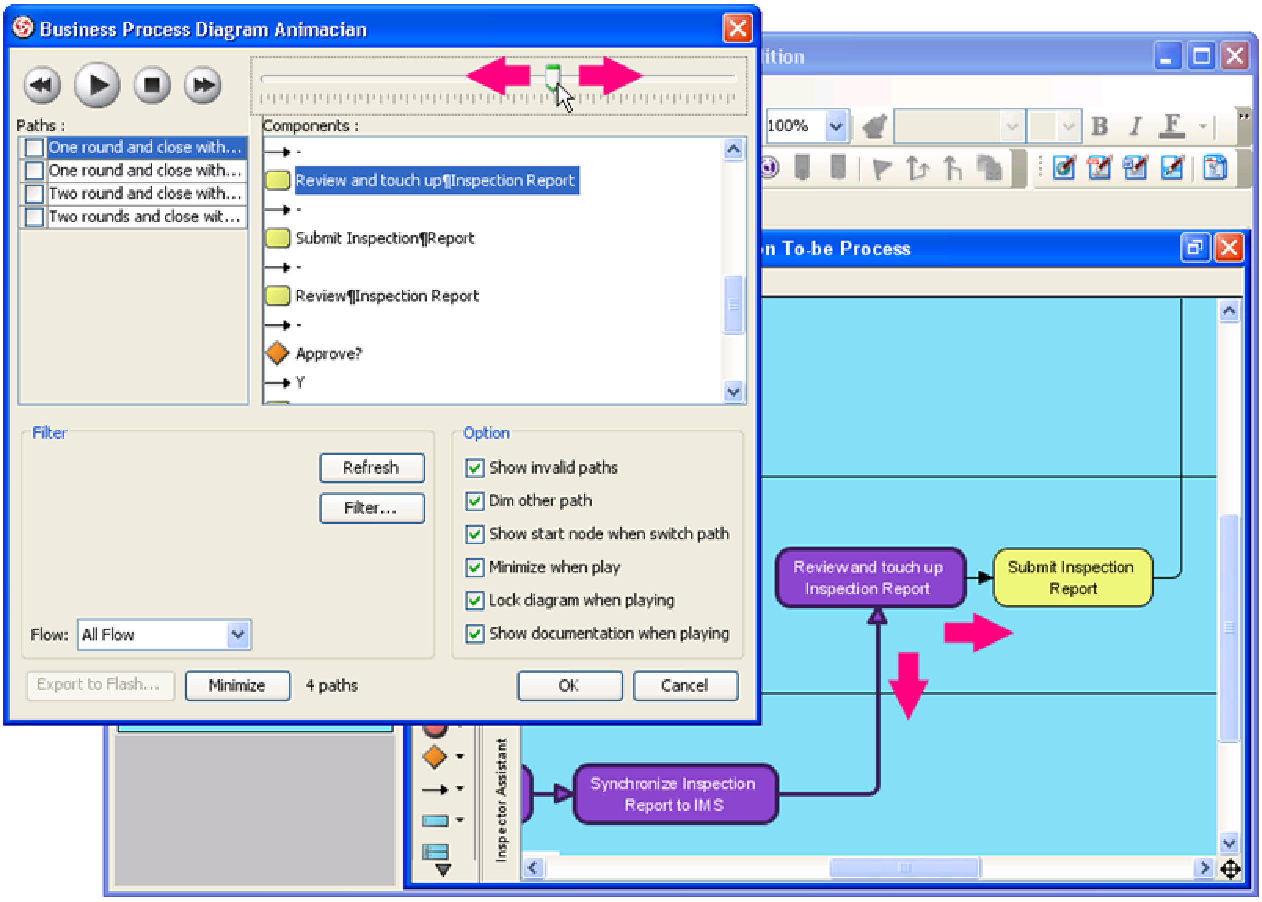

Paths :
□ One round and close with...
□ One round and close with...
□ Two round and close with...
□ Two rounds and close wit...

Figure 12-14 *Path named*

6. User can navigate a path manually by any of the ways below. Note that the shapes on a path will be highlighted in the diagram accordingly.

- Drag the slider back and forth.



Business Process Diagram Animacian

Paths :

- ☐ One round and close with...
- ☐ One round and close with...
- ☐ Two round and close with...
- ☐ Two rounds and close wit...

Components :

- Review and touch up¶Inspection Report
- Submit Inspection¶Report
- Review¶Inspection Report
- Approve?
- Y

Option

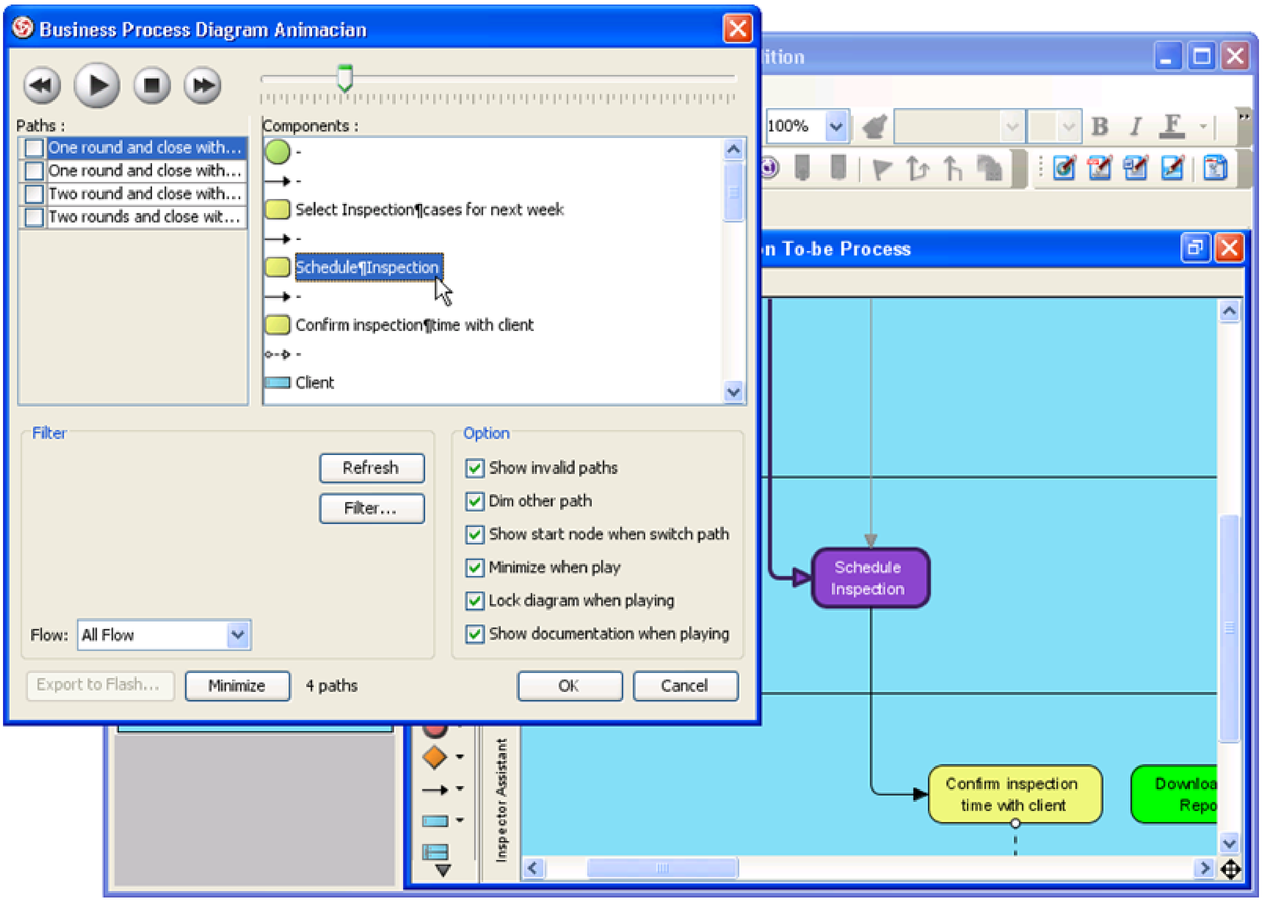
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

Flow: All Flow

4 paths

Figure 12-15 *Dragging a slider back and forth*

- Select component in the **Components** list.



Business Process Diagram Animacian

Paths :

- ☐ One round and close with...
- ☐ One round and close with...
- ☐ Two round and close with...
- ☐ Two rounds and close wit...

Components :

- Select Inspection¶cases for next week
- Schedule¶Inspection
- Confirm inspection¶time with client
- Client

Option

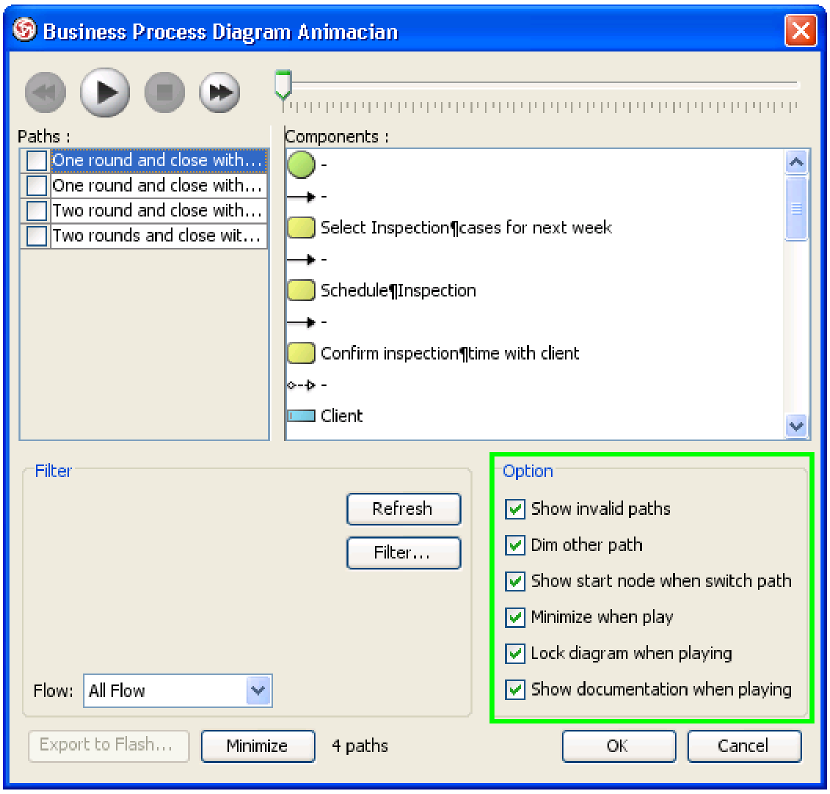
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

Flow: All Flow

4 paths

Figure 12-16 *Select a Component in Components list*

7. Configure the animation in the **Option** pane.



Business Process Diagram Animacian

Paths :

- □ One round and close with...
- □ One round and close with...
- □ Two round and close with...
- □ Two rounds and close wit...

Components :

- -
- -
- Select Inspection¶cases for next week
- -
- Schedule¶Inspection
- -
- Confirm inspection¶time with client
- -
- Client

Flow: All Flow

Option

- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

4 paths

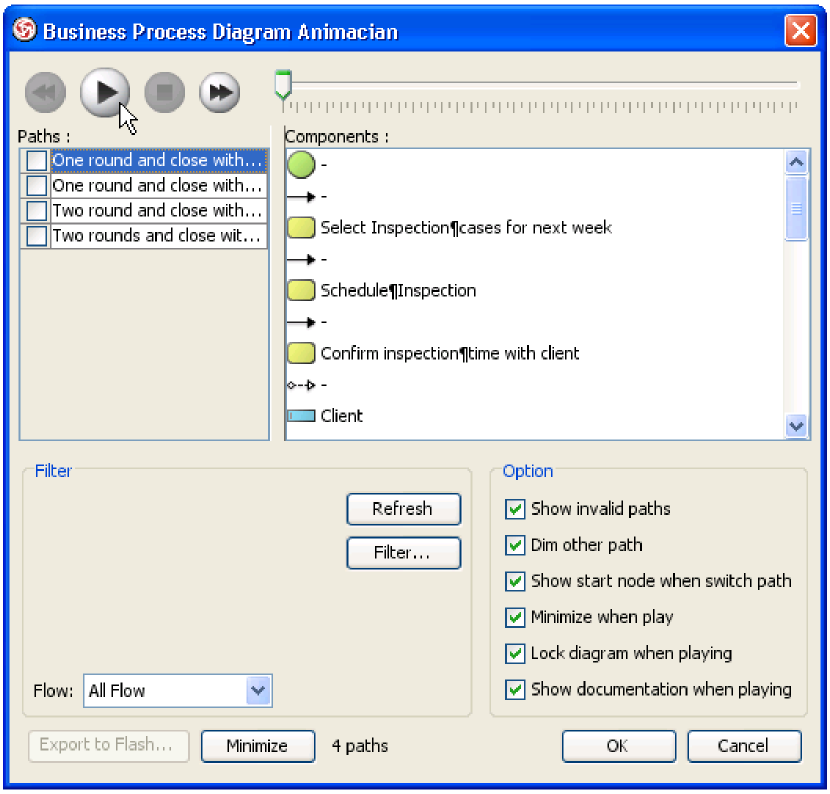
Figure 12-17 *Options for Animacian*

Here is a description of **Animacian** options:

Option	Description
Show invalid paths	List also the invalid, non play-able paths in the Paths list.
Dim other path	Dim the components that are not a part of the selected path.
Show start node when switch path	Jump to the first node of a chosen path upon selection, or keep staying at the current viewing region.
Minimize when play	Minimize this dialog box (Business Process Diagram Animacian) when playing an animation.
Lock diagram when playing	Avoid accidental editing when playing by locking the diagram.
Show documentation when playing	Show documentation of shape at the bottom right of diagram when playing animation.

Table 12-2 *Description of Animacian options*

8. When everything is ready, click the ▶ button to play the animation of the chosen path.



Business Process Diagram Animacian

Paths :

- □ One round and close with...
- □ One round and close with...
- □ Two round and close with...
- □ Two rounds and close wit...

Components :

- ● -
- → -
- Select Inspection¶cases for next week
- → -
- Schedule¶Inspection
- → -
- Confirm inspection¶time with client
- ⇢ -
- Client

Filter

Flow: All Flow

Option

- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

4 paths

Figure 12-18 *Going to play a path*

9. When pressed, Animacian will be minimized to the bottom of diagram, with several buttons and a slider appearing on it.
The animation starts. A black dot will appear at beginning of path, and traverse through the path until the end. When the black dot is reaching a shape, the shape will be highlighted in purple gradually.

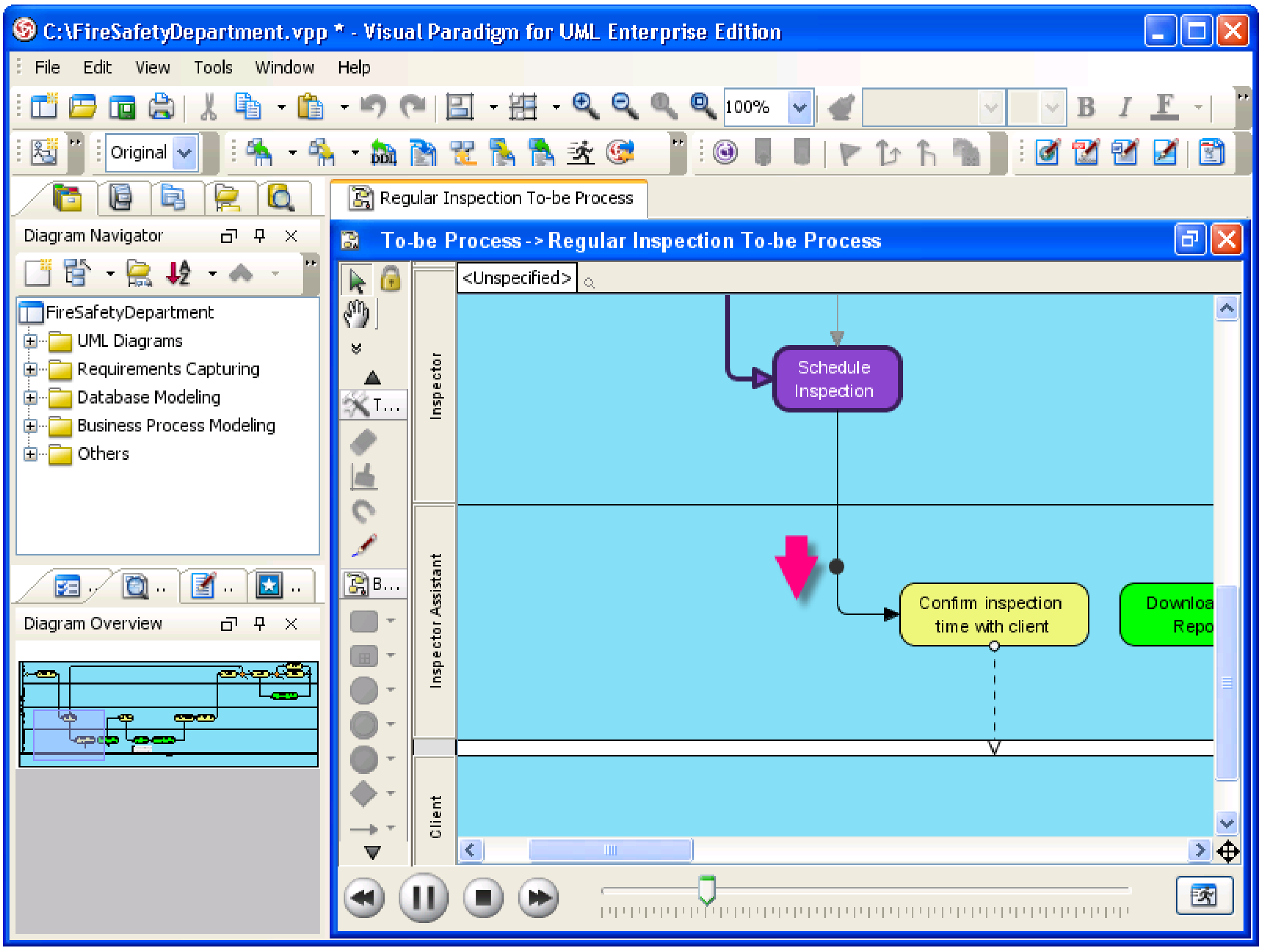

C:\FireSafetyDepartment.vpp * - Visual Paradigm for UML Enterprise Edition
File Edit View Tools Window Help
100%
Original
Regular Inspection To-be Process
Diagram Navigator
FireSafetyDepartment
UML Diagrams
Requirements Capturing
Database Modeling
Business Process Modeling
Others
Diagram Overview
To-be Process -> Regular Inspection To-be Process
<Unspecified>
Inspector
Inspector Assistant
Client
Schedule Inspection
Confirm inspection time with client
Downloa Repo
T...
B...

Figure 12-19 Animation is running

The minimized **Animacian** lets you control the flow of animation. Here is a description of the buttons.

Button	Description
⏪	Backward. Move one shape backward in the flow.
⏸	Pause. Temporary stop playing the movie. Can play again by pressing the ▶ button.
▶	Play. Play or continue to play the movie content.
⏩	Forward. Advance to the next shape in the flow.
⏹	Stop. Terminate the movie.
	Maximize **Animacian**.

Table 12-3 Description of Animacian bar

1. To launch **Animacian**, select **Tools > Animacian** from the main menu.

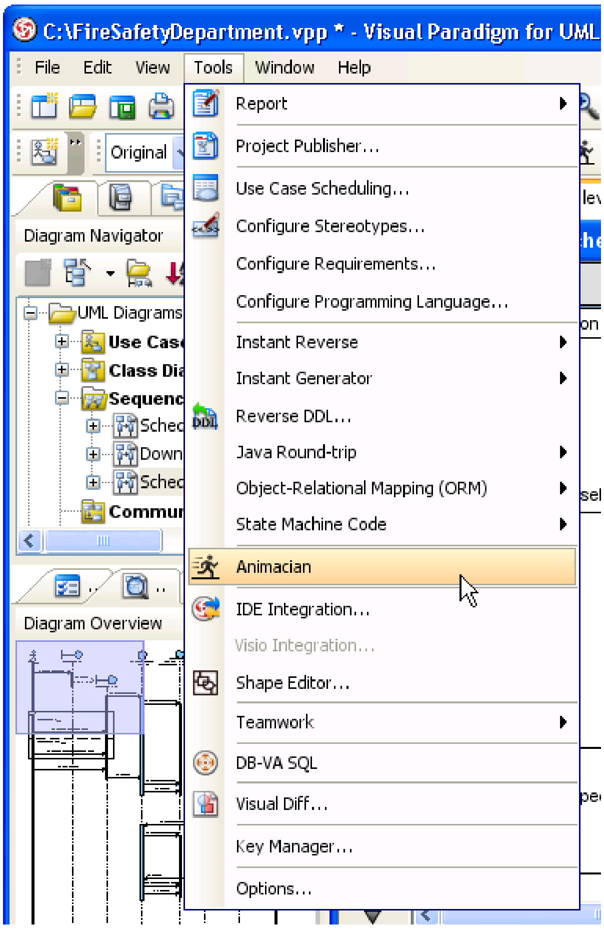
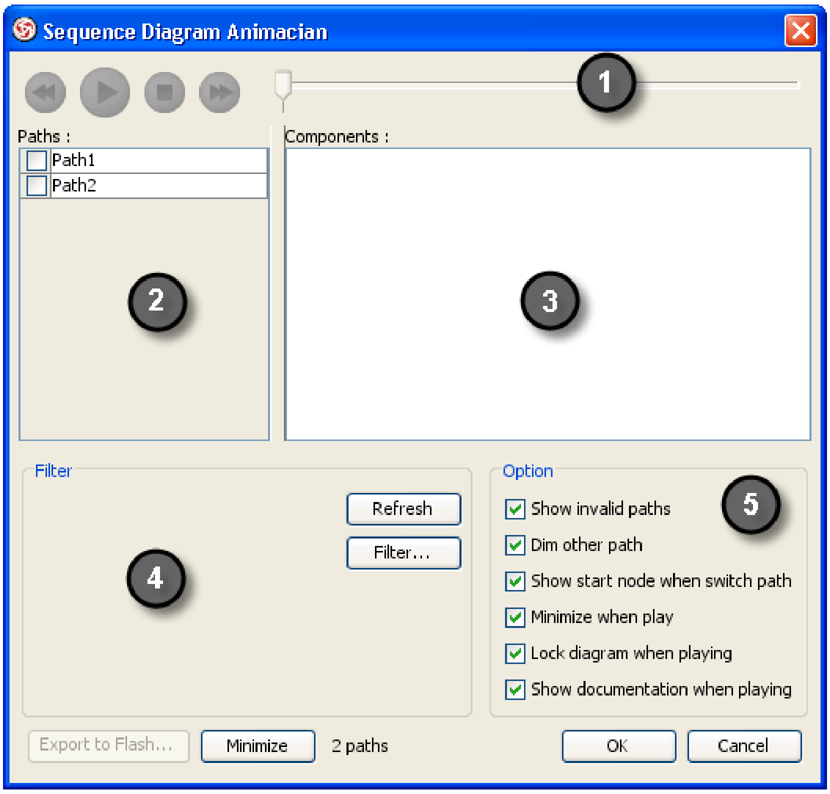


Figure 12-20 Opening Animacian through the main menu

NOTE: Animacian can also be started by any of the ways below:

- Right-click on the diagram background and select **Utilities > Animacian...** from the popup menu.
- Click on the Animacian button in the toolbar.

This starts Animacian. The **Animacian** dialog box is where you can select an execution path to play an animation.



□ Path1
□ Path2

☑ Show invalid paths
☑ Dim other path
☑ Show start node when switch path
☑ Minimize when play
☑ Lock diagram when playing
☑ Show documentation when playing

2 paths

Figure 12-21 The Animacian dialog box

Below is a description of different parts of dialog box.

Region	Description
1	The buttons and slider are for controlling the flow of animation. Once the Play button is pressed, the Animacian dialog box will

2. The **Paths** list lists the available animation paths of the diagram. Each path represents a possible way to go walk through the diagram. By default, paths are named as Path1, Path2, and so forth. It is recommended to give meaningful name to the paths for better clarification. To rename a path, move the mouse cursor over a path in the list.

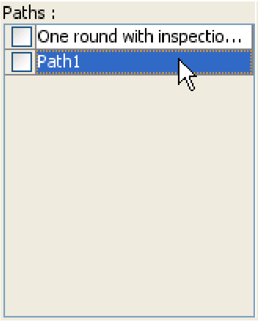


Figure 12-22 *Going to rename a path*

3. Double-click on the path to enter editing mode.

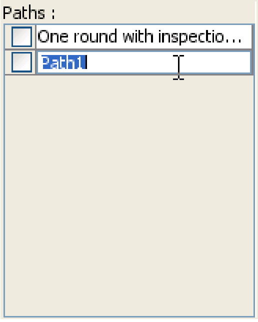


Figure 12-23 *Entered editing mode of path name*

4. Enter the name of path.

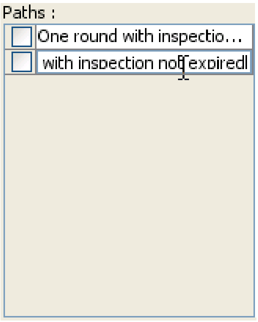


Figure 12-24 *Enter name of path*

5. Press the **Enter** key to confirm editing.

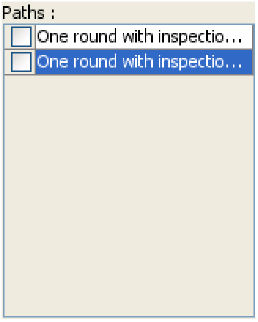
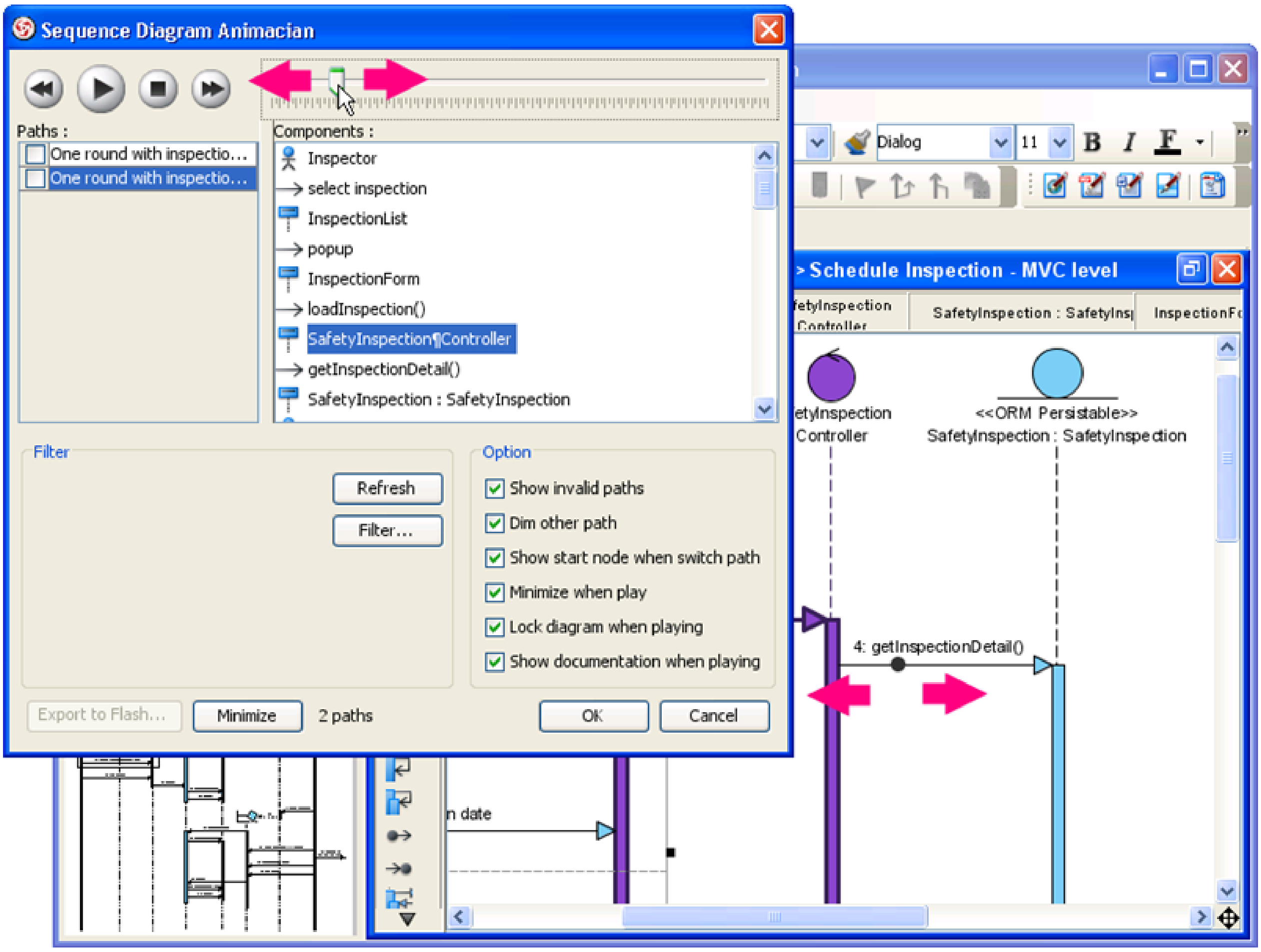


Figure 12-25 *Path named*

6. User can navigate a path manually by any of the ways below. Note that the shapes on a path will be highlighted in the diagram accordingly.

- Drag the slider back and forth.



Sequence Diagram Animacian

Paths :
- □ One round with inspectio...
- □ One round with inspectio...

Components :
- Inspector
- select inspection
- InspectionList
- popup
- InspectionForm
- loadInspection()
- SafetyInspection¶Controller
- getInspectionDetail()
- SafetyInspection : SafetyInspection

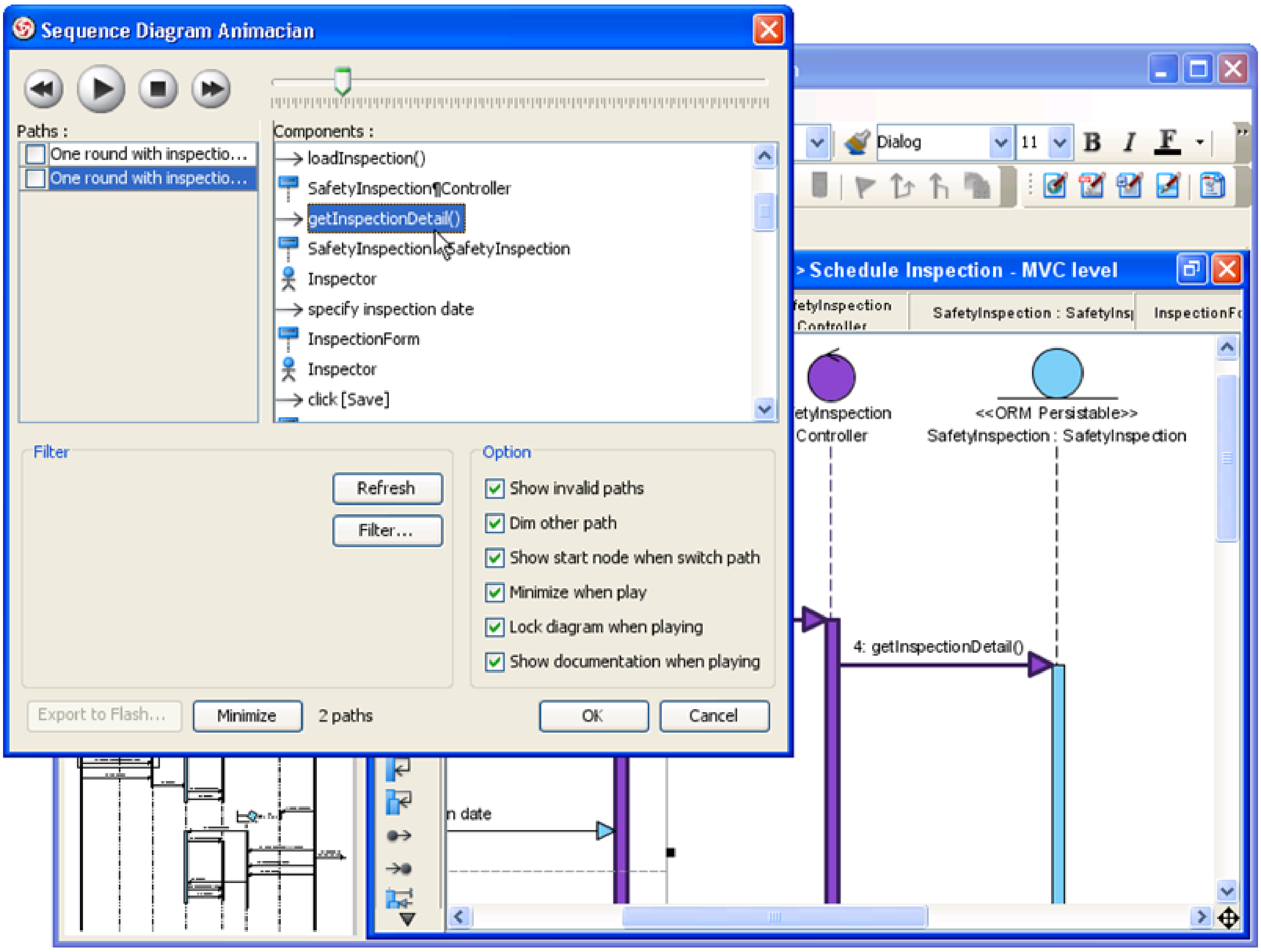
Option
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

2 paths

Filter
Refresh
Filter...
Export to Flash...
Minimize
OK
Cancel
Dialog
11
Schedule Inspection - MVC level
SafetyInspection Controller
<<ORM Persistable>> SafetyInspection : SafetyInspection
SafetyInspection : SafetyIns
InspectionF
4: getInspectionDetail()
n date

Figure 12-26 Dragging a slider back and forth

- Select component in the **Components** list.



Sequence Diagram Animacian

Paths :
- □ One round with inspectio...
- □ One round with inspectio...

Components :
- loadInspection()
- SafetyInspection¶Controller
- getInspectionDetail()
- SafetyInspection : SafetyInspection
- Inspector
- specify inspection date
- InspectionForm
- Inspector
- click [Save]

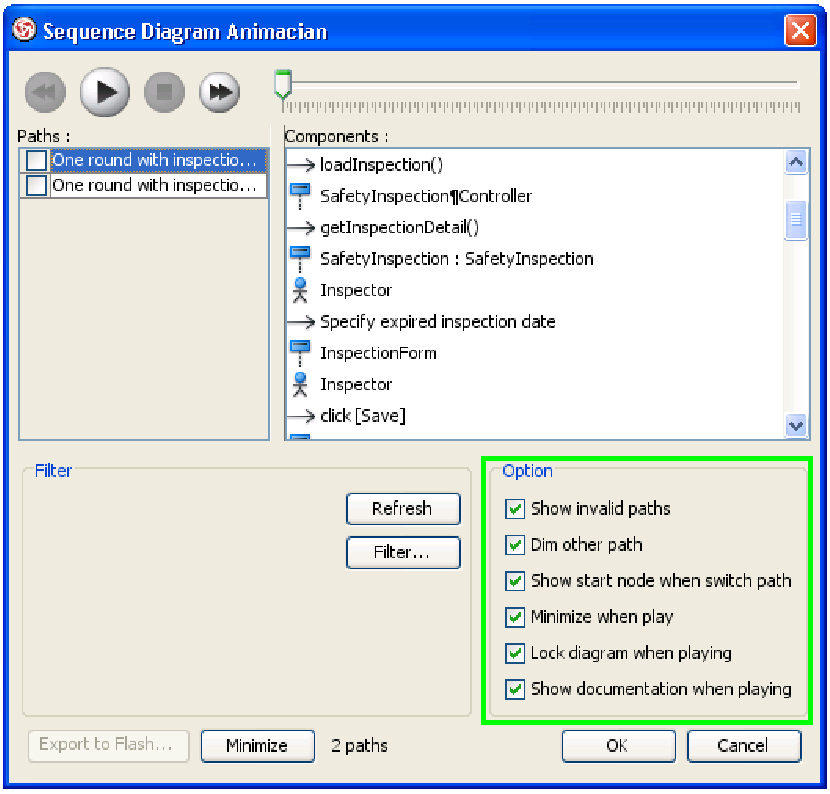
Option
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

2 paths

Filter
Refresh
Filter...
Export to Flash...
Minimize
OK
Cancel
Dialog
11
Schedule Inspection - MVC level
SafetyInspection Controller
<<ORM Persistable>> SafetyInspection : SafetyInspection
SafetyInspection : SafetyIns
InspectionF
4: getInspectionDetail()
n date

Figure 12-27 Select a Component in Components list

7. Configure the animation in the **Option** pane.



Paths :

□ One round with inspectio...
□ One round with inspectio...

Components :

loadInspection()
SafetyInspection¶Controller
getInspectionDetail()
SafetyInspection : SafetyInspection
Inspector
Specify expired inspection date
InspectionForm
Inspector
click [Save]

Filter

Option

☑ Show invalid paths
☑ Dim other path
☑ Show start node when switch path
☑ Minimize when play
☑ Lock diagram when playing
☑ Show documentation when playing

2 paths

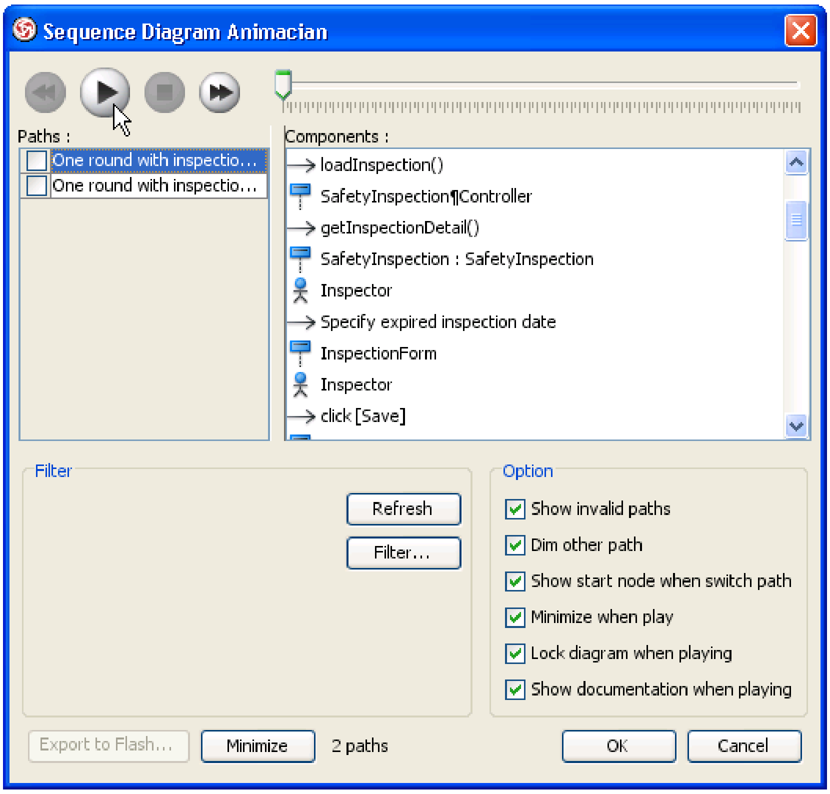
Figure 12-28 *Options for Animacian*

Here is a description of **Animacian** options:

Option	Description
Show invalid paths	List also the invalid, non play-able paths in the Paths list.
Dim other path	Dim the components that are not a part of the selected path.
Show start node when switch path	Jump to the first node of a chosen path upon selection, or keep staying at the current viewing region.
Minimize when play	Minimize this dialog box (Sequence Diagram Animacian) when playing an animation.
Lock diagram when playing	Avoid accidental editing when playing by locking the diagram.
Show documentation when playing	Show documentation of shape at the bottom right of diagram when playing animation.

Table 12-5 *Description of Animacian options*

8. When everything is ready, click the ▶ button to play the animation of the chosen path.



Paths :

- □ One round with inspectio...
- □ One round with inspectio...

Components :

- loadInspection()
- SafetyInspection¶Controller
- getInspectionDetail()
- SafetyInspection : SafetyInspection
- Inspector
- Specify expired inspection date
- InspectionForm
- Inspector
- click [Save]

Filter

Option

- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

2 paths

Figure 12-29 *Going to play a path*

9. When pressed, Animacian will be minimized to the bottom of diagram, with several buttons and a slider appearing on it.
The animation starts. A black dot will appear at beginning of path, and traverse through the path until the end. When the black dot is reaching a shape, the shape will be highlighted in purple gradually.

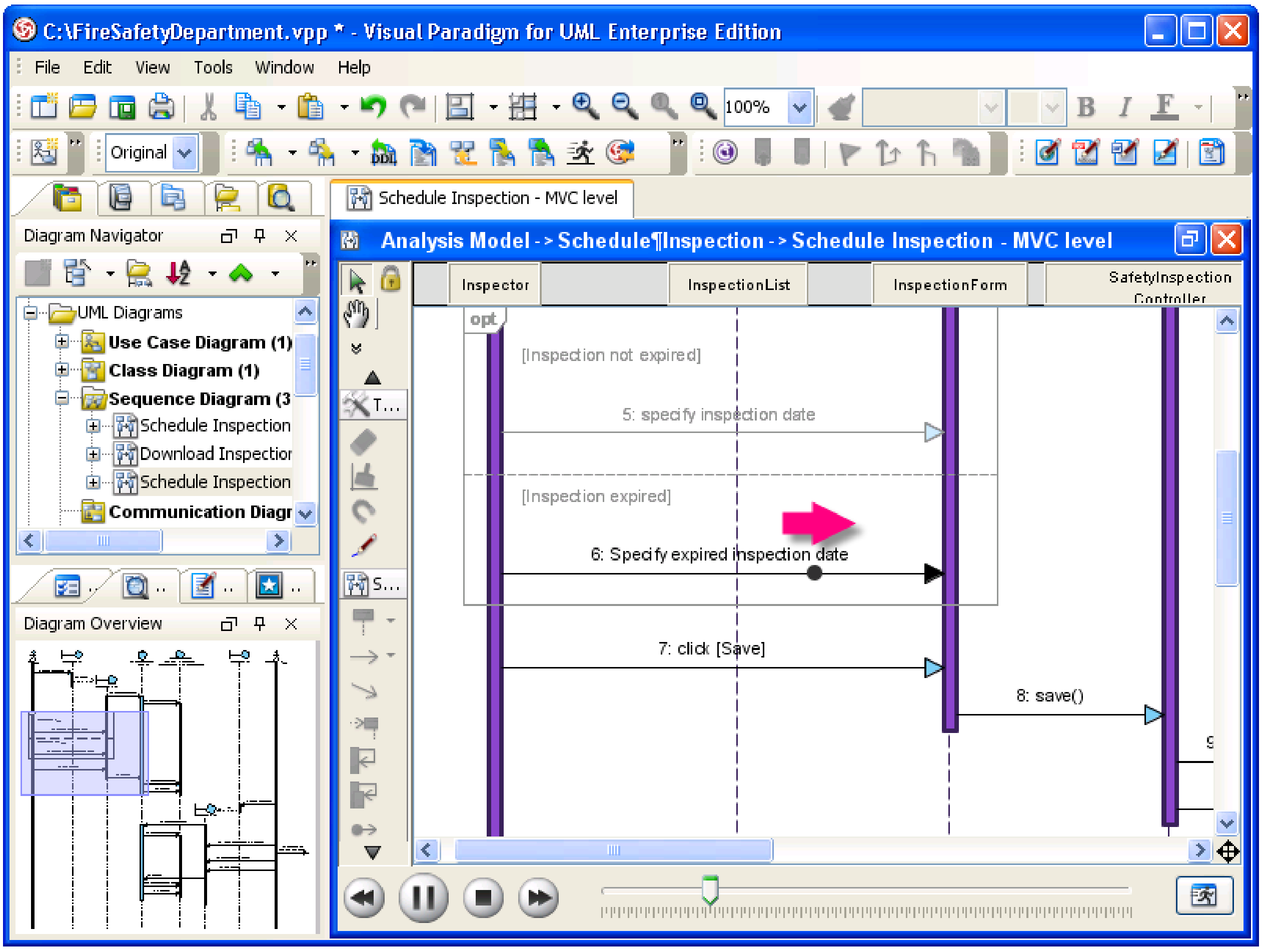


Figure 12-30 Animation is running

The minimized **Animacian** lets you control the flow of animation. Here is a description of the buttons.

Button	Description
⏪	Backward. Move one shape backward in the flow.
⏸	Pause. Temporary stop playing the movie. Can play again by pressing the ▶ button.
▶	Play. Play or continue to play the movie content.
⏩	Forward. Advance to the next shape in the flow.
⏹	Stop. Terminate the movie.
	Maximize **Animacian**.

Table 12-6 Description of Animacian bar

1. To launch **Animacian**, select **Tools > Animacian** from the main menu.

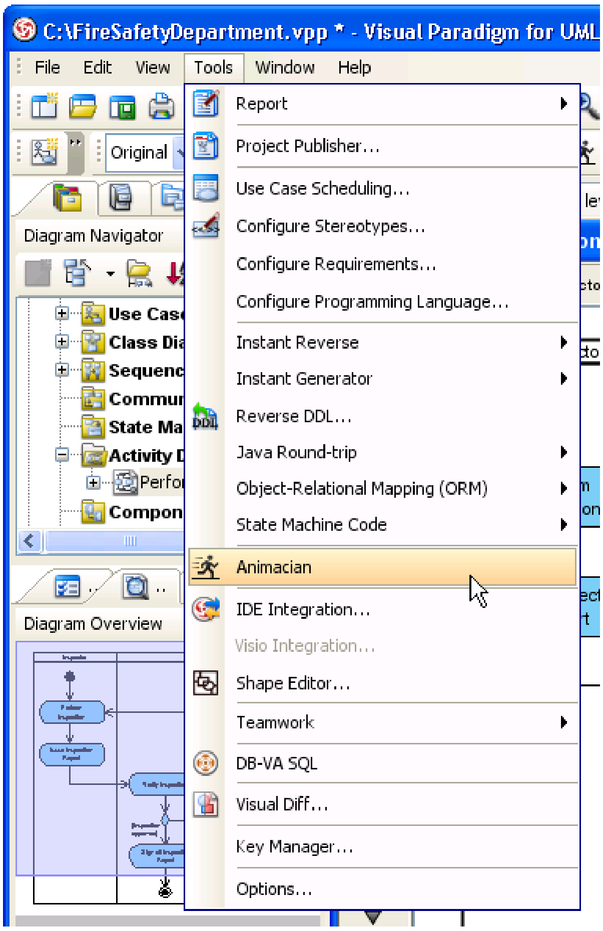
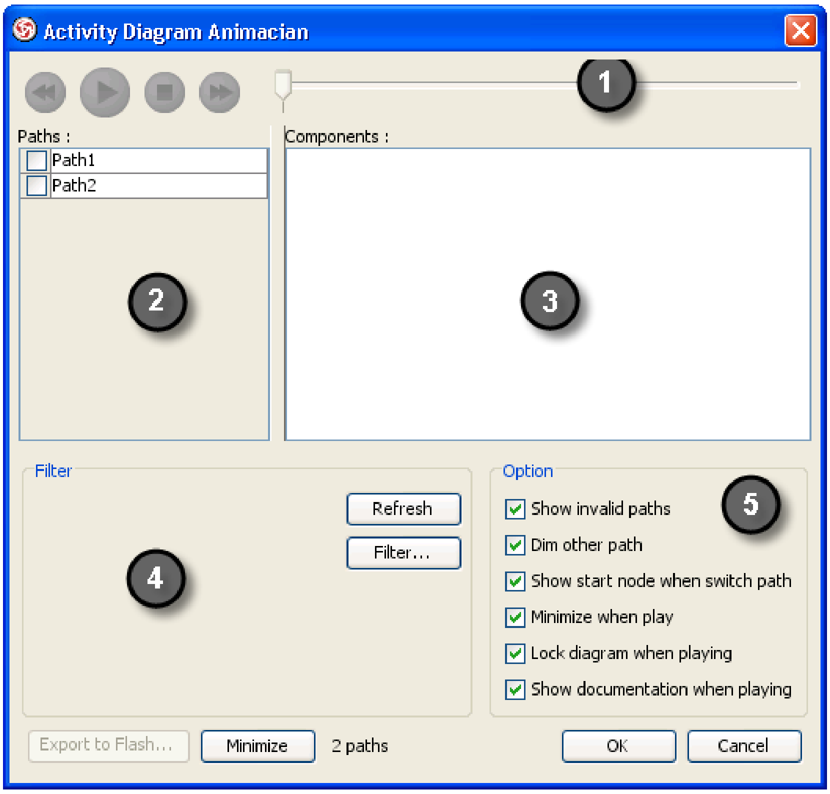


Figure 12-31 Opening Animacian through the main menu

NOTE: Animacian can also be started by any of the ways below:

- Right-click on the diagram background and select **Utilities > Animacian...** from the popup menu.
- Click on the Animacian button in the toolbar.

This starts Animacian. The **Animacian** dialog box is where you can select an execution path to play an animation..



□ Path1
□ Path2

☑ Show invalid paths
☑ Dim other path
☑ Show start node when switch path
☑ Minimize when play
☑ Lock diagram when playing
☑ Show documentation when playing

2 paths

Figure 12-32 The Animacian dialog box

Below is a description of different parts of dialog box.

Region	Description
1	The buttons and slider are for controlling the flow of animation. Once the Play button ▶ is pressed, the Animacian dialog box will

2. The **Paths** list lists the available animation paths of the diagram. Each path represents a possible way to go walk through the diagram. By default, paths are named as Path1, Path2, and so forth. It is recommended to give meaningful name to the paths for better clarification. To rename a path, move the mouse cursor over a path in the list.

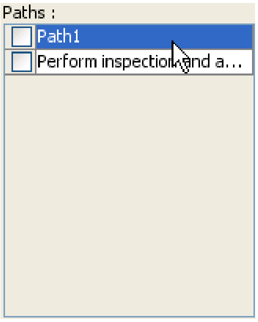


__Figure 12-33__ Going to rename a path

3. Double-click on the path to enter editing mode.

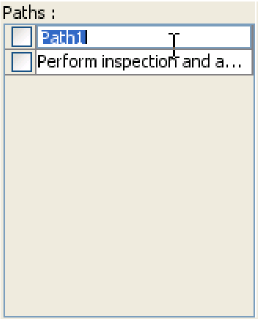


__Figure 12-34__ Entered editing mode of path name

4. Enter the name of path.

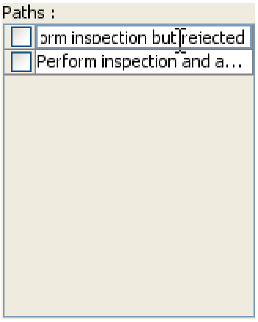


__Figure 12-35__ Enter name of path

5. Press the **Enter** key to confirm editing.

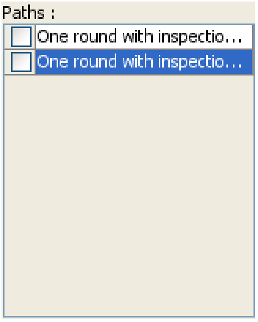
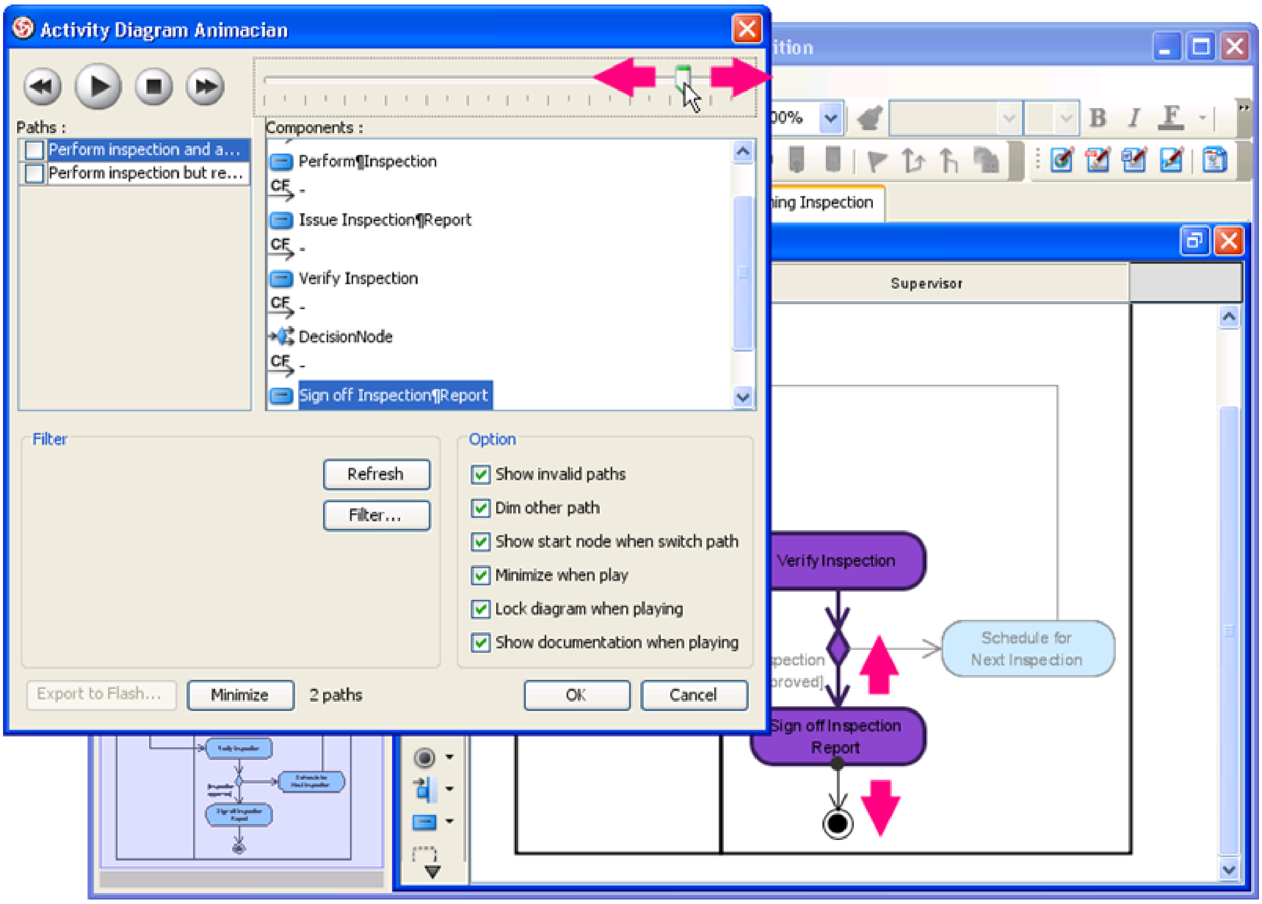


__Figure 12-36__ Path named

6. User can navigate a path manually by any of the ways below. Note that the shapes on a path will be highlighted in the diagram accordingly.

- Drag the slider back and forth.



Activity Diagram Animacian

Paths :
- □ Perform inspection and a...
- □ Perform inspection but re...

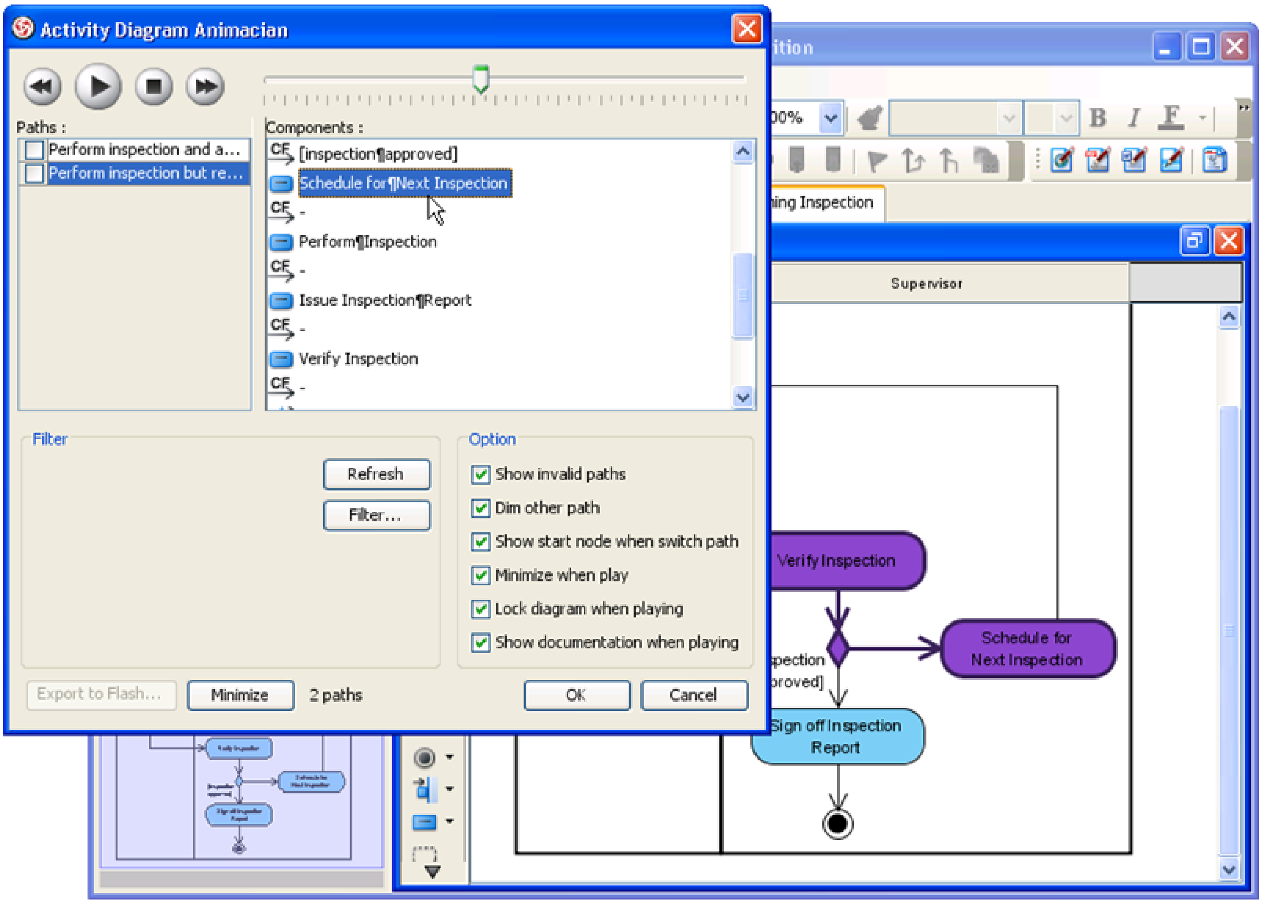
Components :
- Perform¶Inspection
- CF -
- Issue Inspection¶Report
- CF -
- Verify Inspection
- CF -
- DecisionNode
- CF -
- Sign off Inspection¶Report

Option
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

2 paths

Figure 12-37 *Dragging a slider back and forth*

- Select component in the **Components** list.



Activity Diagram Animacian

Paths :
- □ Perform inspection and a...
- □ Perform inspection but re...

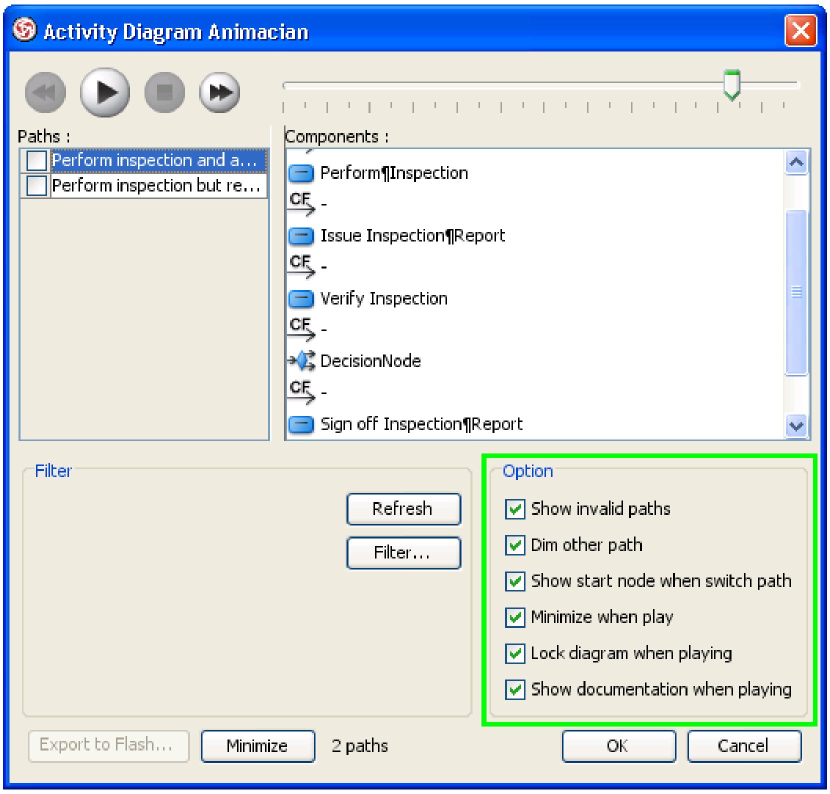
Components :
- CF [inspection¶approved]
- Schedule for¶Next Inspection
- CF -
- Perform¶Inspection
- CF -
- Issue Inspection¶Report
- CF -
- Verify Inspection
- CF -

Option
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

2 paths

Figure 12-38 *Select a Component in Components list*

7. Configure the animation in the **Option** pane.



Activity Diagram Animacian

Paths :

□ Perform inspection and a...
□ Perform inspection but re...

Components :

- Perform¶Inspection
- CF -
- Issue Inspection¶Report
- CF -
- Verify Inspection
- CF -
- DecisionNode
- CF -
- Sign off Inspection¶Report

Option

☑ Show invalid paths
☑ Dim other path
☑ Show start node when switch path
☑ Minimize when play
☑ Lock diagram when playing
☑ Show documentation when playing

2 paths

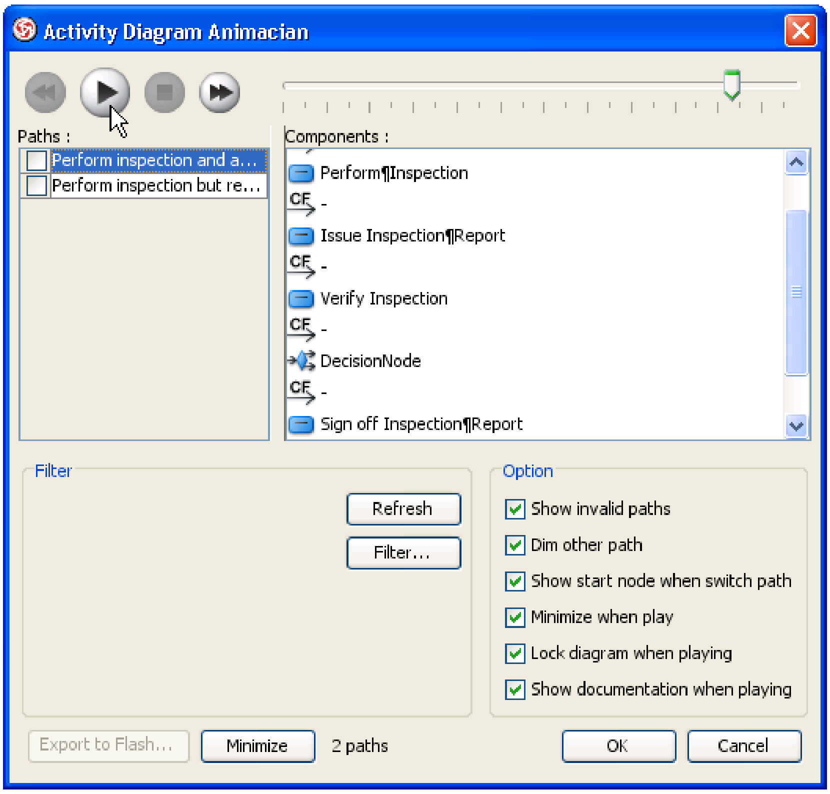
Figure 12-39 *Options for Animacian*

Here is a description of **Animacian** options:

Option	Description
Show invalid paths	List also the invalid, non play-able paths in the Paths list.
Dim other path	Dim the components that are not a part of the selected path.
Show start node when switch path	Jump to the first node of a chosen path upon selection, or keep staying at the current viewing region.
Minimize when play	Minimize this dialog box (Activity Diagram Animacian) when playing an animation.
Lock diagram when playing	Avoid accidental editing when playing by locking the diagram.
Show documentation when playing	Show documentation of shape at the bottom right of diagram when playing animation.

Table 12-8 *Description of Animacian options*

8. When everything is ready, click the ▶ button to play the animation of the chosen path.



Paths :

- □ Perform inspection and a...
- □ Perform inspection but re...

Components :

- Perform¶Inspection
- CF -
- Issue Inspection¶Report
- CF -
- Verify Inspection
- CF -
- DecisionNode
- CF -
- Sign off Inspection¶Report

Filter

Option

- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

2 paths

Figure 12-40 *Going to play a path*

9. When pressed, Animacian will be minimized to the bottom of diagram, with several buttons and a slider appearing on it.
The animation starts. A black dot will appear at beginning of path, and traverse through the path until the end. When the black dot is reaching a shape, the shape will be highlighted in purple gradually.

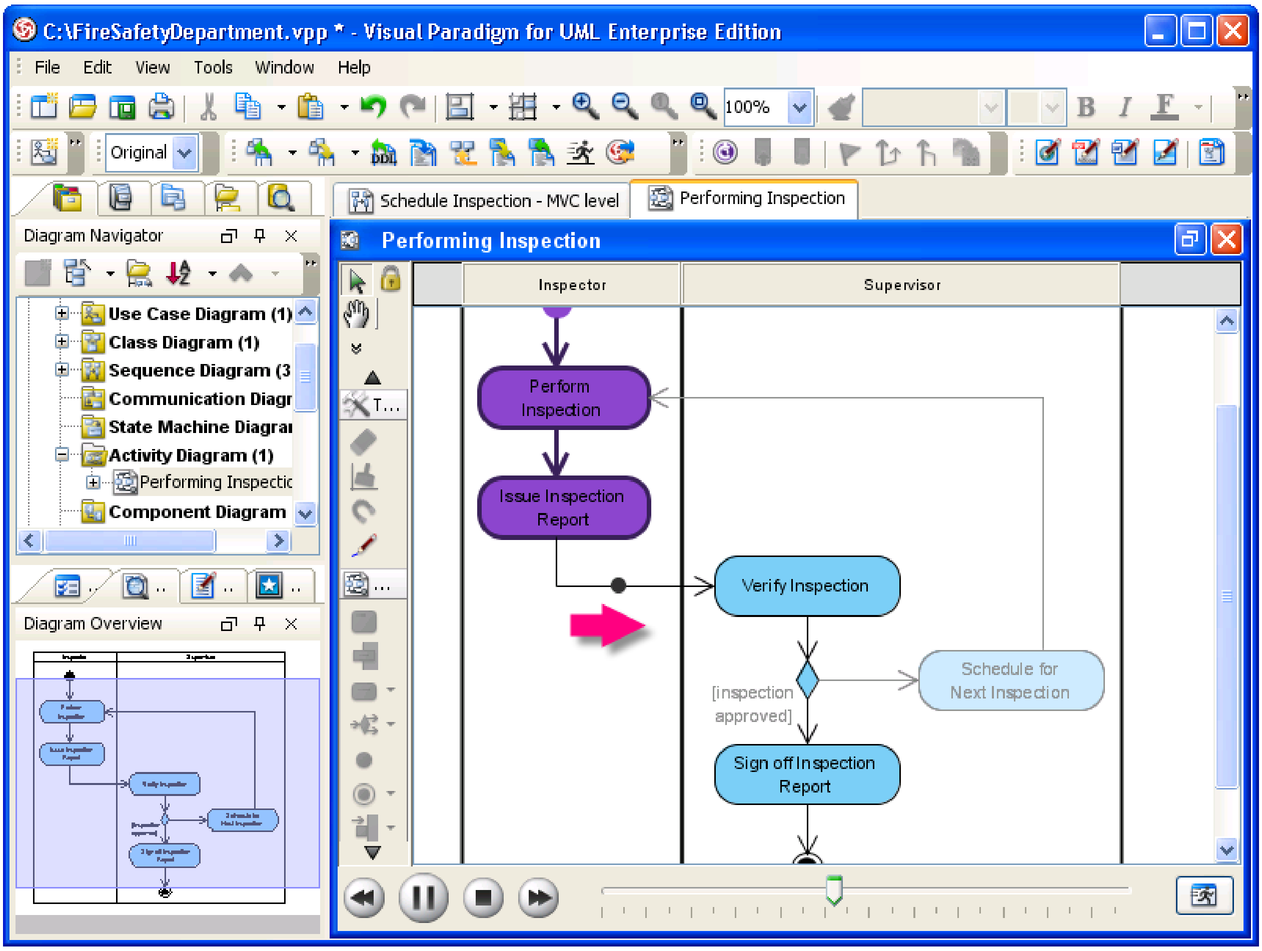


Figure 12-41 Animation is running

The minimized **Animacian** lets you control the flow of animation. Here is a description of the buttons.

Button	Description
⏪	Backward. Move one shape backward in the flow.
⏸	Pause. Temporary stop playing the movie. Can play again by pressing the ▶ button.
▶	Play. Play or continue to play the movie content.
⏩	Forward. Advance to the next shape in the flow.
■	Stop. Terminate the movie.
	Maximize **Animacian**.

Table 12-9 Description of Animacian bar

Export Animation to Adobe Flash

Exporting Flash Movie from Animacian

1. To launch **Animacian**, select **Tools > Animacian** from the main menu.

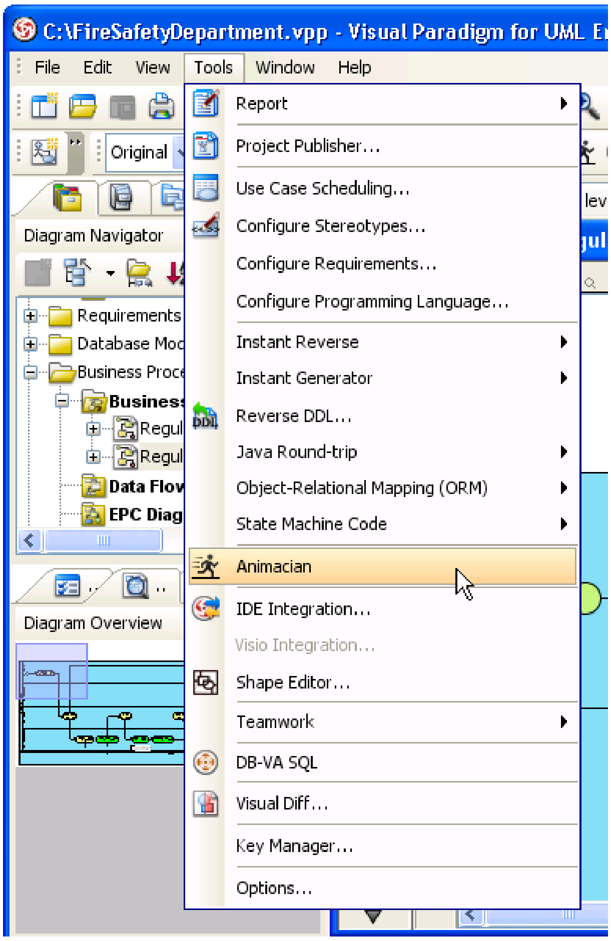
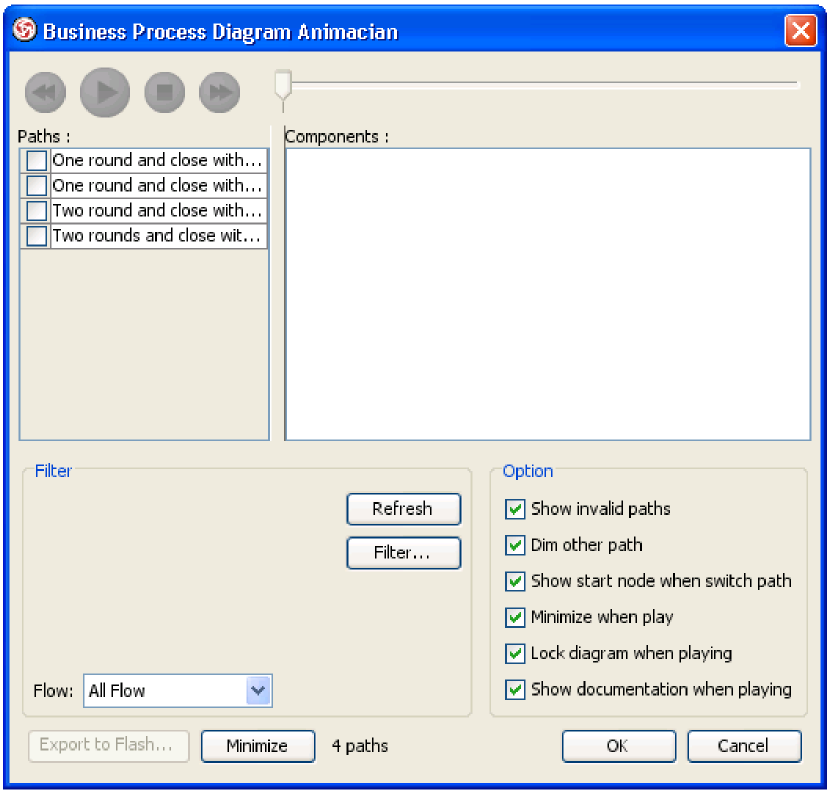


Figure 12-42 *Opening Animacian through the main menu*

NOTE: Animacian can also be started by any of the ways below:

- Right-click on the diagram background and select **Utilities > Animacian...** from the popup menu.
- Click on the Animacian button in the toolbar.

This starts Animacian. The **Animacian** dialog box is where you can select an execution path to play an animation.



Paths :

- □ One round and close with...
- □ One round and close with...
- □ Two round and close with...
- □ Two rounds and close wit...

Option

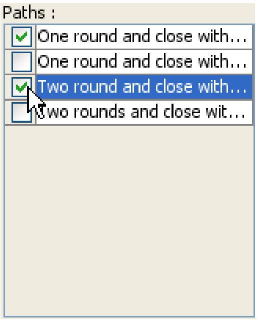
- ☑ Show invalid paths
- ☑ Dim other path
- ☑ Show start node when switch path
- ☑ Minimize when play
- ☑ Lock diagram when playing
- ☑ Show documentation when playing

Flow: All Flow

4 paths

Figure 12-43 *The Animacian dialog box*

2. From the **Paths** list, select the execution paths to export as Flash movie.



Paths :

☑ One round and close with...

□ One round and close with...

☑ Two round and close with...

□ Two rounds and close wit...

Figure 12-44 *Select paths to animate*

3. Click the **Export to Flash...** button.

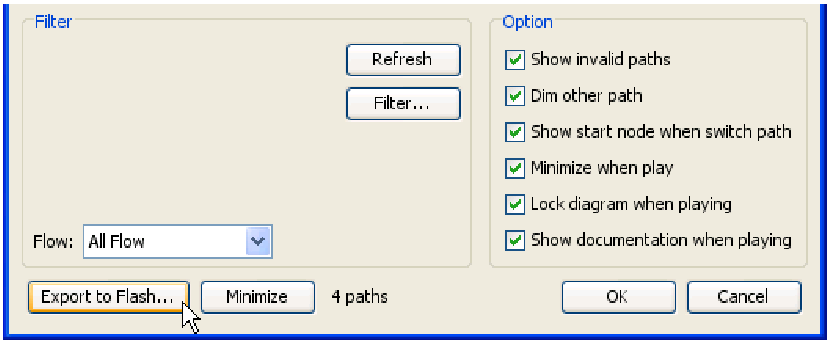
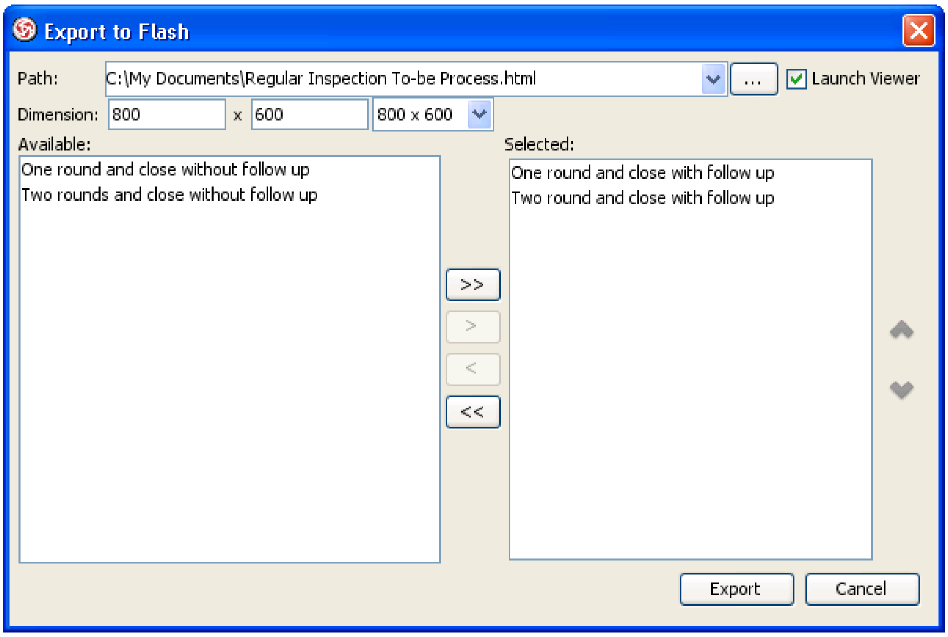


Figure 12-45 *Clicking the Export to Flash button*

This shows the **Export to Flash** dialog box.


Export to Flash

Path: C:\My Documents\Regular Inspection To-be Process.html ☑ Launch Viewer

Dimension: 800 x 600 800 x 600

Available:
One round and close without follow up
Two rounds and close without follow up

Selected:
One round and close with follow up
Two round and close with follow up

Figure 12-46 *The Export to Flash dialog box*

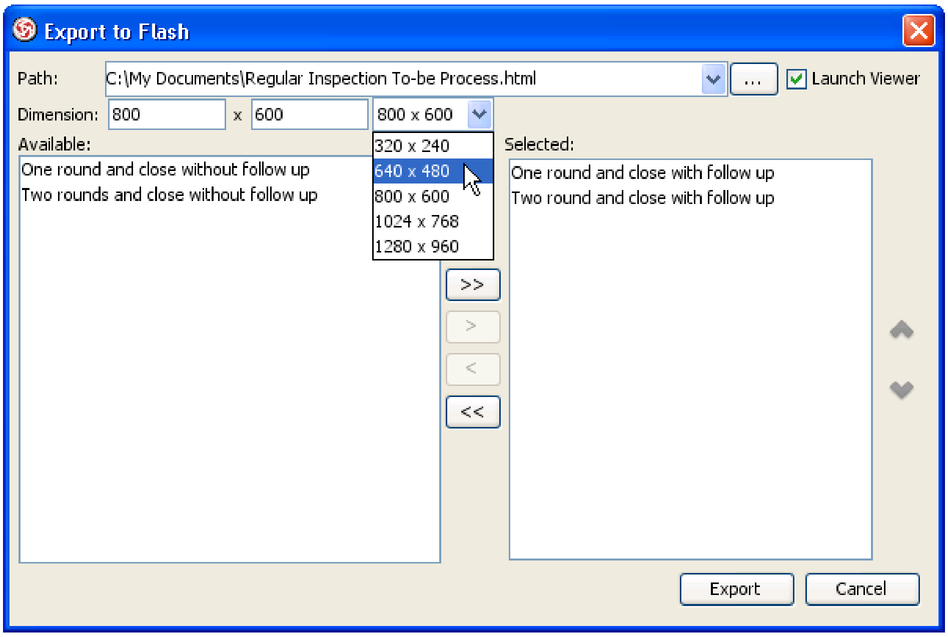
Here is a description of the **Export to Flash** dialog box.

Region	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.

Table 12-10 *Description of the Export to Flash dialog box*

4. 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.

5. 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.



Export to Flash

Path: C:\My Documents\Regular Inspection To-be Process.html

☑ Launch Viewer

Dimension: 800 x 600 800 x 600

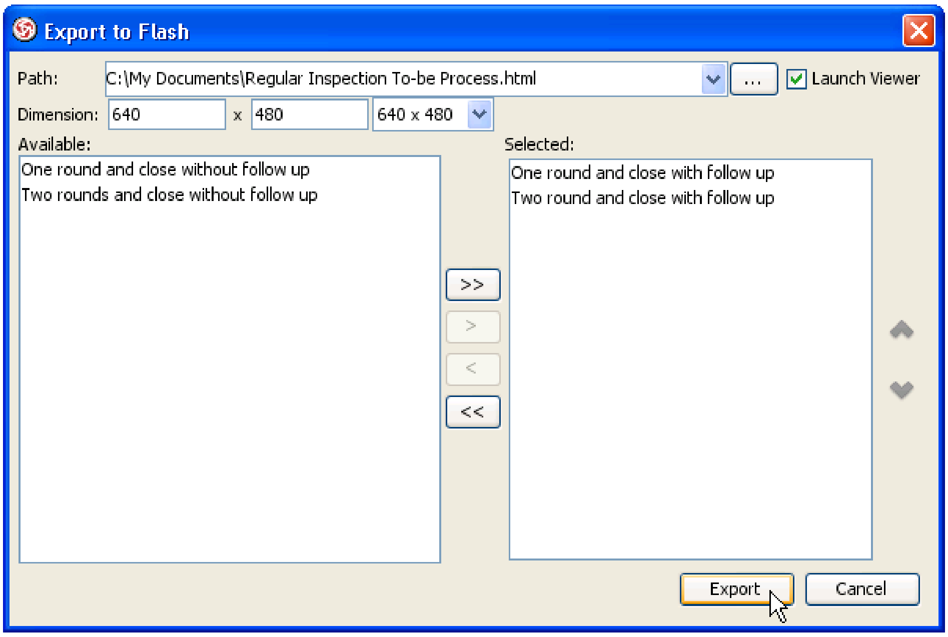
320 x 240
640 x 480
800 x 600
1024 x 768
1280 x 960

Available:
One round and close without follow up
Two rounds and close without follow up

Selected:
One round and close with follow up
Two round and close with follow up

Figure 12-47 *Changing the Flash movie dimension*

6. Click **Export**.



Export to Flash

Path: C:\My Documents\Regular Inspection To-be Process.html

☑ Launch Viewer

Dimension: 640 x 480 640 x 480

Available:
One round and close without follow up
Two rounds and close without follow up

Selected:
One round and close with follow up
Two round and close with follow up

Figure 12-48 *Clicking the Export button*

7. Open the HTML file in the web browser to play the movie.

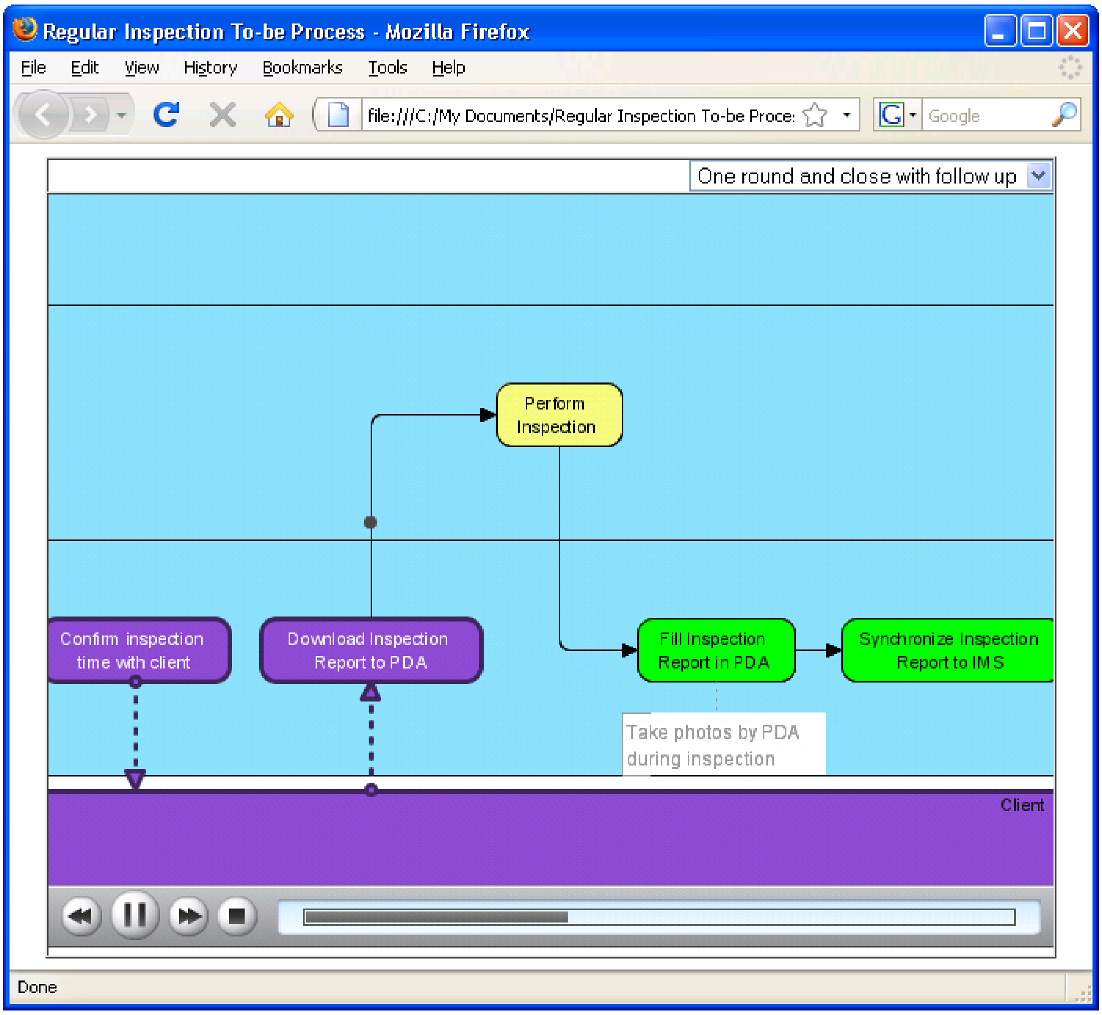

Regular Inspection To-be Process - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
file:///C:/My Documents/Regular Inspection To-be Proce:
Google
One round and close with follow up
Perform Inspection
Confirm inspection time with client
Download Inspection Report to PDA
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS
Take photos by PDA during inspection
Client
Done

Figure 12-49 *Animation is playing*

8. 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.

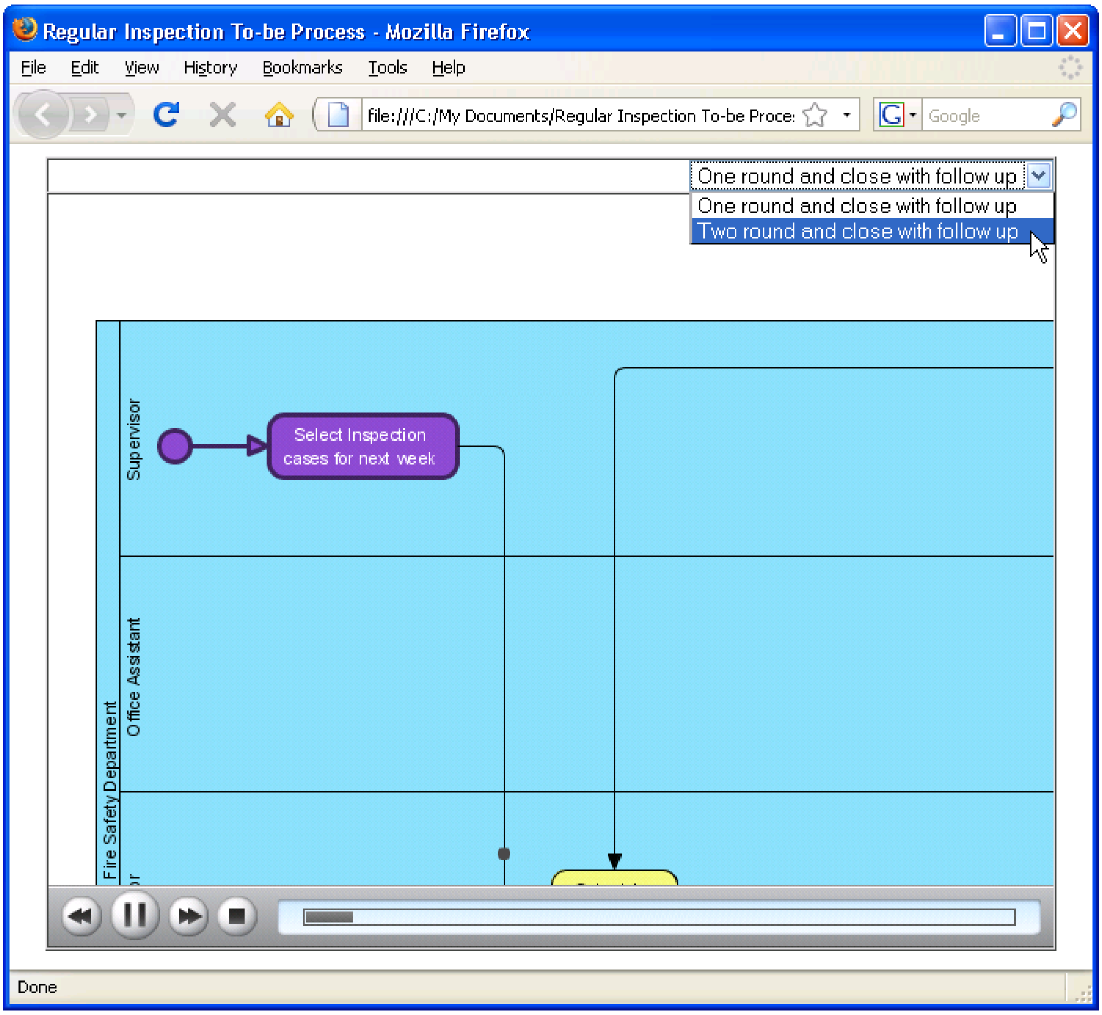

Regular Inspection To-be Process - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
file:///C:/My Documents/Regular Inspection To-be Proce:
Google
One round and close with follow up
One round and close with follow up
Two round and close with follow up
Fire Safety Department
Supervisor
Office Assistant
Select Inspection cases for next week
Done

Figure 12-50 To play another path

Navigation in Exported Movie

You can control the flow of movie by clicking on the buttons at the bottom of the movie or by pressing shortcut keys. Here is a list of buttons.

Button	Description
⏪	Backward. Move one shape backward in the flow.
⏸	Pause. Temporary stop playing the movie. Can play again by pressing the ▶ button.
▶	Play. Play or continue to play the movie content.
⏩	Forward. Advance to the next shape in the flow.
⏹	Stop. Terminate the movie.

Table 12-11 Buttons in exporting Flash movie

You can also click on the slider to move the animation to a desired position.

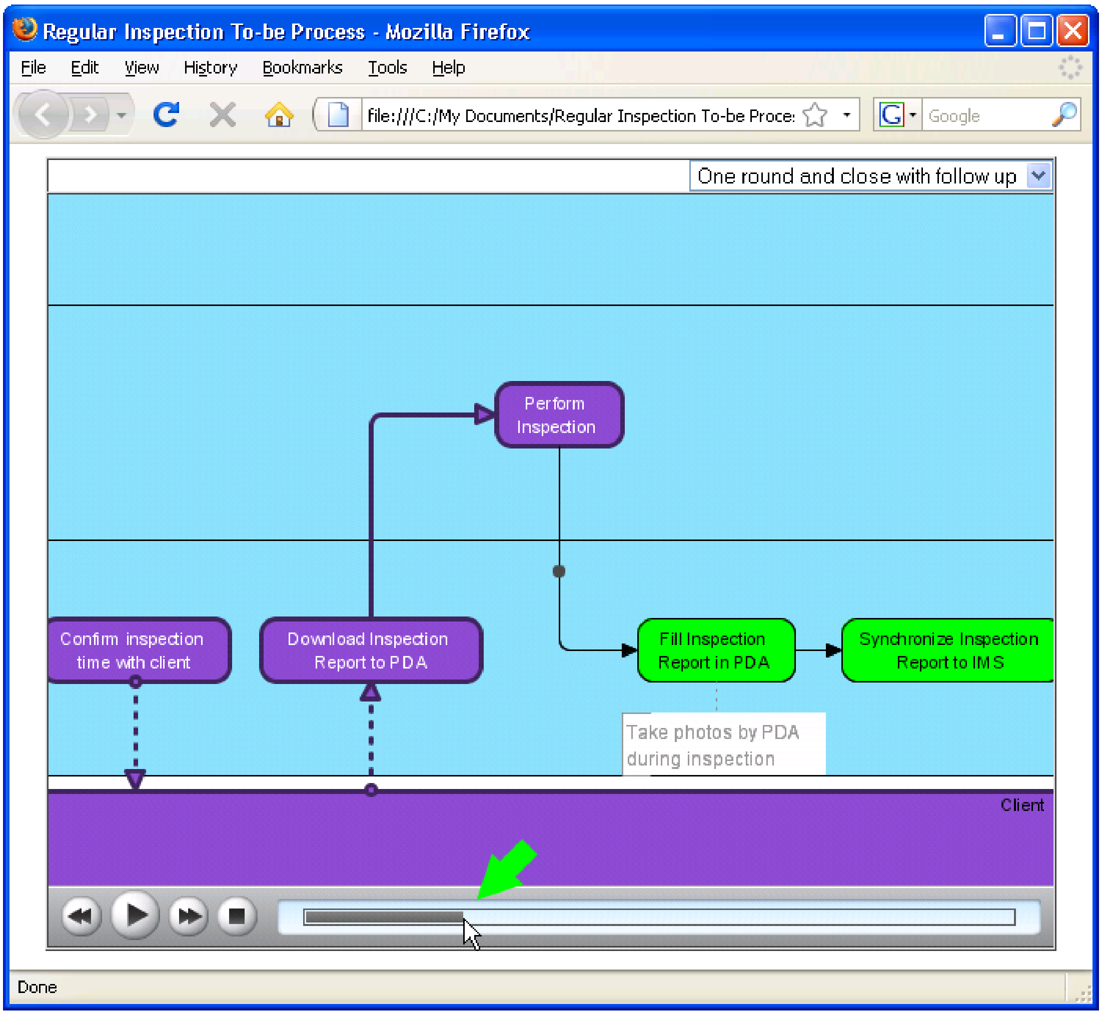

Regular Inspection To-be Process - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/My Documents/Regular Inspection To-be Proce:
Google
One round and close with follow up
Perform Inspection
Confirm inspection time with client
Download Inspection Report to PDA
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS
Take photos by PDA during inspection
Client
Done

Figure 12-51 *Navigate in the Flash movie by dragging slider*

Drawing Use Case Diagrams

Let's see how you can draw use case diagram in Visual Paradigm for UML through an example.

It is recommended to group related diagrams and model elements in a Model, this could improve the performance when save and load the project.

Switching to Model Explorer

To switch to Model Explorer, select menu **View** > **Panes** > **Model Explorer**.

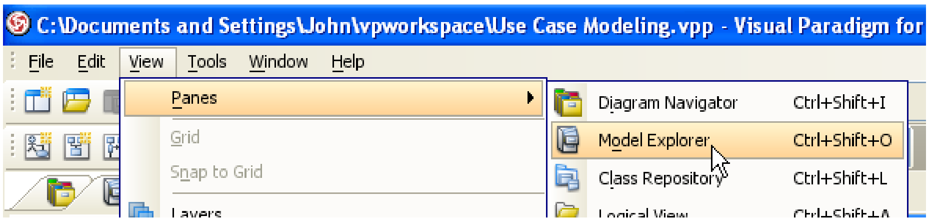

C:\Documents and Settings\John\vpworkspace\Use Case Modeling.vpp - Visual Paradigm for
File
Edit
View
Tools
Window
Help
Panes
Grid
Snap to Grid
Layers
Diagram Navigator Ctrl+Shift+I
Model Explorer Ctrl+Shift+O
Class Repository Ctrl+Shift+L
Logical View Ctrl+Shift+A

Figure 1-1 *Switch to Model Explorer*

Creating "Use Case Model"

Right-click on empty space of Model Explorer and select **Model** > **New Model...** from the popup menu.

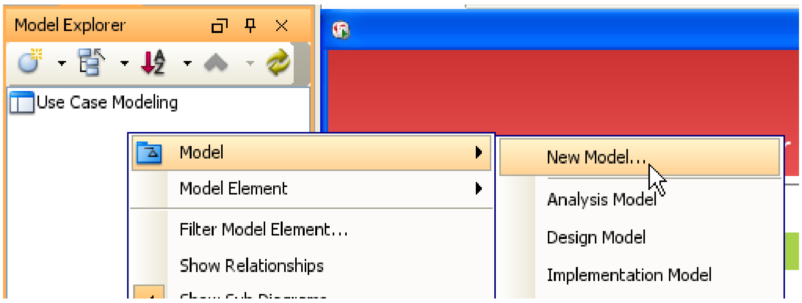


Figure 1-2 *Create new Model*

When the **Model Specification** dialog box appears, rename the Model to *Use Case Model* and click **OK**.

Creating Use Case Diagram

Right-click on the Model and select **Diagram** > **UML Diagrams** > **Use Case Diagram**.

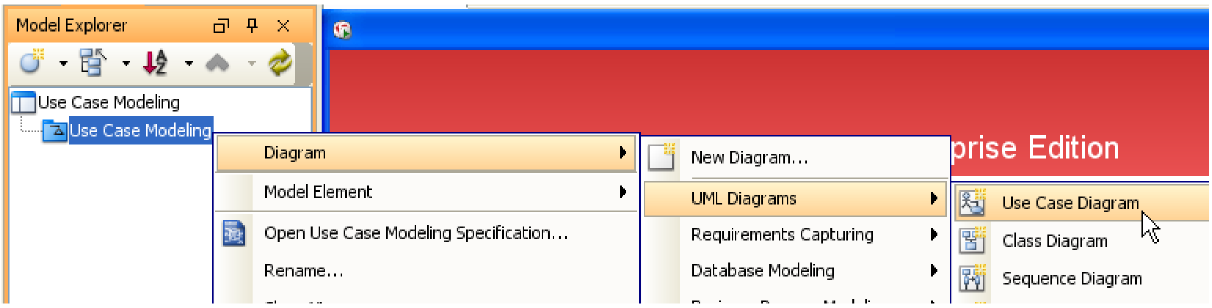


Figure 1-3 *Create use case diagram*

Rename the diagram to *Regular Inspection Use Case Diagram*.

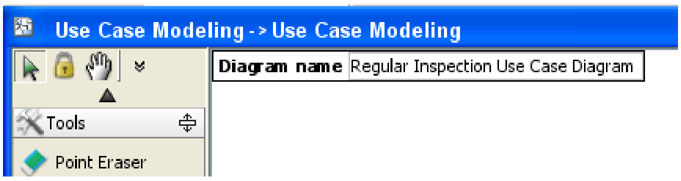


Diagram name Regular Inspection Use Case Diagram

Figure 1-4 *Rename diagram*

Drawing System

To draw a System, click **System** on the diagram toolbar and then click on the diagram.

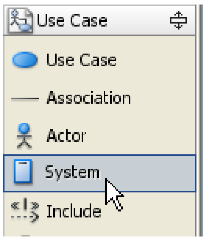


Figure 1-5 *Create System*

Name the System *Inspection Management System (IMS)*.

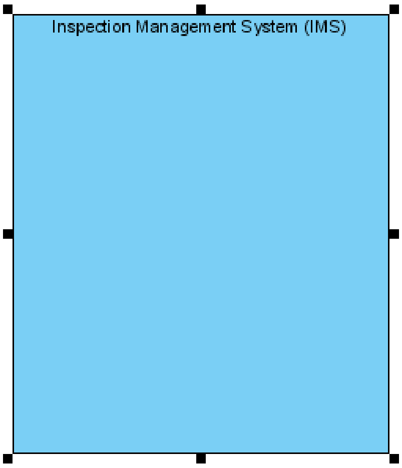


Figure 1-6 *Rename System*

Drawing Actors

To draw a Actor, click **Actor** on the diagram toolbar and then click on the diagram. Note that an Actor should be placed outside the System.

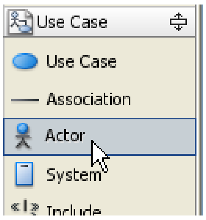


Figure 1-7 *Create Actor*

Name the Actor *Inspector*.

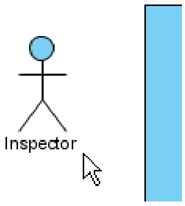


Figure 1-8 *Inspector Actor*

Create three more Actors namely **Inspector Assistant**, **Office Assistant** and **Supervisor**.

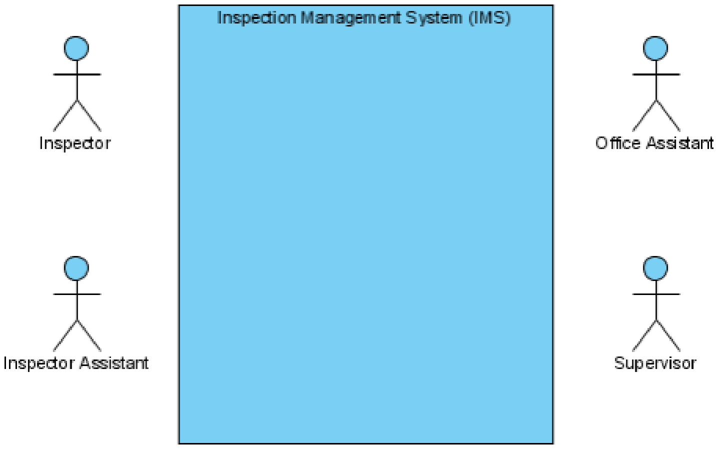


__Figure 1-9__ Actors Inspector Assistant, Office Assistant and Supervisor

Drawing Use Cases

Mouse over the Actor **Inspector**, click on the resource **Association -> Use Case** and drag.

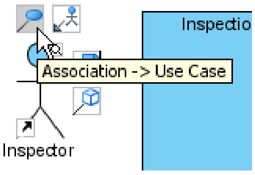


__Figure 1-10__ Create Use Case with resource

Move the mouse over the System and then release the mouse button. A Use Case together with an Association are created. Name the Use Case **Review and touch up Inspection Report.**

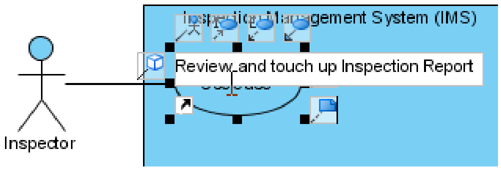

Inspection Management System (IMS)
Review and touch up Inspection Report
Inspector

__Figure 1-11__ Rename Use Case

Follow the same steps to create other Use Cases until the diagram looks like the picture below.

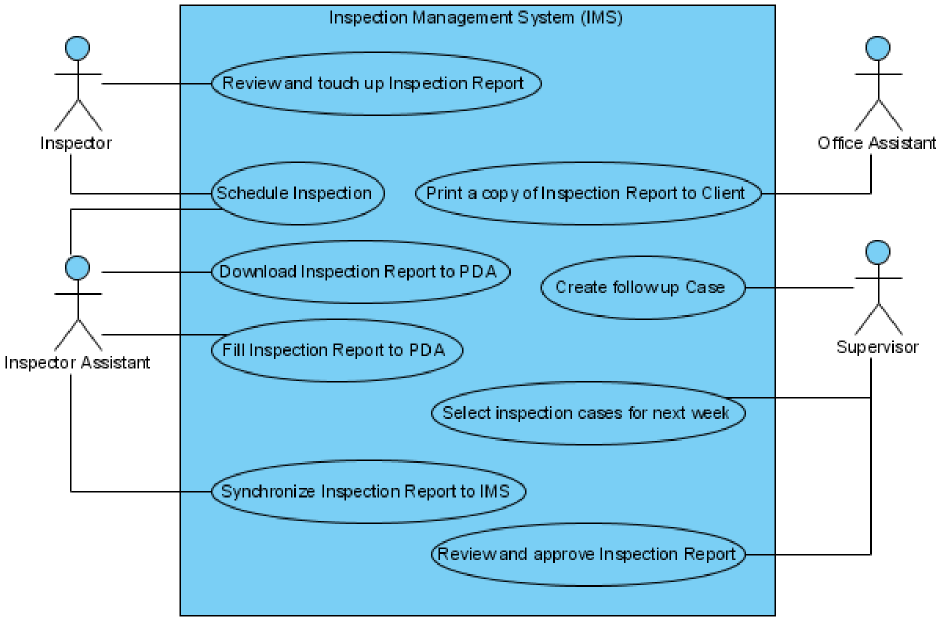

Inspection Management System (IMS)
Inspector
Office Assistant
Inspector Assistant
Supervisor
Review and touch up Inspection Report
Schedule Inspection
Print a copy of Inspection Report to Client
Download Inspection Report to PDA
Create follow up Case
Fill Inspection Report to PDA
Select inspection cases for next week
Synchronize Inspection Report to IMS
Review and approve Inspection Report

Figure 1-12 *Use Cases created*

Line Wrapping Use Case Name

There is a problem with the diagram that the Use Cases are too wide which make them look strange.

We can solve this by simply resizing the Use Case, and then the name will be line-wrapped automatically.

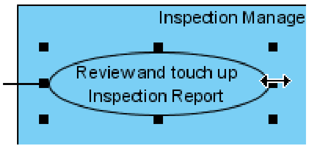


Figure 1-13 *Resize Use Case to wrap caption*

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

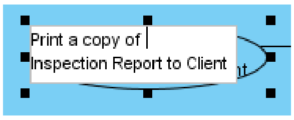


Figure 1-14 *Force new line by Alt + Enter*

Resize Use Cases until the diagram looks like the picture below.

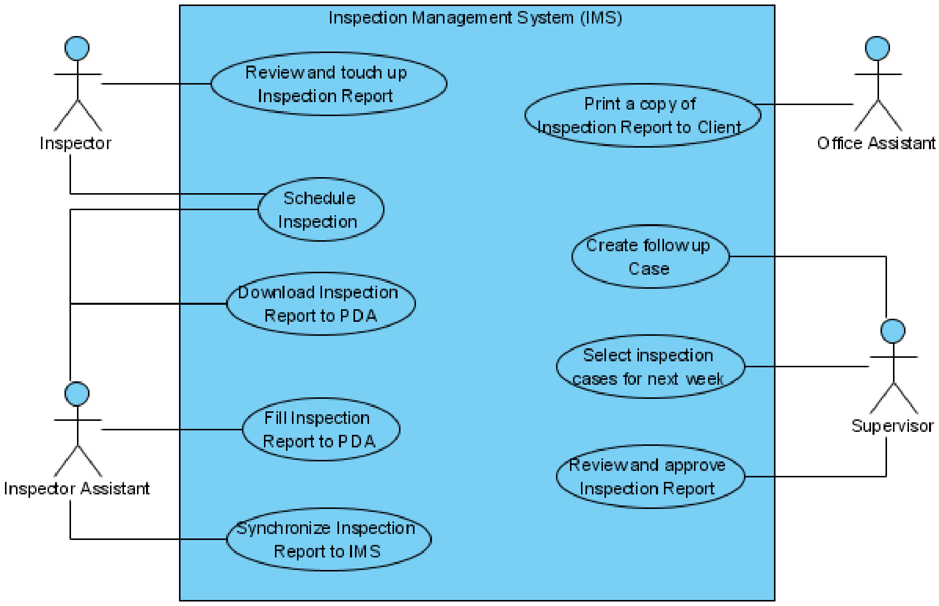

Inspection Management System (IMS)
Inspector
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select inspection cases for next week
Supervisor
Inspector Assistant
Fill Inspection Report to PDA
Review and approve Inspection Report
Synchronize Inspection Report to IMS

Figure 1-15 *Resized Use Cases*

Drawing <<Extend>> Relationship

To create an Extend relationship, mouse over a Use Case (**Review and approve Inspection Report** in this example), click the resource **Extend -> Use Case** and drag.

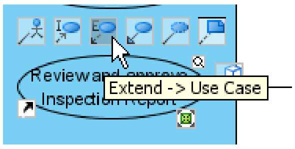


Figure 1-16 *Create Extend relationship*

Move the mouse to empty space of the System and then release the mouse button. A Use Case together with an Extend relationship is created. Name the Use Case *Re-submit Inspection Report .*

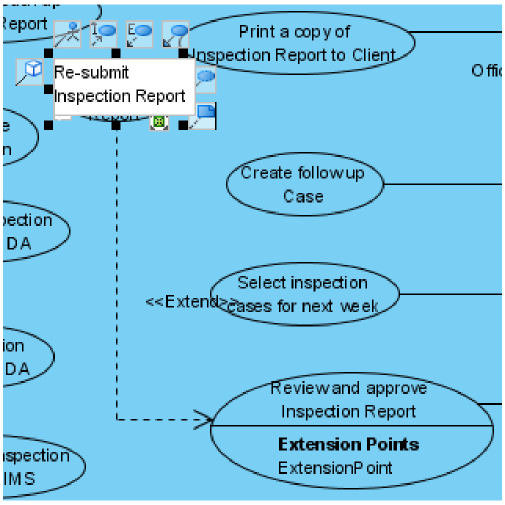

Re-submit Inspection Report
Print a copy of Inspection Report to Client
Offi
Create follow up Case
<<Extend>>
Select inspection cases for next week
Review and approve Inspection Report
Extension Points
ExtensionPoint

Figure 1-17 *Extend relationship created*

An extension point is created automatically for the Use Case **Review and approve Inspection Report**. Double-click it and rename it to *Inspection not completed.*

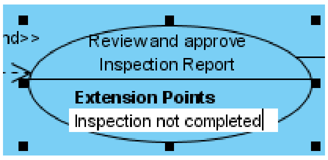


Figure 1-18 *Edit extension point*

Drawing <<Include>> Relationship

To create an Include relationship, mouse over a Use Case (**Review and touch up Inspection Report** in this example), click the resource **Include -> Use Case** and drag.

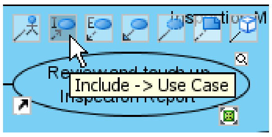


Figure 1-19 *Create Include relationship*

Move the mouse to empty space of the System and then release the mouse button. A Use Case together with an Include relationship is created. Name the Use Case *Login* **.**

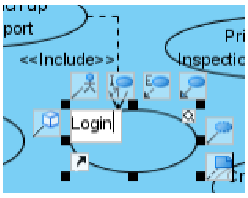


Figure 1-20 *Include relationship created*

Structuring Use Case with Package

When there are many Use Cases, it would be nice to organize them with Package.

Click **Package** on the diagram toolbar (located in the **Common** category).

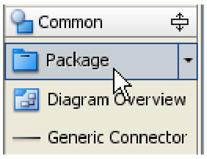


Figure 1-21 *Create a Package*

Click an drag the mouse to create a Package that surrounds the Use Cases related to the Actor **Supervisor** as shown in the picture below.

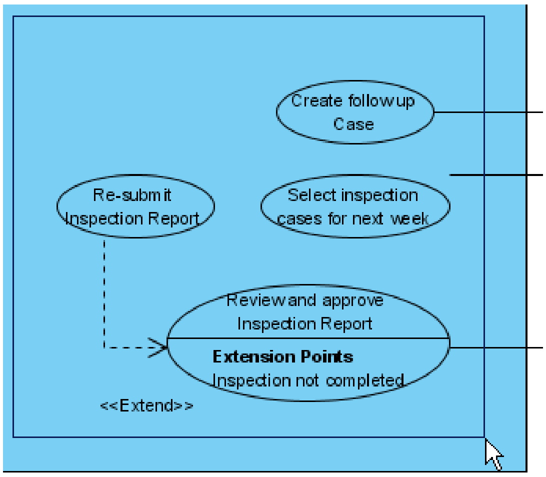

Create follow up Case
Re-submit Inspection Report
Select inspection cases for next week
Review and approve Inspection Report
Extension Points
Inspection not completed
<<Extend>>

Figure 1-22 *Surround Use Cases with Package*

Name the Package *supervisor*.

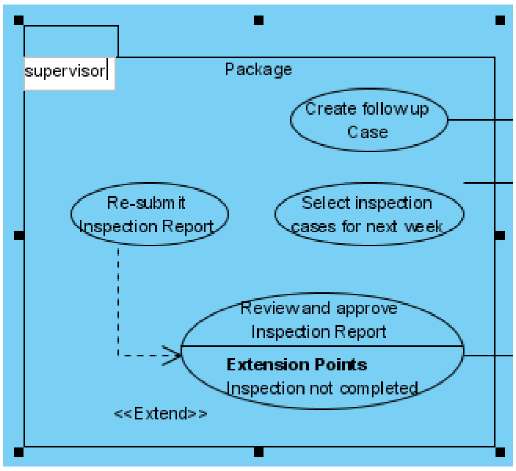

supervisor
Package
Create follow up Case
Re-submit Inspection Report
Select inspection cases for next week
Review and approve Inspection Report
Extension Points
Inspection not completed
<<Extend>>

Figure 1-23 *Rename Package*

Writing Use Case Description

Filling in the use case description is very important in use case modeling. Use case description includes descriptions of the use case, preconditions, post conditions, flow of event and whatever which is important in modeling the user goal.

Opening Use Case Details

To write use case description for a Use Case, you need to open use case details. To do this, right-click on the Use Case and select **Use Case Details...** from the popup menu.

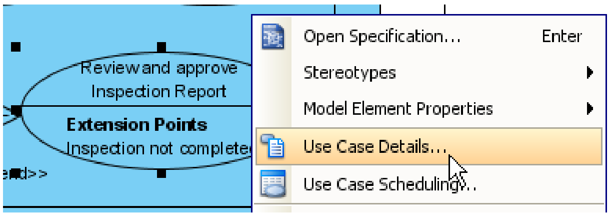
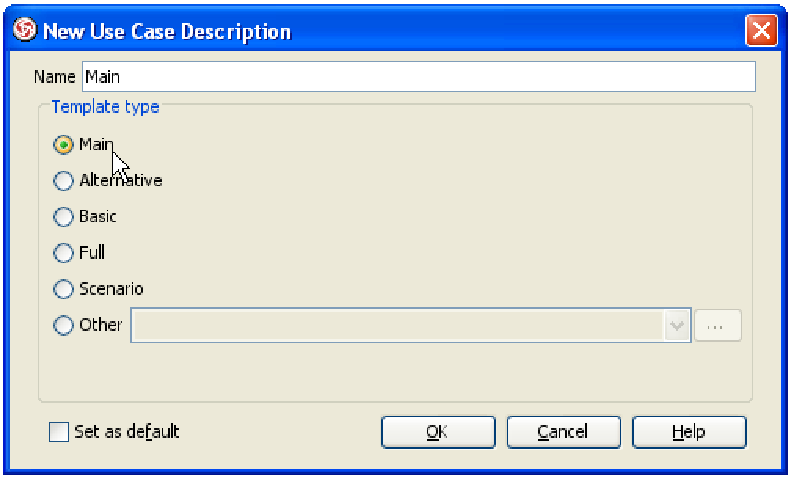


Figure 1-24 *Open use case details*

Selecting Use Case Description Template

The **New Use Case Description** dialog box appears. Enter name and select a template to use.



Name Main

Template type

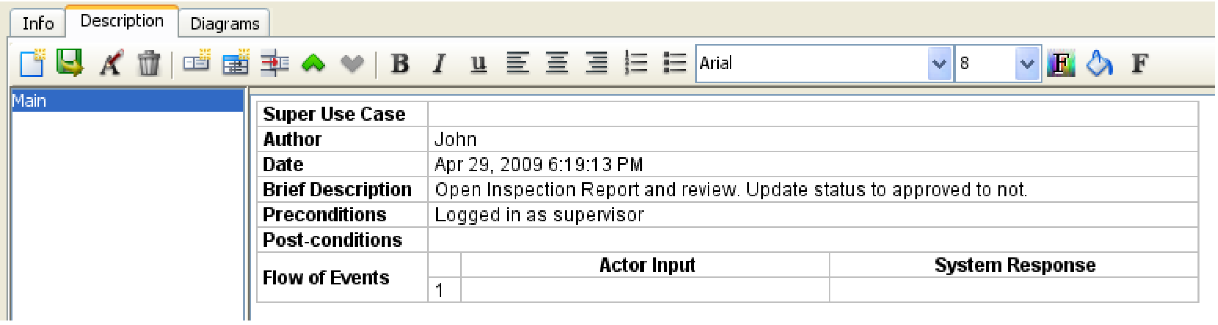
● Main
○ Alternative
○ Basic
○ Full
○ Scenario
○ Other

□ Set as default

Figure 1-25 *Select template*

Editing Use Case Description

To edit use case description, select the **Description** page. Fill in fields like **Super Use Case, Brief Description** where applicable.

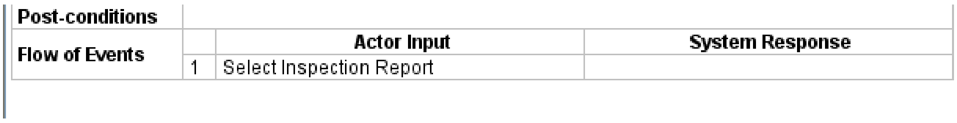


Super Use Case			
Author	John		
Date	Apr 29, 2009 6:19:13 PM		
Brief Description	Open Inspection Report and review. Update status to approved to not.		
Preconditions	Logged in as supervisor		
Post-conditions			
Flow of Events		**Actor Input**	**System Response**
	1		

Figure 1-26 *Edit use case description*

Entering Flow of Events

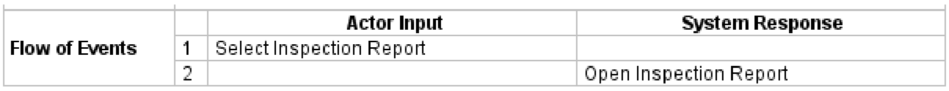
Flow of events are entered as a sequence of actor inputs and system responses alternately. First we enter **Actor Input**.



Post-conditions			
Flow of Events		**Actor Input**	**System Response**
	1	Select Inspection Report	

Figure 1-27 *Enter Actor Input*

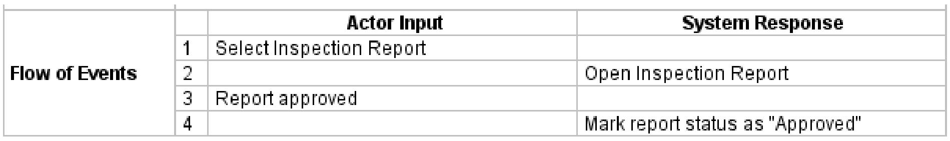
Press **Ctrl** + **Enter**, a new row is created. Enter **System Response**.



		Actor Input	System Response
Flow of Events	1	Select Inspection Report	
	2		Open Inspection Report

Figure 1-28*Enter System Response*

Continue to complete the flow of events.

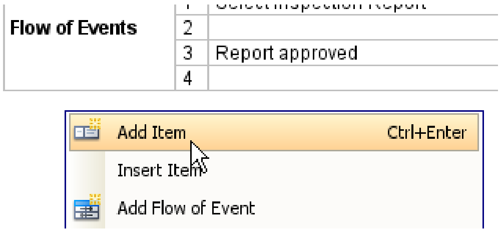


		Actor Input	System Response
Flow of Events	1	Select Inspection Report	
	2		Open Inspection Report
	3	Report approved	
	4		Mark report status as "Approved"

Figure 1-29 *Completed flow of events*

Creating Item

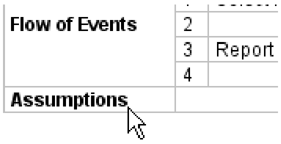
You can create a custom item in the use case description by right-clicking on it and select **Add Item**.



Flow of Events	1	Select Inspection Report
	2	
	3	Report approved
	4	

Figure 1-30 *Create item*

Enter the item name when prompted. A new item will be created.

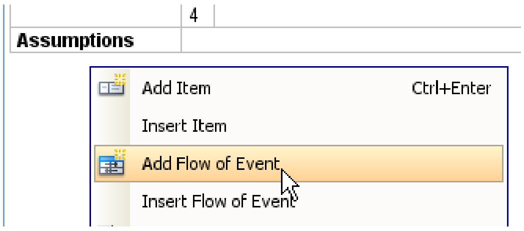


Flow of Events	2	
	3	Report
	4	
Assumptions		

Figure 1-31 *Item created*

Creating Multiple Flow of Events

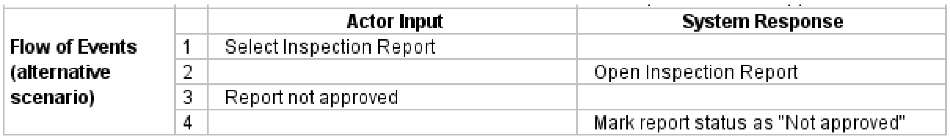
You can create multiple flow of events to model alternative scenarios of the Use Case. To do this, right-click on the use case description and select **Add Flow of Event** from the popup menu.



	4	
Assumptions		

Figure 1-32 *Add flow of events*

Enter name of the flow of events when prompted, and then continue to complete the flow of events.

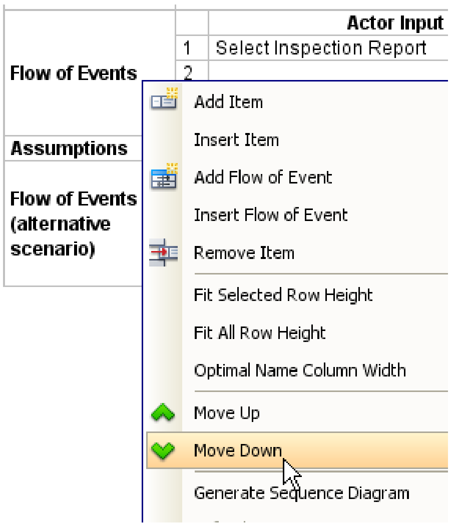


		Actor Input	System Response
Flow of Events (alternative scenario)	1	Select Inspection Report	
	2		Open Inspection Report
	3	Report not approved	
	4		Mark report status as "Not approved"

Figure 1-33 *Multiple flow of events*

Reordering Items

To reorder items, right-click on an item and select **Move Up** or **Move Down**.

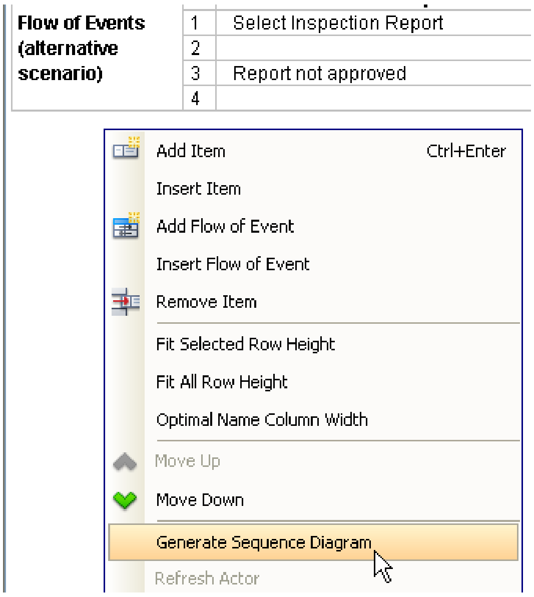


		Actor Input
Flow of Events	1	Select Inspection Report
	2	
Assumptions		
Flow of Events (alternative scenario)		

__Figure 1-34__ Reorder items

Generating Sequence Diagram from Flow of Events

You can generate sequence diagram from flow of events. To do this, right-click on the use case description and select **Generate Sequence Diagram** from the popup menu.



Flow of Events (alternative scenario)	1	Select Inspection Report
	2	
	3	Report not approved
	4	

__Figure 1-35__ Generate sequence diagram from flow of events

A sequence diagram will be generated, with each flow of events item generated as a combined fragment.

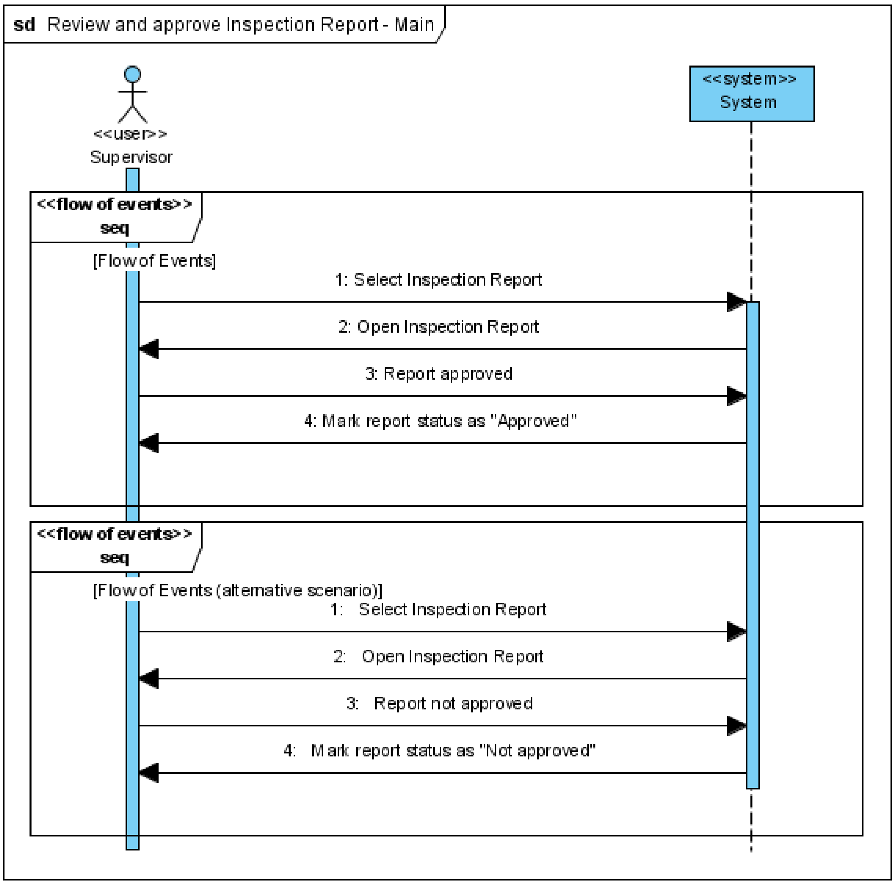

sd Review and approve Inspection Report - Main
<<user>>
Supervisor
<<system>>
System
<<flow of events>>
seq
[Flow of Events]
1: Select Inspection Report
2: Open Inspection Report
3: Report approved
4: Mark report status as "Approved"
<<flow of events>>
seq
[Flow of Events (alternative scenario)]
1: Select Inspection Report
2: Open Inspection Report
3: Report not approved
4: Mark report status as "Not approved"

Figure 1-36 *Generated sequence diagram*

Elaborating Use Case

A Use Case can be elaborated using sequence diagrams and activity diagrams to model its interactions and activity flows respectively.

Elaborating Use Case by Sequence Diagrams

To elaborate a Use Case by sequence diagrams, create a sequence diagram as sub diagram to the Use Case.

Creating Sub Diagram

Right-click on the Use Case and select **Sub Diagrams** > **Sequence Diagram** > **Create Sequence Diagram** from the popup menu.

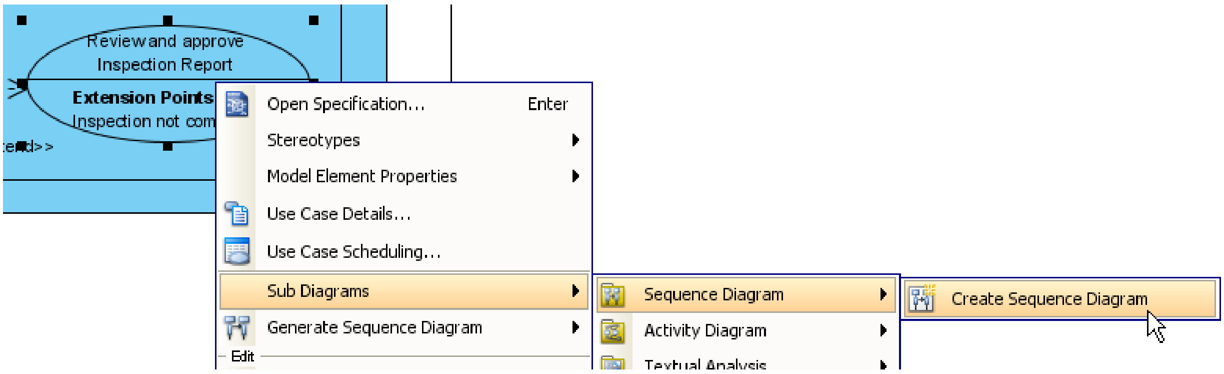


Figure 1-37 *Create sequence diagram as sub diagram*

Drawing Sequence Diagram

Draw the sequence diagram to show how user interacts with the system in the Use Case.

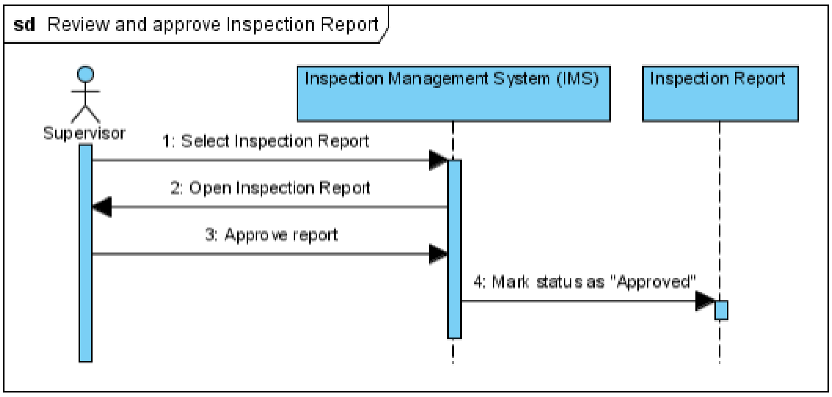

sd Review and approve Inspection Report
Supervisor
Inspection Management System (IMS)
Inspection Report
1: Select Inspection Report
2: Open Inspection Report
3: Approve report
4: Mark status as "Approved"

Figure 1-38 *Drawn sequence diagram*

Navigate to Parent Use Case

To navigate back to the parent Use Case, click the Use Case name (name that appears before diagram name) on the diagram title bar.

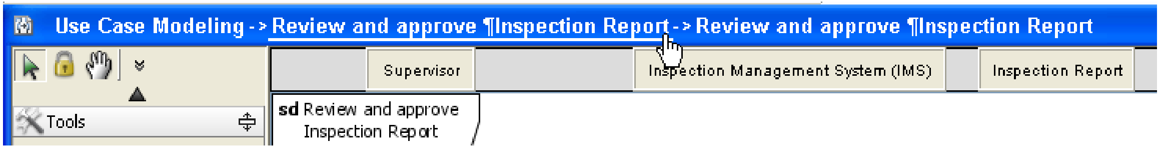

Use Case Modeling -> Review and approve ¶Inspection Report -> Review and approve ¶Inspection Report
Tools
Supervisor
Inspection Management System (IMS)
Inspection Report
sd Review and approve Inspection Report

Figure 1-39 *Navigate to parent Use Case*

Elaborating Use Case by Activity Diagrams

To elaborate a Use Case by activity diagrams, create an activity diagram as sub diagram to the Use Case.

Creating Sub Diagram

Right-click on the Use Case and select **Sub Diagrams** > **Activity Diagram** > **Create Activity Diagram** from the popup menu.

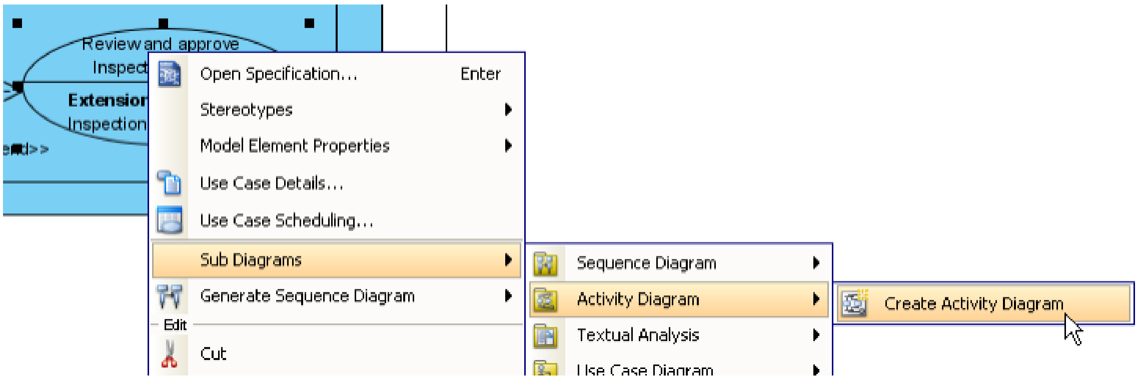


Figure 1-40 *Create activity diagram as sub diagram*

Drawing Activity Diagram

Draw the activity diagram to show the activity flows involved in the Use Case.

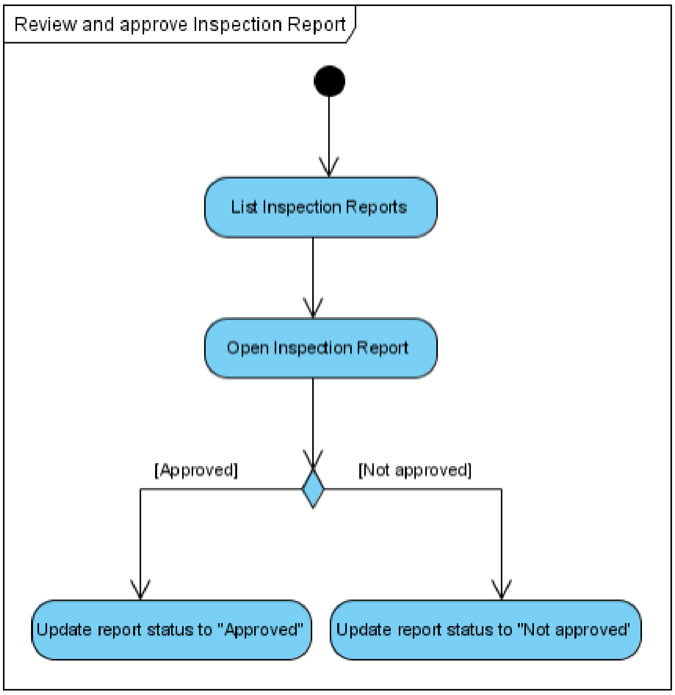

Review and approve Inspection Report
List Inspection Reports
Open Inspection Report
[Approved]
[Not approved]
Update report status to "Approved"
Update report status to "Not approved'

Figure 1-41 *Drawn activity diagram*

Navigate to Parent Use Case

To navigate back to the parent Use Case, click the Use Case name (name that appears before diagram name) on the diagram title bar.

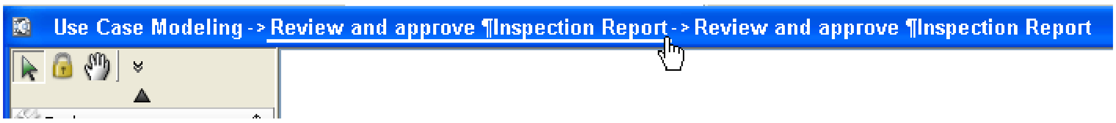

Use Case Modeling -> Review and approve ¶Inspection Report -> Review and approve ¶Inspection Report

Figure 1-42 *Navigate to parent Use Case*

Performing Textual Analysis

Using textual analysis, you can identify candidate objects from a problem statement, create model elements from the candidate objects, and finally draw diagram using the created model elements.

Creating Textual Analysis

To create textual analysis, select menu **File** > **New Diagram** > **Requirements Capturing** > **Textual Analysis**.

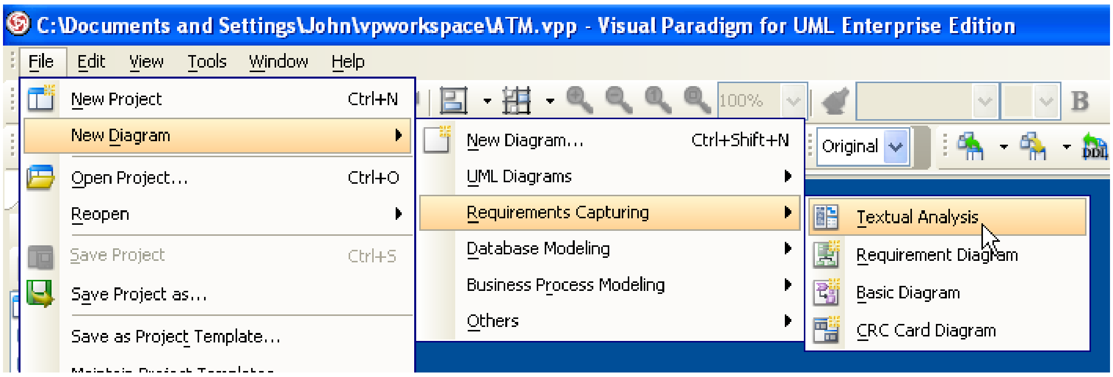


Figure 2-1 *Create textual analysis*

Entering Problem Statement

You can either enter problem statement by typing in the text area, or by importing a text file. To import a text file, click **Import Text File** on the toolbar.

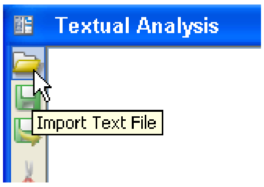
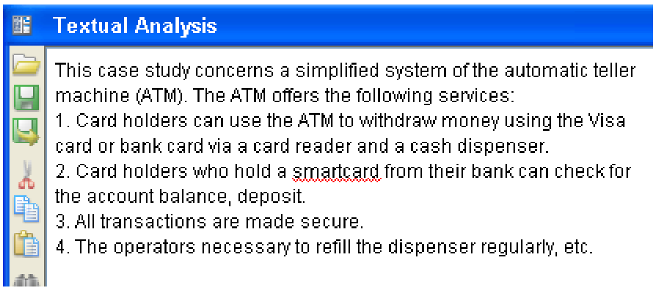


Figure 2-2 *Import text file*

When the file dialog box appears, select the text file to import. After that, the imported problem statement will appear in the text area.

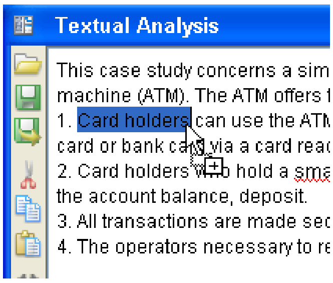


This case study concerns a simplified system of the automatic teller machine (ATM). The ATM offers the following services:
1. Card holders can use the ATM to withdraw money using the Visa card or bank card via a card reader and a cash dispenser.
2. Card holders who hold a smartcard from their bank can check for the account balance, deposit.
3. All transactions are made secure.
4. The operators necessary to refill the dispenser regularly, etc.

Figure 2-3 *Imported problem statement*

Creating Candidate Object by Drag and Drop

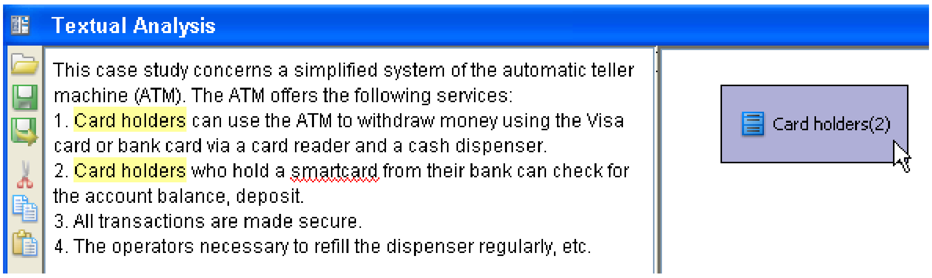
To create candidate object by drag and drop, select the words (**Card holders** in this example) in the text area and drag.



This case study concerns a sim
machine (ATM). The ATM offers t
1. Card holders can use the ATM
card or bank card via a card reac
2. Card holders who hold a sma
the account balance, deposit.
3. All transactions are made sec
4. The operators necessary to re

Figure 2-4 *Drag words in text area*

Move the mouse over the pane on the right of the text area and then release the mouse button. A candidate object is created. The occurrences of the candidate object name are highlighted in the text area.



This case study concerns a simplified system of the automatic teller machine (ATM). The ATM offers the following services:
1. Card holders can use the ATM to withdraw money using the Visa card or bank card via a card reader and a cash dispenser.
2. Card holders who hold a smartcard from their bank can check for the account balance, deposit.
3. All transactions are made secure.
4. The operators necessary to refill the dispenser regularly, etc.

Figure 2-5 Candidate object created

Candidate object created by drag and drop is set to "Class" type by default. However "Card holders" should be Actors. To change the candidate object type to Actor, right-click on the candidate object and select **Actor** from the popup menu.

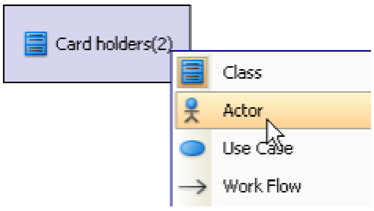


Figure 2-6 Change candidate object type to Actor

Creating Candidate Object by Popup Menu

To create candidate object (e.g. of type "Use Case") by popup menu, select the words (e.g. **withdraw money**) in the text area, right-click on the selection and select **Add text as Use Case** from the popup menu.

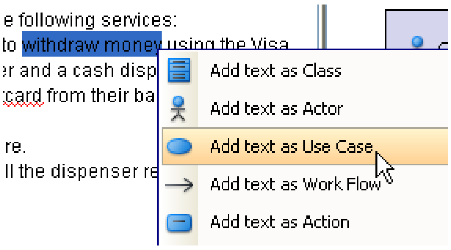


Figure 2-7 Change candidate object type to Actor

A candidate object "withdraw money" of type "Use Case" is created.

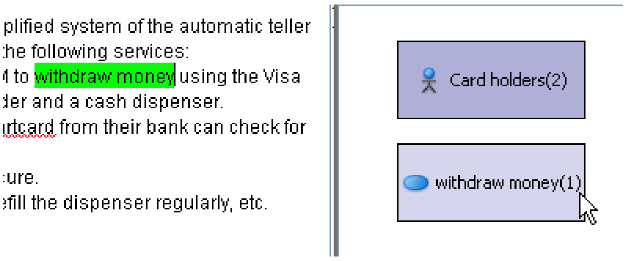
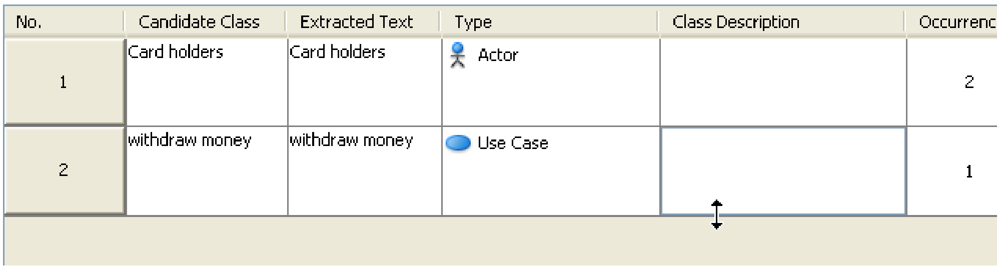

plified system of the automatic teller
he following services:
to withdraw money using the Visa
der and a cash dispenser.
rtcard from their bank can check for
ure.
efill the dispenser regularly, etc.
Card holders(2)
withdraw money(1)

Figure 2-8 Candidate object created for "withdraw money"

Writing Description for Candidate Object

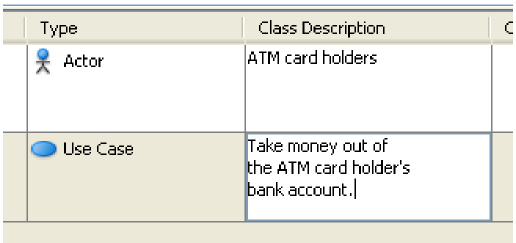
The table in the textual analysis editor allows you to view and edit details of candidate objects. You can edit description of a candidate object by typing in its **Class Description** cell. Before you edit the description, you may resize the row to make more room for the content.



No.	Candidate Class	Extracted Text	Type	Class Description	Occurrenc
1	Card holders	Card holders	Actor		2
2	withdraw money	withdraw money	Use Case		1

Figure 2-9 *Resize rows*

Double-click on the **Class Description** cell to enter description.



Type	Class Description
Actor	ATM card holders
Use Case	Take money out of the ATM card holder's bank account.

Figure 2-10 *Enter description*

Creating Model Elements

To create model element for candidate object, say **Card holders** in this example, right-click it and select **Create Actor Model Element** from the popup menu.

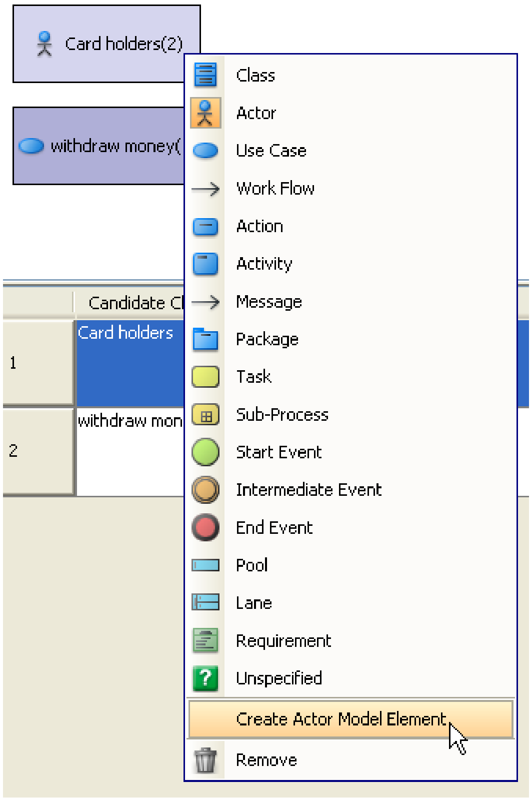


Figure 2-11 *Create Actor model element from candidate object*

Follow the same step to create model element for **withdraw money**. You should see the created model elements in the Model Explorer.

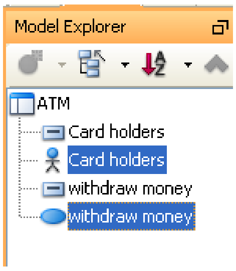


Figure 2-12 *Model elements created*

Creating Use Case Diagram

Now we are going to use the model elements in use case diagram. To create a use case diagram, select menu **File** > **New Diagram** > **UML Diagrams** > **Use Case Diagram**.

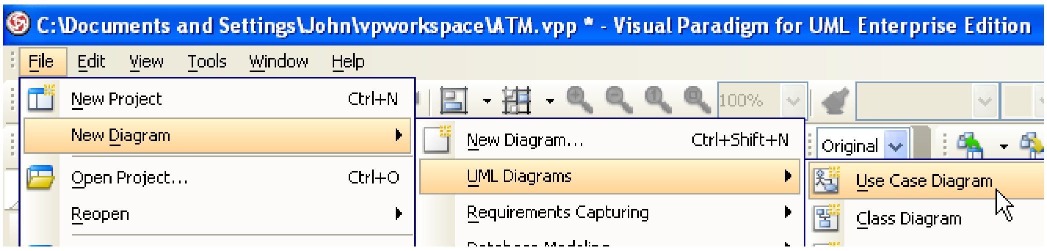


Figure 2-13 *Create use case diagram*

Visualize Created Model Elements

To visualize the created model elements, simply select them in Model Explorer, drag and drop to the use case diagram.

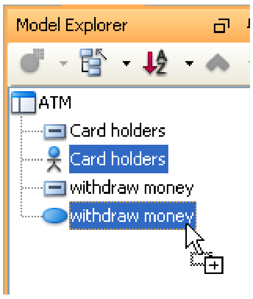


Figure 2-14 *Drag model elements in Model Explorer*

Shapes will be created for the dropped model elements.

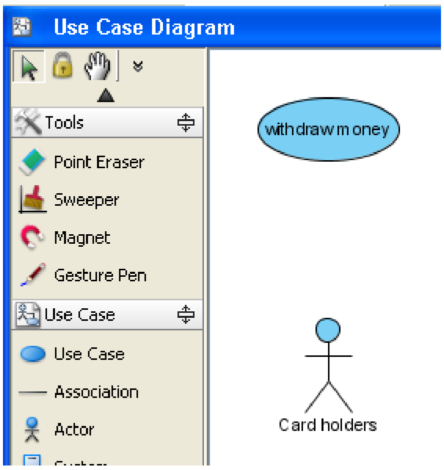


Figure 2-15 *Model elements visualized*

Drawing Requirement Diagrams

Creating Requirement Diagram

To create requirement diagram, select menu **File** > **New Diagram** > **Requirements Capturing** > **Requirement Diagram**.

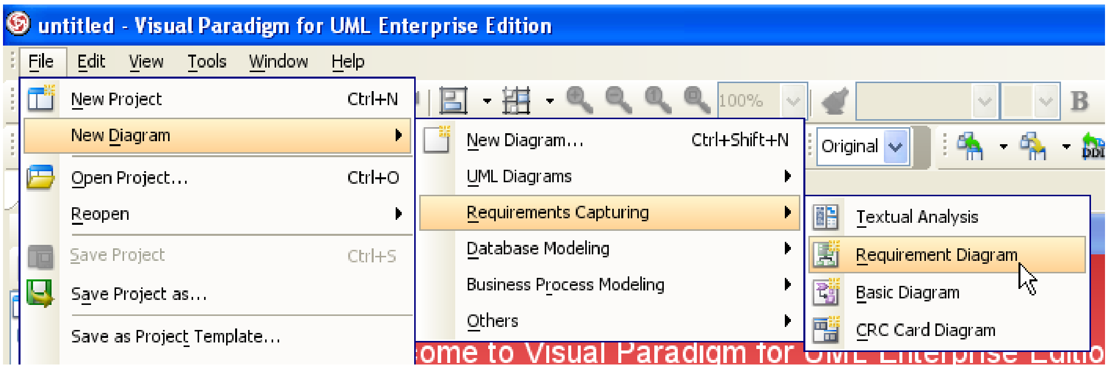


***Figure 2-16** Creating requirement diagram*

Creating Requirement

To create a Requirement, click the **Requirement** button on the diagram toolbar and then click on the diagram.

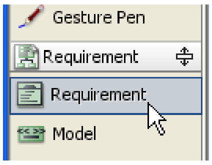
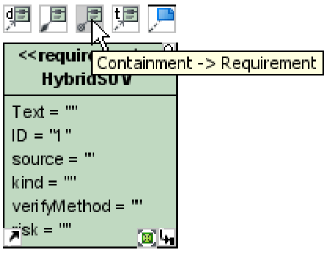


***Figure 2-17** Creating Requirement*

Decomposing Requirement

To decompose a Requirement, click the **Containment** -> **Requirement** resource and drag.



Text = ""
ID = "1"
source = ""
kind = ""
verifyMethod = ""
risk = ""

***Figure 2-18** Decomposing 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.

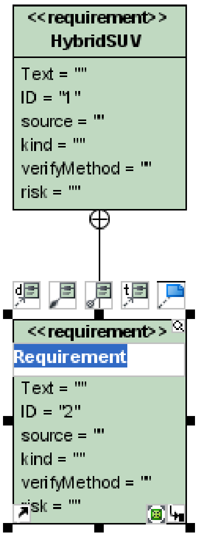


Figure 2-19 *Requirement and Containment created*

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.

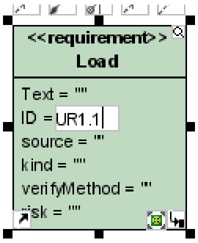


Figure 2-20 *Inline editing Requirement properties*

Edit 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.

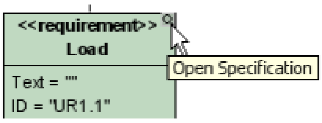
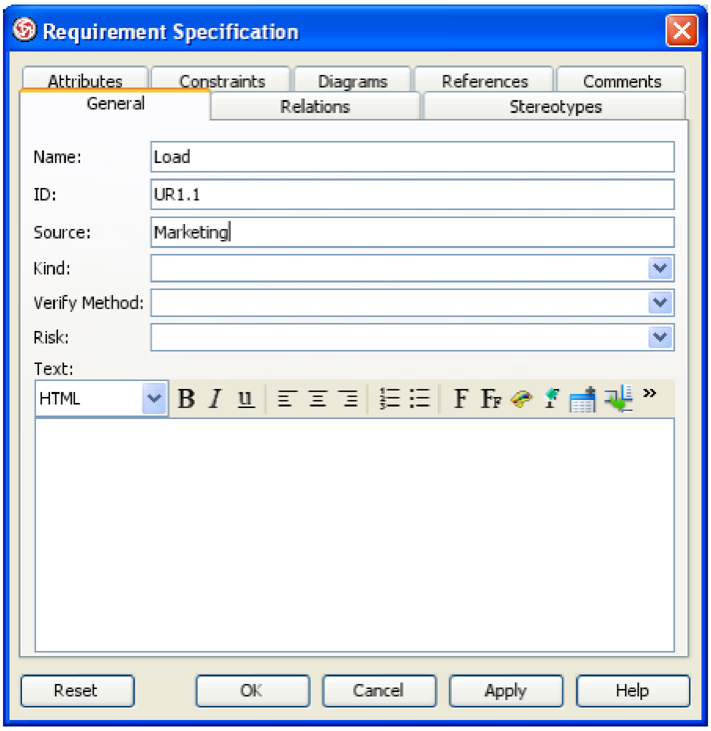


Figure 2-21 *Open specification of Requirement*

The **Requirement Specification** dialog box shows. Edit the properties and click **OK** to apply the changes.



Name: Load
ID: UR1.1
Source: Marketing
Kind:
Verify Method:
Risk:
Text:
HTML

__Figure 2-22__ Requirement Specification

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.

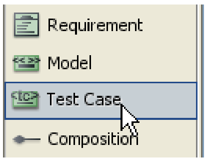


__Figure 2-23__ Creating Test Case

Click **Verify** -> **Requirement** resource of Test Case and drag.

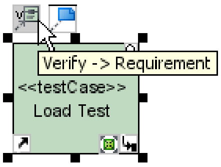


__Figure 2-24__ Linking Requirement with Test Case

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.

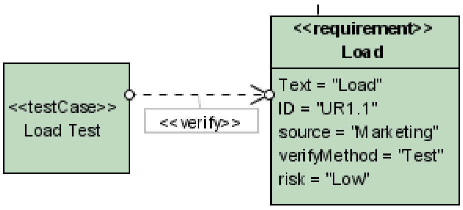


__Figure 2-25__ Verify relationship created

Defining Your Own Set of Requirement Types

Creating New Requirement Type

To create new Requirement type, select menu **Tools** > **Configure Requirements...**.

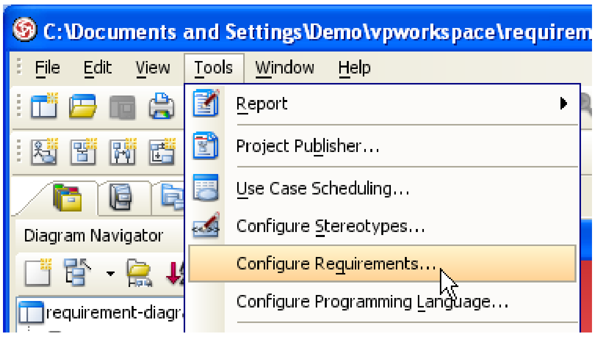


Figure 2-26 *Configure Requirements*

The **Configure Requirements** dialog box appears. Click **Add** to add a new requirement type.

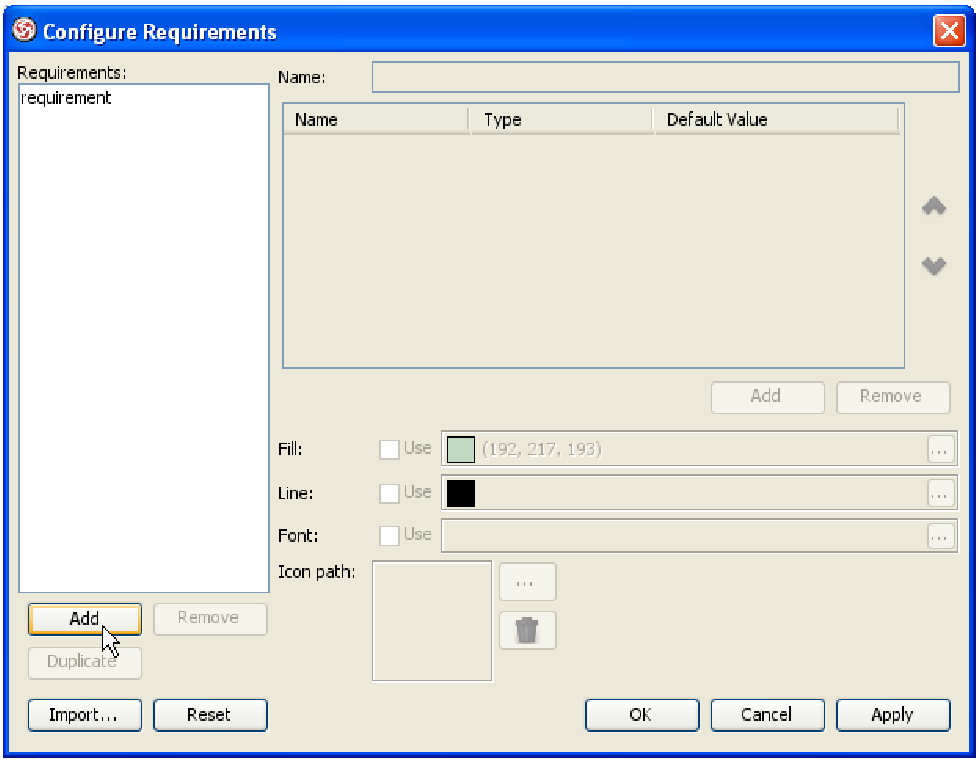
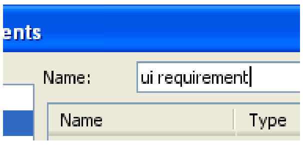


Figure 2-27 *Configure Requirements dialog box*

Enter name of the Requirement type.

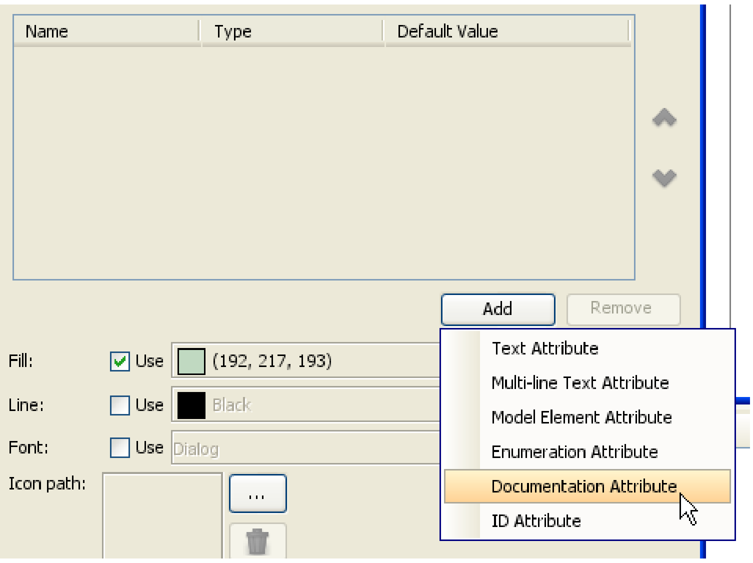


Name: ui requirement

Figure 2-28 *Enter name of Requirement type*

The "ui requirement" will have four attributes: **Text** (HTML Text), **ID** (Text), **source** (Text), **risk** (Enumeration). Let's add them.

Click **Add** below the attribute table and select **Documentation Attribute** (an attribute with rich text content).



Name	Type	Default Value

Fill: ☑ Use (192, 217, 193)

Line: □ Use Black

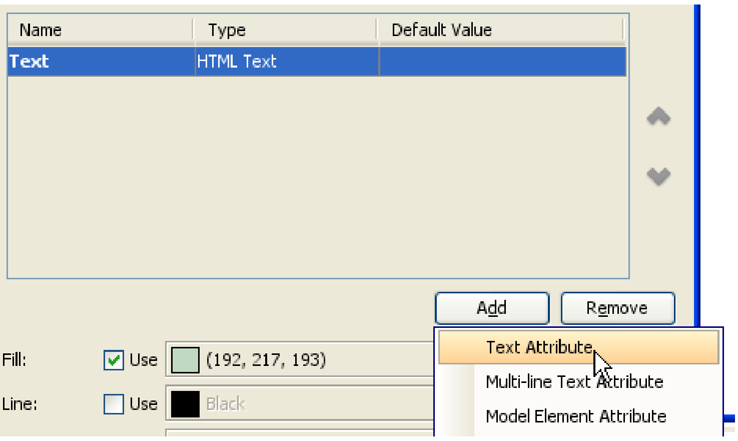
Font: □ Use Dialog

Icon path:

Figure 2-29 *Add documentation attribute*

An attribute named *Text* of type *HTML Text* is created.

Let's add the **ID** attribute. Click **Add** and select **Text Attribute** (an attribute with plain text value).



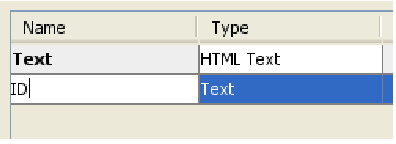
Name	Type	Default Value
Text	HTML Text	

Fill: ☑ Use (192, 217, 193)

Line: □ Use Black

Figure 2-30 *Add text attribute*

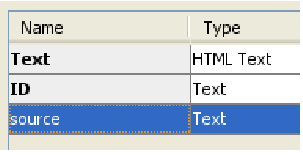
Name the new attribute *ID*.



Name	Type
Text	HTML Text
ID	Text

Figure 2-31 *ID attribute*

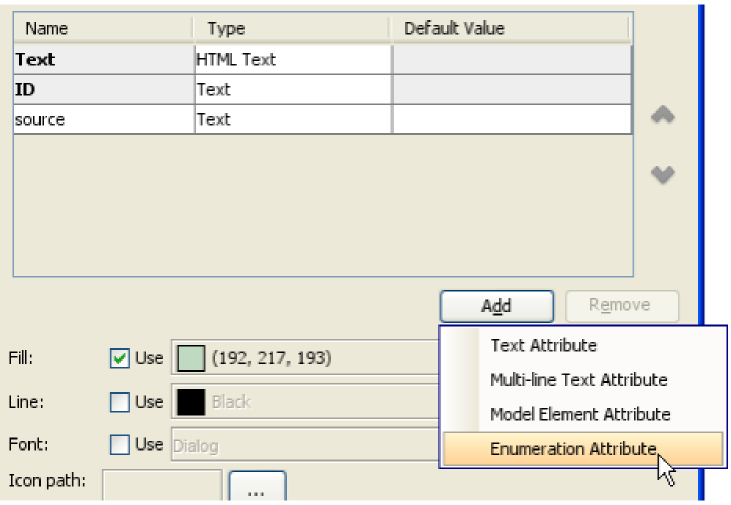
Follow the previous steps to add a *Text* attribute named *source*.



Name	Type
Text	HTML Text
ID	Text
source	Text

Figure 2-32 *source attribute*

Let's add the **risk** attribute. Click **Add** and select **Enumeration Attribute** (an attribute with a list of selectable values).



Name	Type	Default Value
Text	HTML Text	
ID	Text	
source	Text	

Fill: ☑ Use 🟩 (192, 217, 193)

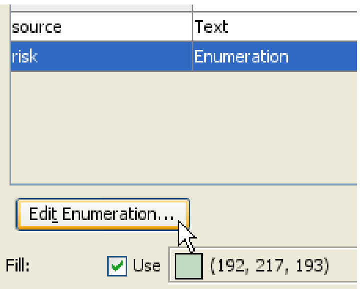
Line: □ Use ■ Black

Font: □ Use Dialog

Icon path:

Figure 2-33 *Add enumeration attribute*

Name the attribute *risk*, and then click **Edit Enumeration...**.

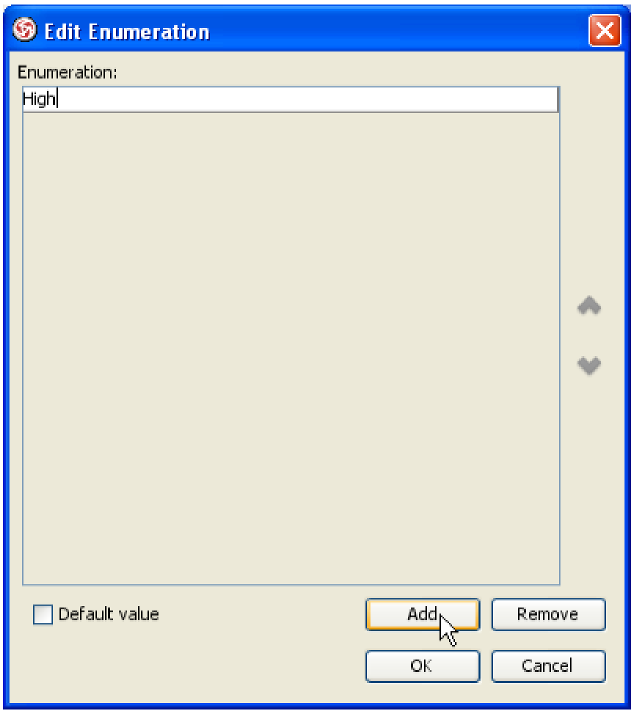


source	Text
risk	Enumeration

Fill: ☑ Use 🟩 (192, 217, 193)

Figure 2-34 *Edit enumeration*

The **Edit Enumeration** dialog box appears. Click **Add** and then name the new item *High*.



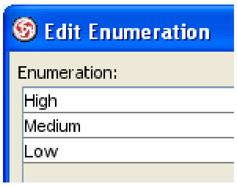
Enumeration:

High

□ Default value

Figure 2-35 *Add enumeration*

Continue to create other items *Medium* and *Low*.

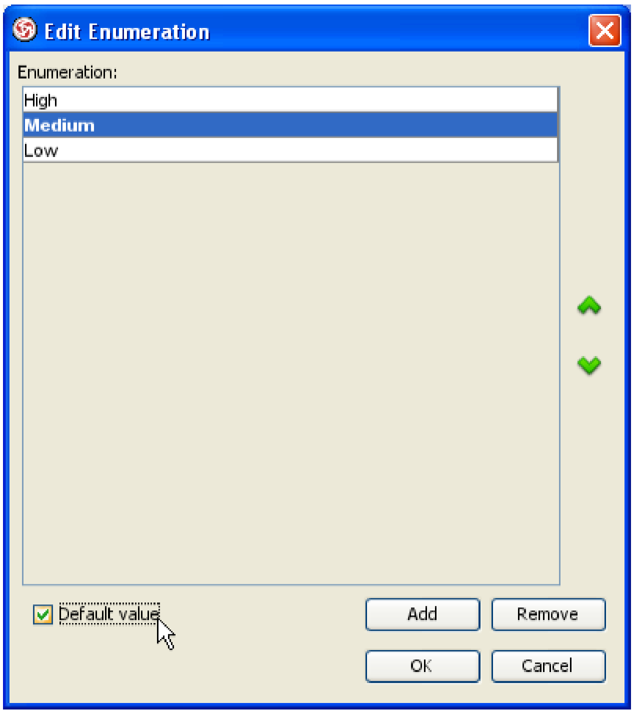

Edit Enumeration

Enumeration:

High
Medium
Low

Figure 2-36 *Enumeration values*

Select **Medium** and select **Default value** to make it the default value for this attribute.


Edit Enumeration

Enumeration:

High
Medium
Low

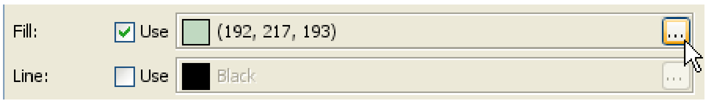
☑ Default value

Figure 2-37 *Set default value*

Click **OK** to apply the changes.

You can also specify appearance of the shape that belongs to this Requirement type. Let's change the fill color and font of "ui requirement".

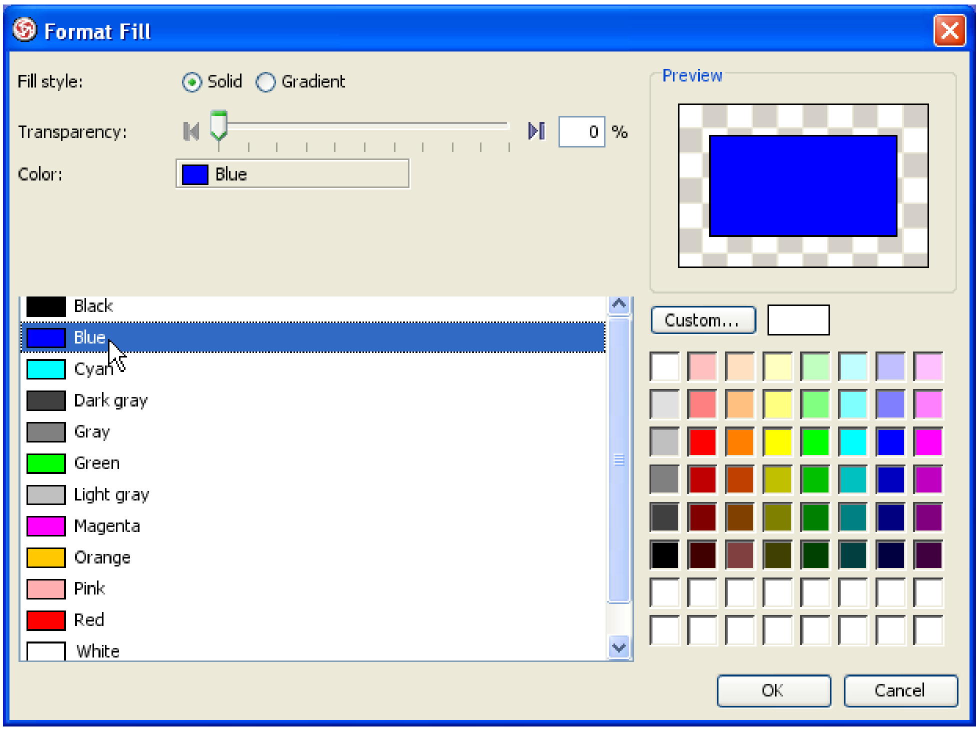
Click on the **...** button of the **Fill** property.


Fill: ☑ Use (192, 217, 193) ...

Line: □ Use Black ...

Figure 2-38 *Edit fill property*

Select **Blue** as fill color and click **OK**.



Format Fill

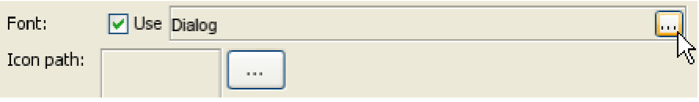
Fill style: ● Solid ○ Gradient

Transparency: 0 %

Color: Blue

Figure 2-39 *Format Fill dialog box*

Select **Use** of the **Font** property to indicate the font settings will be applied to shape, and then click the **...** button.

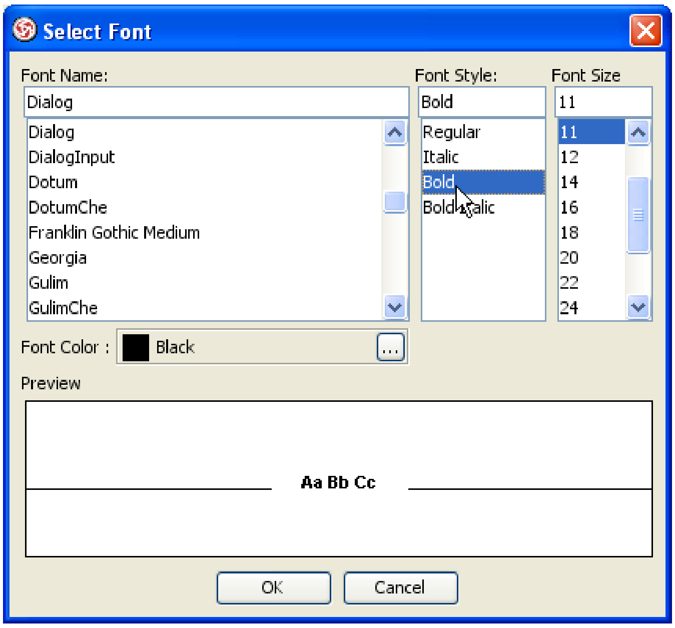


Font: ☑ Use Dialog

Icon path:

Figure 2-40 *Edit font property*

Select **Bold** in **Font Style** and click **OK**.



Select Font

Font Name: Dialog

Font Style: Bold

Font Size 11

Font Color : Black

Figure 2-41 *Select Font dialog box*

Click **OK** in **Configure Requirements** dialog box to apply the changes.

After that, click the drop down button beside the **Requirement** button on the diagram toolbar of a requirement diagram, select **Ui requirement**.

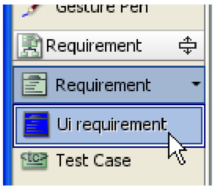
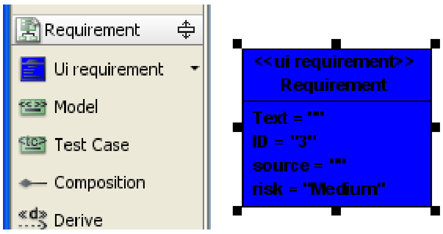


Figure 2-42 *Create Requirement of new type*

Click on the diagram to create the shape, you can see the Requirement created has the attributes you defined, and the styles and formats are applied to the shape.



<<ui requirement>>
Requirement

Text = ""
ID = "3"
source = ""
risk = "Medium"

Figure 2-43 *Requirement of new type created*

Drawing CRC Card Diagram

Creating CRC Card Diagram

Select menu **File** > **New Diagram** > **Requirements Capturing** > **CRC Card Diagram** to create a CRC card diagram.

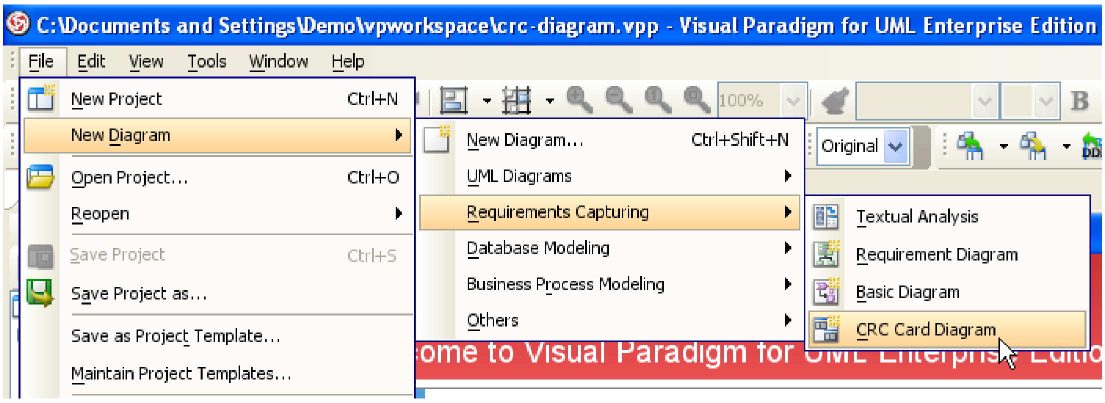

C:\Documents and Settings\Demo\vpworkspace\crc-diagram.vpp - Visual Paradigm for UML Enterprise Edition
File
Edit
View
Tools
Window
Help
New Project Ctrl+N
New Diagram
Open Project... Ctrl+O
Reopen
Save Project Ctrl+S
Save Project as...
Save as Project Template...
Maintain Project Templates...
New Diagram... Ctrl+Shift+N
UML Diagrams
Requirements Capturing
Database Modeling
Business Process Modeling
Others
Textual Analysis
Requirement Diagram
Basic Diagram
CRC Card Diagram
100%
Original
B
ome to Visual Paradigm for UML Enterpris Editio

Figure 2-44 *Create CRC card diagram*

Creating CRC Card

Click **CRC Card** on the diagram toolbar and then click on the diagram to create a CRC card. Name it *Shipment* **.**

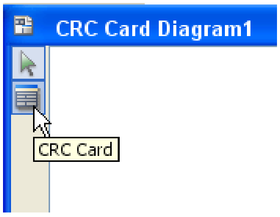
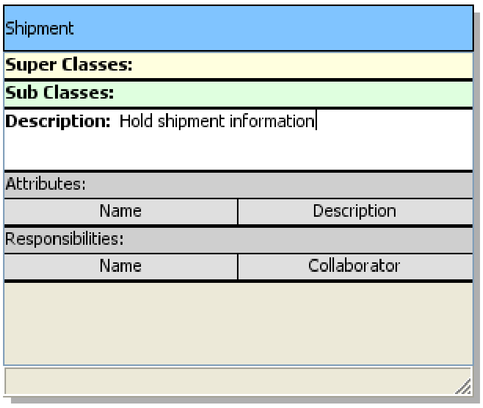


Figure 2-45 *Create CRC card*

Editing CRC Card Properties

Double-click **Description** to edit it and enter *Hold shipment information*.



Shipment

Super Classes:

Sub Classes:

Description: Hold shipment information

Attributes:

Name	Description

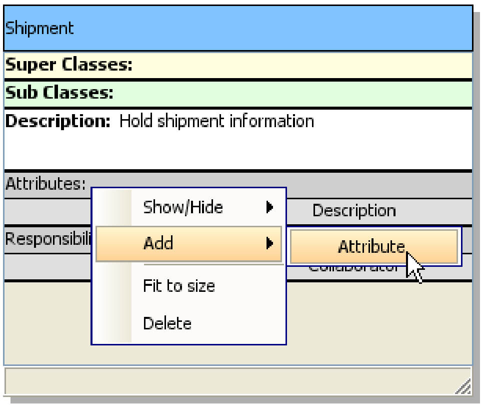
Responsibilities:

Name	Collaborator

Figure 2-46 *Edit description*

Adding Attributes

Right-click on the **Attributes** heading and select **Add** > **Attribute** from the popup menu.



Shipment

Super Classes:

Sub Classes:

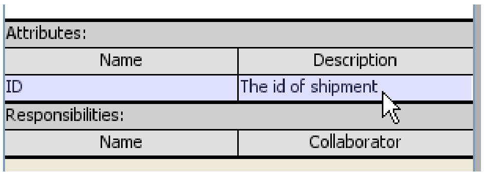
Description: Hold shipment information

Attributes:

Responsibili

Figure 2-47 *Add attribute*

Edit attribute name to *ID* and description to *The id of shipment*.



Attributes:

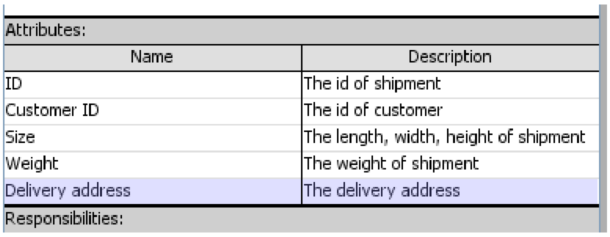
Name	Description
ID	The id of shipment

Responsibilities:

Name	Collaborator

Figure 2-48 *Attribute added*

Continue to add other attributes.



Attributes:

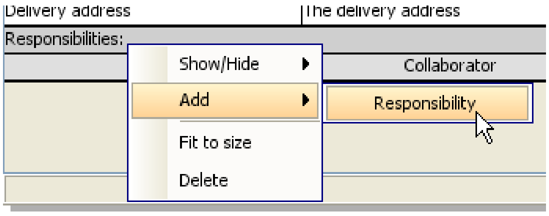
Name	Description
ID	The id of shipment
Customer ID	The id of customer
Size	The length, width, height of shipment
Weight	The weight of shipment
Delivery address	The delivery address

Responsibilities:

Figure 2-49 *All attributes added*

Adding Responsibilities

Right-click on the **Responsibilities** heading and select **Add** > **Responsibility** from the popup menu.

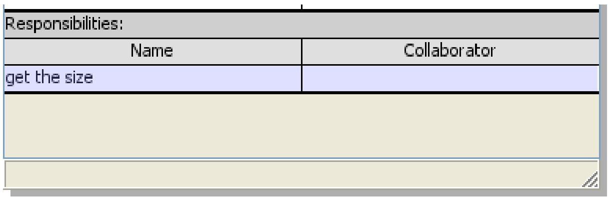


Delivery address	The delivery address

Responsibilities:

Figure 2-50 *Add responsibility*

Edit responsibility name to *get the size*.

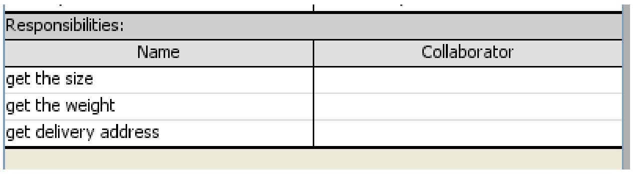


Responsibilities:

Name	Collaborator
get the size	

Figure 2-51 *Responsibility added*

Continue to add other responsibilities.



Responsibilities:

Name	Collaborator
get the size	
get the weight	
get delivery address	

Figure 2-52 *All responsibilities added*

Drawing Activity Diagrams

Creating Activity Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Activity Diagram** to create an activity diagram.

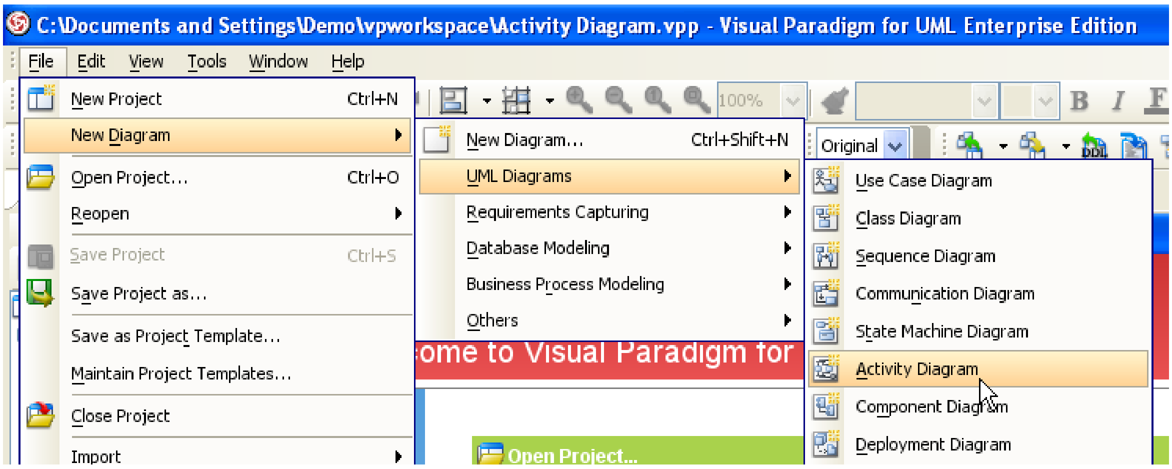


Figure 3-1 *Create activity diagram*

Creating Swimlane

Let's create a vertical swimlane. Click the drop down button beside **Horizontal Swimlane** on the diagram toolbar and then select **Vertical Swimlane**.

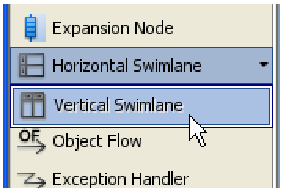


Figure 3-2 *Create swimlane*

Click on the diagram to create the swimlane.

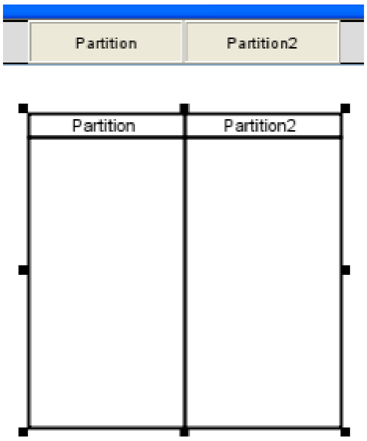


Figure 3-3 *Swimlane created*

Double-click the partition name to rename it.

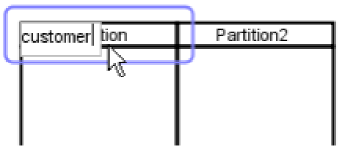


Figure 3-4 *Rename partition*

Inserting Partition to Swimlane

To insert partition to swimlane, right-click on a partition and select **Insert Partition After** from the popup menu.

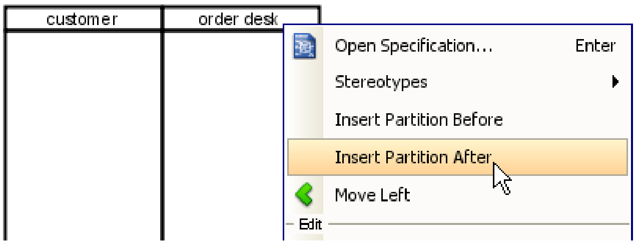


Figure 3-5 *Insert partition to swimlane*

A partition is inserted.

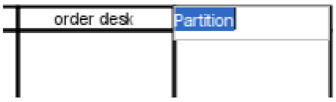


Figure 3-6 *Partition inserted*

Creating Initial Node

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

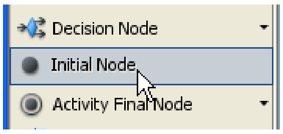


Figure 3-7 *Create initial node*

Click inside the partition to create the initial node there.

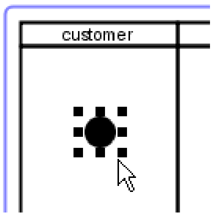


Figure 3-8 *Initial node created*

Creating Action

Mouse over the initial node until its resources are visible. Click on the **Control Flow** -> **Action** resource and drag.

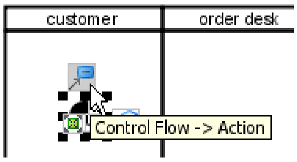


Figure 3-9 *Create action*

Move the mouse to where you want to place the action to, and then release the mouse button. An action is created and is connected to the initial node with a control flow.

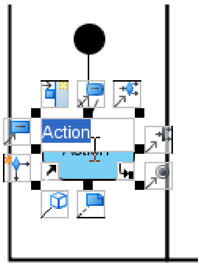


Figure 3-10 *Action created*

Similarly you can create a new action using the **Control Flow** -> **Action** resource of an action.

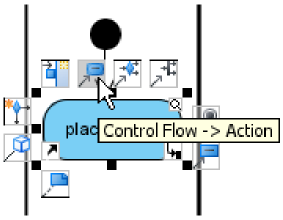


Figure 3-11 *Create a new action from an action*

A new action is created and is connected to the action with a control flow.

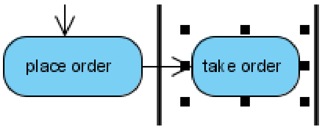


Figure 3-12 *Action created*

Continue to complete the activity diagram.

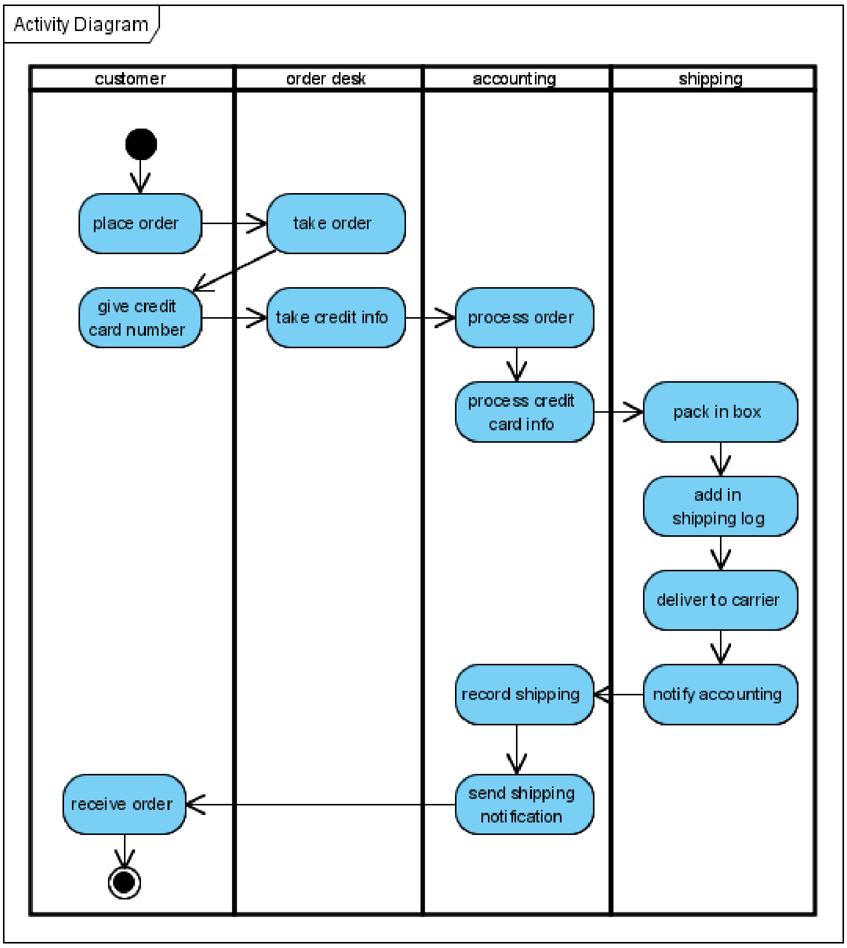


Figure 3-13 *Completed activity diagram*

Drawing State Machine Diagrams

Creating State Machine Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **State Machine Diagram** to create a state machine diagram.

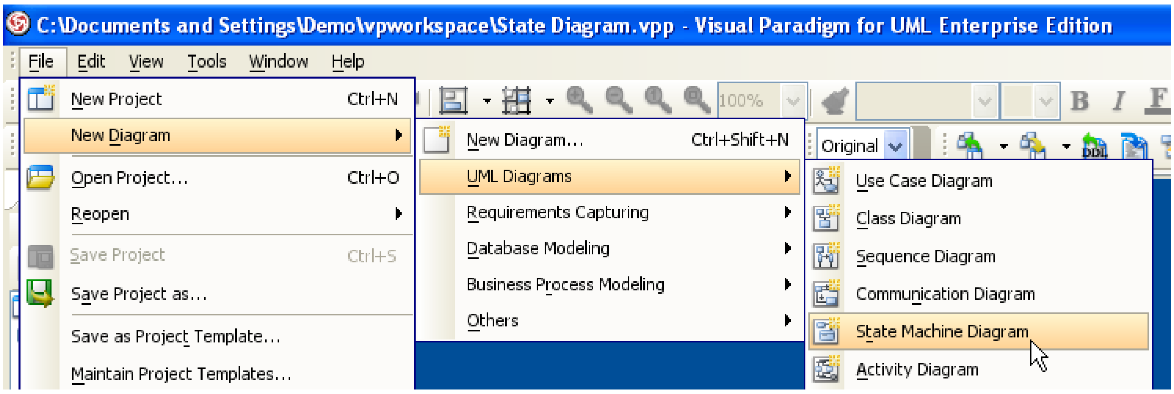


Figure 3-14 *Create state machine diagram*

Before we draw the diagram, let's change the connector style to curve. Right-click on the diagram and select **Connectors** > **Curve** from the popup menu.

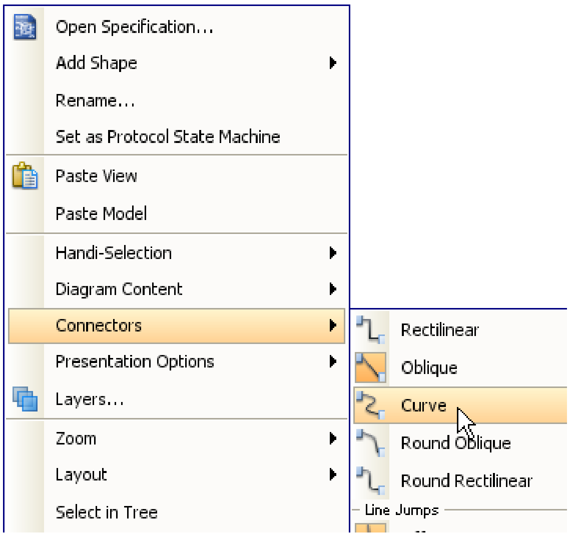


Figure 3-15 *Selecting curve connector style*

Creating States and Transitions

Move the mouse over the initial pseudo state until its resources are visible. Click the **Transition** -> **State** resource and drag.

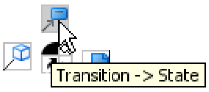


Figure 3-16 *Create state from initial pseudo state*

Move the mouse to where you want to place the state to, and then release the mouse button. A state is created and is connected to the initial pseudo state with a transition.

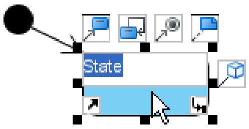


Figure 3-17 *State and transition created*

Drag the connector to make it curved.

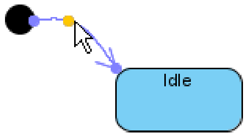


Figure 3-18 Drag the connector to make it curved

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

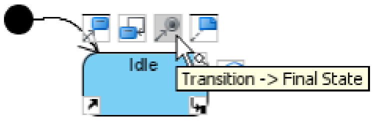


Figure 3-19 Create final state

Final state does not show caption by default. To show it, right-click on the diagram and select **Presentation Options** > **Show Shape Caption** > **Final State**.

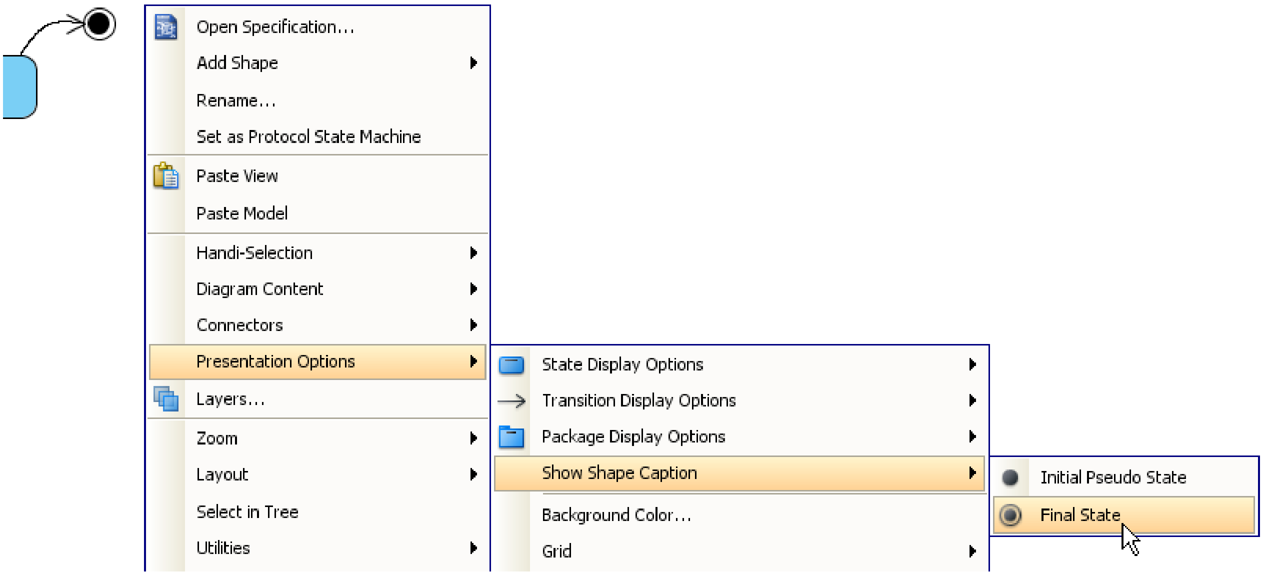


Figure 3-20 Show caption of final state

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 **Add Horizontal Region** from the popup menu.

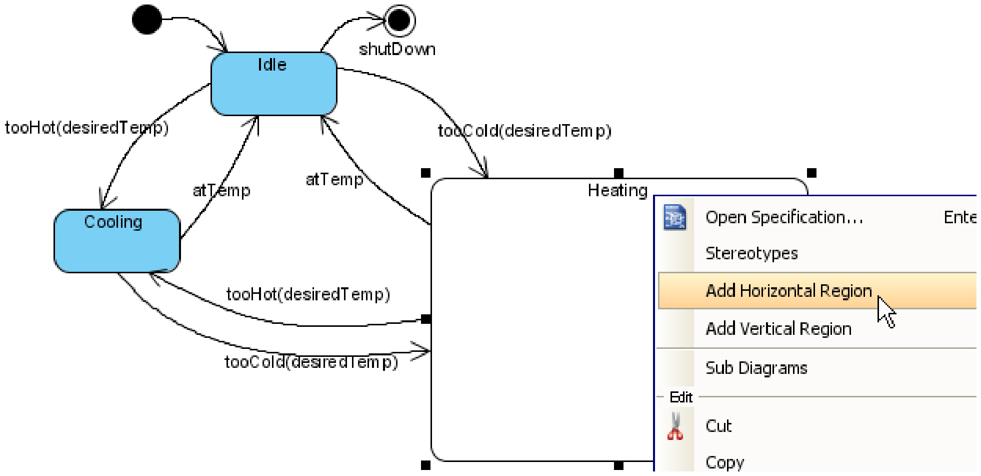


Figure 3-21 Add region to state

Then you can draw the substates inside the region.

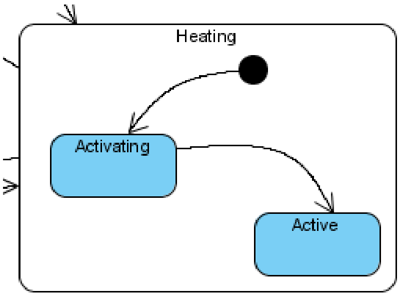


Figure 3-22 Substates in a composite state

Modeling Properties of Transition

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

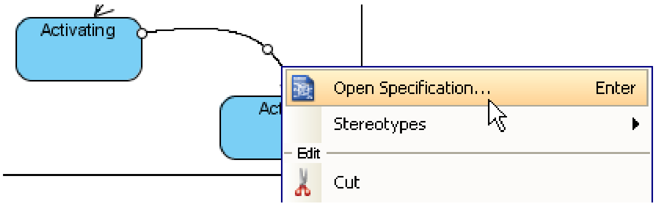
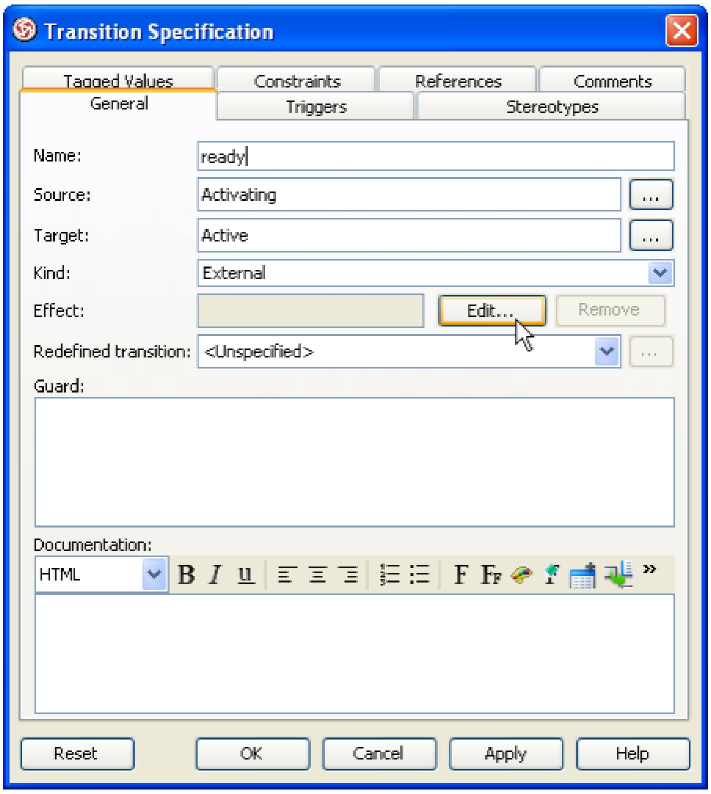


Figure 3-23 Open specification of transition

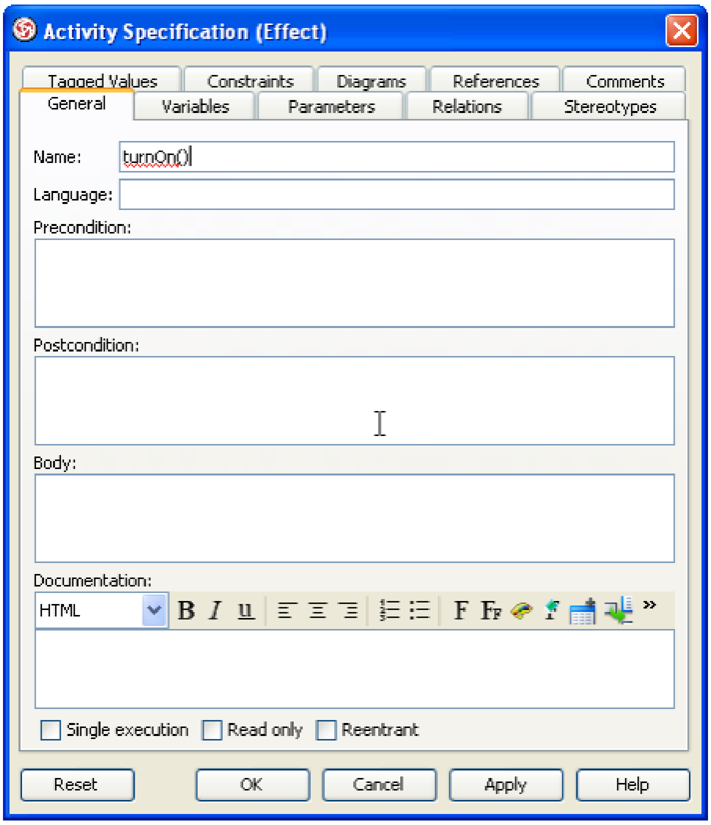
The **Transition Specification** dialog box appears. Here you can edit its name, effect, guard, etc. Let's click **Edit...** of the **Effect** property.



Name: ready
Source: Activating
Target: Active
Kind: External
Effect:
Redefined transition: <Unspecified>
Guard:
Documentation:

Figure 3-24 Transition Specification dialog box

The **Activity Specification (Effect)** dialog box apppears. Let's change its name, and then click **OK** to apply the change.



Name: turnOn()

Language:

Precondition:

Postcondition:

Body:

Documentation: HTML

□ Single execution □ Read only □ Reentrant

Figure 3-25 *Edit effect property of transition*

Click **OK** in the **Transition Specification** dialog box to close it. The name (ready) and effect (turnOn()) are shown on the transition caption.

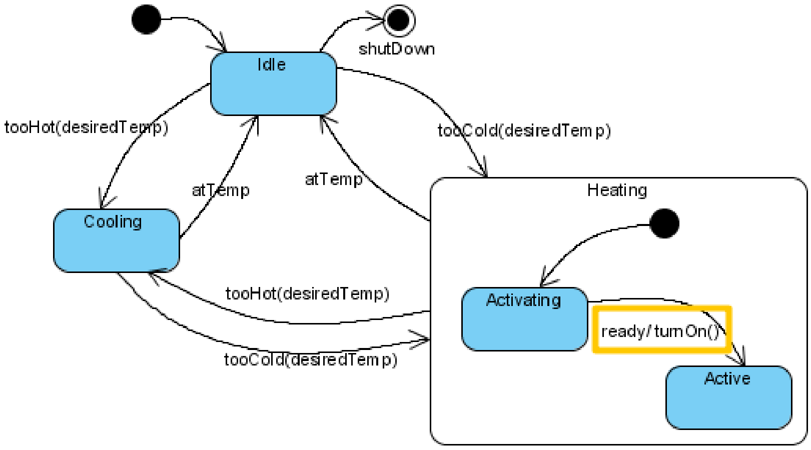


Figure 3-26 *Name and effect shown in caption of transition*

Drawing Timing Diagrams

Creating Timing Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Timing Diagram** to create a timing diagram.

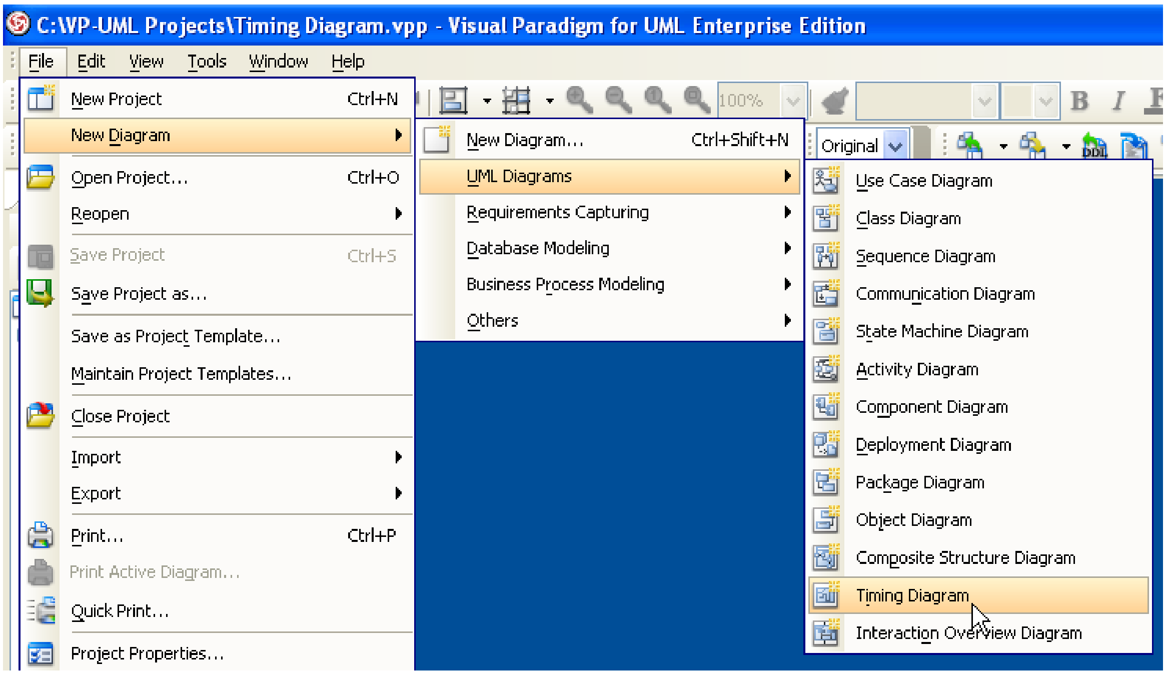


Figure 3-27 *Create timing diagram*

Creating Timing Frame

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

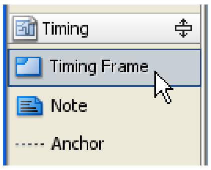


Figure 3-28 *Create timing frame*

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

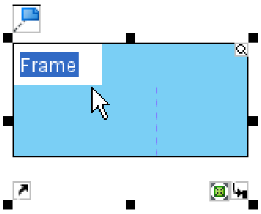


Figure 3-29 *Rename frame*

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

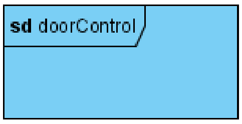


Figure 3-30
Frame renamed

Adding Lifeline to Frame

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

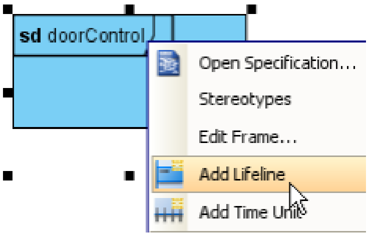


Figure 3-31 *Add lifeline*

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 popup menu.

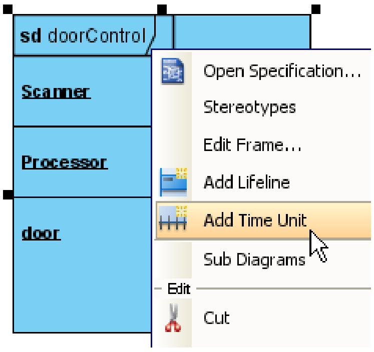


Figure 3-32 *Add time unit*

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

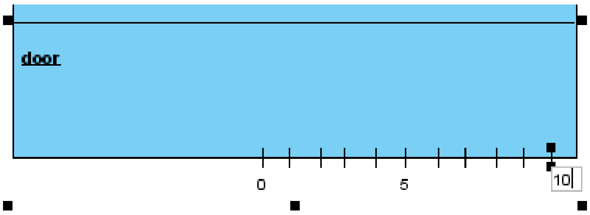


Figure 3-33 *Rename time unit*

Adding State/Condition to Lifeline

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

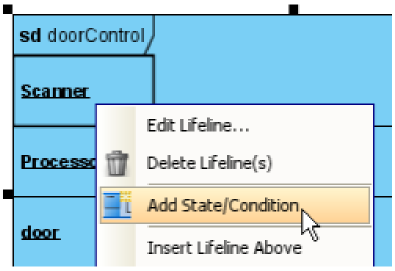


Figure 3-34 *Add state/condition*

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.

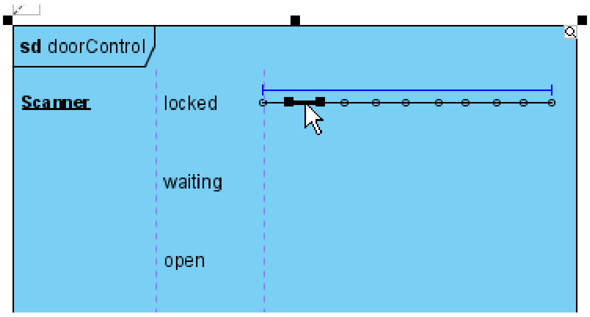


Figure 3-35 *Drag time instance*

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

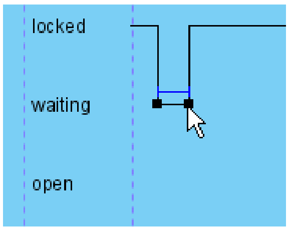


Figure 3-36 *Dragged time instance*

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.

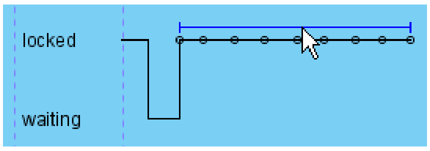


Figure 3-37 *Move a group of time instances*

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

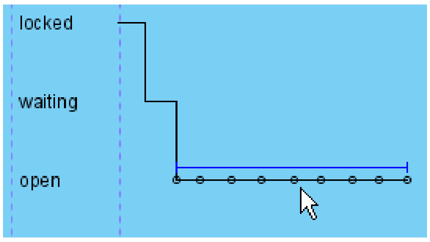


Figure 3-38 *Moved group of time instances*

Adding Time Messages to Frame

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

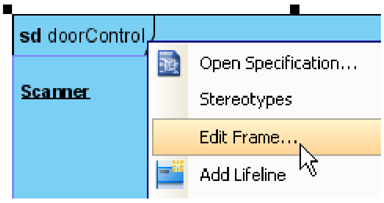


Figure 3-39 *Edit frame*

The **Edit Frame** dialog box appears. Select the **Time Messages** tab and click **Add...**.

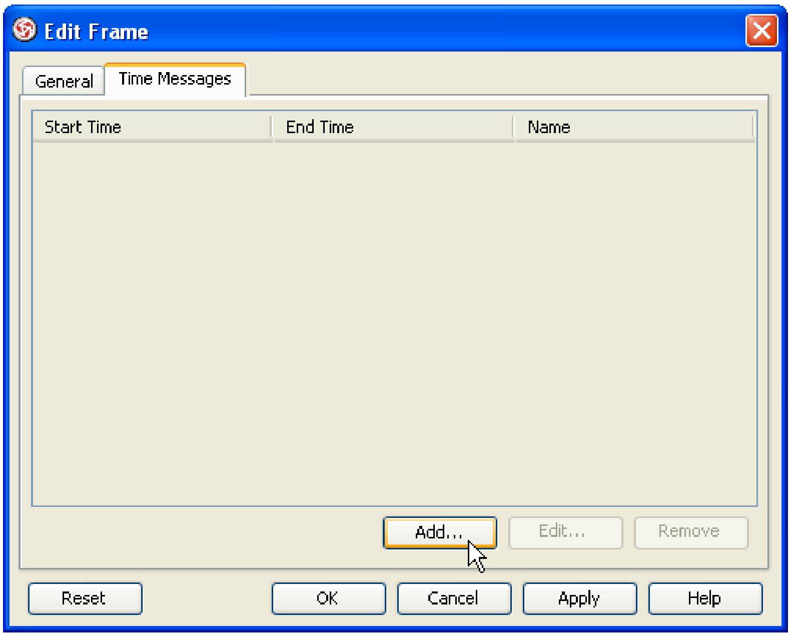
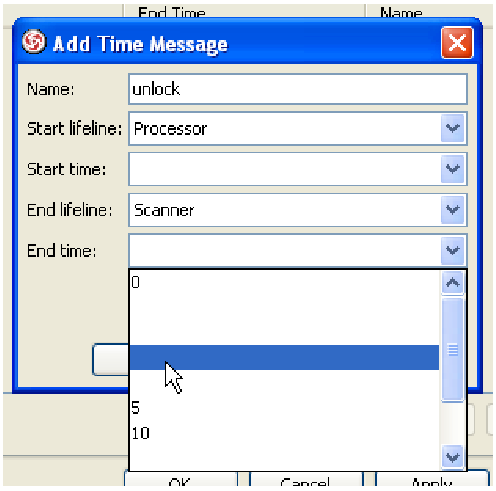


Figure 3-40 *Add time message*

The **Add Time Message** dialog box appears. 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.



Name: unlock

Start lifeline: Processor

Start time:

End lifeline: Scanner

End time:

Figure 3-41 *Select end time of time message*

The time message is shown on the frame.

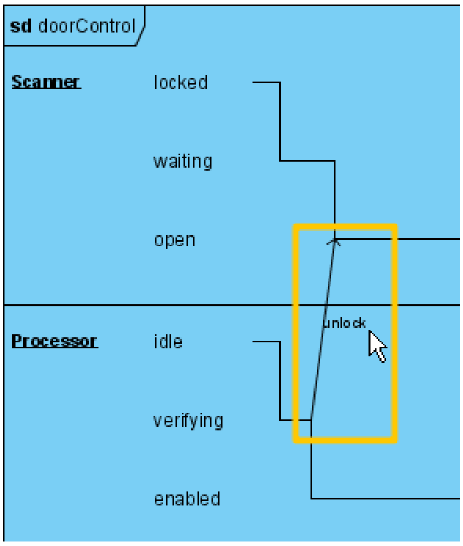


Figure 3-42 Time message

Switching to Compact View Mode

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

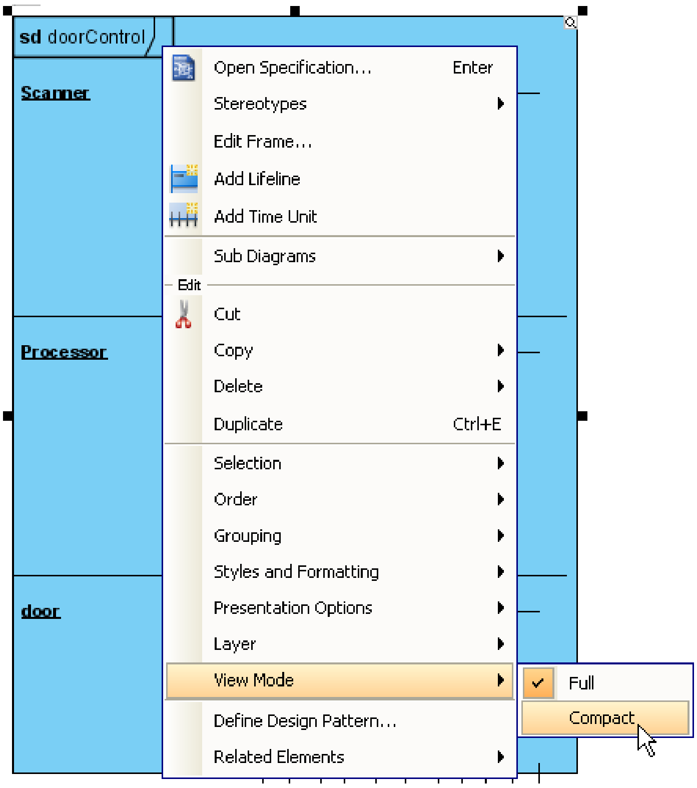


Figure 3-43 Switch to compact view mode

The frame will be shown in compact mode.

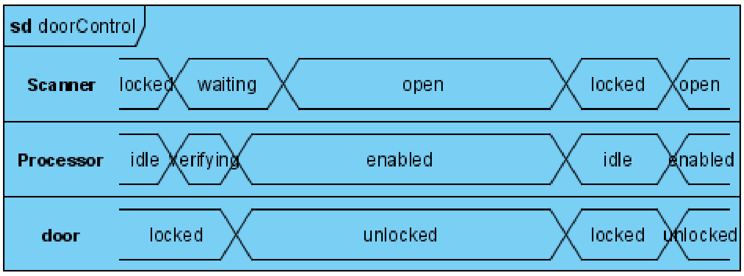


Figure 3-44 *Frame shown in compact mode*

Drawing Sequence Diagrams

Creating Sequence Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Sequence Diagram** to create a sequence diagram.

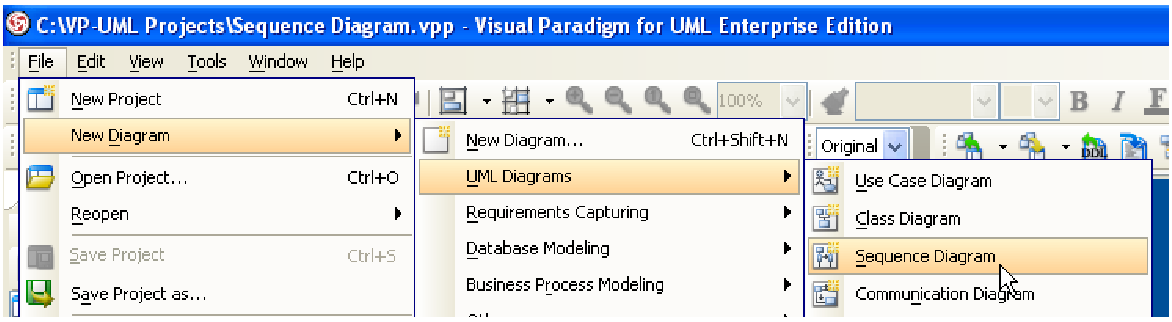


Figure 4-1 Create sequence diagram

Creating Actor

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

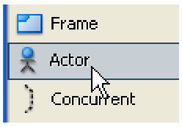


Figure 4-2 Create actor

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.

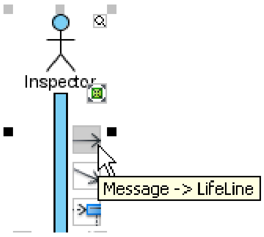


Figure 4-3 Create lifeline

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.

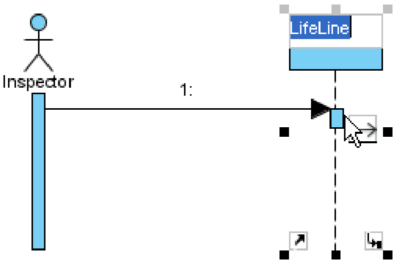


Figure 4-4 Lifeline and message created

Auto Extending Activation

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

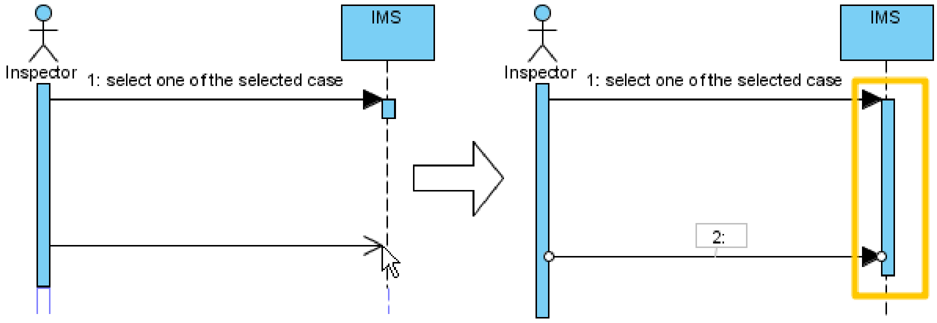

Inspector
IMS
1: select one of the selected case
Inspector
IMS
1: select one of the selected case
2:

Figure 4-5 *Auto extending 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).

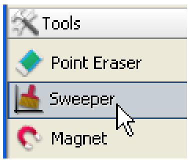


Figure 4-6
Sweeper

Click on empty space of the diagram and drag towards either 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.

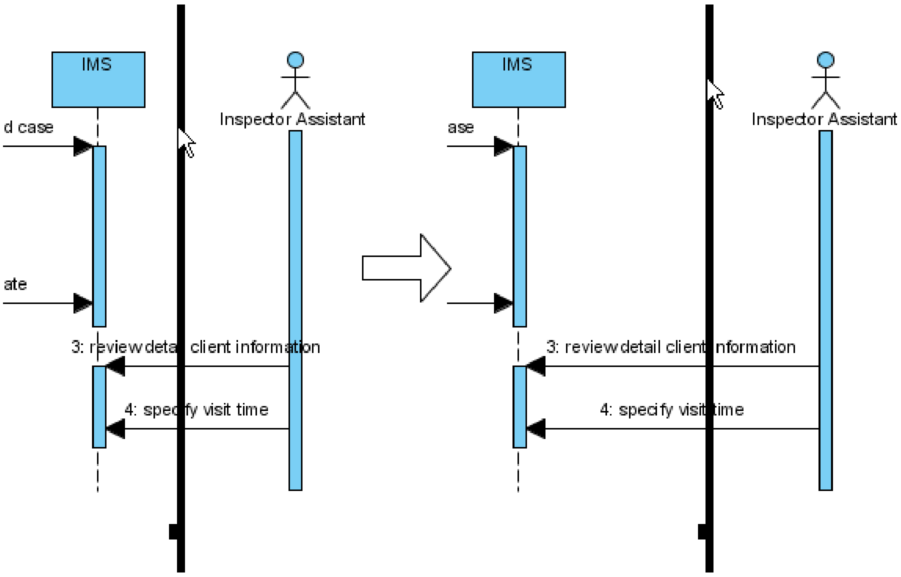


Figure 4-7 *Sweep towards right*

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

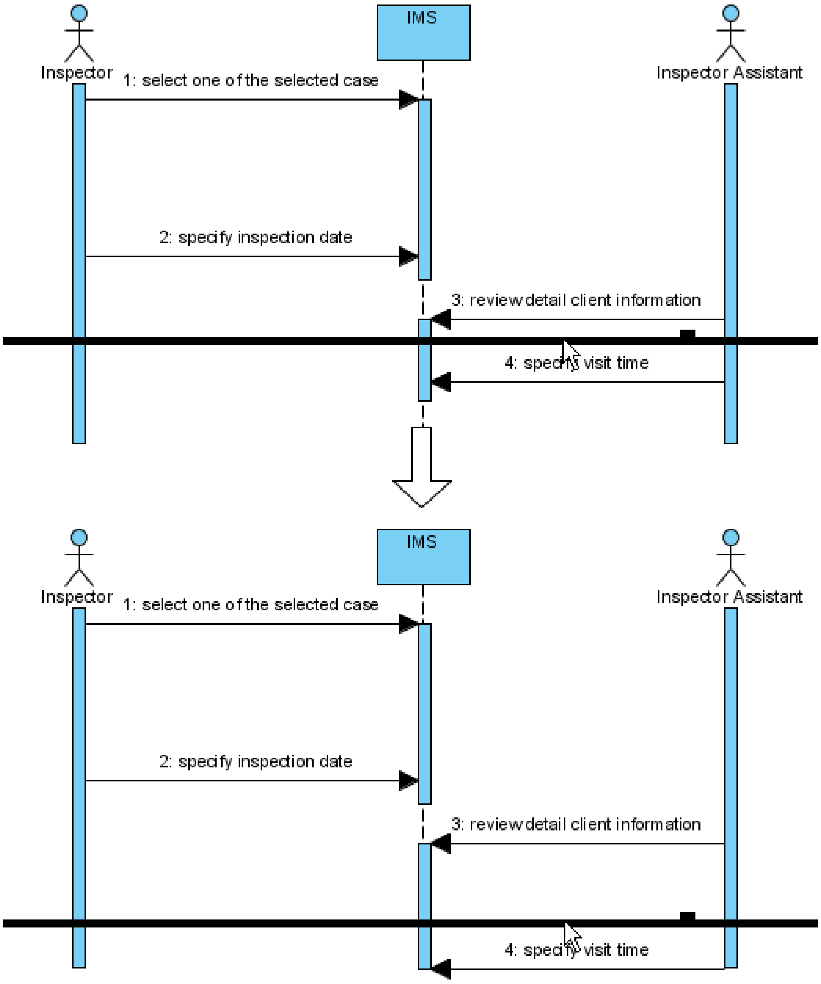

IMS
Inspector
Inspector Assistant
1: select one of the selected case
2: specify inspection date
3: review detail client information
4: specify visit time

Figure 4-8 *Sweep downwards*

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

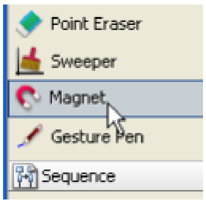


Figure 4-9 *Magnet*

Click on empty space of the diagram and drag towards either 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.

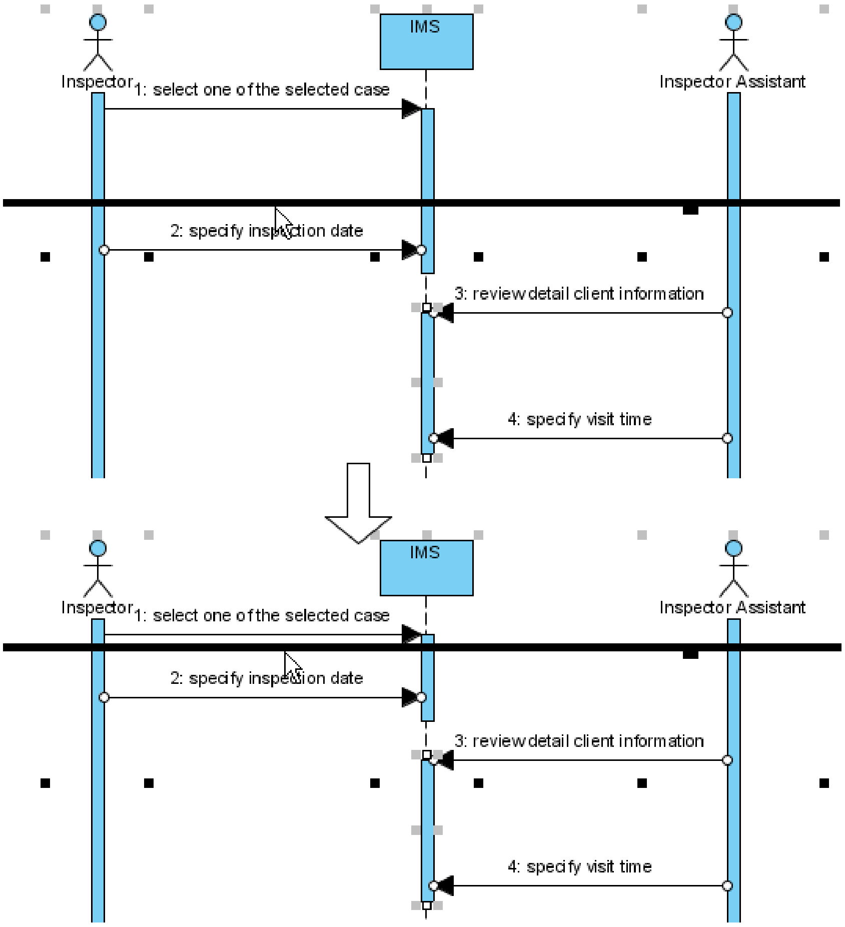

IMS
Inspector
Inspector Assistant
1: select one of the selected case
2: specify inspection date
3: review detail client information
4: specify visit time

Figure 4-10 *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.

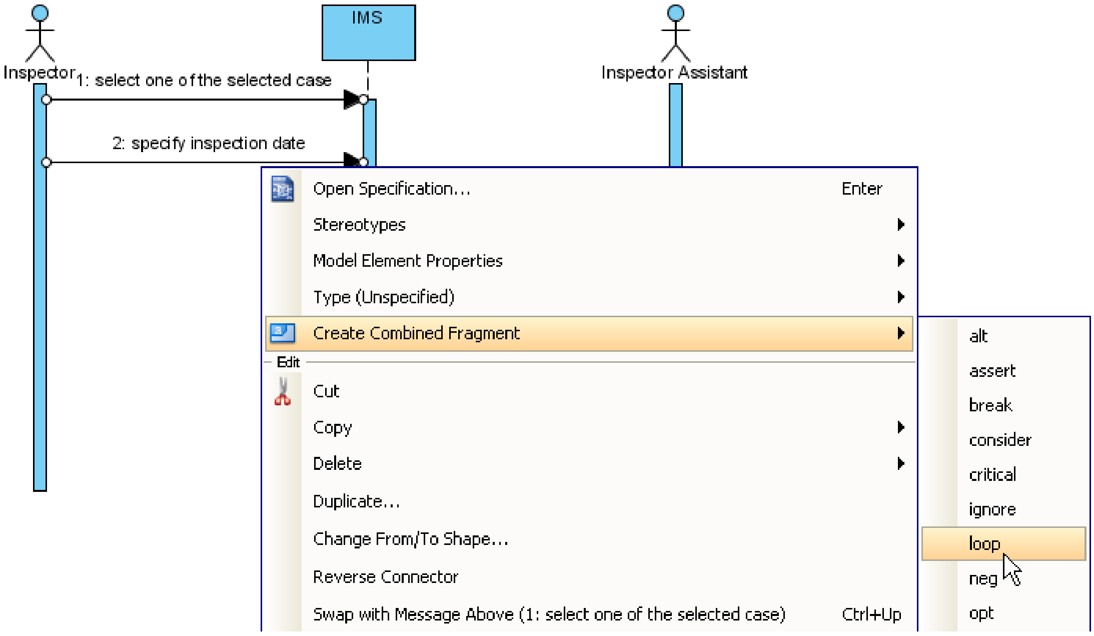


Figure 4-11 *Create combined fragment for messages*

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

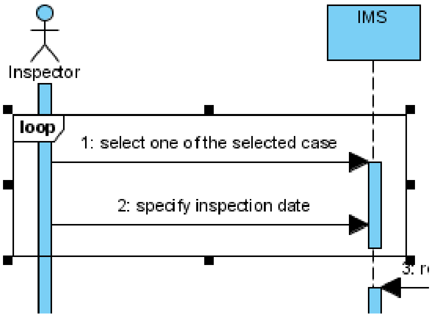

IMS
Inspector
loop
1: select one of the selected case
2: specify inspection date
3: r

Figure 4-12 *Combined fragment created*

Drawing Communication Diagrams

Creating Communication Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Communication Diagram** to create a communication diagram.

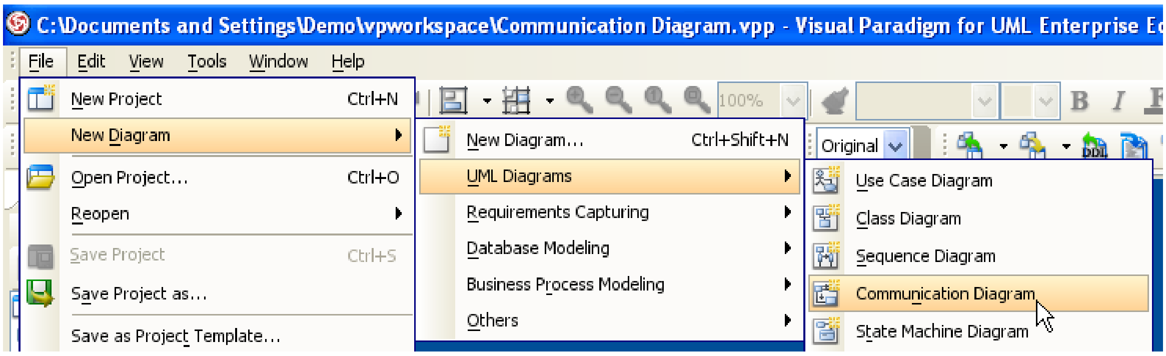


Figure 4-13 Create communication diagram

Creating Actor

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

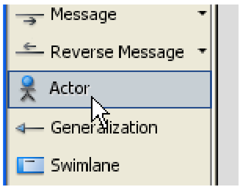


Figure 4-14 Create actor

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.

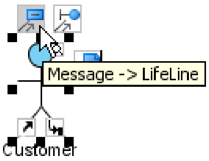


Figure 4-15 Create lifeline

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 link (the line) and a message (the arrow).

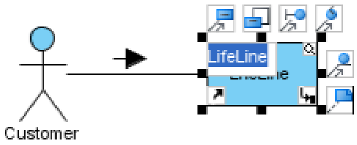


Figure 4-16 Lifeline, link and message created

Creating Message on Link

To create message on link, click its **Create Message** resource.

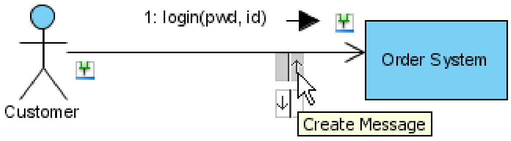


Figure 4-17 Create message on link

A message will be created on the link.

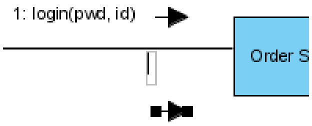


Figure 4-18 *Message created on link*

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 and select **Reorder Messages ...** from the popup menu.

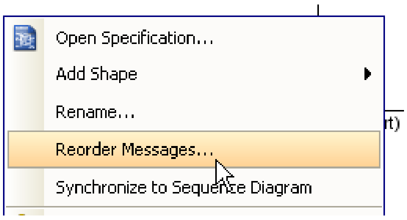
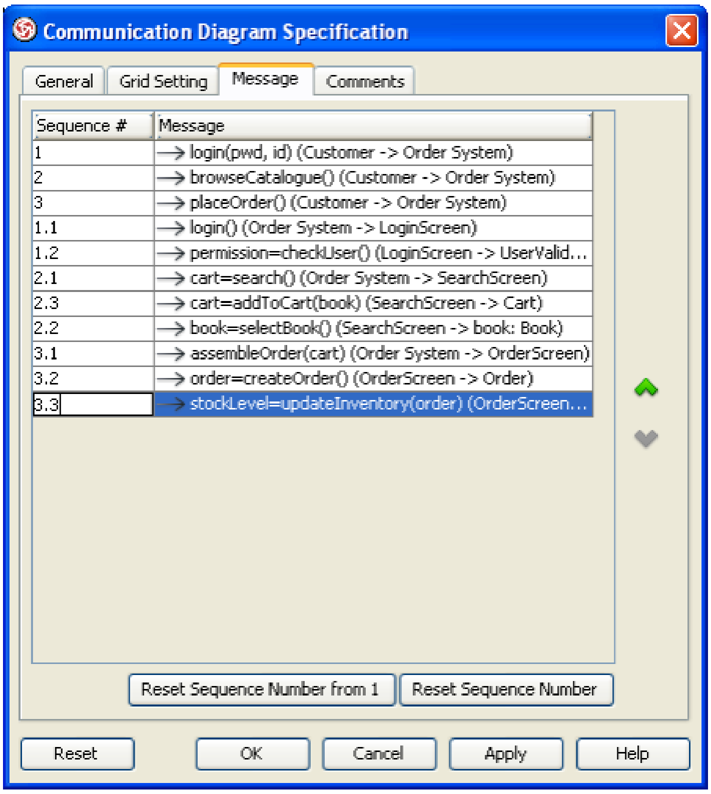


Figure 4-19 *Reorder messages*

The **Communication Diagram Specification** dialog box appears with the **Message** tab selected. Double-click on the **Sequence #** cell of an message to edit it. Click **OK** to apply the changes.



Sequence #	Message
1	→ login(pwd, id) (Customer -> Order System)
2	→ browseCatalogue() (Customer -> Order System)
3	→ placeOrder() (Customer -> Order System)
1.1	→ login() (Order System -> LoginScreen)
1.2	→ permission=checkUser() (LoginScreen -> UserValid...
2.1	→ cart=search() (Order System -> SearchScreen)
2.3	→ cart=addToCart(book) (SearchScreen -> Cart)
2.2	→ book=selectBook() (SearchScreen -> book: Book)
3.1	→ assembleOrder(cart) (Order System -> OrderScreen)
3.2	→ order=createOrder() (OrderScreen -> Order)
3.3	→ stockLevel=updateInventory(order) (OrderScreen...

Figure 4-20 *Edit sequence number of messages*

The sequence number of messages are updated.

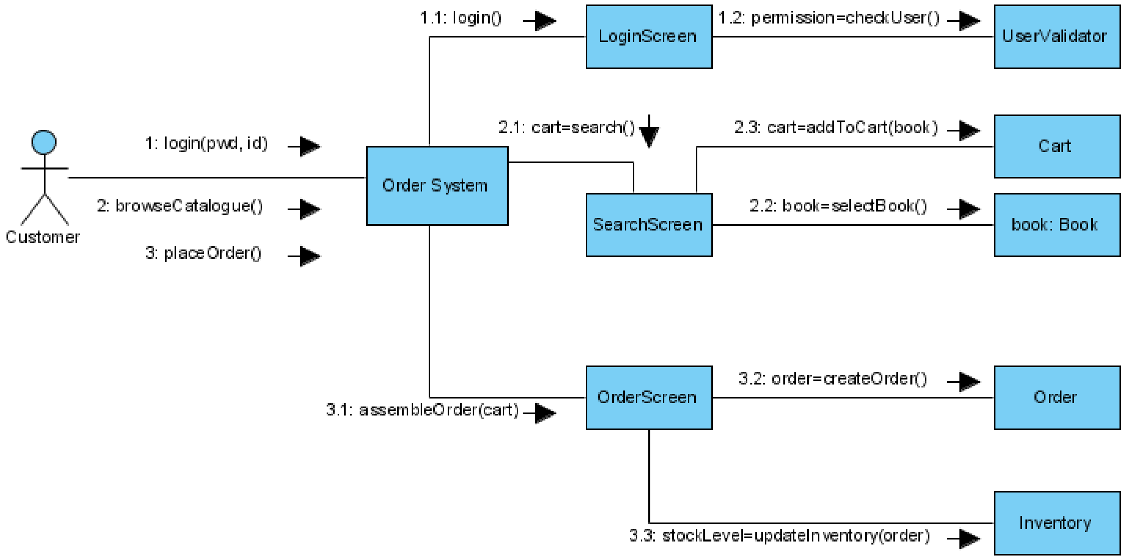


Figure 4-21 *Sequence number of messages updated*

Drawing Interaction Overview Diagrams

Creating Interaction Overview Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Interaction Overview Diagram** to create an interaction overview diagram.

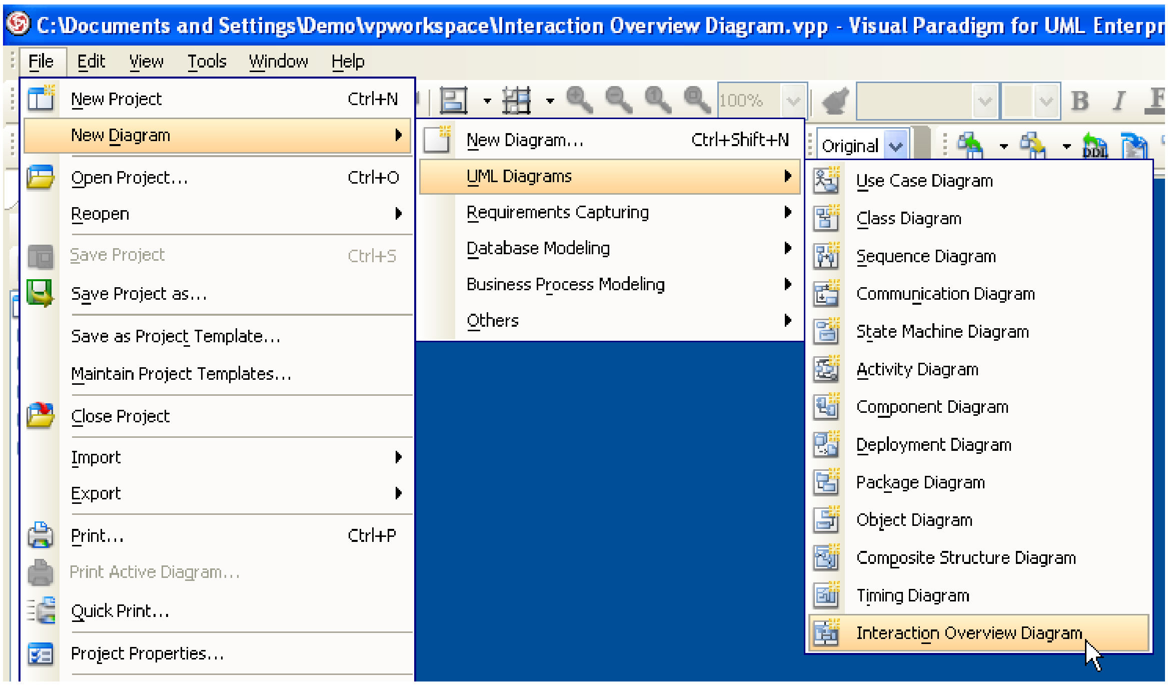


Figure 4-22 *Create interaction overview diagram*

Creating Initial Node

To create initial node, click **Initial Node** on the diagram toolbar and then click on the diagram.

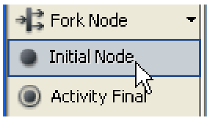


Figure 4-23 *Create initial node*

An initial node is created. The caption of initial node is hidden by default, to show it, right-click on the diagram and select **Presentation Options** > **Show Shape Caption** > **Initial Node** from the popup menu.

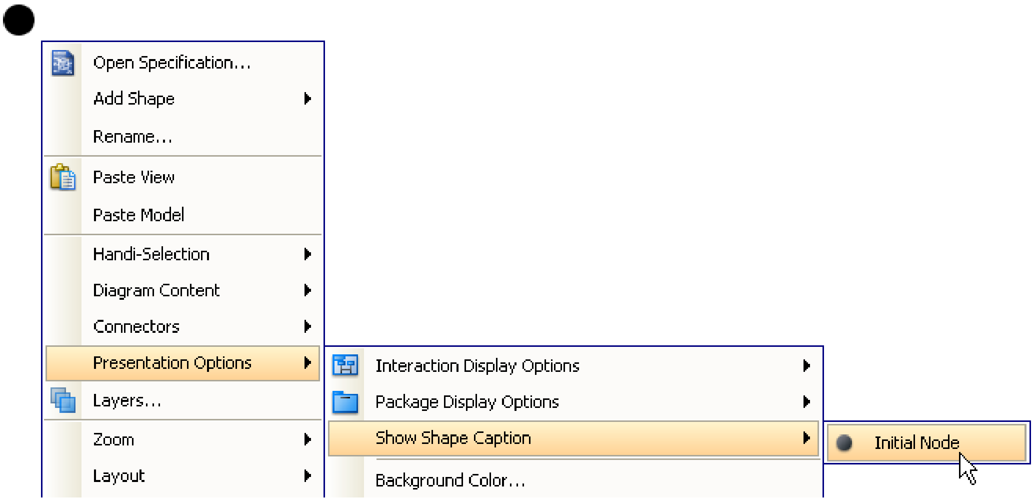


Figure 4-24 *Show caption of initial node*

Creating Decision Node

To create a decision node from an initial node, click on the **Generic Resource** beside it and drag.

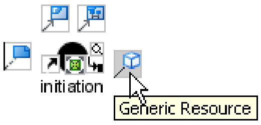


Figure 4-25 *Generic resource*

Move the mouse to empty space of the diagram and then release the mouse button. When a popup menu shows, select **Control Flow** -> **Decision Node**.

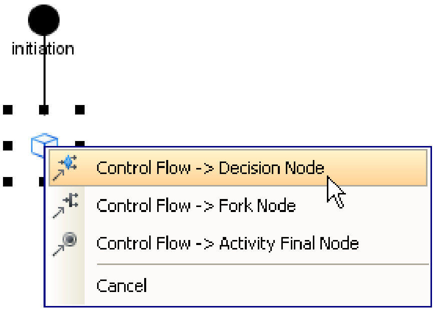

initiation
Control Flow -> Decision Node
Control Flow -> Fork Node
Control Flow -> Activity Final Node
Cancel

Figure 4-26 *Create decision node*

Right-click the diagram and select **Presentation Options** > **Show Shape Caption** > **Decision Node** from the popup menu to show caption of decision node.

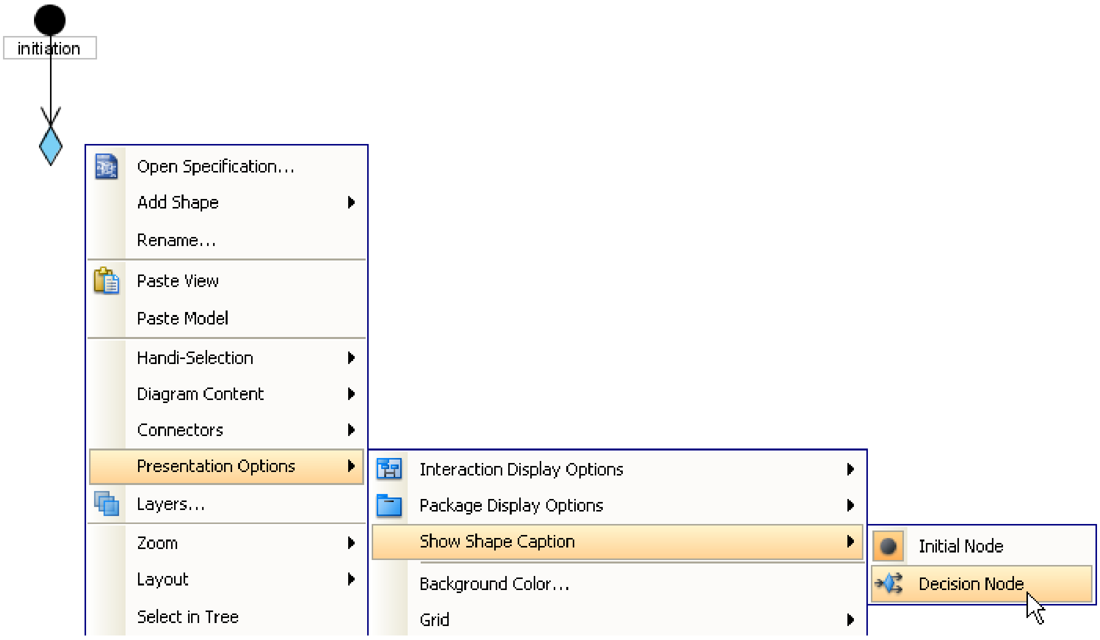


Figure 4-27 *Show caption of decision node*

Creating Interaction Use

To create an interaction use from a decision node, click on the **Control Flow** -> **Interaction Use** resource beside it and drag.

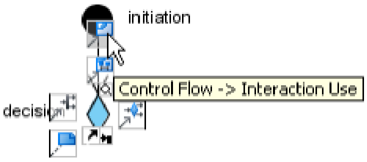


Figure 4-28 *Create interaction use*

Move the mouse to empty space of the diagram and then release the mouse button. An interaction use is created and connected to the decision node with a control flow.

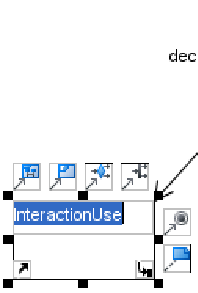


__Figure 4-29__ Interaction use and control flow created

Make the interaction use refers to a diagram by right-clicking on it and select **Refers to** > **New Sequence Diagram**.

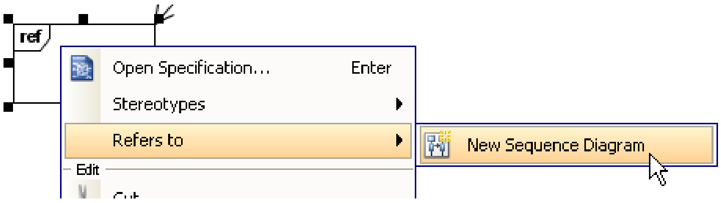


__Figure 4-30__ Make interaction use refers to diagram

When sequence diagram is created, rename the diagram.

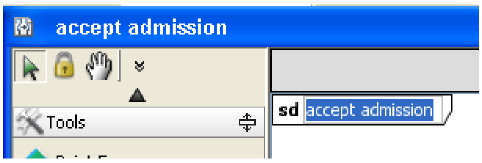


__Figure 4-30__ Rename sequence diagram

Return to the interaction overview diagram, the interaction use caption now reflects the diagram it refers to.

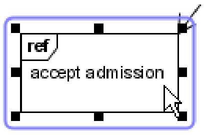


__Figure 4-31__ Interaction use caption updated

Creating Fork Node

To create a fork node from an interaction use, click on the **Control Flow** -> **Fork Node** resource beside it and drag.

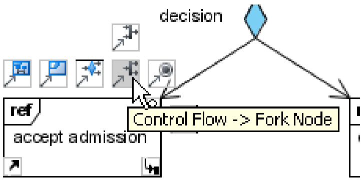


__Figure 4-32__ Create fork node

Move the mouse to empty space of the diagram and then release the mouse button. A fork node is created and connected to the interaction use with a control flow.

The fork node created is vertical by default, to change it to horizontal, right-click on the fork node and select **Orientation** > **Horizontal** from the popup menu.

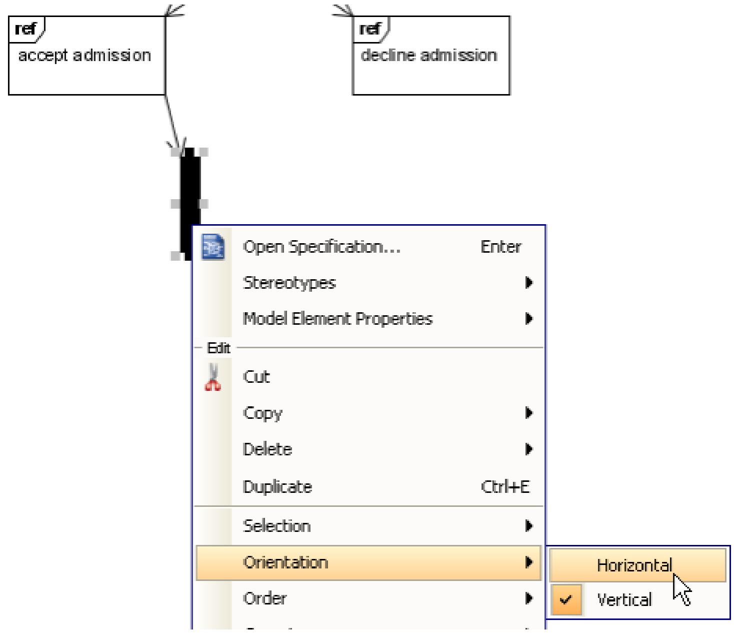


Figure 4-33 Change orientation of fork node

Right-click the diagram and select **Presentation Options** > **Show Shape Caption** > **Fork Node** from the popup menu to show caption of fork node.

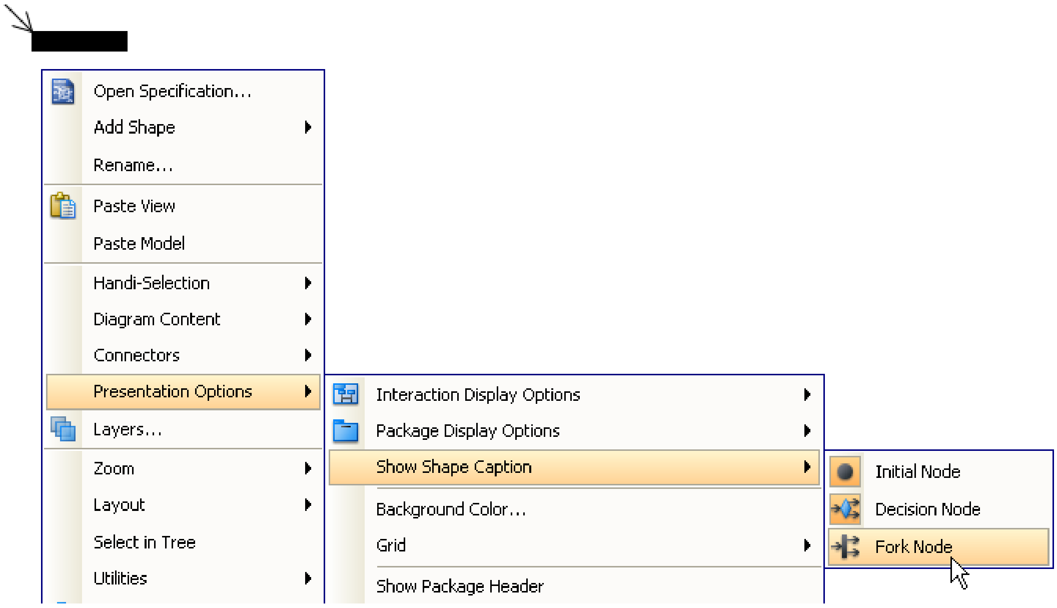


Figure 4-34 Show caption of fork node

Creating Interaction

To create an interaction from a fork node, click on the **Control Flow** -> **Interaction** resource beside it and drag.

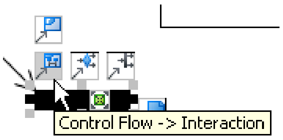


Figure 4-35 Create interaction

Move the mouse to empty space of the diagram and then release the mouse button. An interaction is created and connected to the fork node 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** > *<diagram name>*.

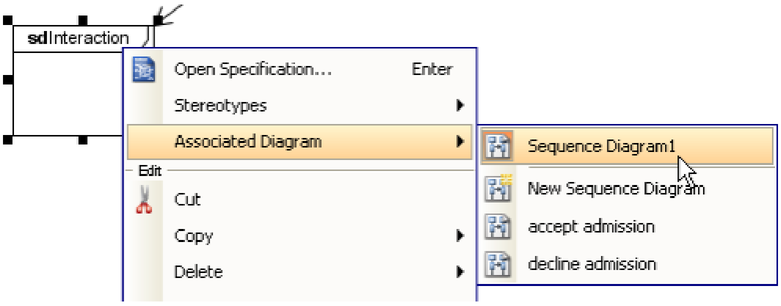


Figure 4-36 *Open associated diagram of interaction*

Draw the sequence diagram.

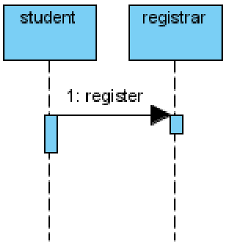


Figure 4-37
Sequence diagram

Return to the interaction overview diagram, the interaction now shows the thumbnail of the sequence diagram.

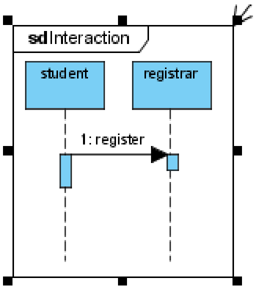


Figure 4-38 *Updated diagram thumbnail in interaction*

Creating Join Node

To create a join node from an interaction, click on the **Control Flow** -> **Join Node** resource beside it and drag.

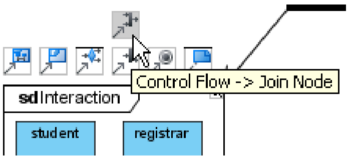


Figure 4-39 *Create join node*

Move the mouse to empty space of the diagram and then release the mouse button. A join node is created and connected to the interaction 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 popup menu.

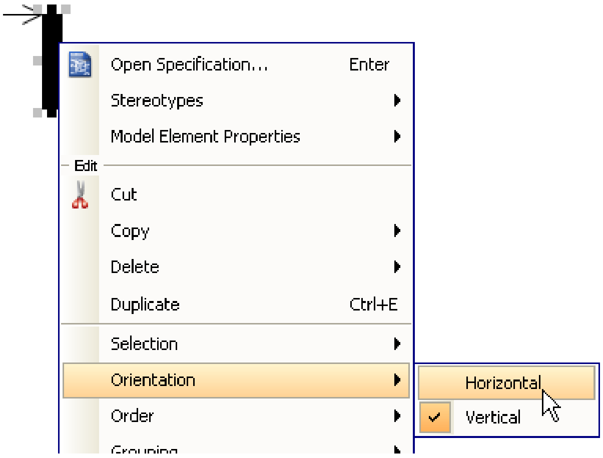


Figure 4-40 Change orientation of join node

Right-click the diagram and select **Presentation Options** > **Show Shape Caption** > **Join Node** from the popup menu to show caption of join node.

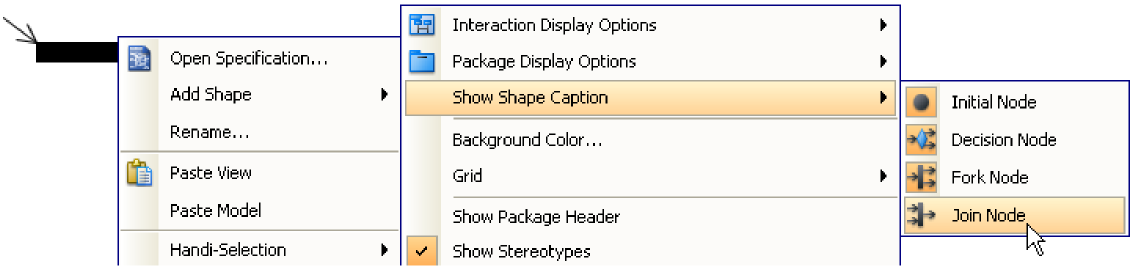


Figure 4-41 Show caption of join node

Creating Activity Final Node

To create an activity final node from an interaction use, click on the **Control Flow** -> **Activity Final Node** resource beside it and drag.

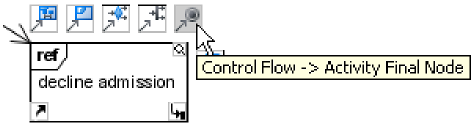


Figure 4-42 Create activity final node

Move the mouse to empty space of the diagram and then release the mouse button. An activity final node is created and connected to the interaction use with a control flow.

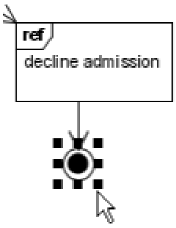


Figure 4-43 Activity final node and control flow created

Right-click the diagram and select **Presentation Options** > **Show Shape Caption** > **Activity Final Node** from the popup menu to show caption of activity final node.

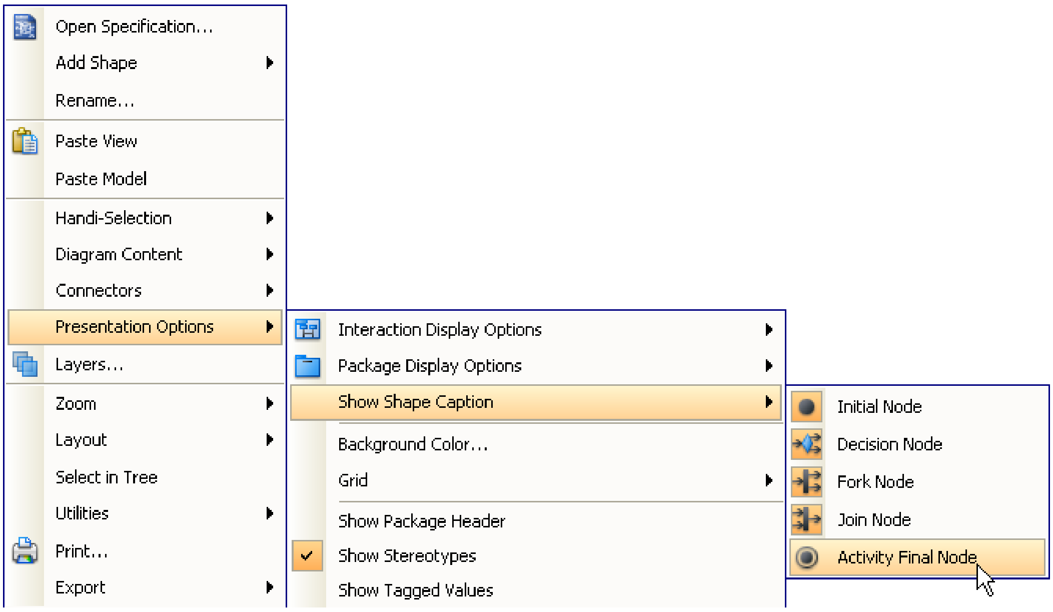


Figure 4-44 *Show caption of activity final node*

Continue to complete the diagram.

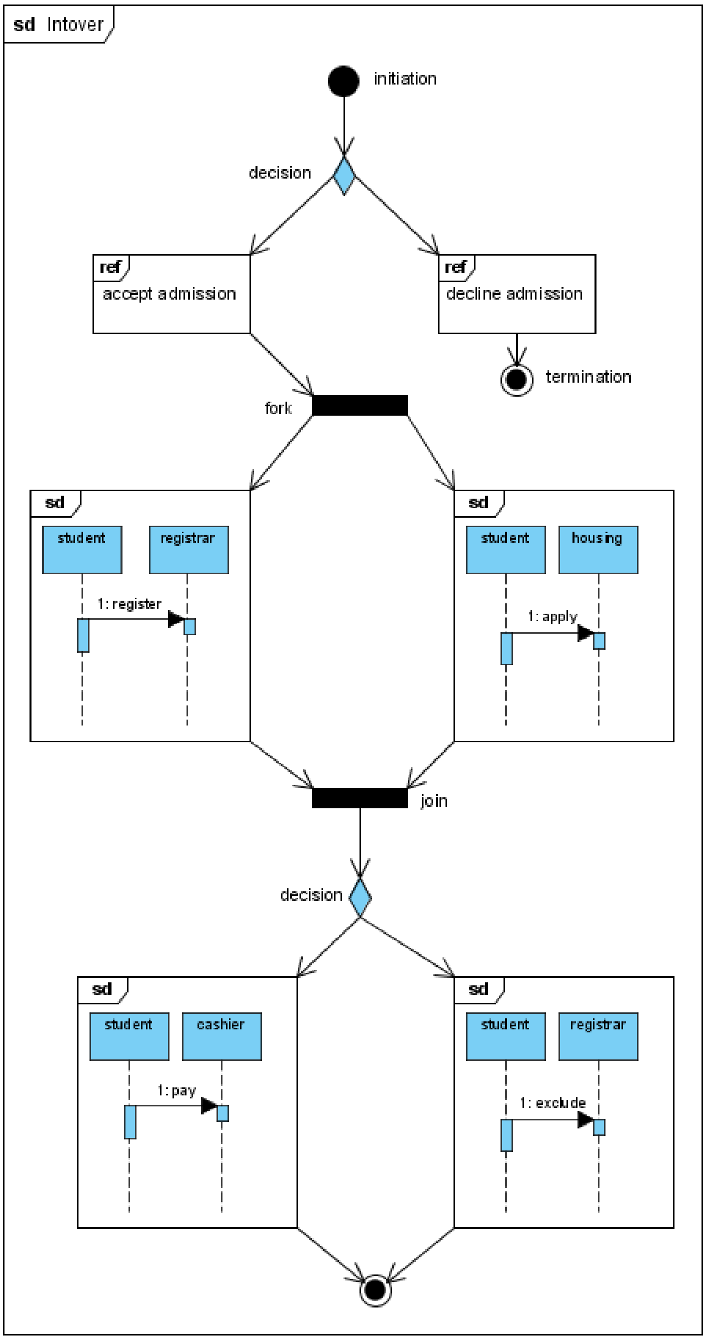


Figure 4-45 *Completed diagram*

Drawing Class Diagrams

Creating Class Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Class Diagram** to create a class diagram.

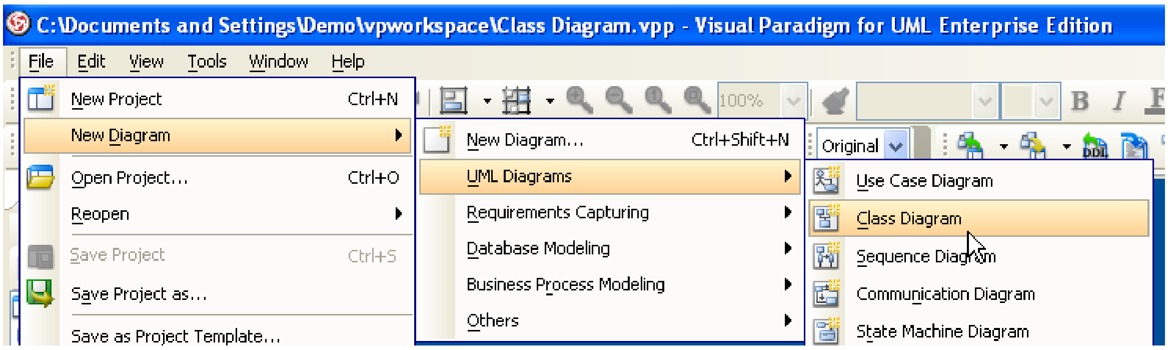


Figure 5-1 *Create class diagram*

Creating Class

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

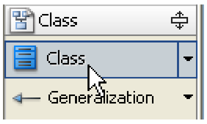


Figure 5-2 *Create class*

A class will be created.

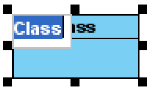


Figure 5-3 *Class created*

Creating Association

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

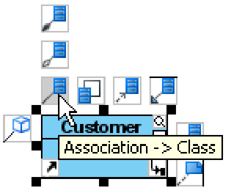


Figure 5-4 *Create association*

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.

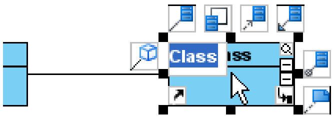


Figure 5-5 *Association created*

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

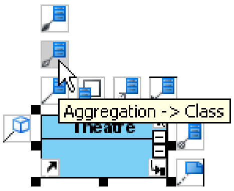


Figure 5-6 *Create aggregation*

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

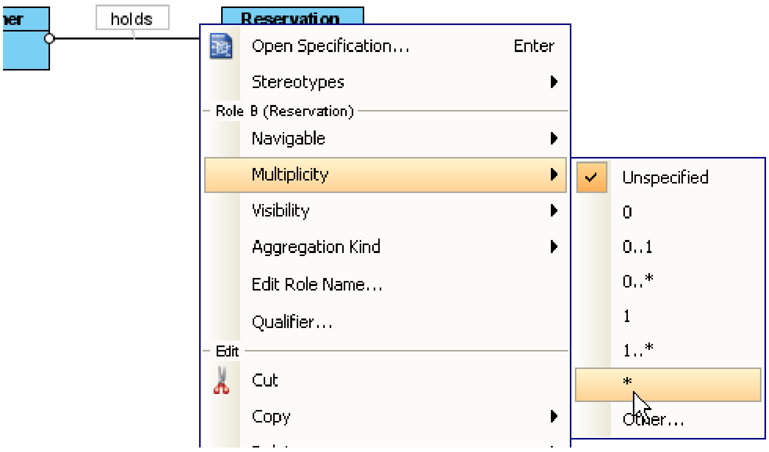


Figure 5-7 Edit multiplicity

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

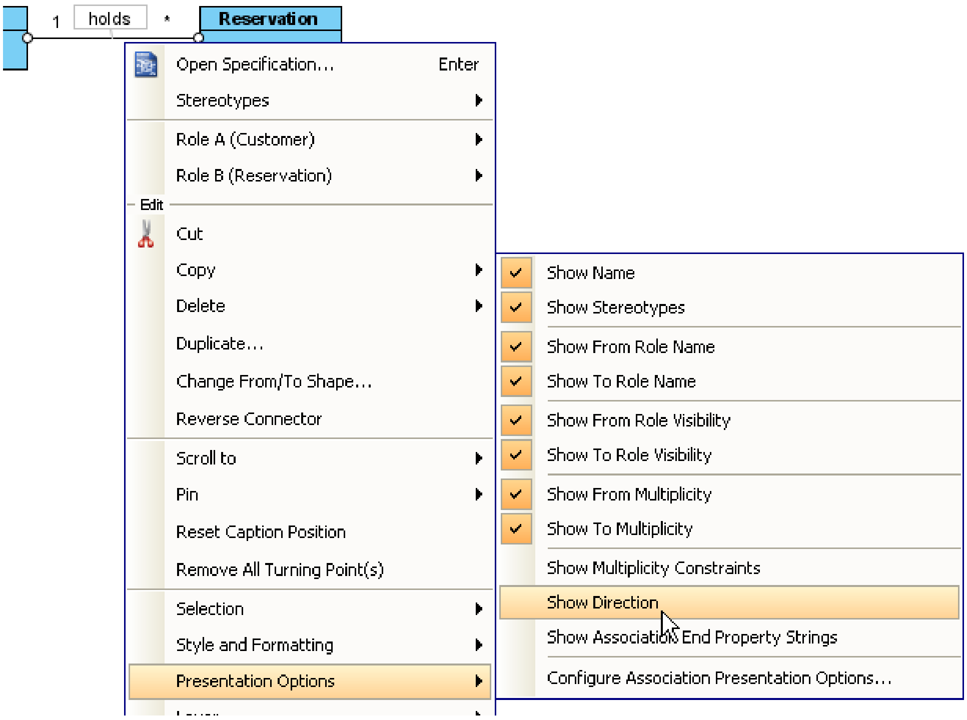


Figure 5-8 Show direction

The direction arrow is shown beside the association.

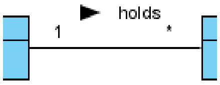


Figure 5-9 Direction shown

Creating Generalization

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

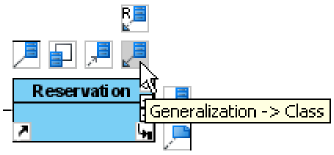


Figure 5-10 Create generalization

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.

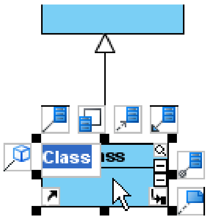


Figure 5-11 Generalization created

Creating Attribute

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

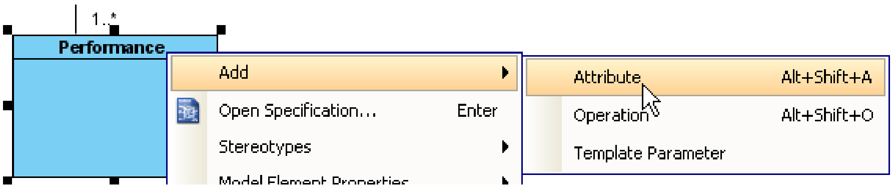


Figure 5-12 Create attribute

An attribute is created.

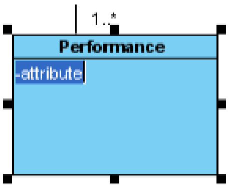


Figure 5-13 Attribute created

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.

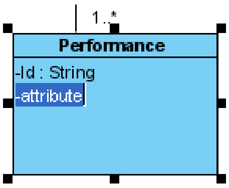


Figure 5-14 Create attribute with Enter key

Creating Operation

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

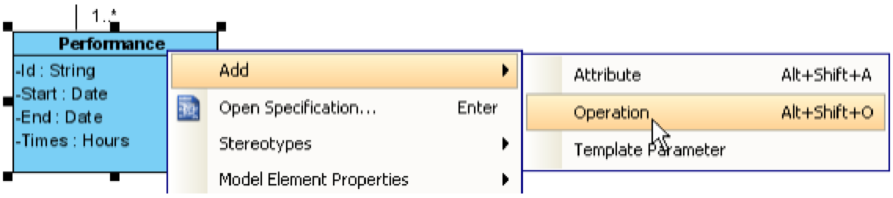


Figure 5-15 Create operation

An operation is created.

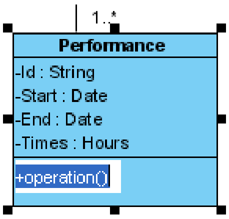


Figure 5-16 *Operation created*

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.

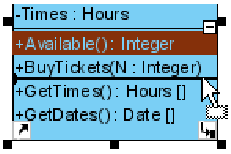


Figure 5-17 *Reorder class member*

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

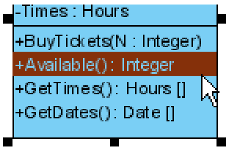


Figure 5-18 *Class member reordered*

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.

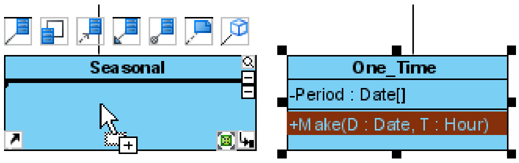


Figure 5-19 *Copy class member*

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

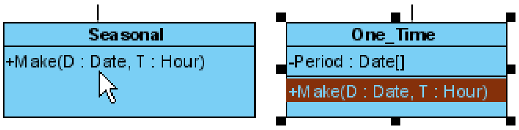


Figure 5-20 *Class member copied*

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.

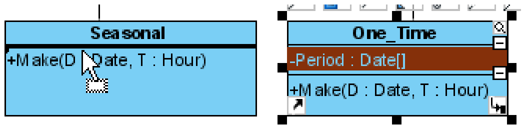


Figure 5-21 *Move class member*

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

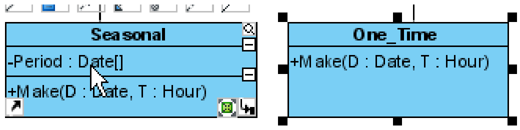


Figure 5-22 Class member moved

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.

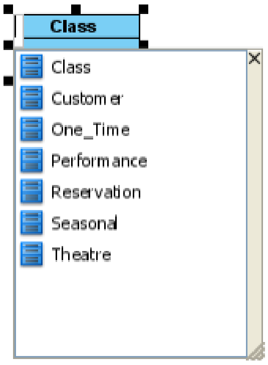


Figure 5-23 Model name completion

Type text to filter classes in the list.

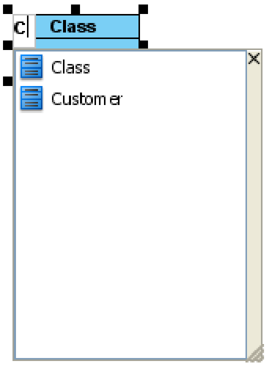


Figure 5-24 List filtered by typed text

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.

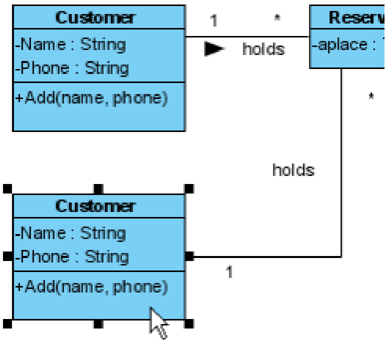


Figure 5-25 Multiple views of the same model

Continue to complete the diagram.

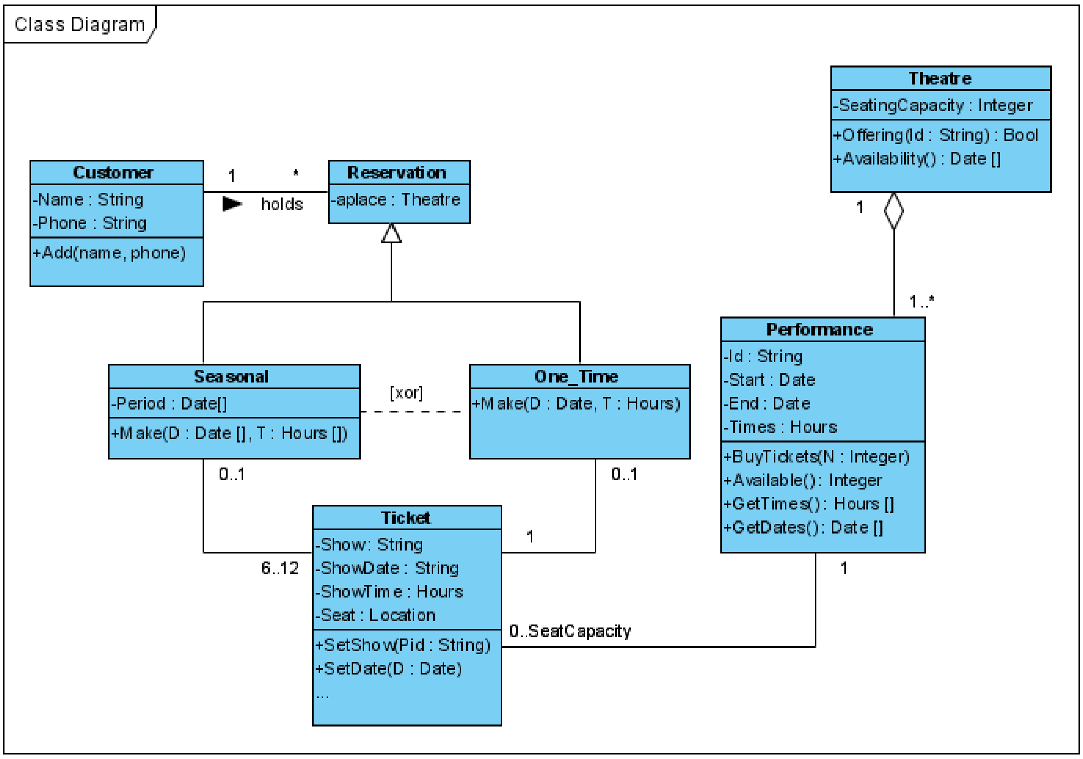


Figure 5-26 *Completed diagram*

Drawing Object Diagrams

Creating Object Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Object Diagram** to create an object diagram.

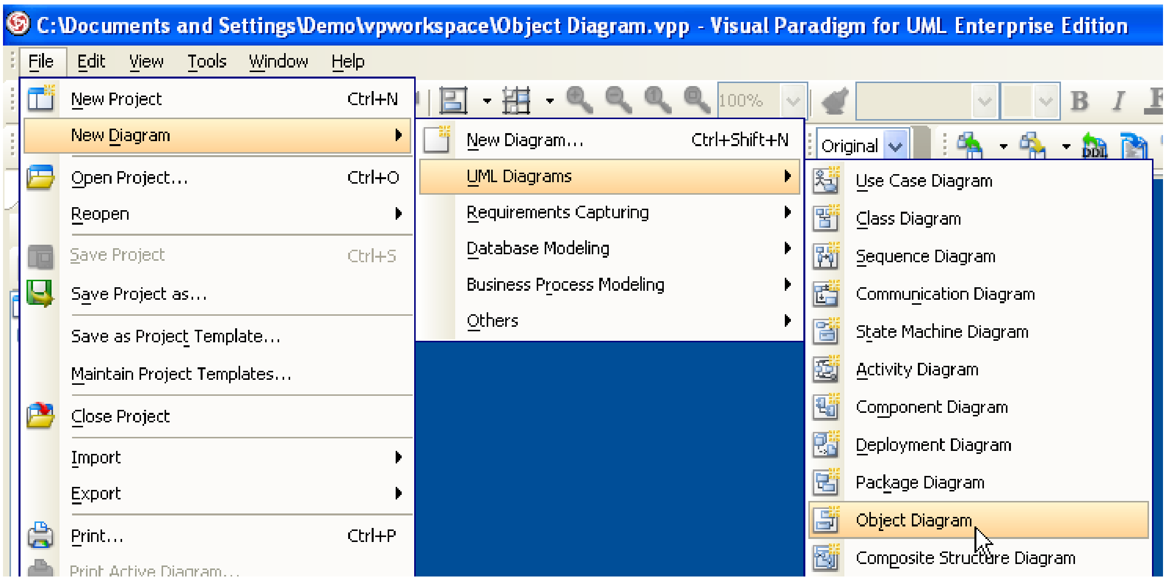


Figure 5-27 *Create object diagram*

Creating Instance Specification

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

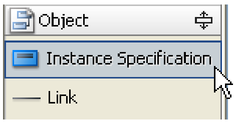


Figure 5-28 *Create instance specification*

An instance specification will be created.

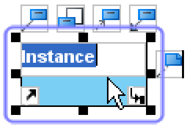


Figure 5-29 *Instance specification created*

Selecting Classifiers

To specify classifiers for an instance specification, right-click it and select **Select Classifier** > **Select Classifier...** from the popup menu.

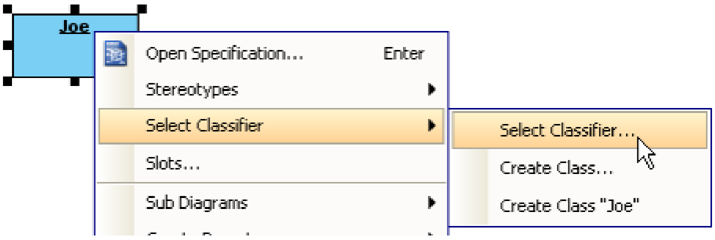


Figure 5-30 *Select classifier*

The **Instance Specification Specification** dialog box appears with the **Classifiers** tab selected. Select the classifiers on the left and click **Add Selected** to add them.

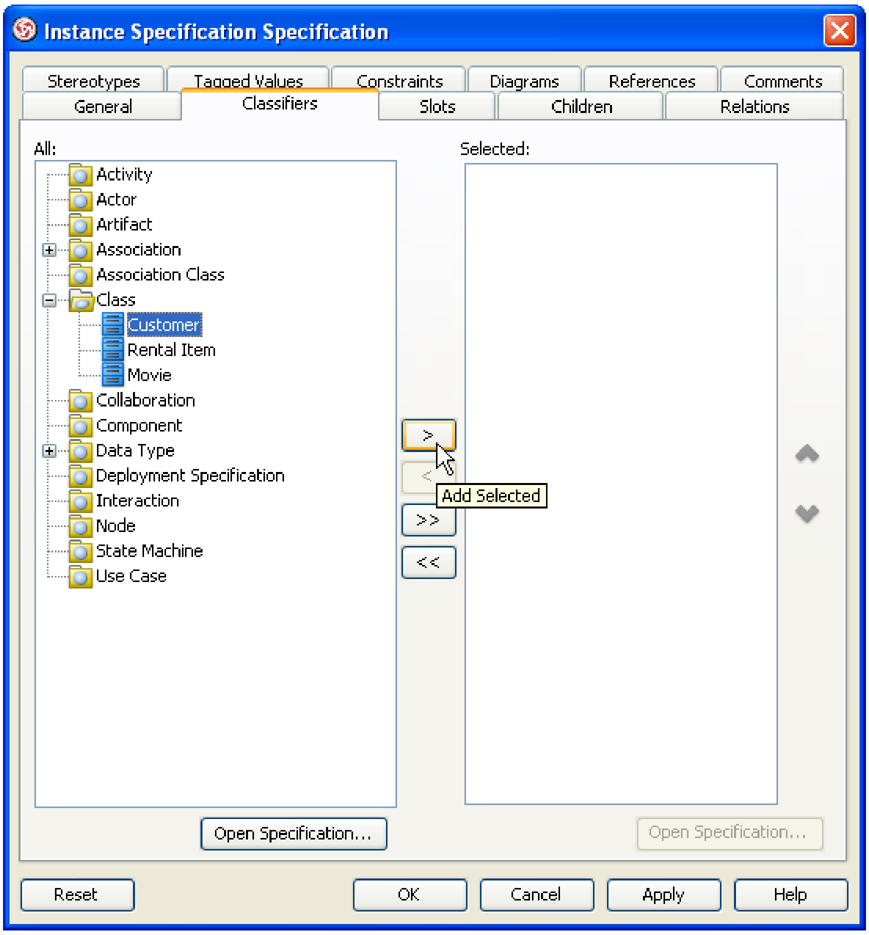


***Figure 5-31** Add selected classifiers*

Click **OK** to close the specification dialog box. The selected classifiers are assigned to the instance specification.

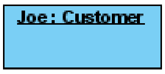


***Figure 5-32** Classifiers assigned*

Defining Slots

To define slots for an instance specification, right-click it and select **Slots. ..** from the popup menu.

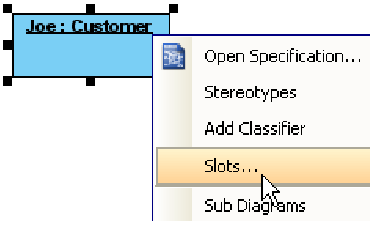
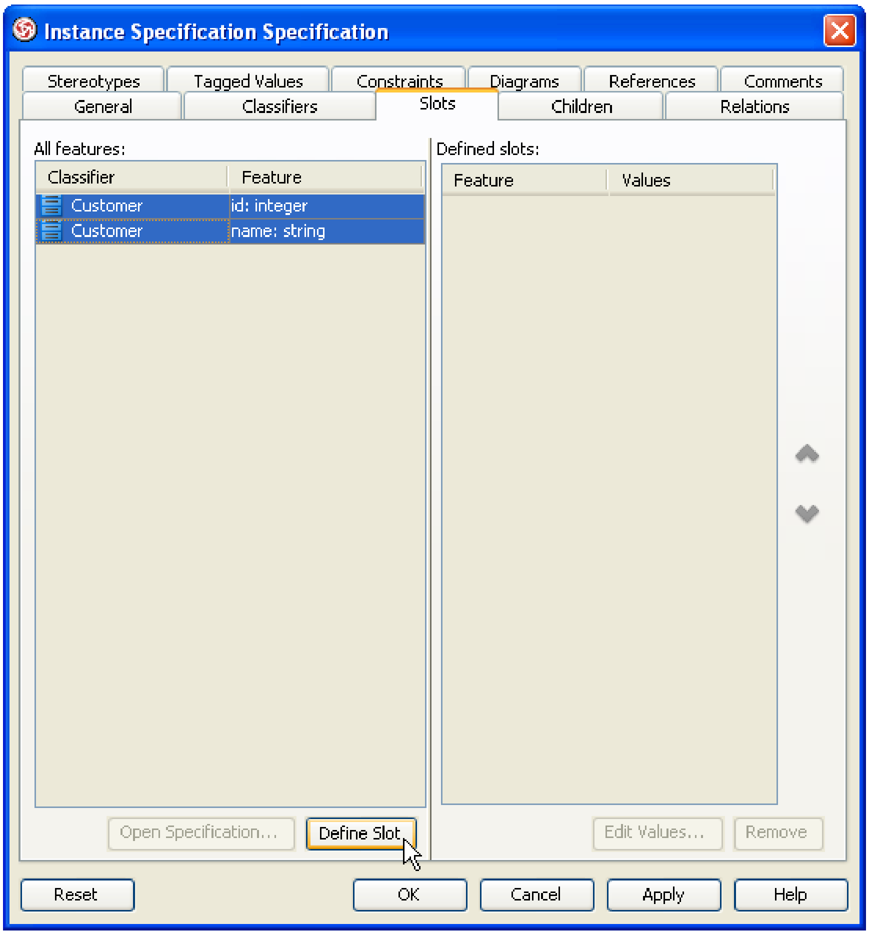


***Figure 5-33** Defining slots*

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**.



All features:

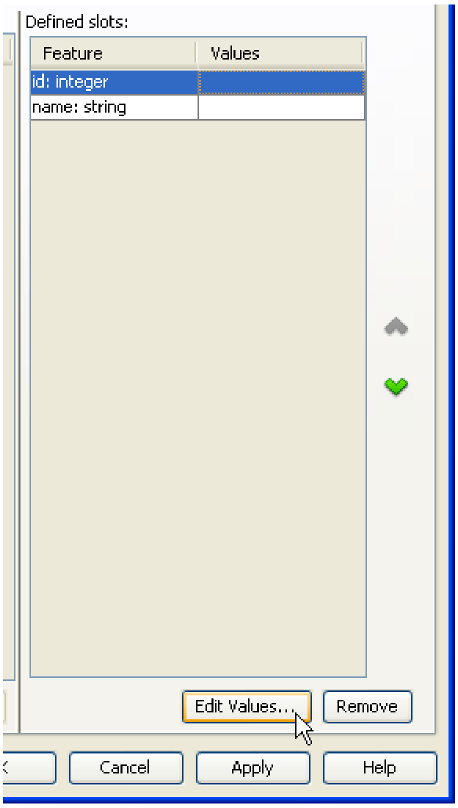
Classifier	Feature
Customer	id: integer
Customer	name: string

Defined slots:

Feature	Values

Figure 5-34 Select features to define slots

Select a slot in **Defined slots** and click **Edit Values...**.



Defined slots:

Feature	Values
id: integer	
name: string	

Figure 5-35 *Edit values of slot*

The **Slot Specification** dialog box appears with the **Values** tab selected. Click **Add** and select **Text** from the menu.

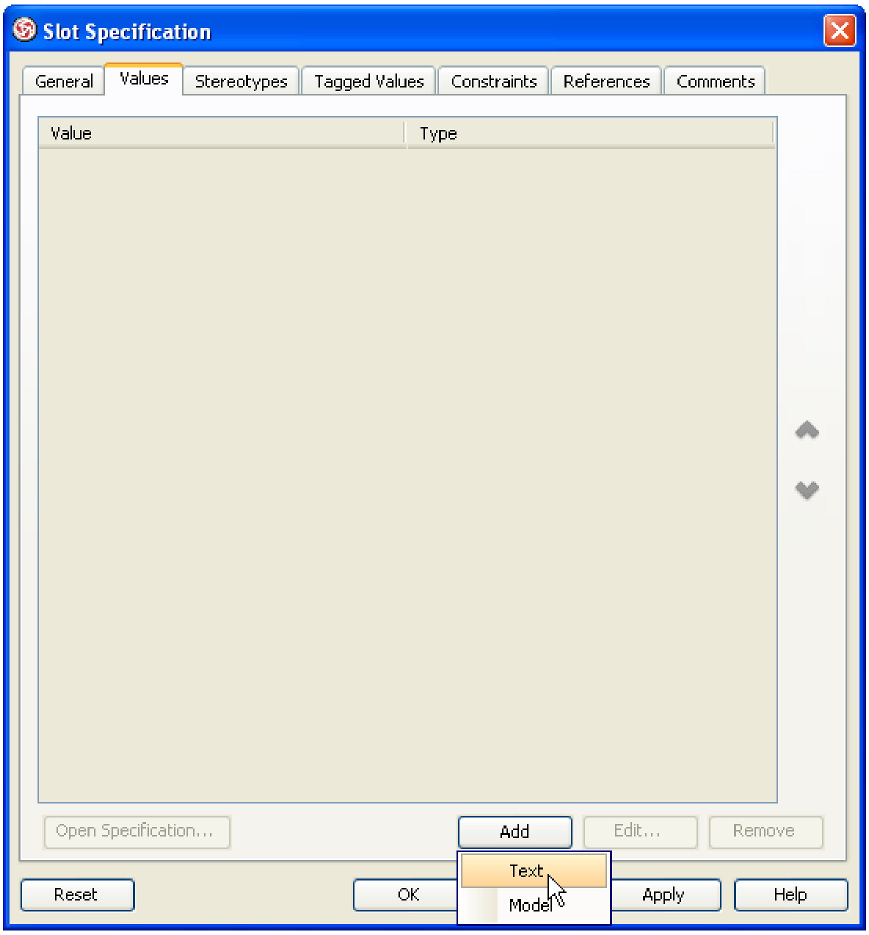


Figure 5-36 *Add text value*

Enter the value when prompted. Close the specification dialog boxes to apply the changes, defined slots will be shown in the instance specification.

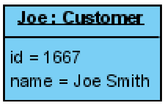


Figure 5-37 *Defined slots shown*

Creating Link

To create link from instance specification, click the **Link** -> **Instance Specification** resource beside it and drag.

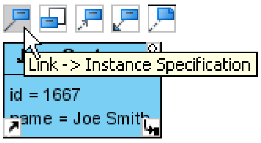


Figure 5-38 *Create link*

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.

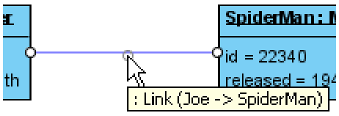


Figure 5-39 *Link created*

Continue to complete the diagram.

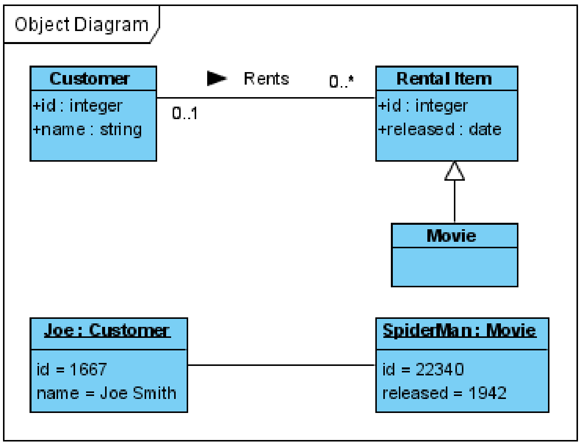


Figure 5-40 *Completed diagram*

Drawing Package Diagrams

Creating Package Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > Package **Diagram** to create a package diagram.

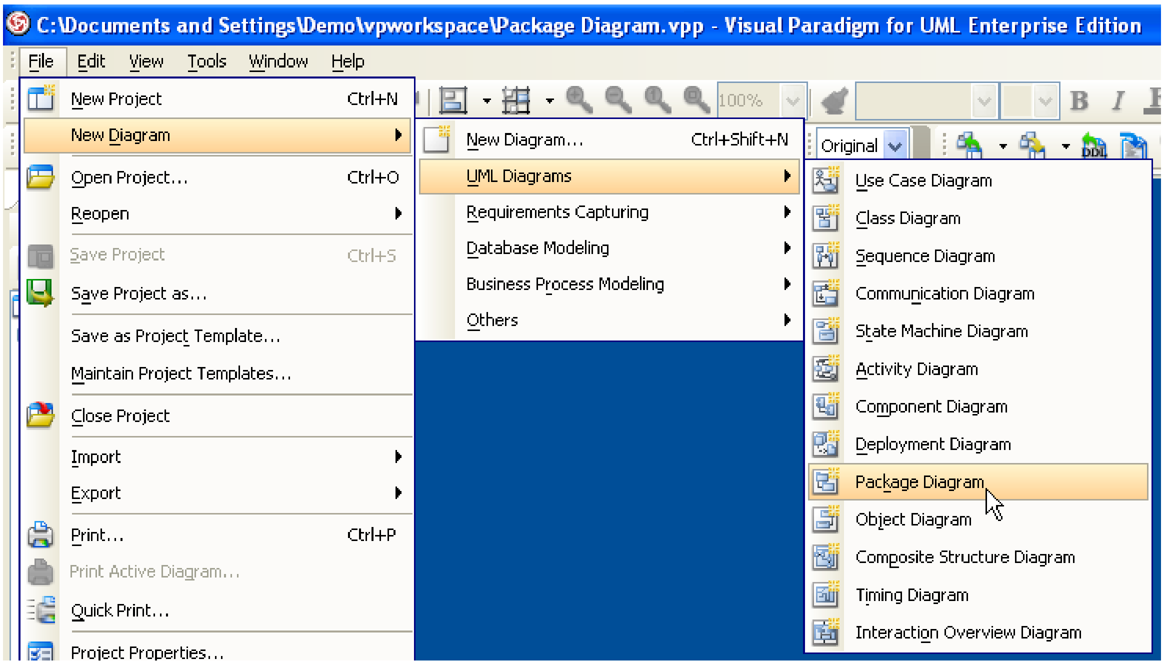


Figure 5-41 *Create package diagram*

Creating Package

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

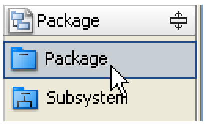


Figure 5-42 *Create package*

A package will be created.

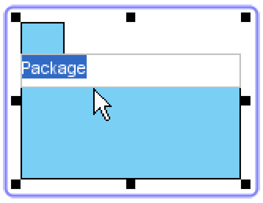


Figure 5-43 *Package created*

Assigning Stereotypes

Right-click on the package and select **Stereotypes** > **Stereotypes**... from the popup menu.

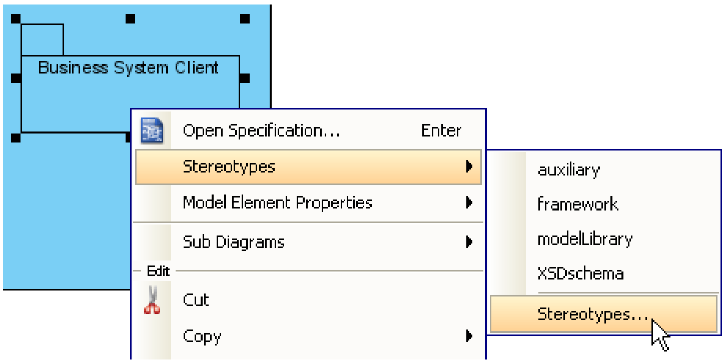


Figure 5-44 *Assigning stereotypes*

The **Package Specification** dialog box appears with the **Stereotypes** tab selected. The list on the left shows the selectable stereotypes.

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

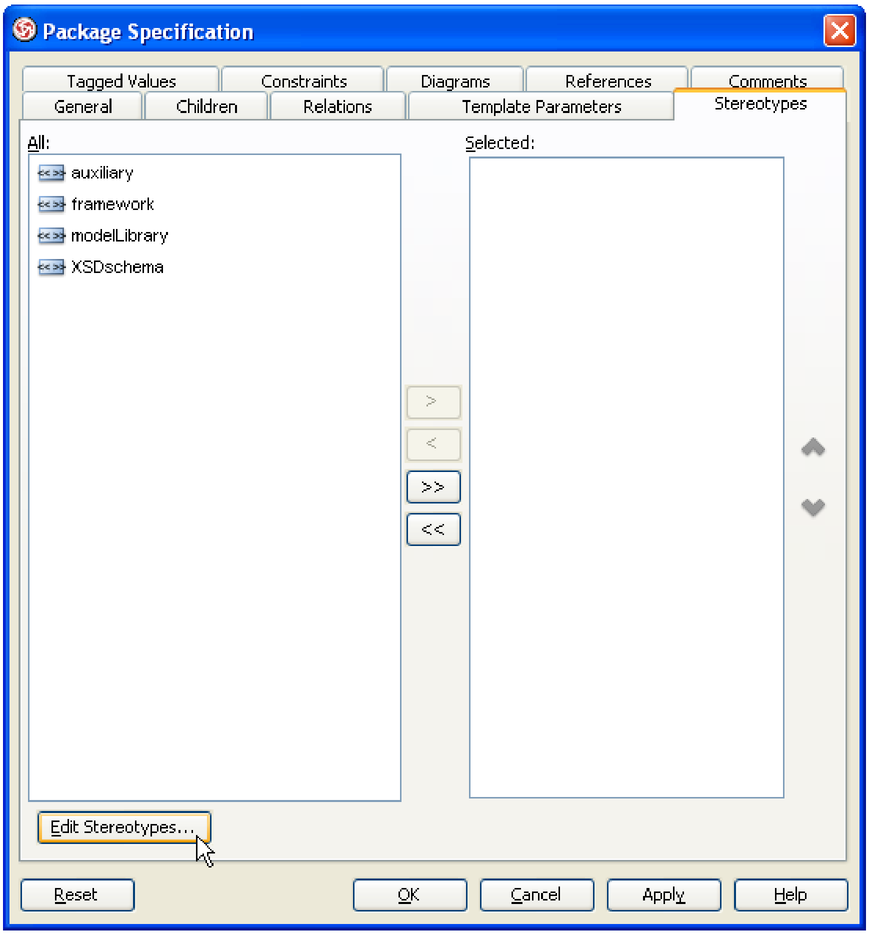
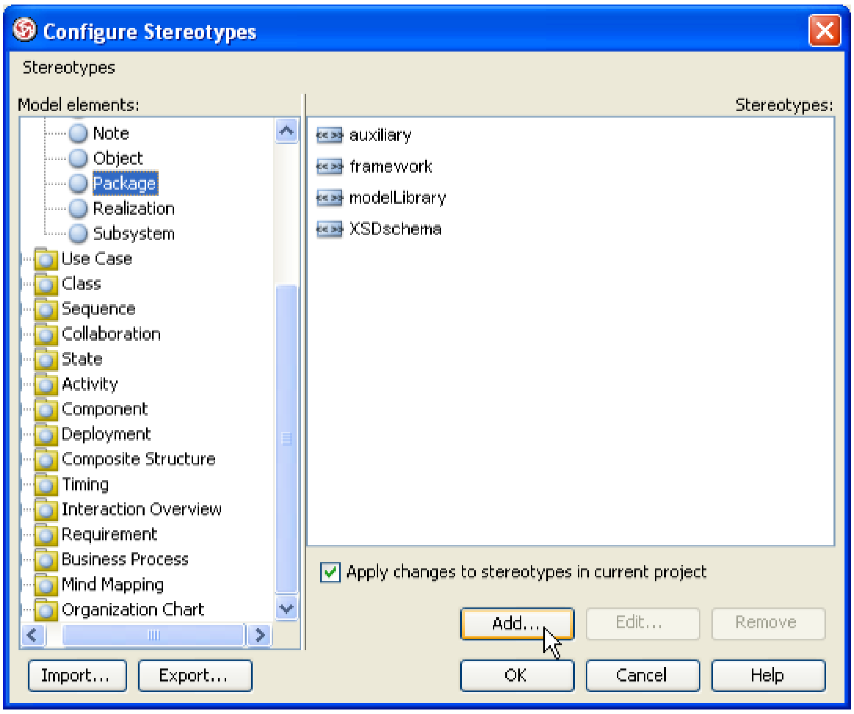


Figure 5-45 *Edit stereotypes*

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



☑ Apply changes to stereotypes in current project

Figure 5-46 Add stereotype

Name the stereotype (e.g. *facade*). Close the Stereotype Specification and the Configure Stereotypes dialog box. You will see the added stereotype appears on the list. Select it and click **Add Selected**.

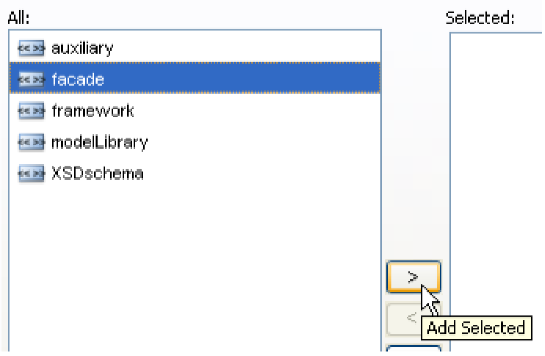


Figure 5-47 Add selected stereotypes

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

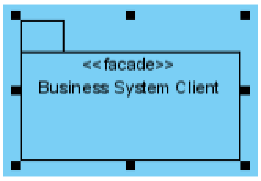


Figure 5-48 Stereotypes assigned

Continue to complete the diagram.

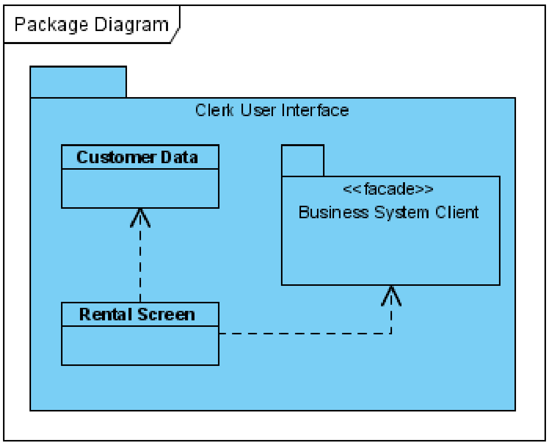


Figure 5-49 *Completed diagram*

Drawing Composite Structure Diagrams

Creating Composite Structure Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > Composite Structure **Diagram** to create a composite structure diagram.

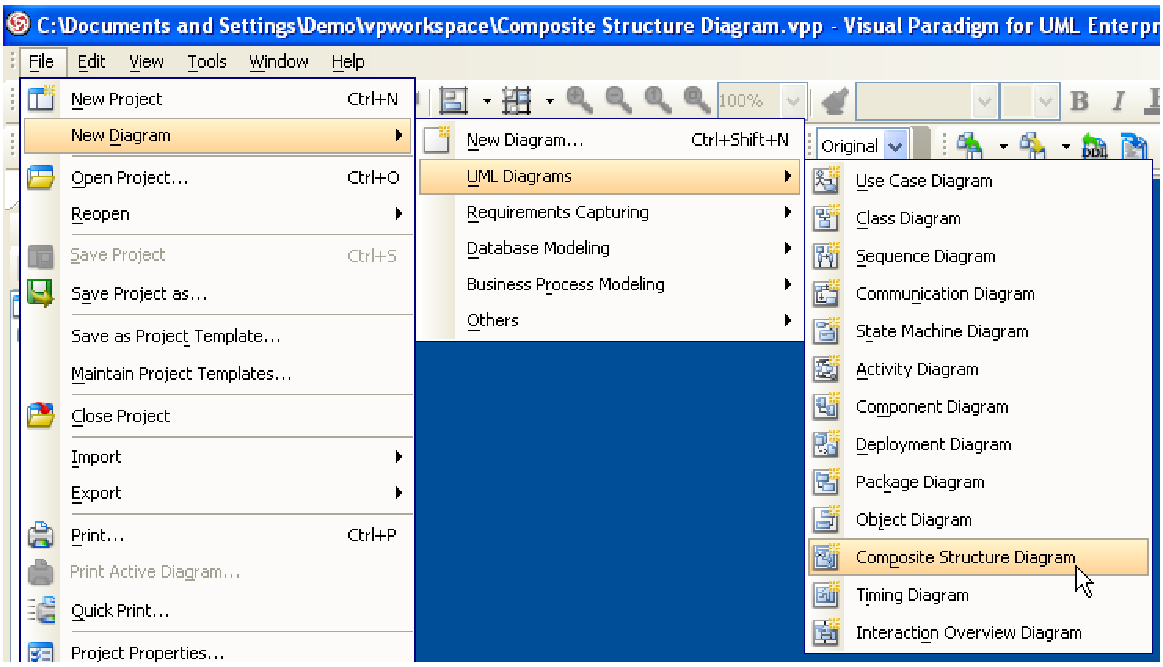


Figure 5-50 *Create composite structure diagram*

Creating Class

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

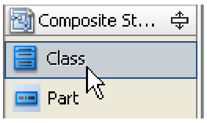


Figure 5-51 *Create class*

A class will be created.

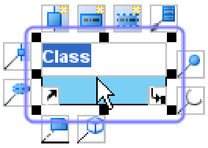


Figure 5-52 *Class created*

Creating Part

To create part, click the **New Part** resource of class.

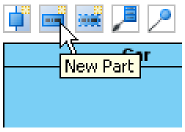


Figure 5-53 *Create part*

A part will be created.

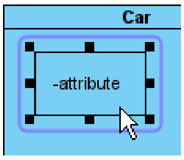


Figure 5-54 *Part created*

Creating Port

To create port, click the **New Port** resource of class.

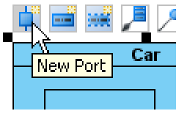


Figure 5-55 *Create port*

A port will be created.

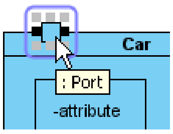


Figure 5-56 *Port created*

Specifying Type of Port

Right-click the port and select **Open Specification...** from the popup menu. The **Port Specification** dialog box appears.

Click the combo box beside the **Type** field, select a class.

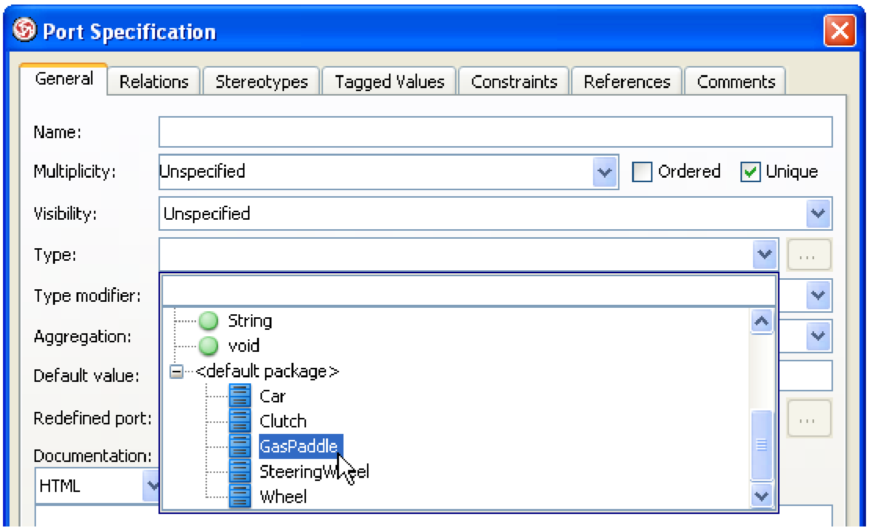


Figure 5-57 *Select type*

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

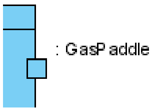


Figure 5-58 *Type shown on port*

Creating Connector

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

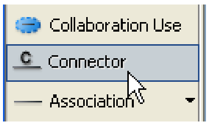


Figure 5-59 *Create connector*

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

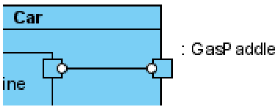


Figure 5-60 *Connector created*

Continue to complete the diagram.

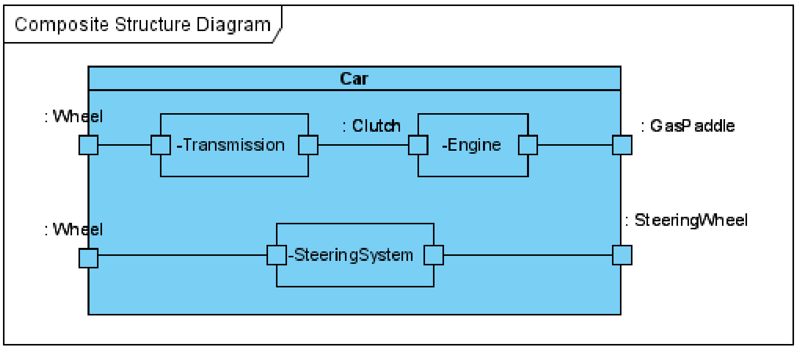


Figure 5-61 Completed diagram

Drawing Component Diagrams

Creating Component Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Component Diagram** to create a component diagram.

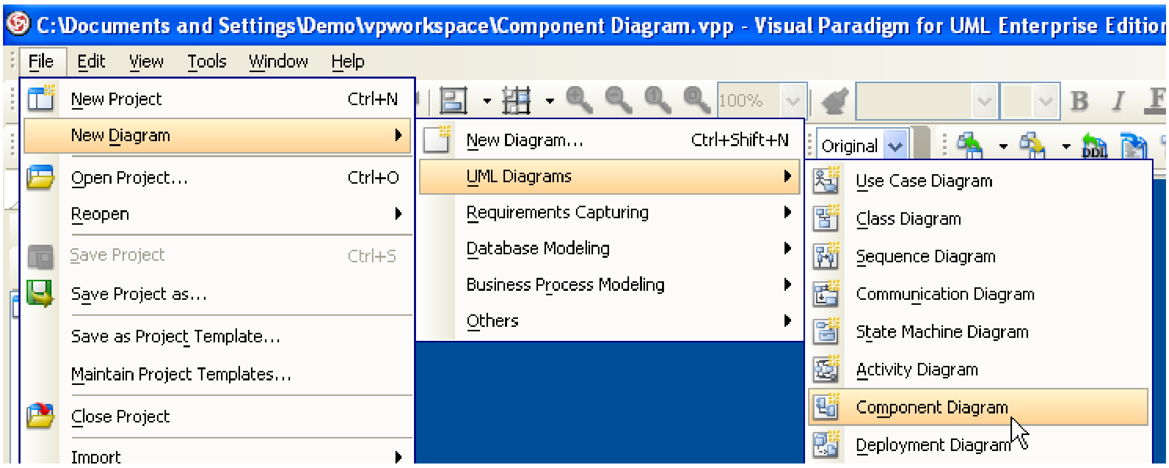


Figure 6-1 Create component diagram

Creating Component

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

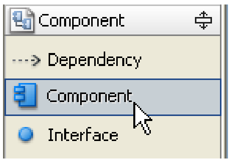


Figure 6-2 Create component

A component will be created.

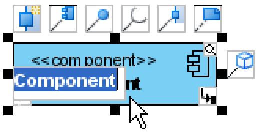


Figure 6-3 Component created

Assigning Stereotypes

Righ t-click on the package and select **Stereotypes** > **Stereotypes**... from the popup menu.

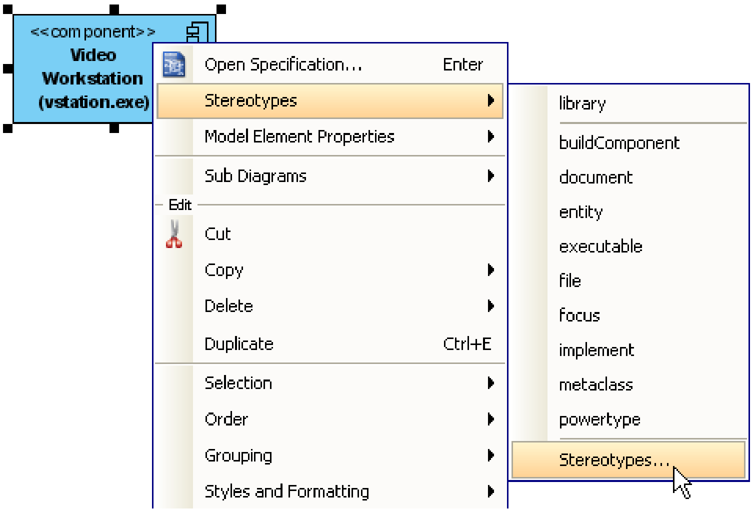


Figure 6-4 Assigning stereotypes

The **Component Specification** dialog box appears with the **Stereotypes** tab selected. The list on the left shows the selectable stereotypes.

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

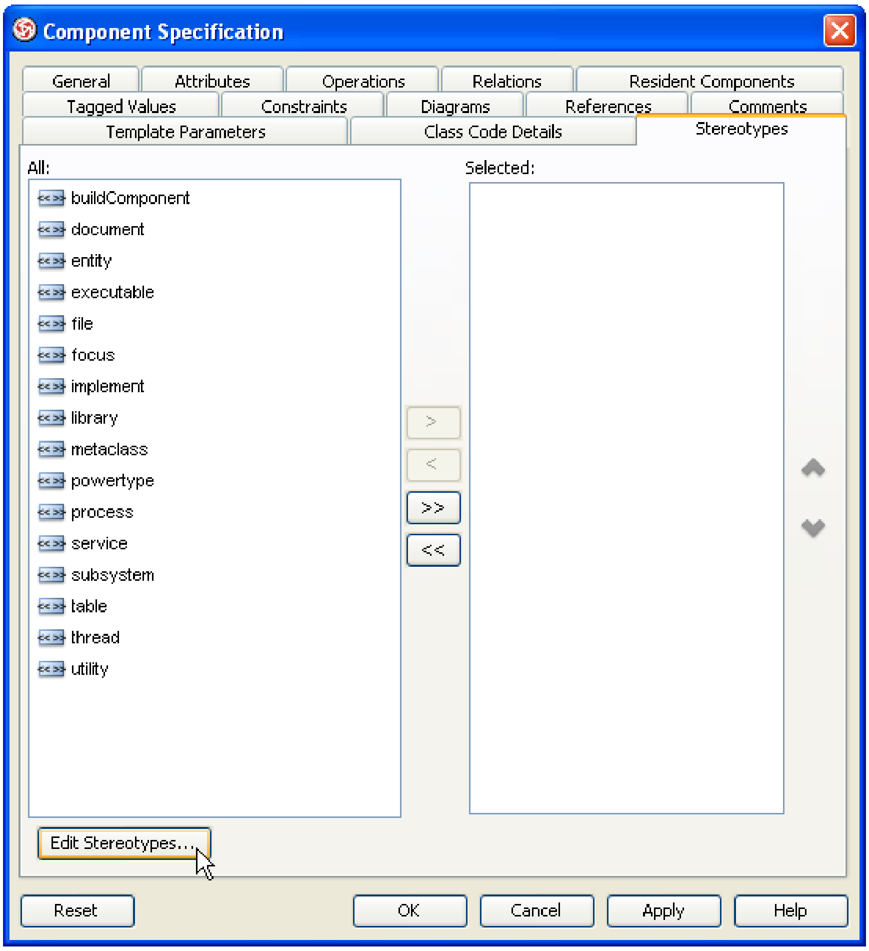


Figure 6-5 *Edit stereotypes*

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

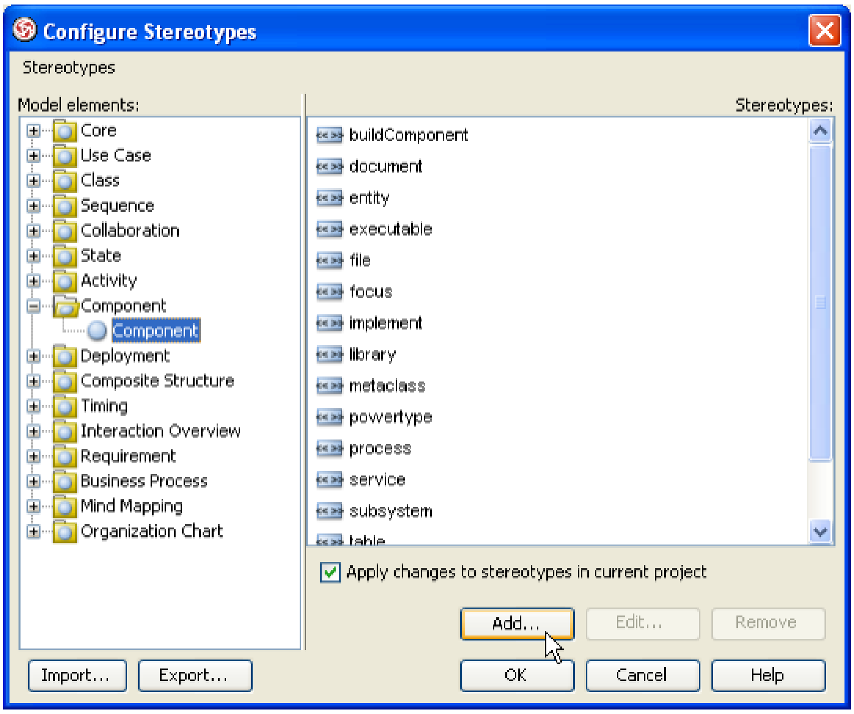


Figure 6-6 *Add stereotype*

Name the stereotype (e.g. *application*). Close the Stereotype Specification and the Configure Stereotypes dialog box. You will see the added stereotype appears on the list. Select it and click **Add Selected**.

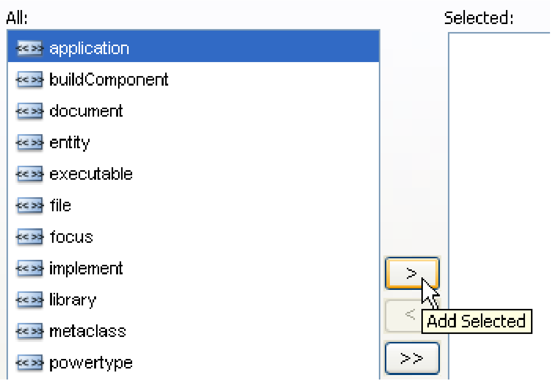


Figure 6-7 *Add selected stereotypes*

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

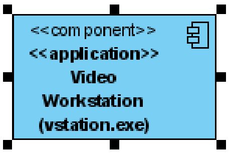


Figure 6-8 *Stereotypes assigned*

Creating Provided Interface

To create provided interface for a component, click on the **Realization** -> **Interface** resource beside it and drag.

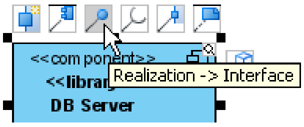


Figure 6-9 *Create provided interface*

Move the mouse to empty space of the diagram and then release the mouse button. The provided interface is created.

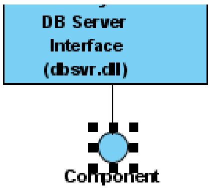


Figure 6-10 *Provided interface created*

Creating Required Interface

To create required interface for a component, click on the **Usage**-> **Interface** resource beside it and drag.

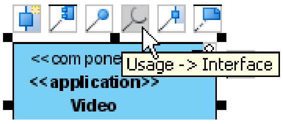


Figure 6-11 *Create required interface*

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

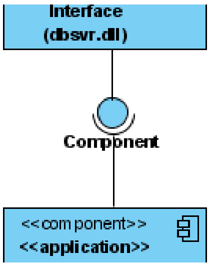


Figure 6-12 *Required interface created*

Creating Dependency

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

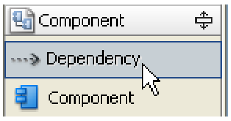


Figure 6-13 *Create dependency*

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

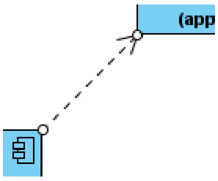


Figure 6-14 *Dependency created*

Continue to complete the diagram

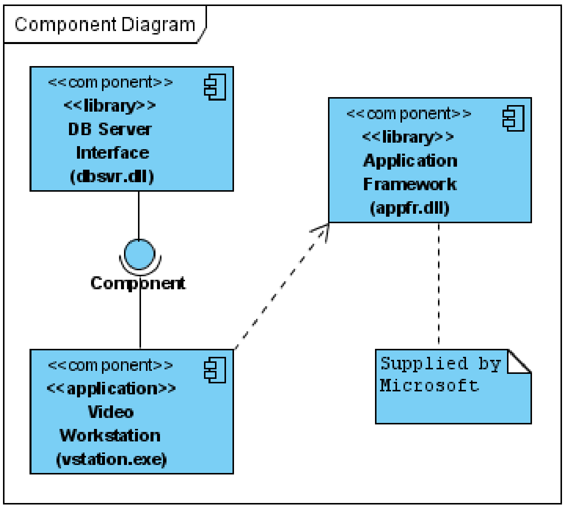


Figure 6-15 *Completed diagram*

Drawing Deployment Diagrams

Creating Deployment Diagram

Select menu **File** > **New Diagram** > **UML Diagrams** > **Deployment Diagram** to create a deployment diagram.

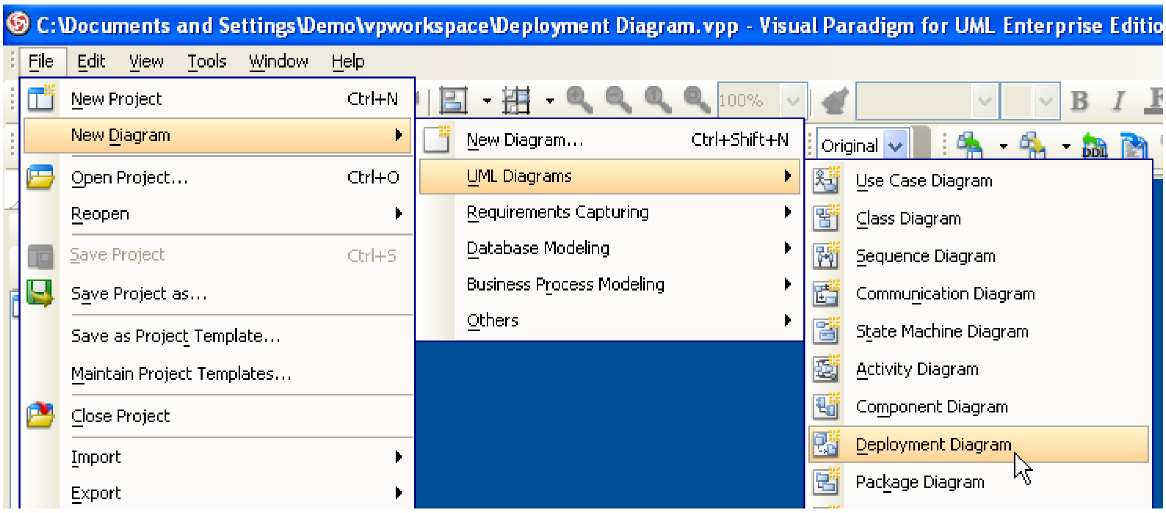


Figure 6-16 *Create deployment diagram*

Creating Node Model Element

To create node model element, right-click on empty space of **Model Explorer** and select **Model Element** > **New Model Element** from the popup menu.

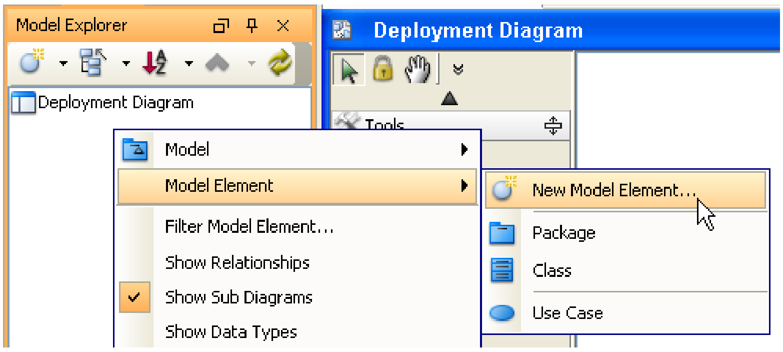


Figure 6-17 *Create model element*

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

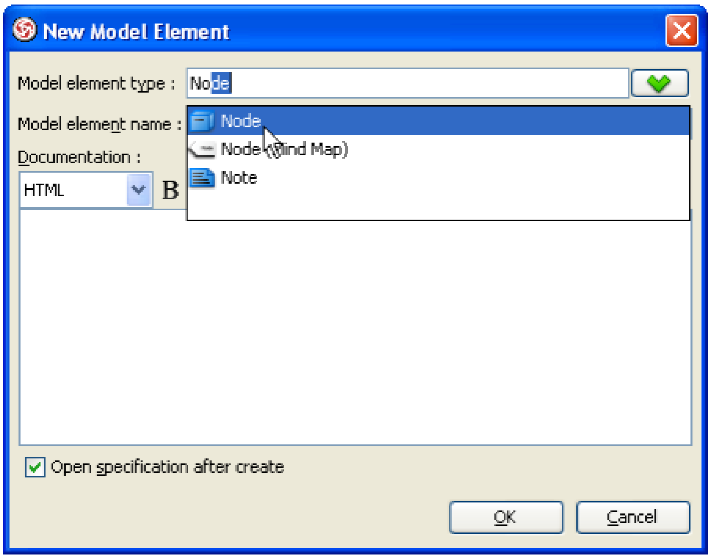


Figure 6-18 *Create node*

A node model element will be created.

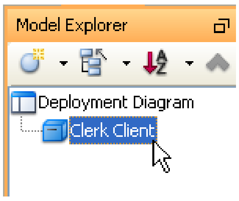


***Figure 6-19** Node created*

Creating Instance of Node

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

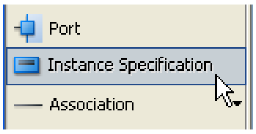


***Figure 6-20** Create instance specification*

An instance specification will be created.

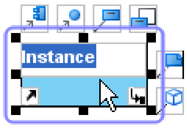


***Figure 6-21** Instance specification created*

Selecting Classifiers

To specify classifiers for an instance specification, right-click it and select **Select Classifier** > **Select Classifier...** from the popup menu.

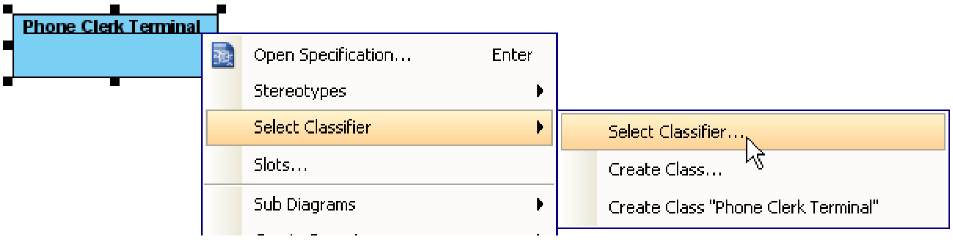


***Figure 6-22** Select classifier*

The **Instance Specification Specification** dialog box appears with the **Classifiers** tab selected. Select the classifiers on the left and click **Add Selected** to add them.

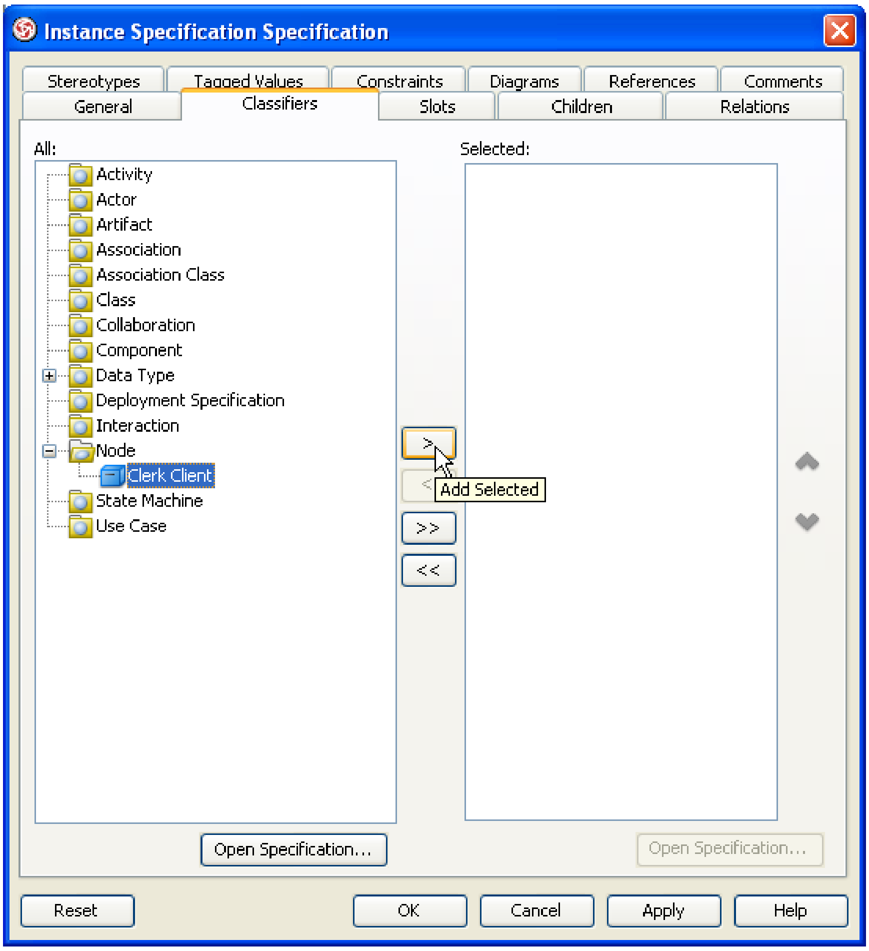


Figure 6-23 *Add selected classifiers*

Click **OK** to close the specification dialog box. The selected classifiers are assigned to the instance specification.

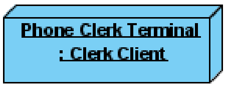


Figure 6-24 *Classifiers assigned*

Creating Link

To create link from instance specification, click the **Link** -> **Instance Specification** resource beside it and drag.

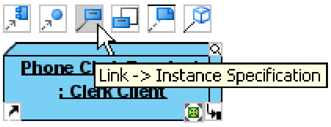


Figure 6-25 *Create link*

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.

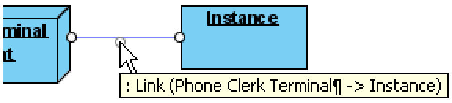


Figure 6-26 *Link created*

Creating Instance of Component

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

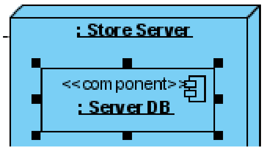


Figure 6-27 *Instance of component*

Creating Dependency

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

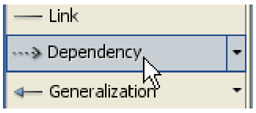


Figure 6-28 *Create dependency*

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

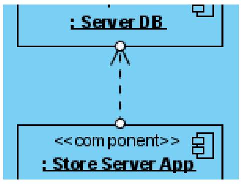


Figure 6-29 *Dependency created*

Continue to complete the diagram.

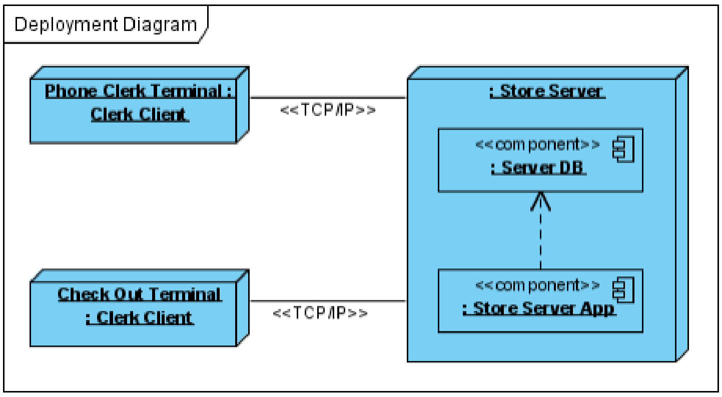


Figure 6-30 *Completed diagram*

Drawing Business Process Diagrams

Creating Business Process Diagram

Select menu **File** > **New Diagram** > **Business Process Modeling** > **Business Process Diagram** to create a business process diagram.

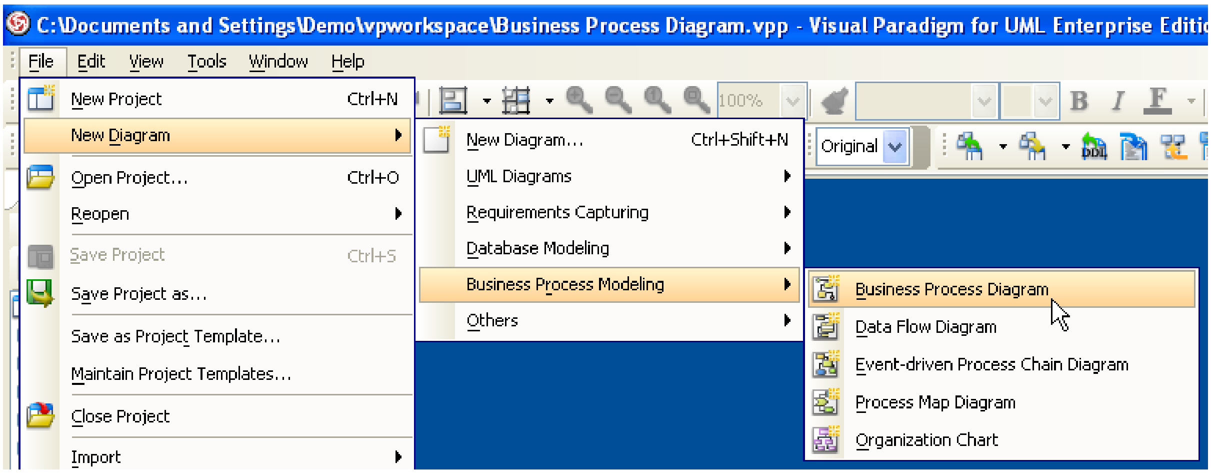


Figure 1-1 *Create business process diagram*

Creating Pool

To create pool, click **Horizontal Pool** on the diagram toolbar and then click on the diagram.

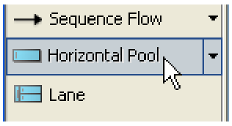


Figure 1-2 *Create pool*

Adding Lane to Pool

To add lane to pool, right-click the pool and select **Add Lane** from the popup menu.

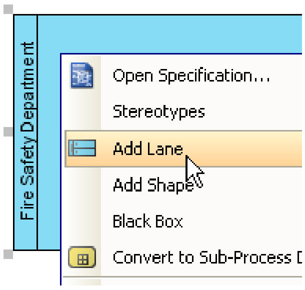


Figure 1-3 *Create lane*

Creating Start Event

To create start event, click **Start Event** on the diagram toolbar and then click on the diagram.

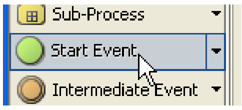


Figure 1-4 *Create start event*

Creating Task

To create task from a start event, click on the **Sequence Flow** -> **Task** resource beside it and drag.

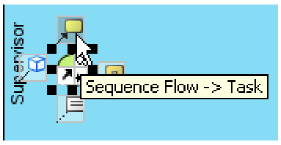


Figure 1-5 *Create task*

Move the mouse to where you want to place the shape to and then release the mouse button. A task is created and connected to the start event with a sequence flow.

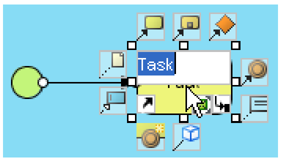


Figure 1-6 *Task and sequence flow created*

Creating Message Flow

To create message flow, click **Message Flow** on the diagram toolbar.

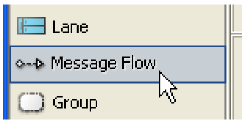


Figure 1-7 *Create message flow*

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

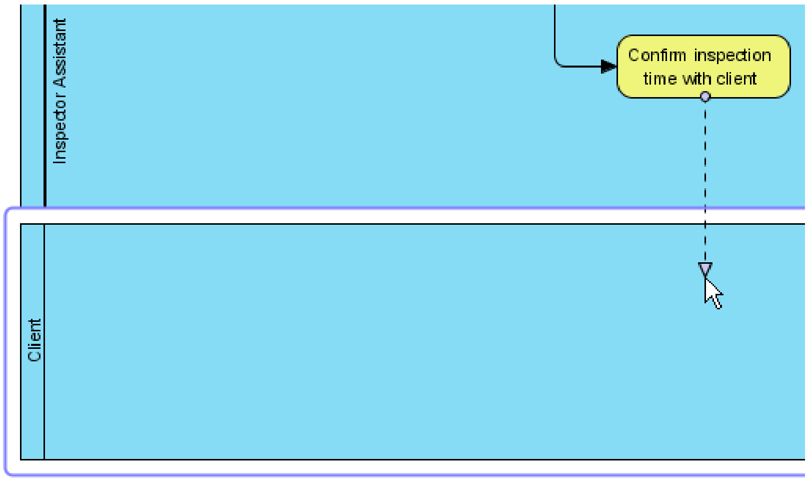
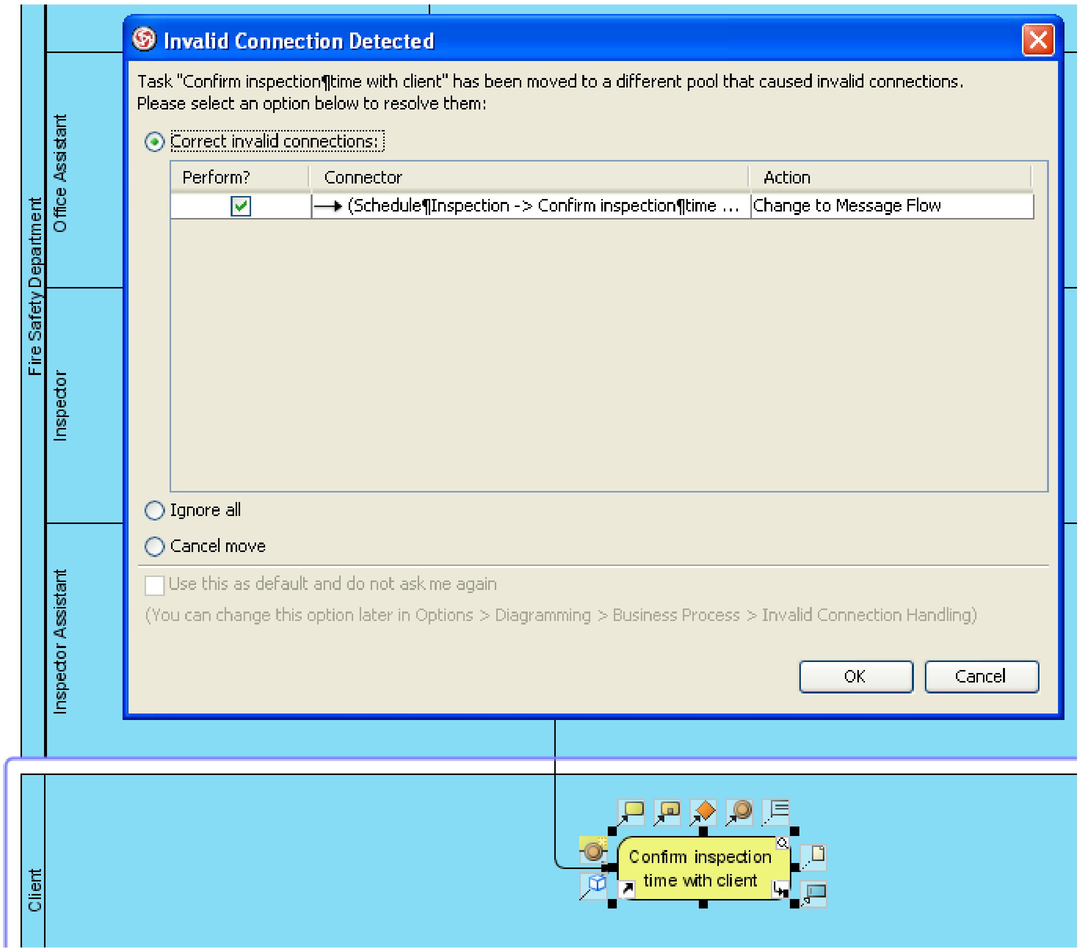


Figure 1-8 *Connect shapes with message flow*

Automatic Connection Rule Checking

VP-UML provides automatic connection rule checking to make sure your business process diagram is in compliance with the BPMN specification.

For example, if you move a task from one pool to another, and if this task is connected with another shape with sequence flow, the **Invalid Connection Detected** dialog box shows to advice you to change sequence flow to message flow.



Invalid Connection Detected

Task "Confirm inspection¶time with client" has been moved to a different pool that caused invalid connections.
Please select an option below to resolve them:

● Correct invalid connections:

Perform?	Connector	Action
☑	→ (Schedule¶Inspection -> Confirm inspection¶time ...	Change to Message Flow

○ Ignore all

○ Cancel move

□ Use this as default and do not ask me again

(You can change this option later in Options > Diagramming > Business Process > Invalid Connection Handling)

Figure 1-9 *Invalid connection detected*

Click **OK** to accept the correction. The sequence flow is changed to message flow automatically.

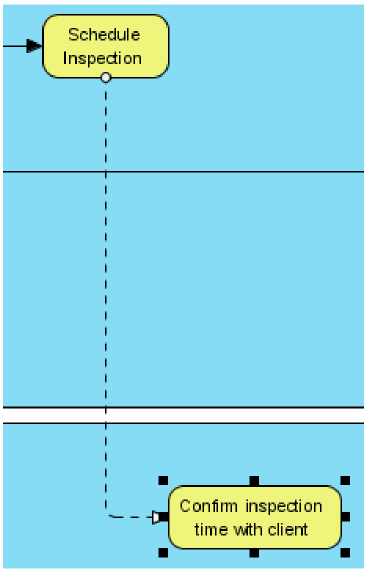


Figure 1-10 *Sequence flow changed to message flow*

Creating Gateway

To create gateway from a task, click on the **Sequence/Message Flow** -> **Gateway** resource beside it and drag.

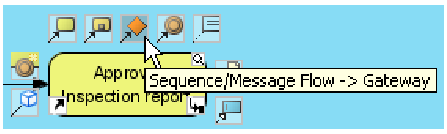


Figure 1-11 *Create gateway*

Move the mouse to where you want to place the shape to and then release the mouse button. A gateway is created and connected to the task with a sequence flow.

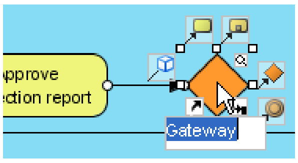


Figure 1-12 *Gateway and sequence flow created*

Creating Intermediate Event

To create intermediate event from a task, click on the **Sequence/Message Flow** -> **Intermediate Event** resource beside it and drag.

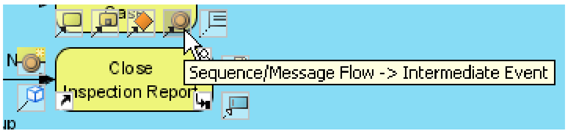


Figure 1-13 *Create intermediate event*

Move the mouse to where you want to place the shape to and then release the mouse button. An intermediate event is created and connected to the task with a sequence flow.

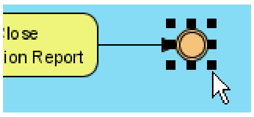


Figure 1-14 *Intermediate event and sequence flow created*

Creating Text Annotation

To create text annotation from a task, click on the **Association** -> **Text Annotation** resource beside it and drag.

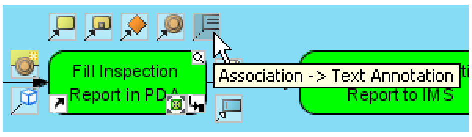


Figure 1-15 *Create text annotation*

Move the mouse to where you want to place the shape to and then release the mouse button. A text annotation is created and connected to the task with an association.

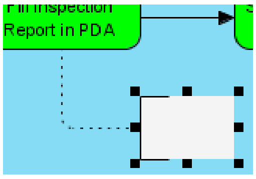


Figure 1-16 *Text annotation and association created*

Spliting Sequence Flow with Shape

You can use the split sequence flow utility to split a sequence flow by task, sub-process, intermediate event or gateway.

Before this, you may want to move some shapes aside to make room for the new shape. Select **Sweeper** on the diagram toolbar.

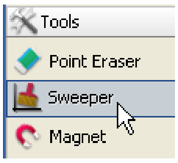


Figure 1-17
Sweeper

Click on the diagram and drag to sweep shapes aside, release the mouse button when enough room is made.

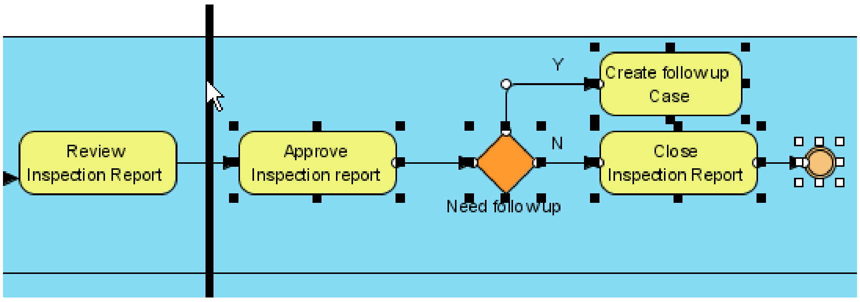


Figure 1-18 *Sweep shapes aside to make more room*

Click the **Split with Shape** resource beside the sequence flow.

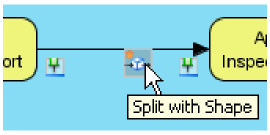


Figure 1-19 *Split with Shape resource*

Select the shape type to split the sequence flow.

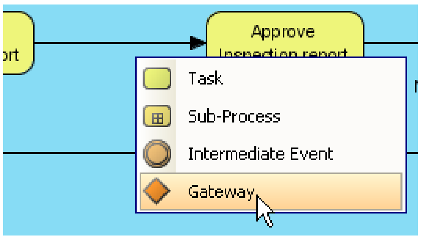


Figure 1-20 *Select shape type to split sequence flow*

The sequence flow is split by the selected type of shape.

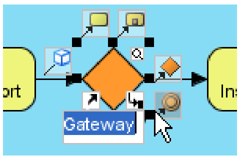


Figure 1-21 *Sequence flow split by the selected type of shape*

Forming Sub-Process Diagram

To create sub-process for shapes, select the shapes in diagram, right-click on the selection and select **Form Sub-Process Diagram** from the popup menu.

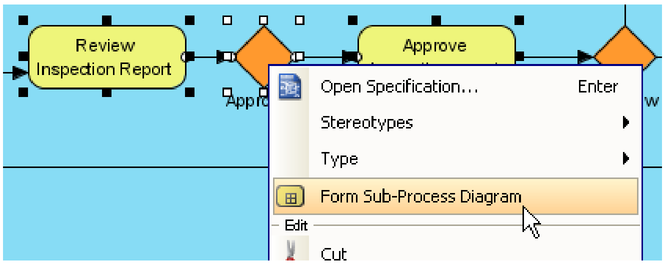


Figure 1-22 *Form sub-process diagram*

A new diagram is created with the selected shapes.

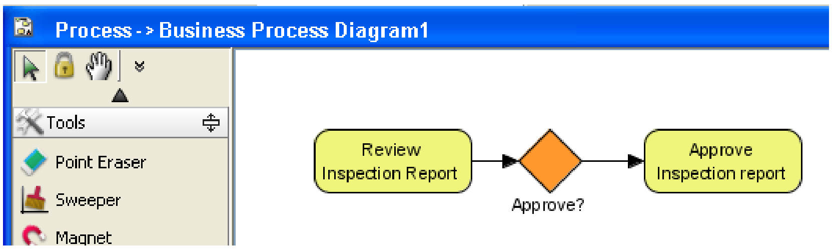


Figure 1-23 *Form sub-process diagram*

Return to the original diagram, you will see the selected shapes are transformed to a sub-process.

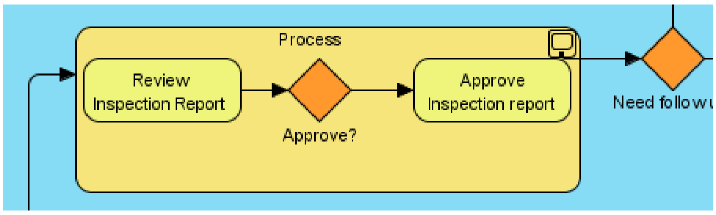


Figure 1-24 *Shapes transformed to sub-process*

Continue to complete the diagram.

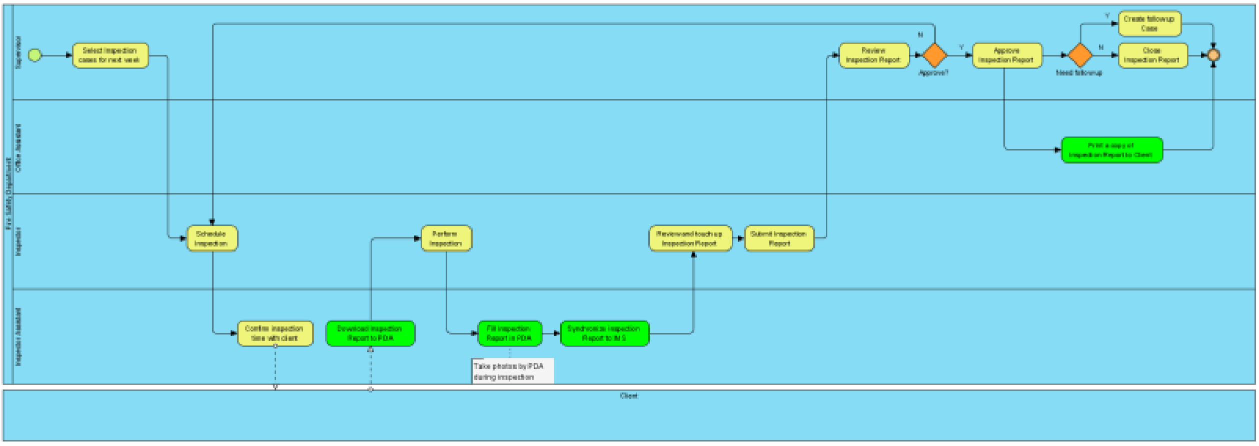

Fire Safety Department
Supervisor
Office Assistant
Inspector
Inspection Assistant
Select inspection cases for next week
Schedule Inspection
Confirm inspection time with client
Download Inspection Report to PDA
Perform Inspection
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS
Take photos by PDA during inspection
Review and touch up Inspection Report
Submit Inspection Report
Review Inspection Report
N
Approve?
Y
Approve Inspection Report
Need followup
Y
Create followup Case
N
Close Inspection Report
Print a copy of Inspection Report to Client
Client

Figure 1-25 *Completed diagram*

Drawing Data Flow Diagrams

Creating Data Flow Diagram

Select menu **File** > **New Diagram** > **Business Process Modeling** > **Data Flow Diagram** to create a data flow diagram.

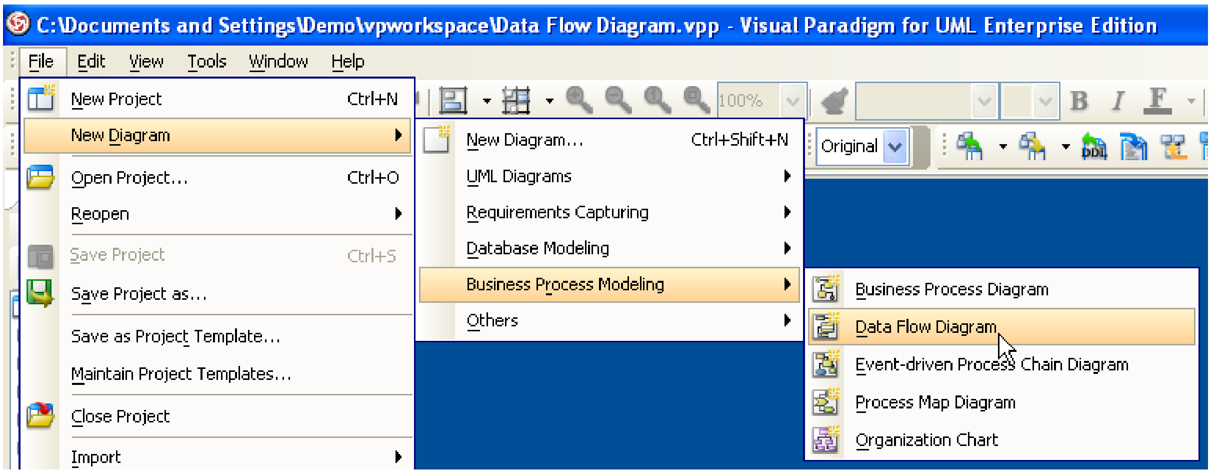


Figure 1-26 Create data flow diagram

Creating External Entity

To create external entity, click **External Entity** on the diagram toolbar and then click on the diagram.

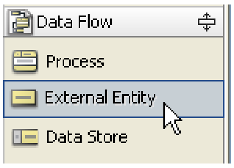


Figure 1-27 Create external entity

An external entity is created.

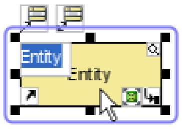


Figure 1-28 External entity created

Creating Process

To create process from external entity, click the **Data Flow** -> **Process** resource beside it and drag.

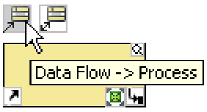


Figure 1-29 Create process

Move the mouse to empty space of the diagram and then release the mouse button, a process will be created and connected to the external entity with a data flow.

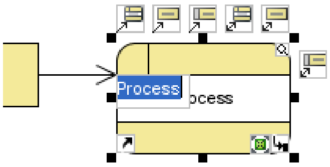
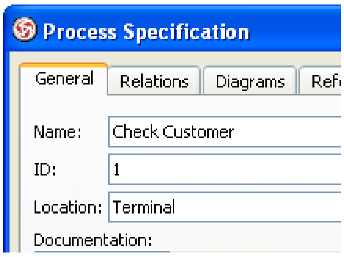


Figure 1-30 Process and data flow created

To edit properties of the process, right-click on it and select **Open Specification** from the popup menu.

Edit properties such as ID and Location in the **Process Specification** dialog box, click **OK** to apply the changes.


Process Specification

General | Relations | Diagrams | Ref

Name: Check Customer

ID: 1

Location: Terminal

Documentation:

__Figure 1-31__ Process specification dialog box

The process is updated.

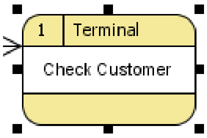


__Figure 1-32__ Process updated

Creating Data Store

To create data store from process, click the **Data Flow** -> **Data Store** resource beside it and drag.

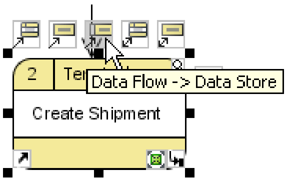


__Figure 1-33__ Create data store

Move the mouse to empty space of the diagram and then release the mouse button, a data store will be created and connected to the process with a data flow.

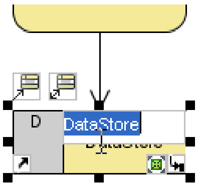


__Figure 1-34__ Data store and data flow created

Continue to complete the diagram.

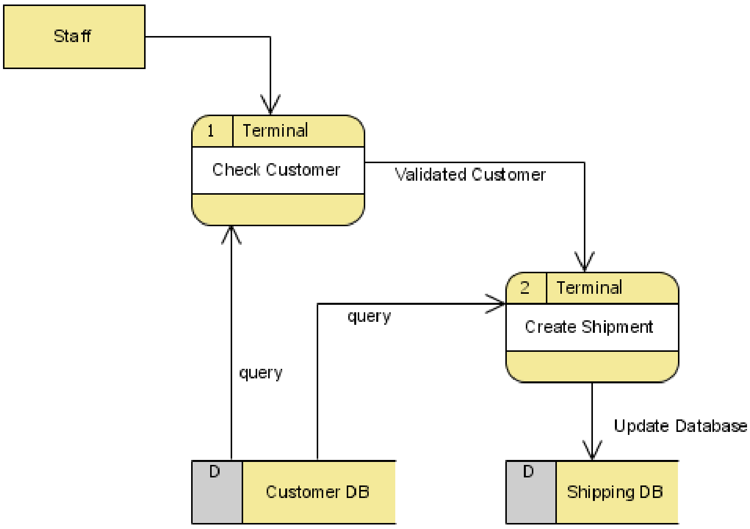


***Figure 1-35** Completed diagram*

Drawing Event-driven Process Chain Diagrams

Creating Data Flow Diagram

Select menu **File** > **New Diagram** > **Business Process Modeling** > **Event-driven Process Chain Diagram** to create an event-driven process chain diagram.

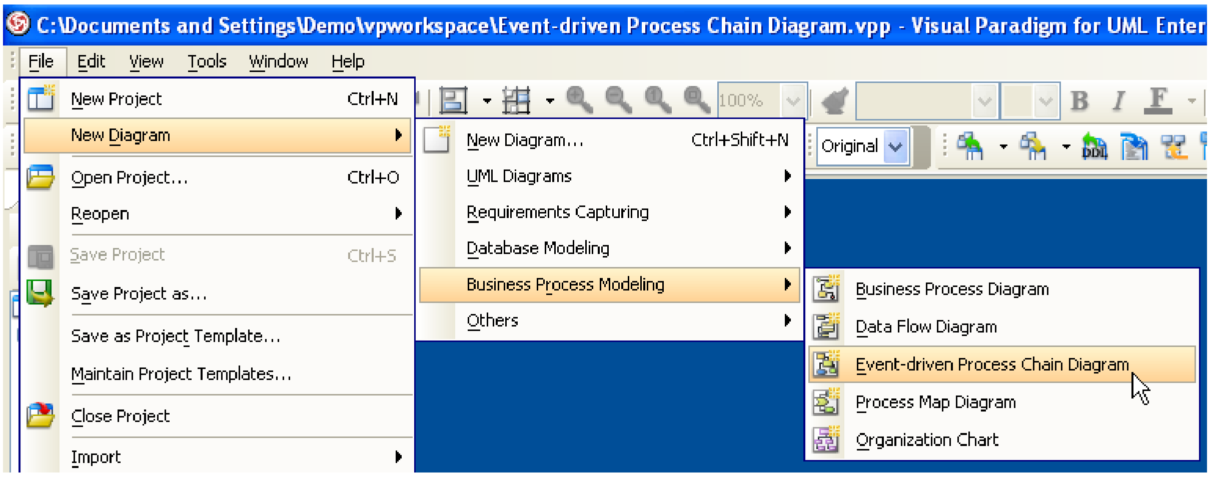


Figure 1-36 Create event-driven process chain diagram

Creating Event

To create event, click **Event** on the diagram toolbar and then click on the diagram.

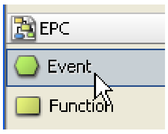


Figure 1-37 Create event

An event will be created.

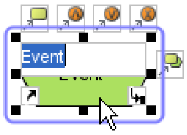


Figure 1-38 Event created

Creating Function

To create function from event, click the **Control Flow** -> **Function** resource beside it and drag **.**

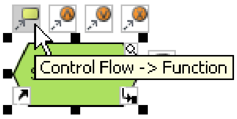


Figure 1-39 Create function

Move the mouse to empty space of the diagram and then release the mouse button, a function will be created and connected to the event with a control flow.

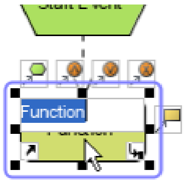


Figure 1-40 Function and control flow created

Creating Operator

To create operator (e.g. XOR operator) from event, click the **Control Flow** -> **XOR Operator** resource beside it and drag **.**

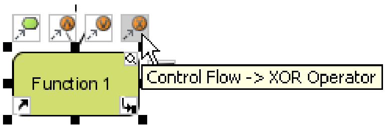


Figure 1-41 *Create operator*

Move the mouse to empty space of the diagram and then release the mouse button, an operator will be created and connected to the function with a control flow.

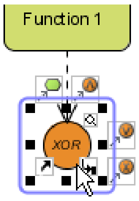


Figure 1-42 *Operator and control flow created*

Creating Process Path

To create process path from event, click the **Control Flow** -> **Process Path** resource beside it and drag **.**

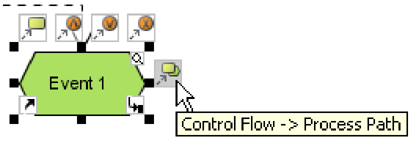


Figure 1-43 *Create process path*

Move the mouse to empty space of the diagram and then release the mouse button, a process path will be created and connected to the event with a control flow.

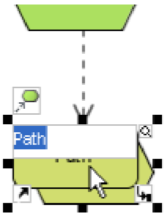


Figure 1-44 *Process path and control flow created*

Creating Information Resource

To create information resource, click **Information Resource** on the diagram toolbar and then click on the diagram.

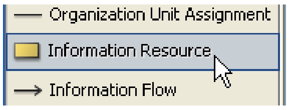


Figure 1-45 *Create information resource*

An information resource will be created. To connect the information resource with a function using information flow, click the **Information Flow** -> **Function** resource and drag.

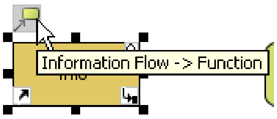


Figure 1-46 *Create information flow*

Mouse over a function and then release the mouse button, an information flow will be created.

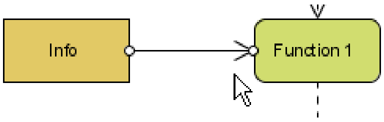


Figure 1-47 *Information flow created*

Continue to complete the diagram.

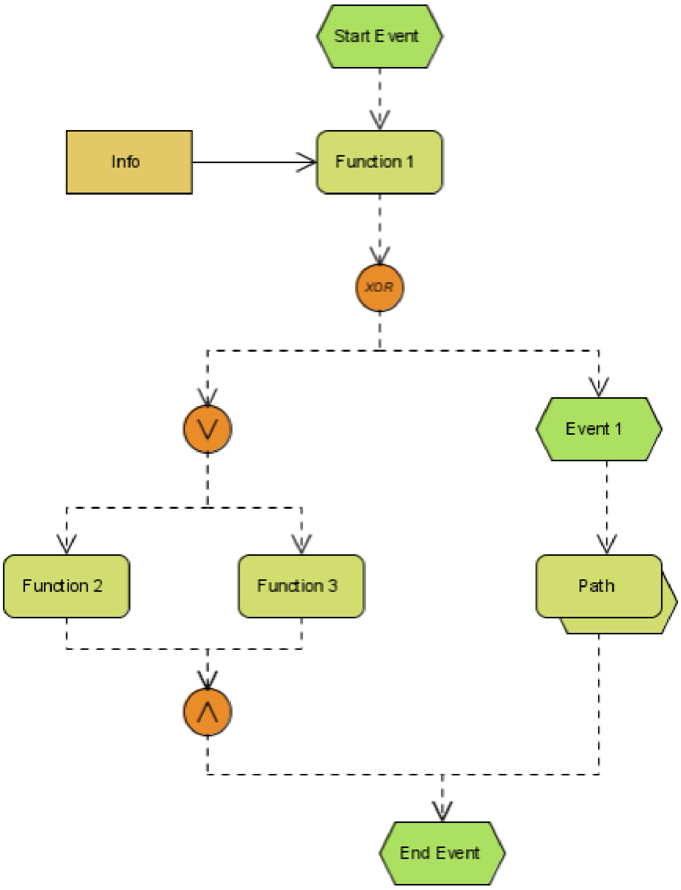


Figure 1-48 *Completed diagram*

Drawing Event-driven Process Chain Diagrams

Creating Data Flow Diagram

Select menu **File** > **New Diagram** > **Business Process Modeling** > **Process Map Diagram** to create a process map diagram.

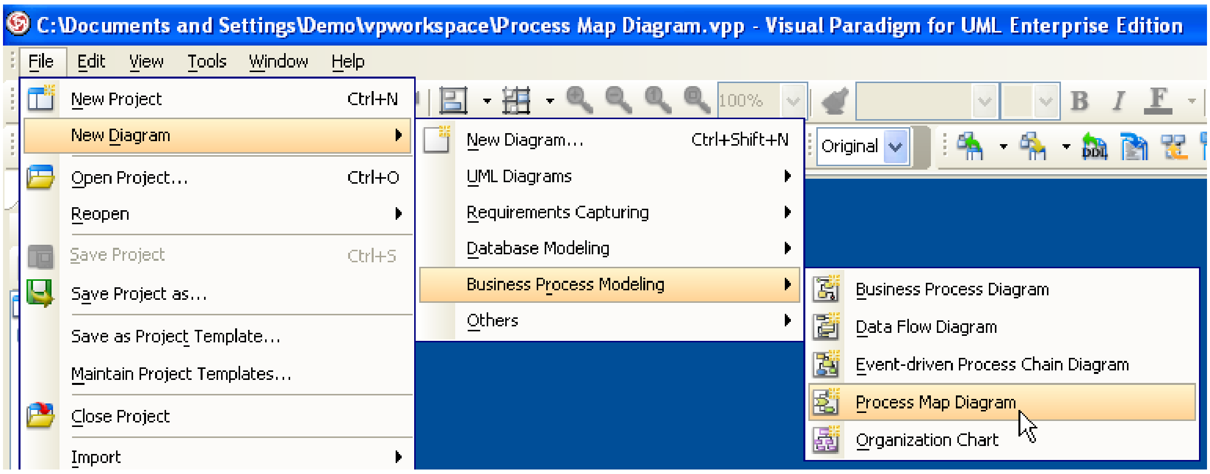


Figure 1-49 Create process map diagram

Creating Receive

To create receive, click **Receive** on the diagram toolbar and then click on the diagram.

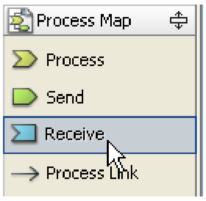


Figure 1-50 Create receive

A receive will be created.

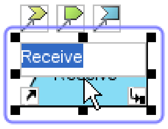


Figure 1-51 Receive created

Creating Process

To create process from receive, click the **Process Link** -> **Process** resource beside it and drag **.**

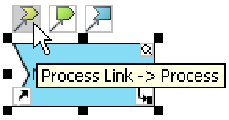


Figure 1-52 Create process

Move the mouse to empty space of the diagram and then release the mouse button, a process will be created and connected to the receive with a process link.

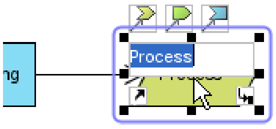


Figure 1-53 Process created

Creating Send

To create send from process, click the **Process Link** -> **Send** resource beside it and drag **.**

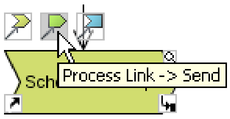


Figure 1-54 *Create send*

Move the mouse to empty space of the diagram and then release the mouse button, a send will be created and connected to the process with a process link.

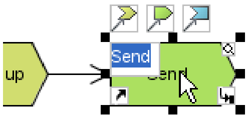


Figure 1-55 *Send created*

Continue to complete the diagram.

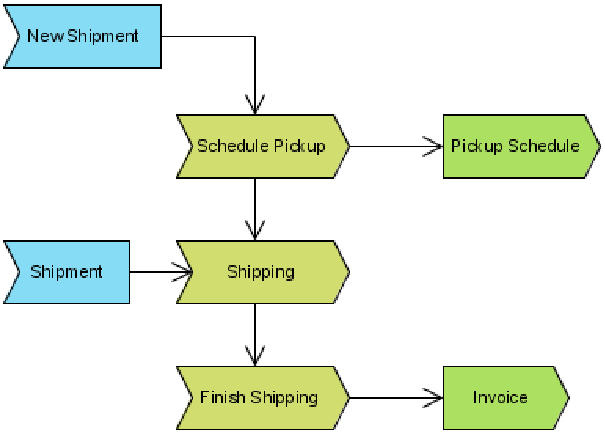


Figure 1-56 *Completed diagram*

Drawing Organization Charts

Creating Organization Charts

Select menu **File** > **New Diagram** > **Business Process Modeling** > **Organization Chart** to create an organization chart.

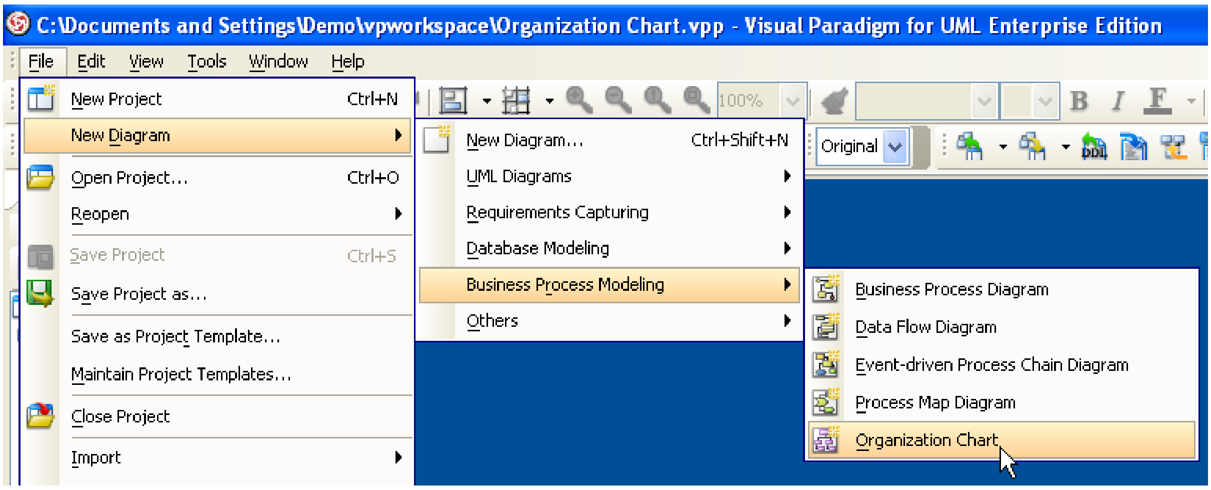


Figure 1-57 *Create organization chart*

A default unit is created for an organization chart.

Creating Subordinate

To create subordinate of a unit, click the **New Subordinate** resource below it.

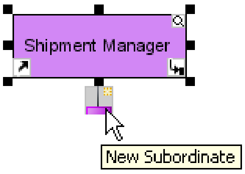


Figure 1-58 *Create subordinate*

A subordinate of the unit will be created.

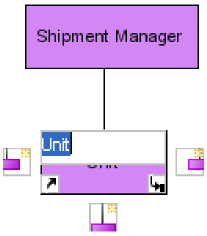


Figure 1-59 *Subordinate created*

Creating Coworker

To create coworker of a unit, click the **New Coworker** resource either on its left or right hand side (click left resource will create coworker to the left of the unit, while click right resource will create coworker to the right of the unit).

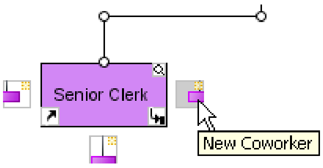


Figure 1-60 *Create coworker*

A coworker of the unit will be created.

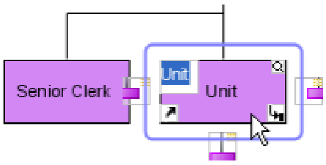


__Figure 1-61__ Coworker created

Continue to complete the diagram.

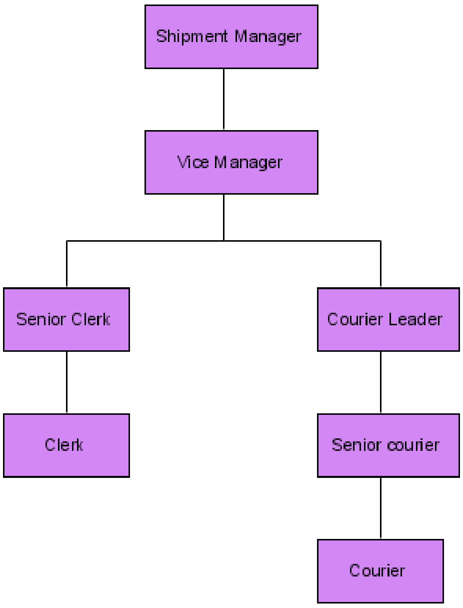


__Figure 1-62__ Completed diagram

Drawing Entity Relationship Diagram

Creating Entity Relationship Diagram

Open **Diagram Navigator**, right click on **Entity Relationship Diagram**, select **New Entity Relationship Diagram**.

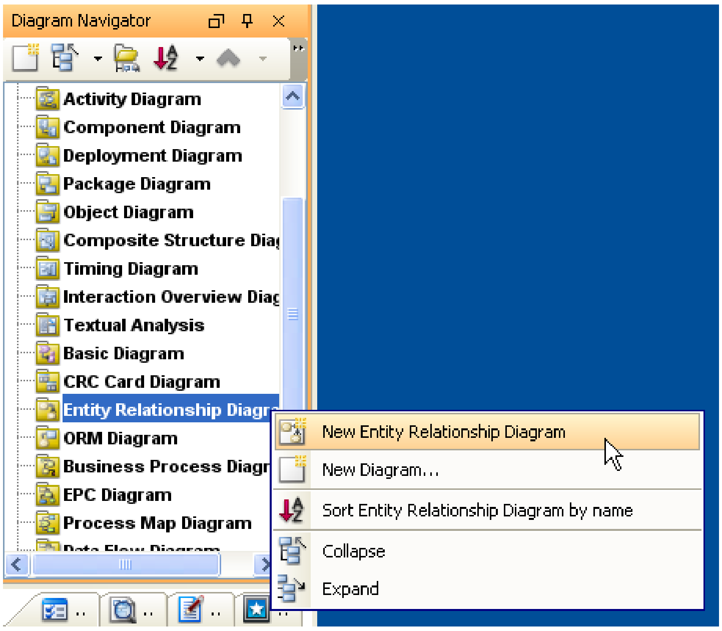


Figure 1-1 Create entity relationship diagram

Enter the name for the diagram. At the same time, a **Data Model** selection box appear on the top right corner of the diagram.

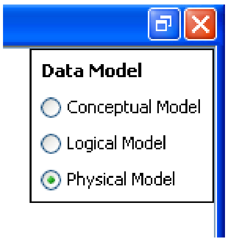


Figure 1-2 Select data model

All model elements created on diagram will follow this **Data Model** setting. Only **Physical Model** will be able to generate SQL. Leave it as default (Physical Model).

Drawing Entity

Select **Entity** from diagram palette.

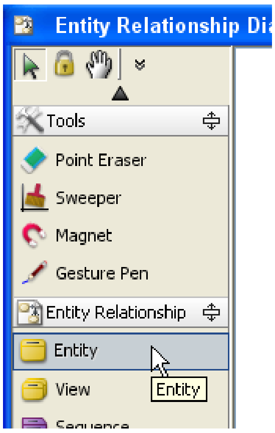


Figure 1-3 Create entity

Click on diagram and specify the entity name.

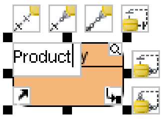


Figure 1-4 *Rename entity*

Create column by select **New Column** from entity popup menu, or **Alt+Shift+C**.

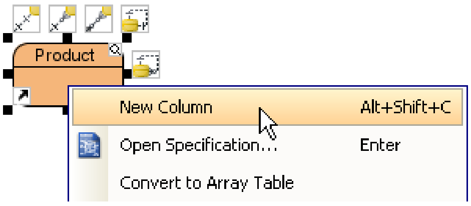


Figure 1-5 *Create column*

Set the column as primary key by select **Include in primary key** from popup menu.

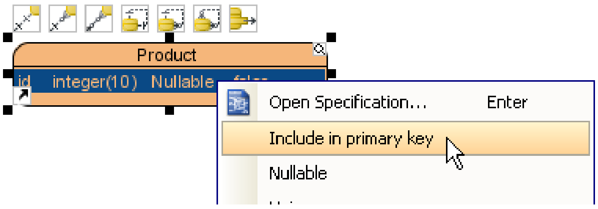


Figure 1-6 *Set as primary key*

Drawing Relationships

Click or drag One-to-One or One-to-Many relationship icon on entity resources.

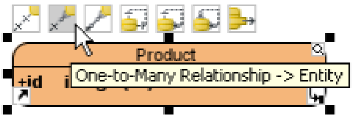


Figure 1-7 *Create relationship*

A new entity is created with foreign key column(s), referencing the original entity's primary key column(s).

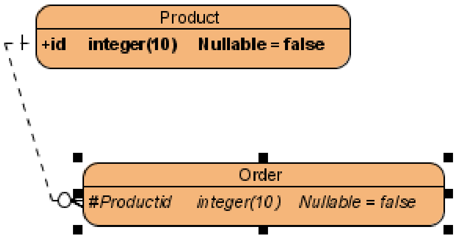
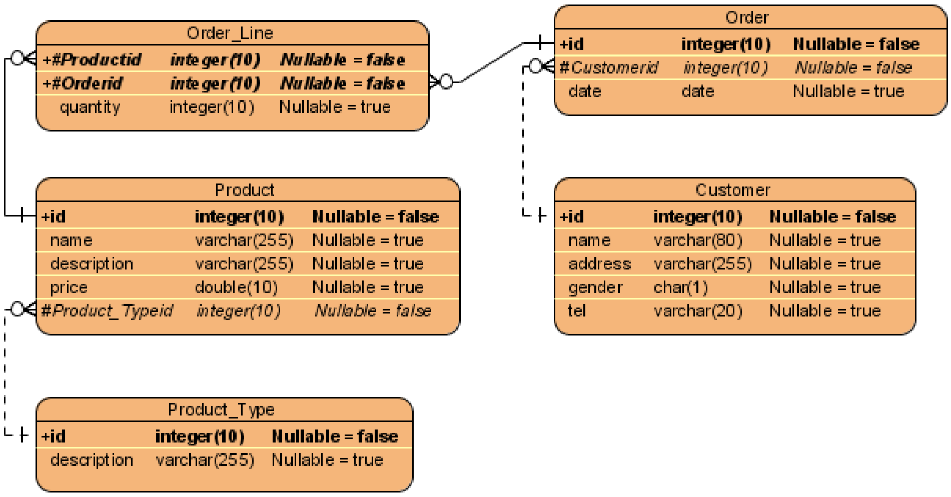

Product
+id integer(10) Nullable = false
Order
#ProductId integer(10) Nullable = false

Figure 1-8 *Rename relationship*

Follow the same steps to create other entities and relationship until the diagram looks like the picture below.



Order_Line

+#Productid	integer(10)	Nullable = false
+#Orderid	integer(10)	Nullable = false
quantity	integer(10)	Nullable = true

Order

+id	integer(10)	Nullable = false
#Customerid	integer(10)	Nullable = false
date	date	Nullable = true

Product

+id	integer(10)	Nullable = false
name	varchar(255)	Nullable = true
description	varchar(255)	Nullable = true
price	double(10)	Nullable = true
#Product_Typeid	integer(10)	Nullable = false

Customer

+id	integer(10)	Nullable = false
name	varchar(80)	Nullable = true
address	varchar(255)	Nullable = true
gender	char(1)	Nullable = true
tel	varchar(20)	Nullable = true

Product_Type

+id	integer(10)	Nullable = false
description	varchar(255)	Nullable = true

Figure 1-9 *Complete ERD*

Drawing Many-to-many Relationship

Click or drag Many-to-Many relationship icon on entity resources.

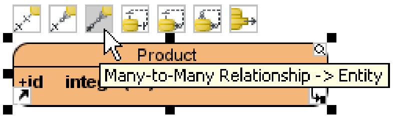
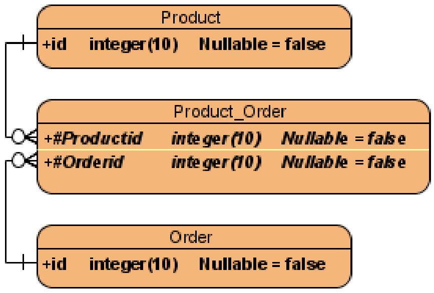


Figure 1-10 *Create many-to-many relationship*

Many-to-Many relationship will auto convert to two One-to-Many relationships and a join table. The primary key both tables will be used to create foreign key columns in join table as composite primary key.



Product

+id	integer(10)	Nullable = false

Product_Order

+#Productid	integer(10)	Nullable = false
+#Orderid	integer(10)	Nullable = false

Order

+id	integer(10)	Nullable = false

Figure 1-11 *Created join table with one-to-many relationships*

Drawing Sequence

Creating Sequence

Select **Sequence** from diagram palette.

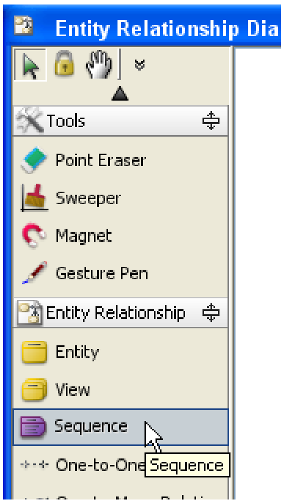


Figure 1-12 *Create sequence*

Click on the diagram and specify the sequence name.

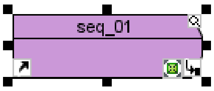
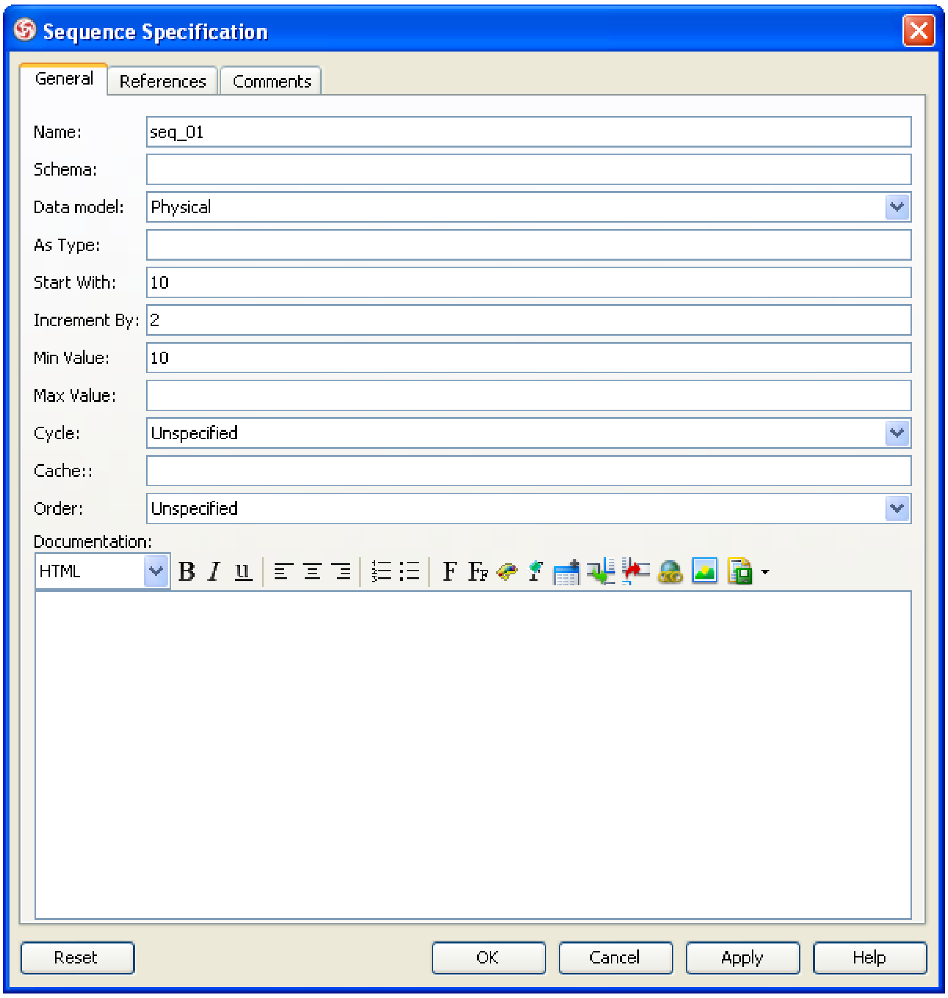


Figure 1-13
Rename sequence

Open specification dialog for the sequence and specify attributes.


Sequence Specification

Name: seq_01
Schema:
Data model: Physical
As Type:
Start With: 10
Increment By: 2
Min Value: 10
Max Value:
Cycle: Unspecified
Cache::
Order: Unspecified
Documentation:

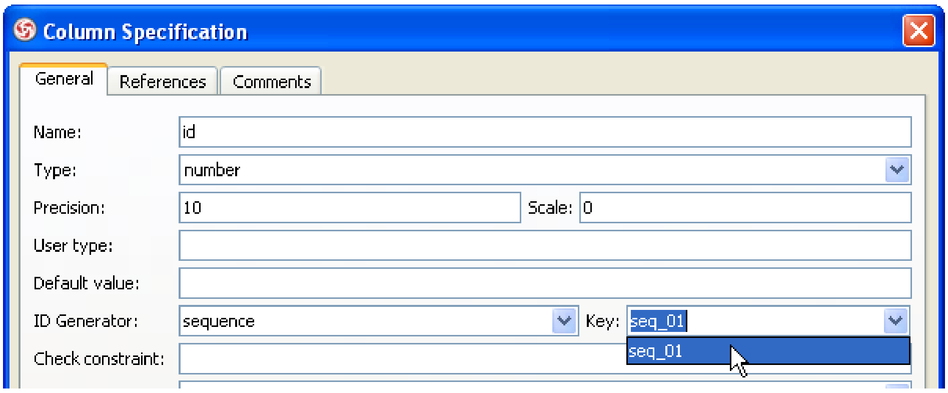
Figure 1-14 *Specify sequence attributes*

NOTE: Generate sequence is only supported in DB2 and Oracle.

Select sequence for entity

Create an entity and primary key column, then open specification dialog for the column.

In **Column Specification** dialog, select **sequence** for **ID Generator**, and select the sequence name created above for **Key**.


Column Specification

Name: id
Type: number
Precision: 10 Scale: 0
User type:
Default value:
ID Generator: sequence Key: seq_01
Check constraint:

Figure 1-15 *Select sequence for entity*

When using Object-Relational Mapping feature, the primary key value will be insert automatically from the sequence.

Drawing Stored Procedures

Creating Stored Procedures Shape

Select **Stored Procedure** from diagram palette.

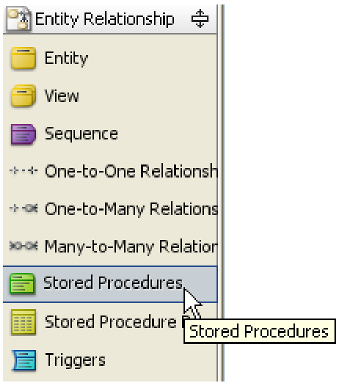


Figure 1-16 *Create stored procedure shape*

Click on the diagram and specify the procedure shape name.

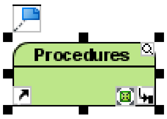


Figure 1-17 *Rename procedure shape*

NOTE: Procedure shape is a virtual container to group a set of stored procedures. It is not a stored procedure.

Creating Procedure

Select **New Procedure** from procedure shape popup menu, or **Alt+Shift+P**.

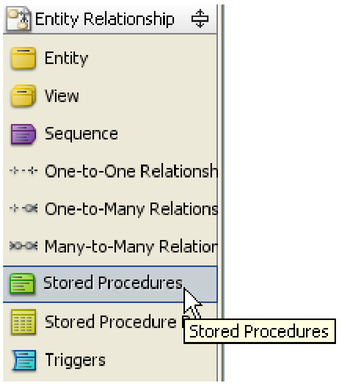
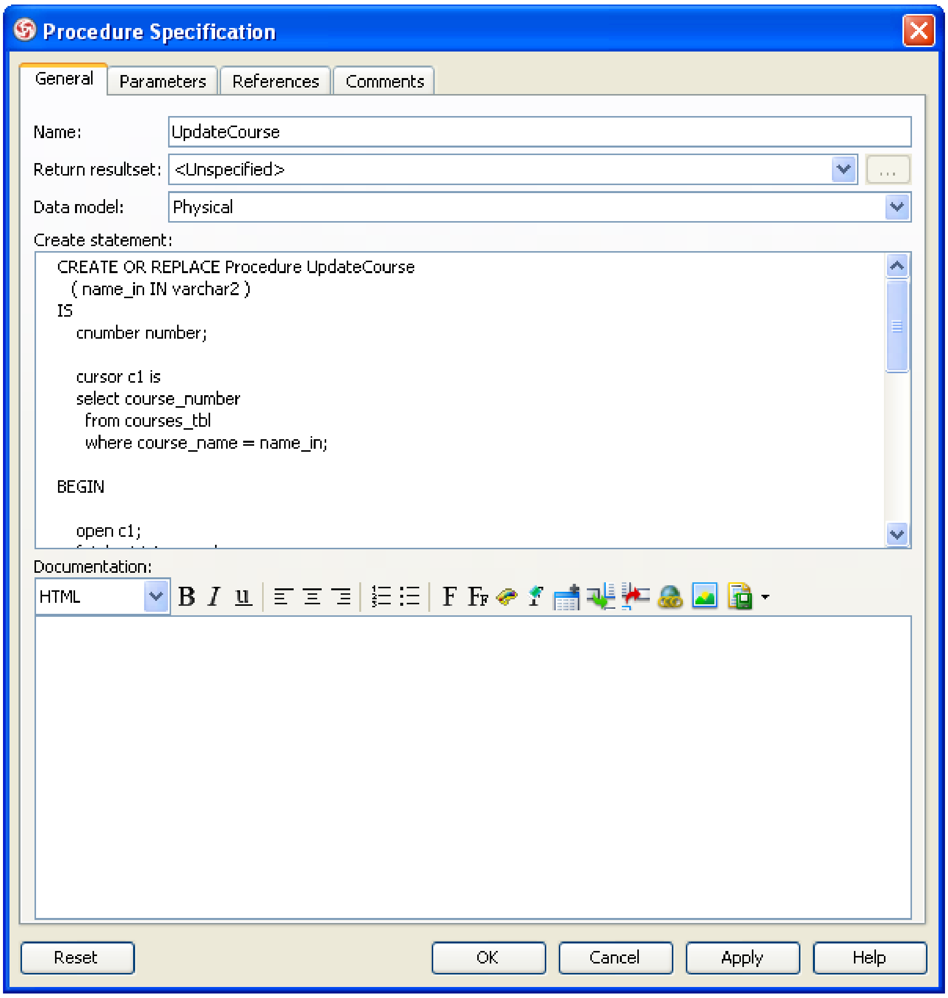


Figure 1-18 *Create stored procedure*

Open specification dialog for procedure, specify the create statement and create parameters if necessary.



Name: UpdateCourse

Return resultset: <Unspecified>

Data model: Physical

Create statement:

```
CREATE OR REPLACE Procedure UpdateCourse
  ( name_in IN varchar2 )
IS
    cnumber number;

    cursor c1 is
    select course_number
      from courses_tbl
      where course_name = name_in;

BEGIN

    open c1;
```

Figure 1-19 Stored procedure specification dialog

Creating Stored Procedure ResultSet

Select **Stored Procedure ResultSet** from diagram palette.

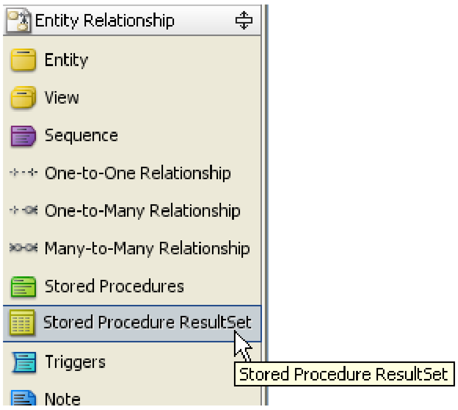


Figure 1-20 Create stored procedure resultset

Click on the diagram and specify the resultset name.

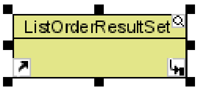
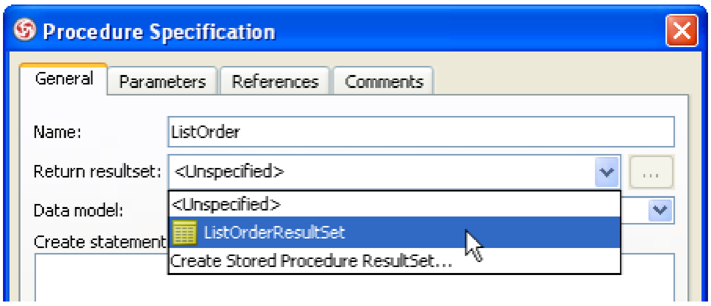


Figure 1-21 Rename stored procedure resultset

Create resultset column in the same way as create entity column.

Specify stored procedure return resultset from stored procedure specification dialog.



Name: ListOrder

Return resultset: <Unspecified>

- <Unspecified>
- ListOrderResultSet
- Create Stored Procedure ResultSet...

Data model:

Create statement

Figure 1-22 Specify stored procedure return resultset

Drawing Triggers

Creating Triggers Shape

Select **Triggers** from diagram palette.

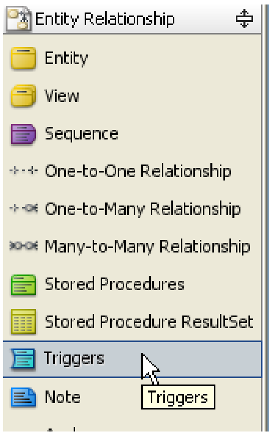


Figure 1-23 *Create trigger shape*

Click on the diagram and specify the trigger shape name.

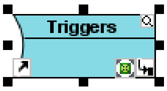


Figure 1-24 *Rename trigger shape*

Creating Trigger

Select **New Trigger** from trigger shape popup menu, or **Alt+Shift+T**.

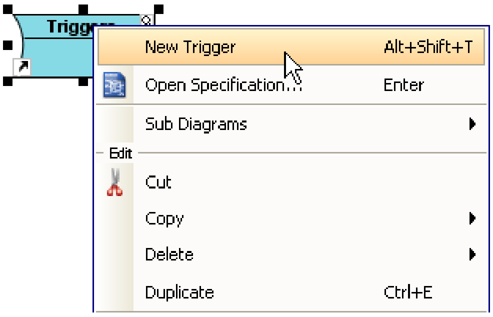
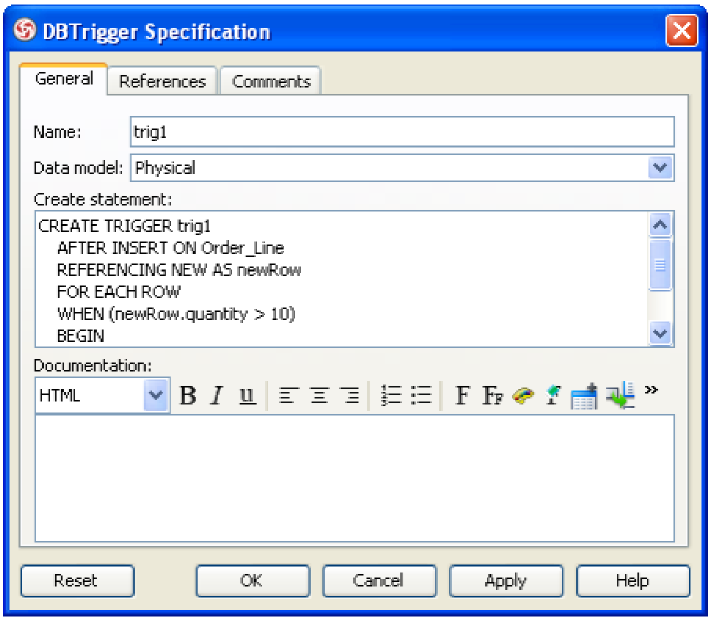


Figure 1-25 *Create trigger*

Open specification dialog for trigger, specify the create statement.



Name: trig1

Data model: Physical

Create statement:

```
CREATE TRIGGER trig1
   AFTER INSERT ON Order_Line
   REFERENCING NEW AS newRow
   FOR EACH ROW
   WHEN (newRow.quantity > 10)
   BEGIN
```

Documentation:

Figure 1-26 *Trigger specification dialog*

Showing Entity in multiple ERD

1. Create a new Entity Relationship Diagram.
2. Drag and drop entities from **Model Explorer** to diagram.

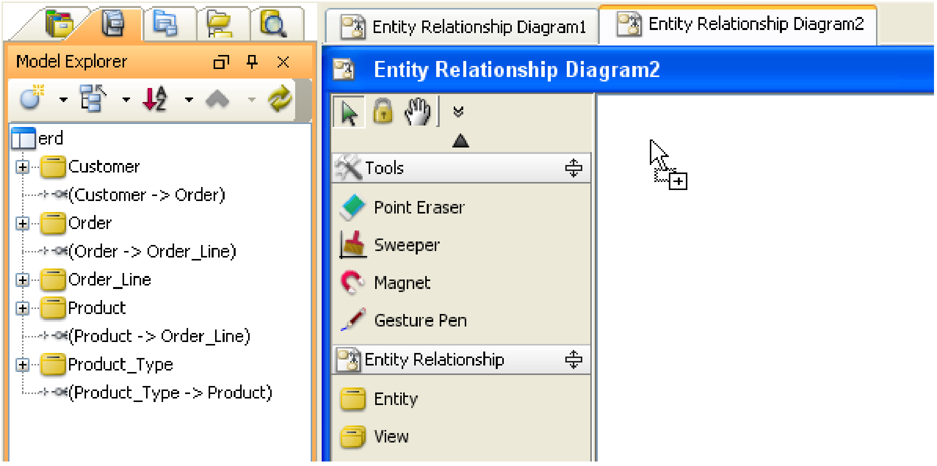
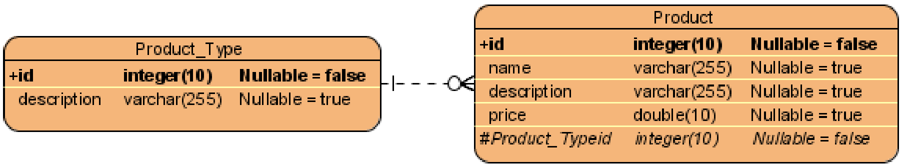


Figure 1-27 *Drag and drop entities*

3. Entities created in new ERD. They reference the same entities in other ERD, modify the entity in any diagram will affect the enitity in other diagrams.



Product_Type

+id	**integer(10)**	**Nullable = false**
description	varchar(255)	Nullable = true

Product

+id	**integer(10)**	**Nullable = false**
name	varchar(255)	Nullable = true
description	varchar(255)	Nullable = true
price	double(10)	Nullable = true
#Product_Typeid	*integer(10)*	*Nullable = false*

Figure 1-28 *Entities created in new ERD*

Controlling Primary Key Values Using ID Generator

ID generator defines how a unique value will be produced for a primary key column. You can assign an ID generator to a primary key column to indicate which strategy will be used when generating an ID in runtime.

1. Open specification for the primary key column that you want to select an ID generator.

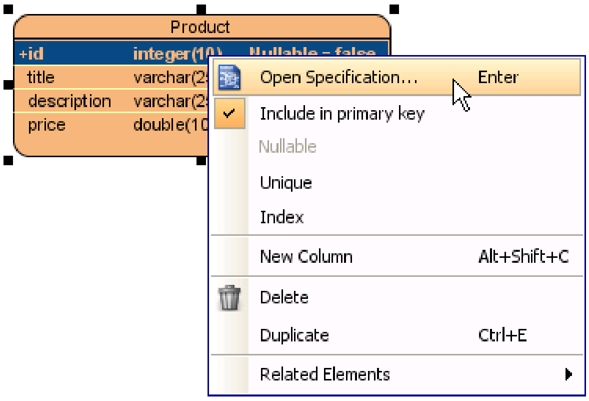
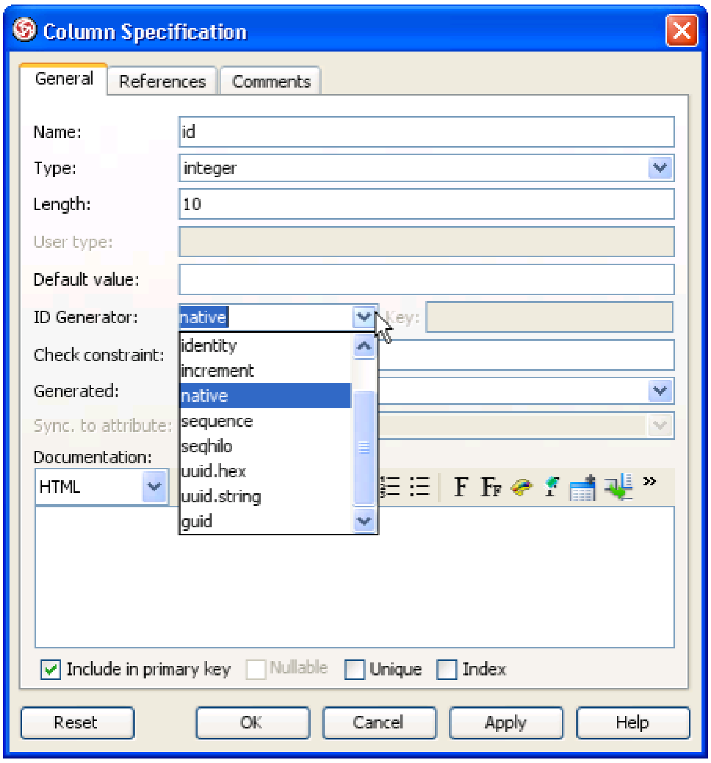


Figure 1-29 *To open column specification*

2. Select the **ID Generator** in the specification dialog box.



Column Specification

Name: id

Type: integer

Length: 10

User type:

Default value:

ID Generator: native

Check constraint:

Generated:

Sync. to attribute:

Documentation: HTML

☑ Include in primary key □ Nullable □ Unique □ Index

Figure 1-30 *Selecting an ID generator*

Below is a description of available ID generator.

ID Generator	Description
assigned	lets the application to assign an identifier to the object before is called.
guid	uses a database-generated GUID string on MS SQL Server and MySQL.
hilo	uses a hi/lo algorithm to efficiently generate identifiers of type , or , given a table and column as a source of hi values. The hi/lo algorithm generates identifiers that are unique only for a particular database.
identity	supports identity columns in DB2, MySQL, MS SQL Server, Sybase and HypersonicSQL. The returned identifier is of type , or .
increment	generates identifiers of type , or that are unique only when no other process is inserting data into the same table. *Do not use in a cluster.*
native	(default) picks , or depending upon the capabilities of the underlying database.
seqhilo	uses a hi/lo algorithm to efficiently generate identifiers of type , or , given a named database sequence.
sequence	uses a sequence in DB2, PostgreSQL, Oracle. The returned identifier is of type , or

Table 1-1 *Available ID Generator*

3. Click **OK** to confirm.

Customizing ID Generator

Besides the built-in strategies for generating ID, user can implement how ID will be generated by customizing an ID generator.

1. In Class Diagram, create the ID generator class, and stereotype it as **ORM ID Generator**.

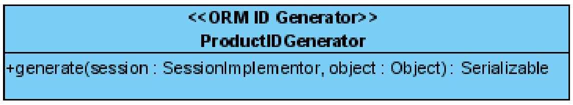

<<ORM ID Generator>>
ProductIDGenerator
+generate(session : SessionImplementor, object : Object) : Serializable

Figure 1-31 *An ID generator class*

2. Open specification for the primary key column that you want to select an ID generator.

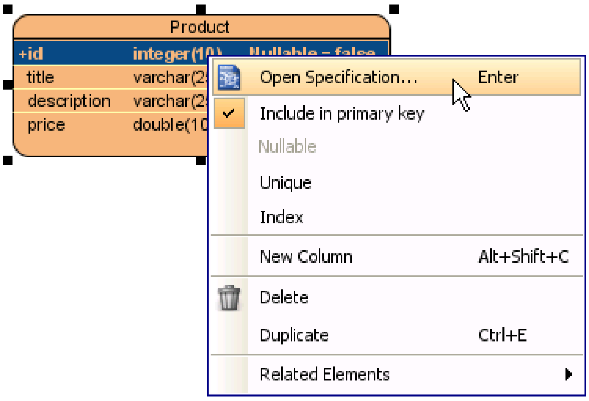
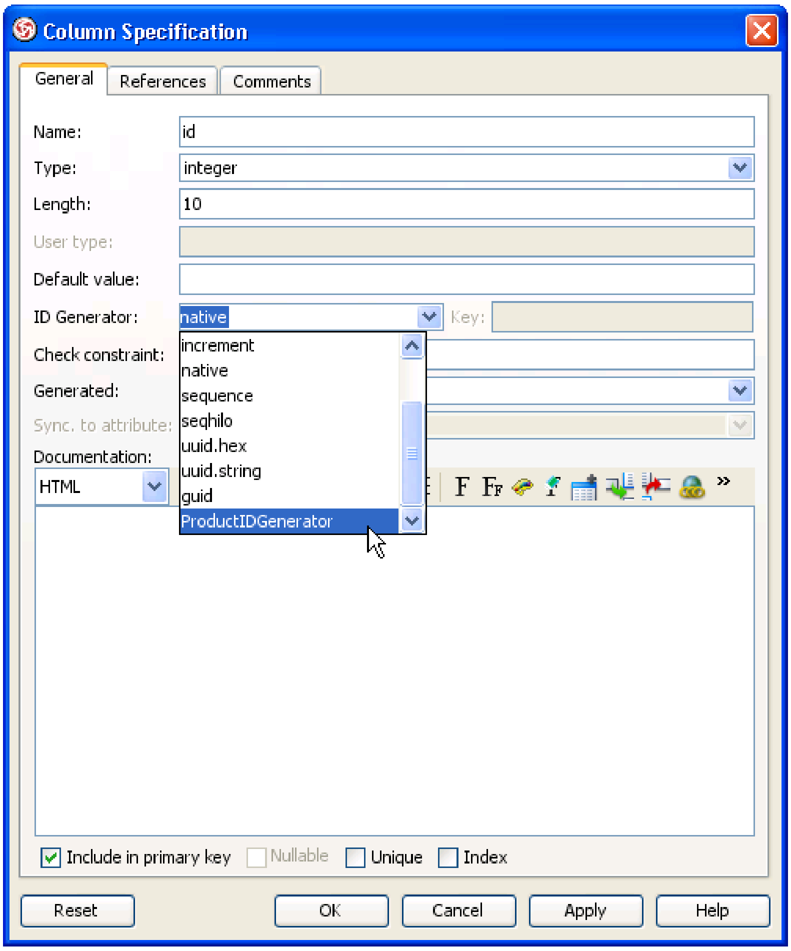


Figure 1-32 *To open column specification*

3. In the specification dialog box, select the class to be the **ID Generator**.


Column Specification

General | References | Comments

Name: id

Type: integer

Length: 10

User type:

Default value:

ID Generator: native Key:

Check constraint:

Generated:

Sync. to attribute:

Documentation:

HTML

increment
native
sequence
seqhilo
uuid.hex
uuid.string
guid
ProductIDGenerator

☑ Include in primary key ☐ Nullable ☐ Unique ☐ Index

Reset | OK | Cancel | Apply | Help

Figure 1-33 *Selecting an ID generator*

4. Click **OK** to confirm.

5. After generated ORM code, look for the ID generator class and implement the **generate** method by return an Integer or Long.

```
/**
* Licensee: VP Development
* License Type: Purchased
*/
import java.io.Serializable;
import org.hibernate.engine.SessionImplementor;
import org.hibernate.id.IdentifierGenerator;
public class ProductIDGenerator implements IdentifierGenerator {
public Serializable generate(SessionImplementor session, Object object) {
//TODO: Implement Method
throw new UnsupportedOperationException();
}
}
//ORM Hash:fae9faed19486e5f2b85c9d2d0d52cd9
```

Database Configuration

Opening database configuration dialog

Select **Tools > Object-Relational Mapping (ORM) > Database Configuration...** from the main menu.

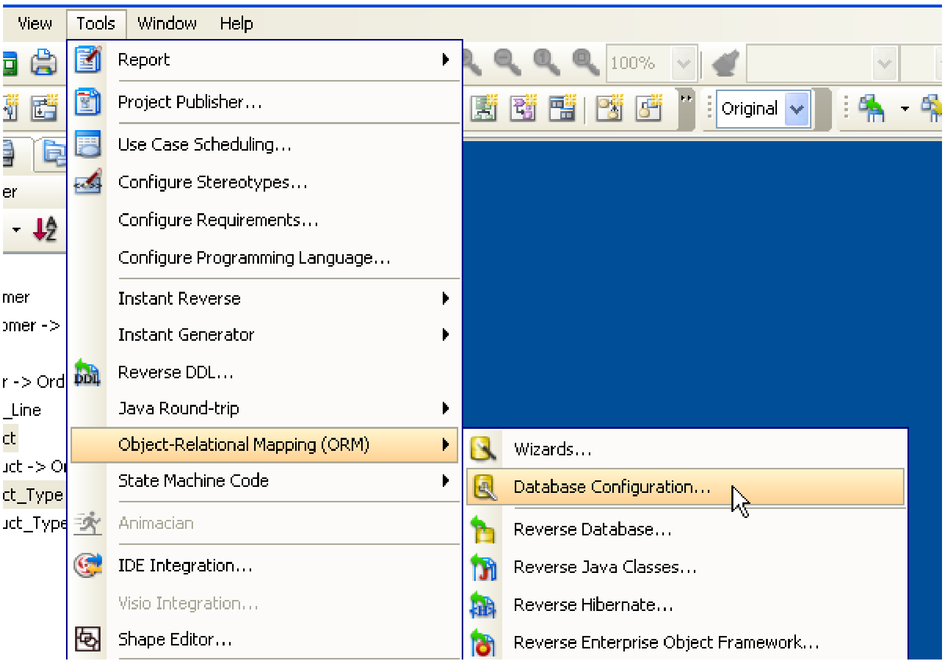
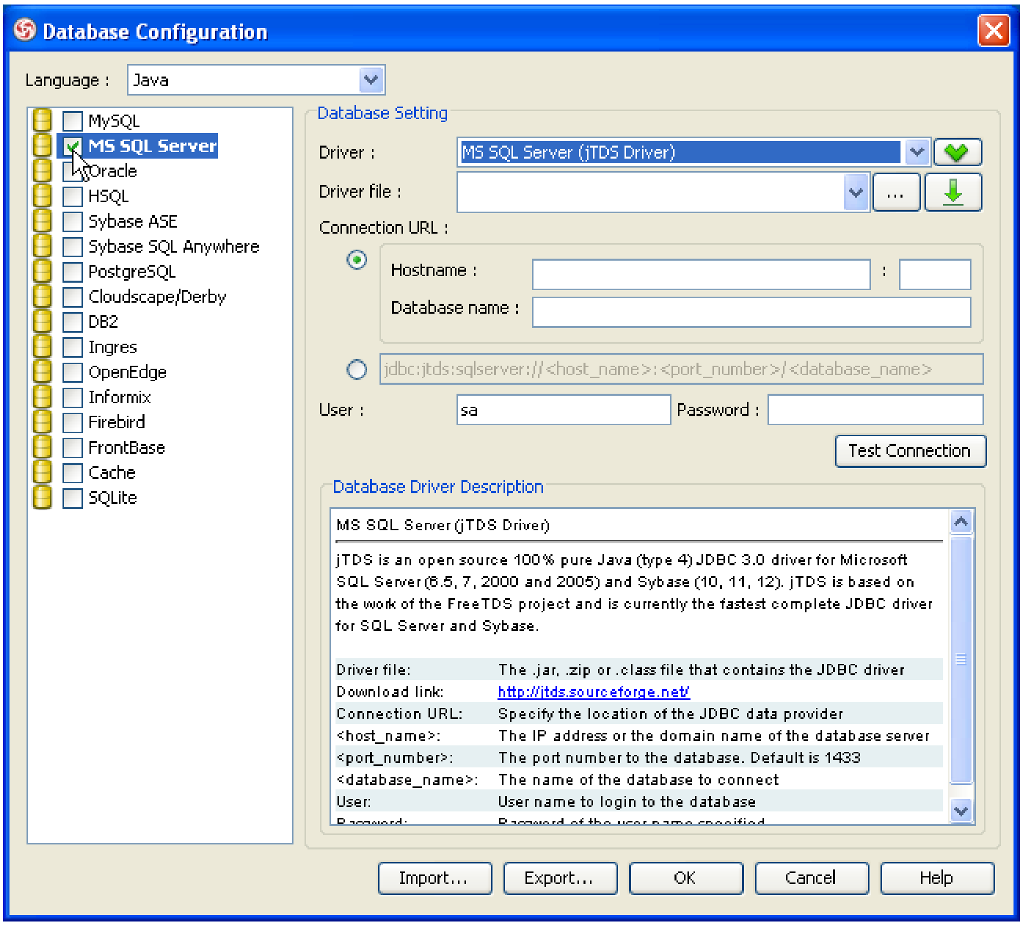


Figure 2-1 Database configuration

Selecting database

Click the checkbox of the database.



MS SQL Server (jTDS Driver)

jTDS is an open source 100% pure Java (type 4) JDBC 3.0 driver for Microsoft SQL Server (6.5, 7, 2000 and 2005) and Sybase (10, 11, 12). jTDS is based on the work of the FreeTDS project and is currently the fastest complete JDBC driver for SQL Server and Sybase.

Driver file:	The .jar, .zip or .class file that contains the JDBC driver
Download link:	http://jtds.sourceforge.net/
Connection URL:	Specify the location of the JDBC data provider
<host_name>:	The IP address or the domain name of the database server
<port_number>:	The port number to the database. Default is 1433
<database_name>:	The name of the database to connect
User:	User name to login to the database
Password:	Password of the user name specified

Figure 2-2 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 popup menu.

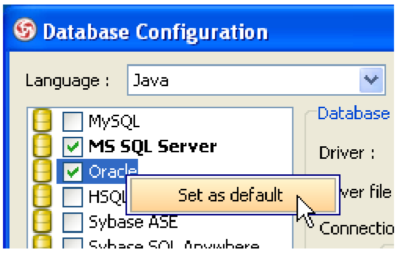


Figure 2-3 *Set as default database*

Downloading JDBC Driver

If the JDBC driver is free and available to public, VP-UML can help you download automatically. Click the **Download or Update** button on **Database Setting**.

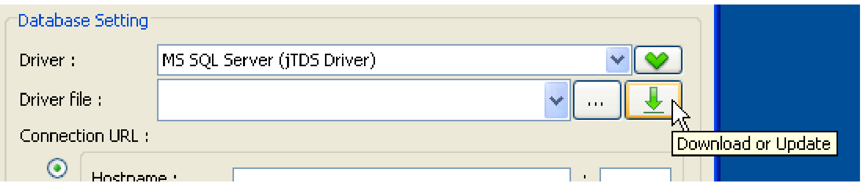


Figure 2-4 *Download JDBC driver*

Configure proxy if required, click **OK** button to continue download.

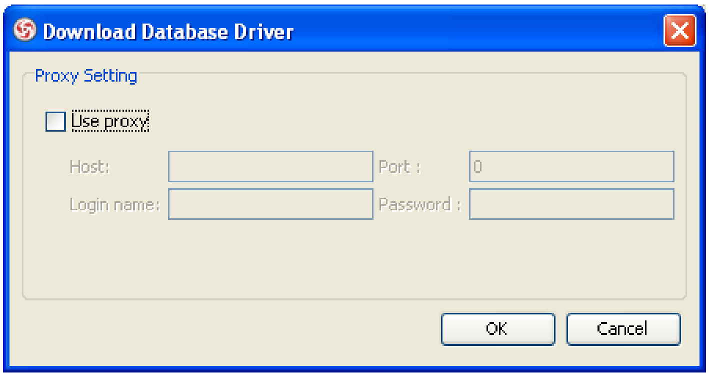
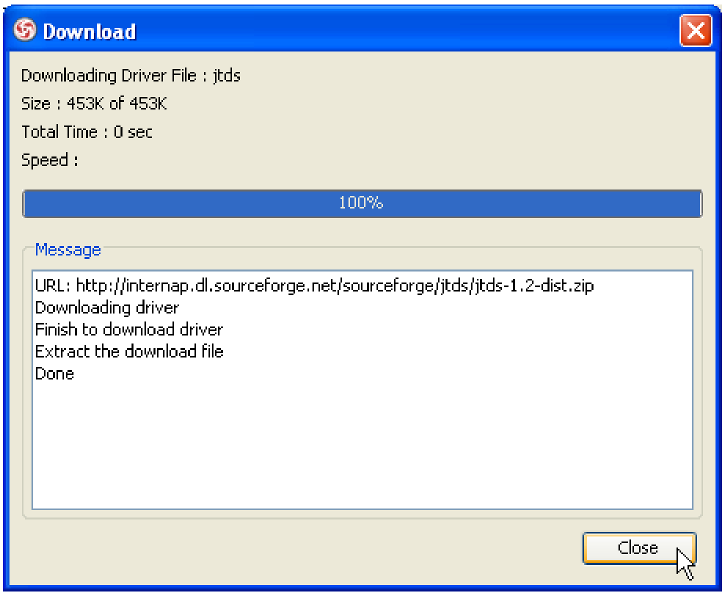


Figure 2-5 *Download JDBC proxy setting*

The **Download** dialog show the URL, file size, speed, progress information, click **Close** button after finish.

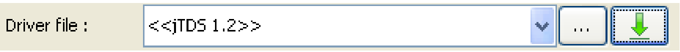


Downloading Driver File : jtds
Size : 453K of 453K
Total Time : 0 sec
Speed :

```
URL: http://internap.dl.sourceforge.net/sourceforge/jtds/jtds-1.2-dist.zip
Downloading driver
Finish to download driver
Extract the download file
Done
```

Figure 2-6 *Download JDBC progress*

The **Driver file** field will becomes <<*Driver Name*>> indicating using the driver downloaded by VP-UML.

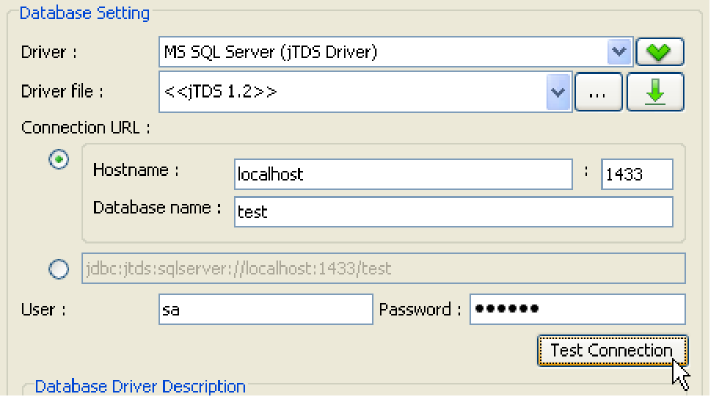


Driver file : <<jTDS 1.2>>

Figure 2-7 Using downloaded JDBC driver

Testing Connection

After configure the database setting, click the **Test Connection** button to confirm the setting is valid.



Database Setting

Driver : MS SQL Server (jTDS Driver)

Driver file : <<jTDS 1.2>>

Connection URL :

● Hostname : localhost : 1433

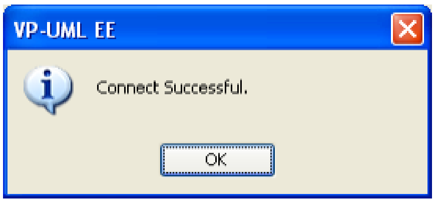
Database name : test

○ jdbc:jtds:sqlserver://localhost:1433/test

User : sa Password : ••••••

Figure 2-8 Test connection

If success, a dialog will show connect successful.



VP-UML EE

Connect Successful.

Figure 2-9 Connect successful

Generating SQL for selected Entities

Generating SQL

Select multiple entities, right click and select **Generate SQL...** from the popup menu.

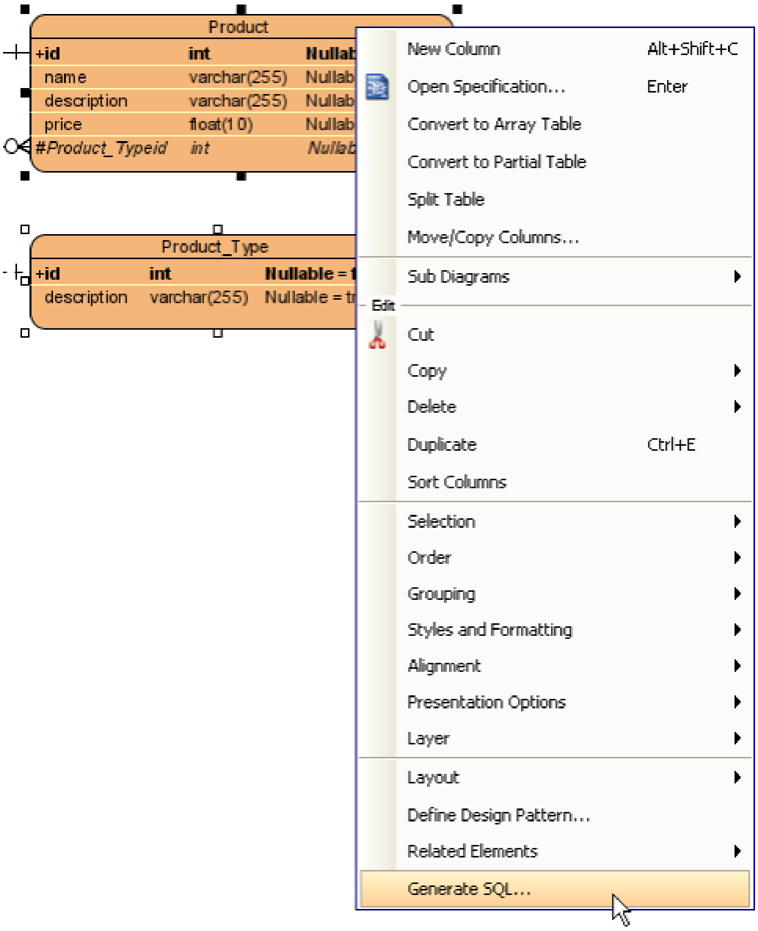
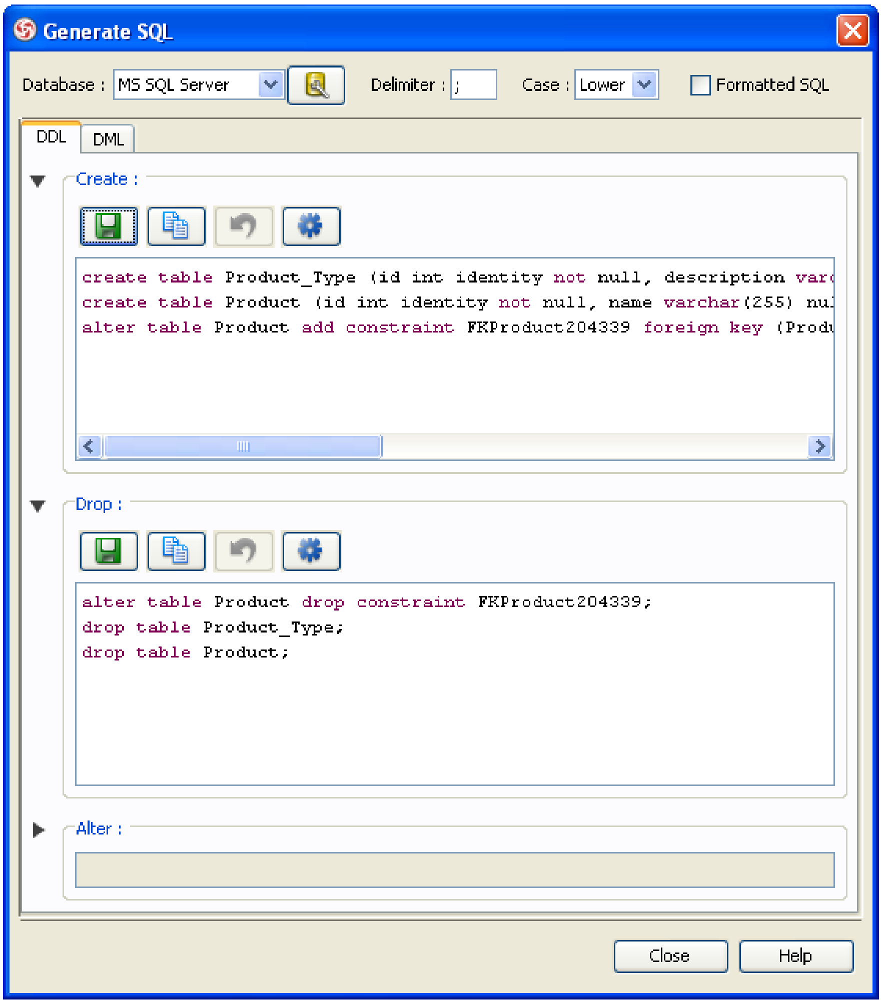


Figure 2-10 *Generate SQL*

The **Generate SQL** dialog shows the DDL and DML for the selected entities.



Database : MS SQL Server   Delimiter : ;   Case : Lower   □ Formatted SQL

```
create table Product_Type (id int identity not null, description var
create table Product (id int identity not null, name varchar(255) nu
alter table Product add constraint FKProduct204339 foreign key (Prod
```

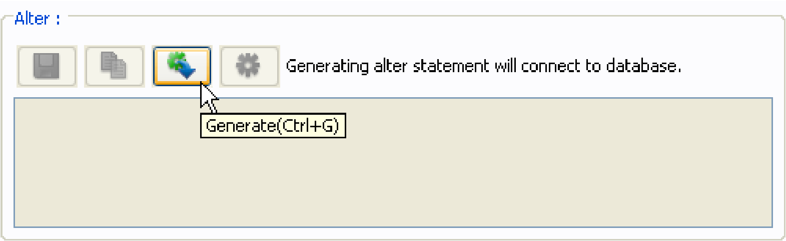
```
alter table Product drop constraint FKProduct204339;
drop table Product_Type;
drop table Product;
```

Figure 2-11 Generate SQL dialog

DDL and DML

DDL stand for Data Definition Language, include create, drop and alter statements. DML stand for Data Manipulating Lanauge, include select, insert, update, delete statements.

The generated DDL statements can directly execute to database without any modification. Generate alter statement require connection to database, query the object from database and compare with the ERD. The alter statements are not generated by default, you can click the Generate button to generate on demand.



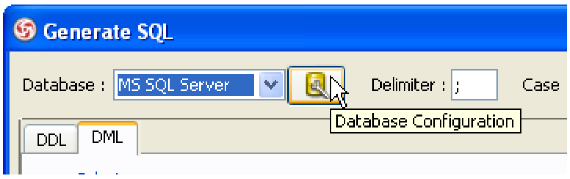
Generating alter statement will connect to database.

Figure 2-12 Generate alter statements

The generate DML is a template for select, insert, update, delete statements, you are required to modify the statement before execute.

Selecting database

Click the **Database Configuration** button to open the database configuration dialog.



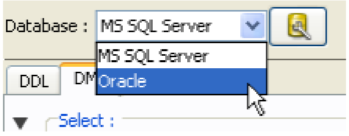
Database : MS SQL Server

Delimiter : ;

Case

Figure 2-13 *Database configuration*

If you selected multiple database, you can select one of the database from the database combo box.



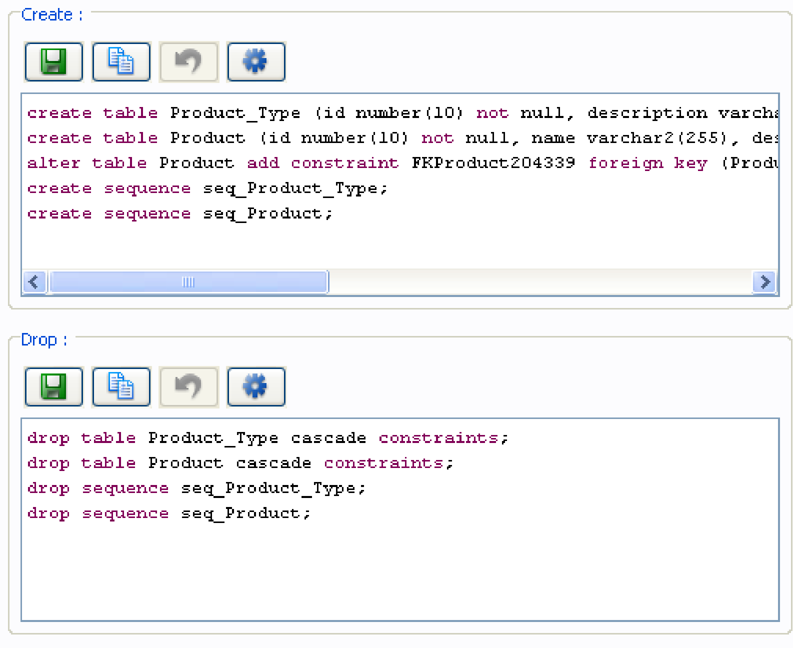
Database : MS SQL Server

MS SQL Server

Oracle

Figure 2-14 *Select database*

After change the database, the SQL will re-generate for the selected database.



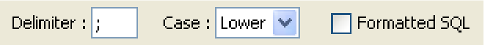
```
create table Product_Type (id number(10) not null, description varcha
create table Product (id number(10) not null, name varchar2(255), de
alter table Product add constraint FKProduct204339 foreign key (Produ
create sequence seq_Product_Type;
create sequence seq_Product;
```

```
drop table Product_Type cascade constraints;
drop table Product cascade constraints;
drop sequence seq_Product_Type;
drop sequence seq_Product;
```

Figure 2-15 *SQL updated*

Generate SQL options

There are several option on the top of the **Generate SQL** dialog.



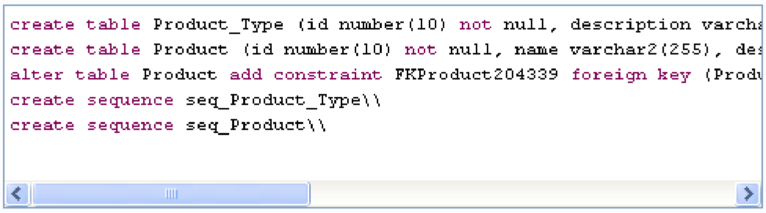
Delimiter : ;

Case : Lower

□ Formatted SQL

Figure 2-16 *Generate SQL options*

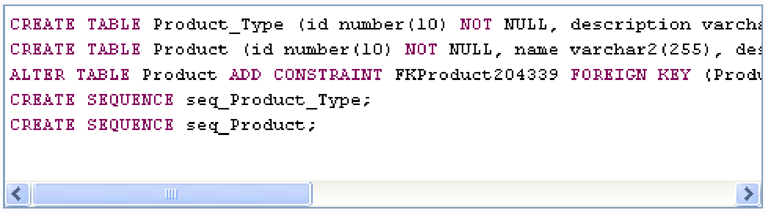
- **Delimiter** - append to end of each statement, used for separate two statement. If change to \\, the SQL will becomes:



```
create table Product_Type (id number(10) not null, description varcha
create table Product (id number(10) not null, name varchar2(255), de
alter table Product add constraint FKProduct204339 foreign key (Produ
create sequence seq_Product_Type\\
create sequence seq_Product\\
```

Figure 2-17 *Delimiter*

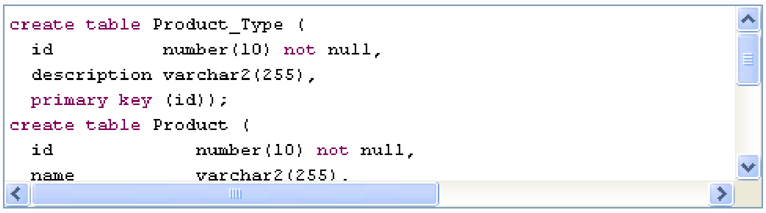
- **Case** - the case for the keyword. If change to **Upper**, the SQL will becomes:



```
CREATE TABLE Product_Type (id number(10) NOT NULL, description varcha
CREATE TABLE Product (id number(10) NOT NULL, name varchar2(255), des
ALTER TABLE Product ADD CONSTRAINT FKProduct204339 FOREIGN KEY (Produ
CREATE SEQUENCE seq_Product_Type;
CREATE SEQUENCE seq_Product;
```

Figure 2-18 *Upper SQL*

- **Formatted SQL** - formatted sql generate high readability SQL statements. If enable, the SQL will becomes:



```
create table Product_Type (
  id          number(10) not null,
  description varchar2(255),
  primary key (id));
create table Product (
  id              number(10) not null,
  name            varchar2(255),
```

Figure 2-19 *Formatted SQL*

Using toolbar

There is a toolbar above each group of statements.

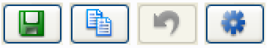


Figure 2-20 *Generate SQL toolbar*

Generating SQL for project

Select **Tools > Object-Relational Mapping (ORM) > Generate Database...** from the main menu,

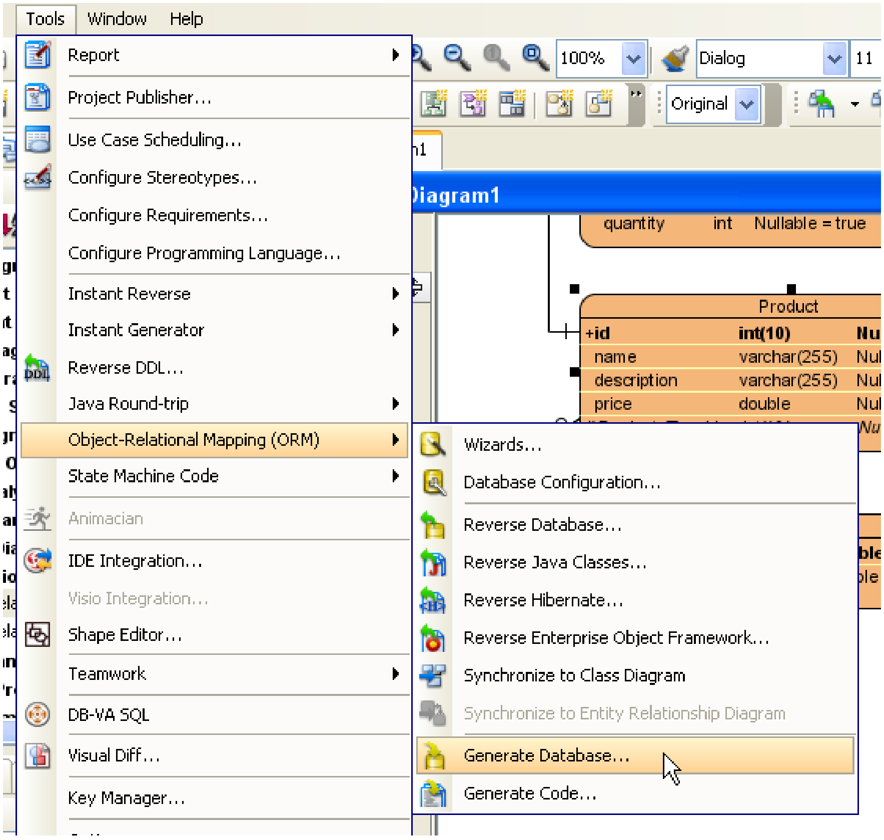
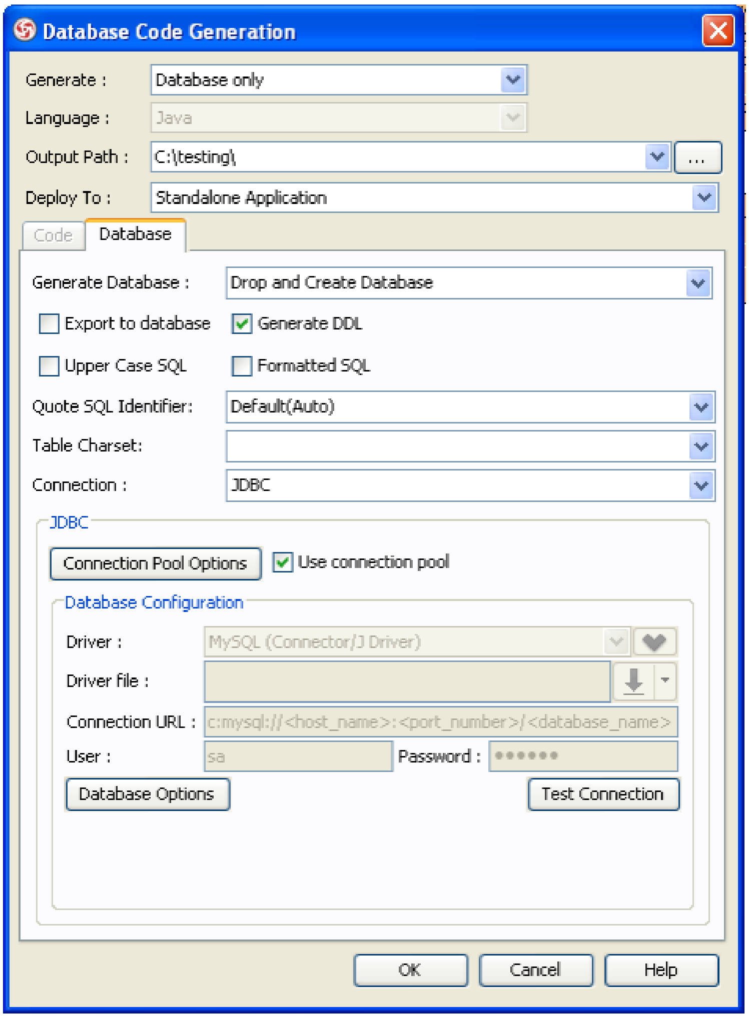


Figure 2-21 Generate database

The **Database Code Generation** dialog provides several options to generate DDL and export to database.



Database Code Generation

Generate : Database only
Language : Java
Output Path : C:\testing\
Deploy To : Standalone Application

Code | Database

Generate Database : Drop and Create Database
□ Export to database ☑ Generate DDL
□ Upper Case SQL □ Formatted SQL
Quote SQL Identifier: Default(Auto)
Table Charset:
Connection : JDBC

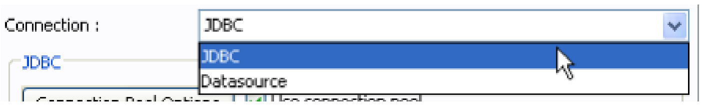
JDBC
Connection Pool Options ☑ Use connection pool

Database Configuration
Driver : MySQL (Connector/J Driver)
Driver file :
Connection URL : c:mysql://<host_name>:<port_number>/<database_name>
User : sa Password : ••••••
Database Options Test Connection

OK Cancel Help

Figure 2-22 *Database code generation*

- Generate Database:
    - Create Database - generate create statements only.
    - Update Database - query existing object in database, generate create and alter statement depends on object not exists or outdated in database, or do nothing if database is up-to-date.
    - Drop and Create Database - generate drop statements first, then generate create statements.
    - Drop Database - generate drop statements only.
- Export to database - execute the generated statements directly to database.
- Generate DDL - save the generated statements to a file.
- Upper Case SQL - generate upper case for keyword (e.g. select, from, update, insert...etc).
- Formatted SQL - generate pretty format, high readability SQL.
- Quote SQL Identifier - if database object name is reserved word, it must be quoted; otherwise, it cannot be used as the object name:
    - Auto - auto detect and quote the name only if it is reserved word.
    - Yes - always quote the name.
    - No - never quote the name.
- Table Charset (Only available for MySQL) - the charset used for database connection.
- Connection:



Connection : JDBC
JDBC
Datasource

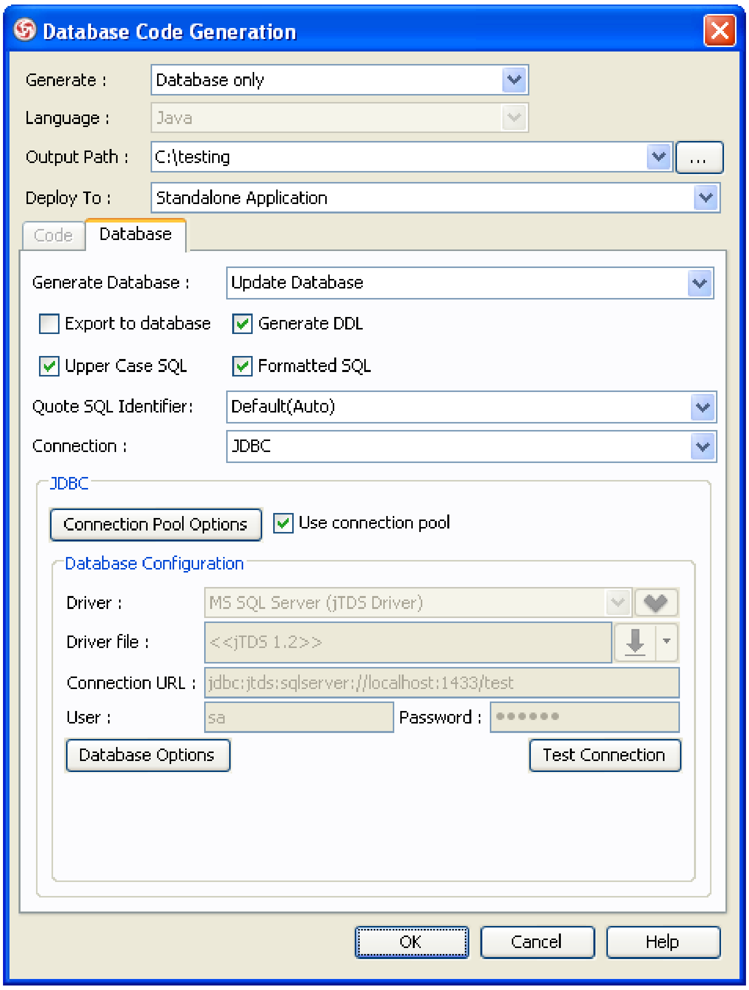
Figure 2-23 *Connection*

- JDBC - a standard way to connect database in Java.
- Datasource - the database connection is managed by application server.

Generating alter statements

Generate alter statements helps you to update the database with the changes on ERD. With the generate DDL option, you can preview a list of alter statements before actually applies to the database.

1. Open **Database Code Generation** dialog.
2. Select the following options:



Database Code Generation

Generate : Database only

Language : Java

Output Path : C:\testing

Deploy To : Standalone Application

Generate Database : Update Database

□ Export to database ☑ Generate DDL

☑ Upper Case SQL ☑ Formatted SQL

Quote SQL Identifier: Default(Auto)

Connection : JDBC

JDBC

☑ Use connection pool

Database Configuration

Driver : MS SQL Server (jTDS Driver)

Driver file : <<jTDS 1.2>>

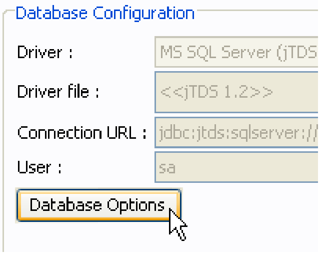
Connection URL : jdbc:jtds:sqlserver://localhost:1433/test

User : sa Password : ●●●●●●

Figure 2-24 *Database code generation*

- **Generate** - Database only
- **Generate Database** - Update Database
- **Export to database** - unchecked
- **Generate DDL** - checked
- **Upper Case SQL** - checked
- **Formatted SQL** - checked
- **Connection** - JDBC

3. Click **Database Options** button to configure JDBC.



Database Configuration

Driver : MS SQL Server (jTDS

Driver file : <<jTDS 1.2>>

Connection URL : jdbc:jtds:sqlserver://

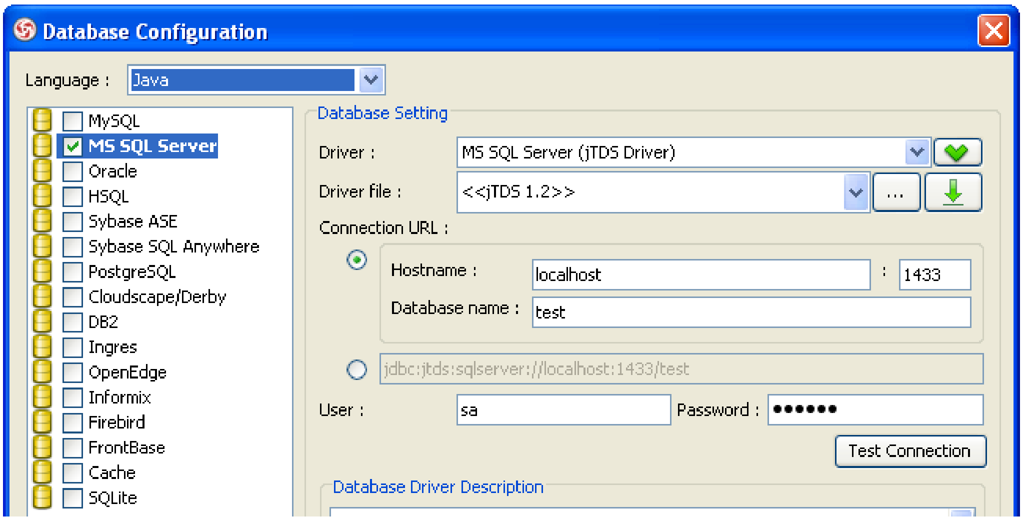
User : sa

Figure 2-25 *Database options*

Export and Import Database Configuration between projects

Export and Import Database Configuration allows you to reuse the same database configuration across projects, without the needs to re-define the configuration in each project.

1. Open **Database Configuration** dialog.
2. Select database(s) and define their settings.



Database Configuration

Language : Java

□ MySQL
☑ MS SQL Server
□ Oracle
□ HSQL
□ Sybase ASE
□ Sybase SQL Anywhere
□ PostgreSQL
□ Cloudscape/Derby
□ DB2
□ Ingres
□ OpenEdge
□ Informix
□ Firebird
□ FrontBase
□ Cache
□ SQLite

Database Setting

Driver : MS SQL Server (jTDS Driver)

Driver file : <<jTDS 1.2>>

Connection URL :

● Hostname : localhost : 1433

Database name : test

○ jdbc:jtds:sqlserver://localhost:1433/test

User : sa Password : ●●●●●●

Test Connection

Database Driver Description

Figure 2-27 *Database configuration*

3. Click the **Export...** button, and specify the filename to save the setting.

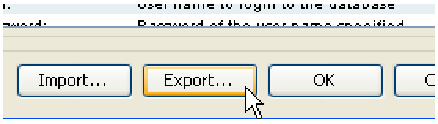

User name to login to the database
Password of the user name specified
Import...
Export...
OK

Figure 2-28 *Export button*

Reversing Database

1. Create a new project.
2. Select **Tools > Object-Relational Mapping (ORM) > Reverse Database...** from the main menu.

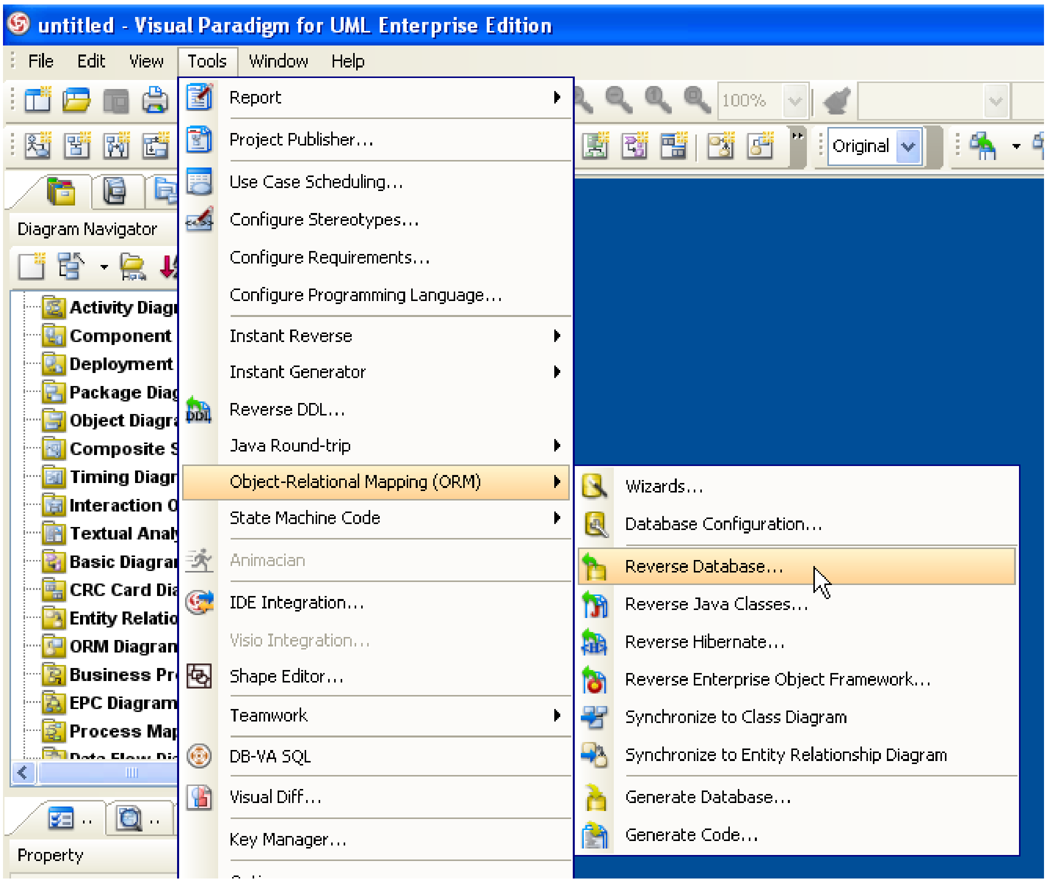
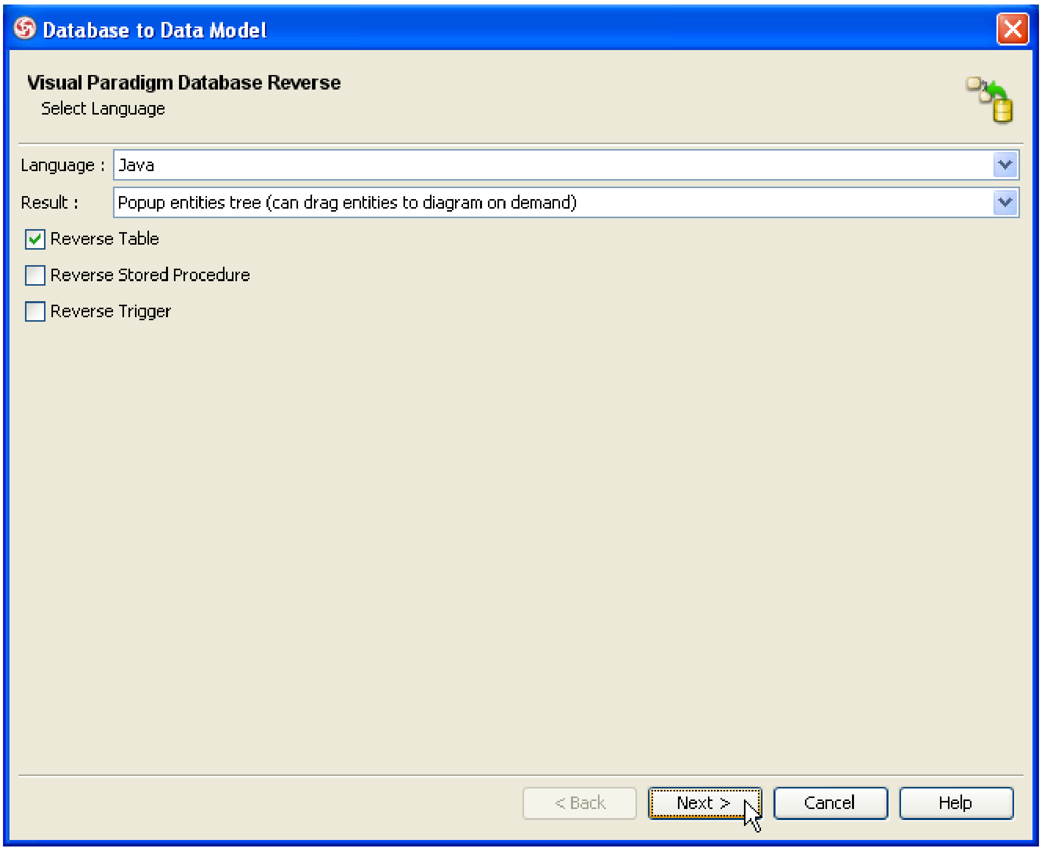


Figure 3-1 *Reverse database*

3. Select **Reverse Table** and click **Next >** button to continue.



Database to Data Model

Visual Paradigm Database Reverse

Select Language

Language : Java

Result : Popup entities tree (can drag entities to diagram on demand)

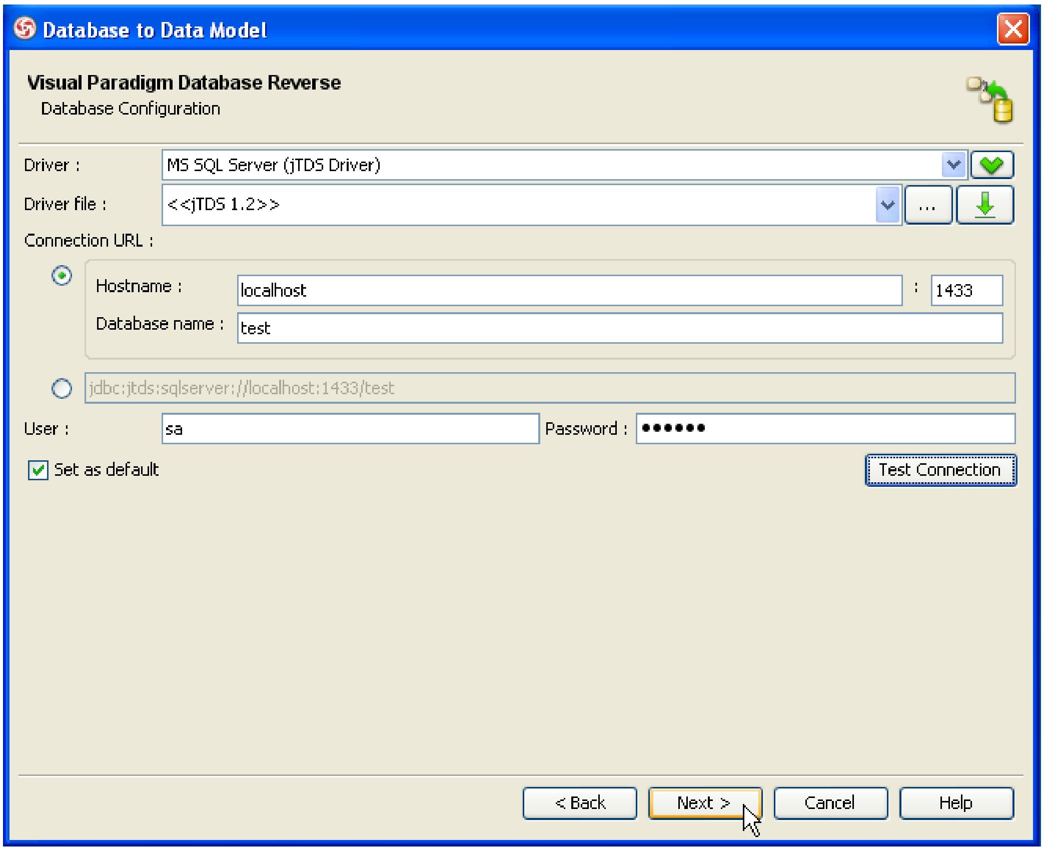
☑ Reverse Table

□ Reverse Stored Procedure

□ Reverse Trigger

Figure 3-2 *Reverse table*

4. Fill in the database setting and click **Next >** button.



Database to Data Model

Visual Paradigm Database Reverse

Database Configuration

Driver : MS SQL Server (jTDS Driver)

Driver file : <<jTDS 1.2>>

Connection URL :

● Hostname : localhost : 1433

Database name : test

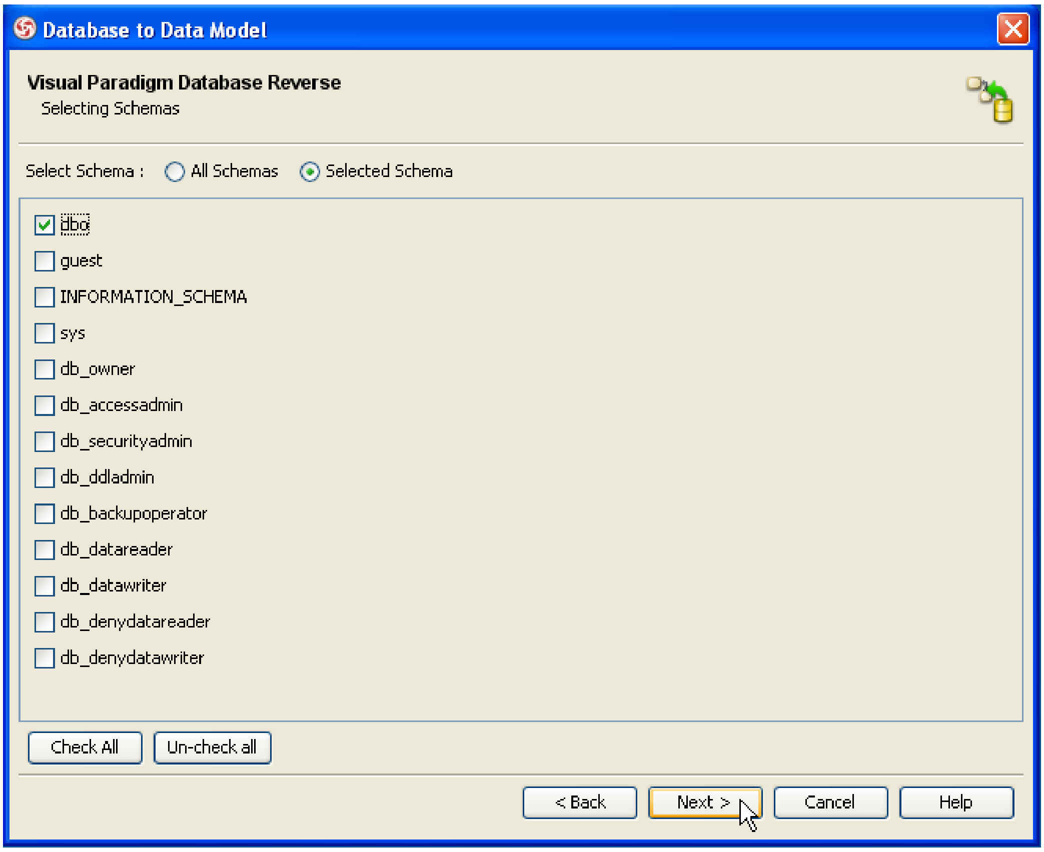
○ jdbc:jtds:sqlserver://localhost:1433/test

User : sa Password : ••••••

☑ Set as default

Figure 3-3 *Database setting*

5. Select the **Selected Schema** and check the schema containing your tables, and click **Next >** button.



Database to Data Model

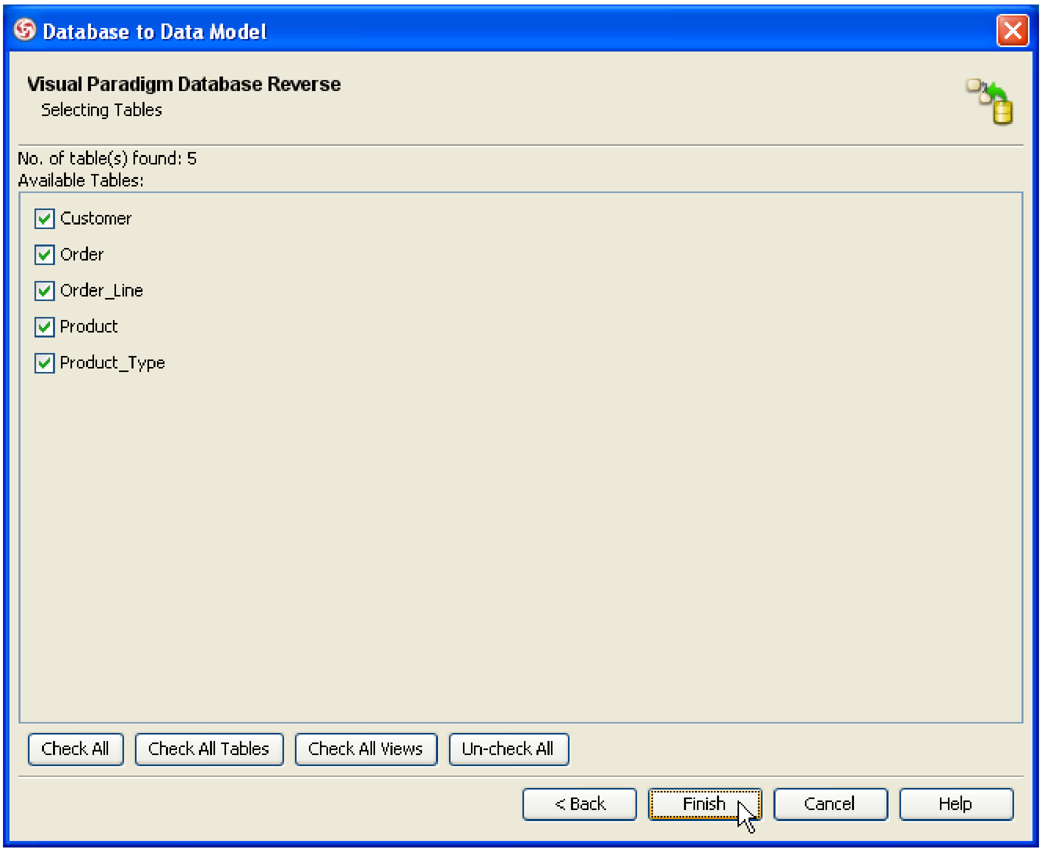
Visual Paradigm Database Reverse
Selecting Schemas

Select Schema : ○ All Schemas ● Selected Schema

☑ dbo
□ guest
□ INFORMATION_SCHEMA
□ sys
□ db_owner
□ db_accessadmin
□ db_securityadmin
□ db_ddladmin
□ db_backupoperator
□ db_datareader
□ db_datawriter
□ db_denydatareader
□ db_denydatawriter

Figure 3-4 *Select schema*

6. Check the tables you want to reverse, and click **Finish** button to start reverse.



Database to Data Model

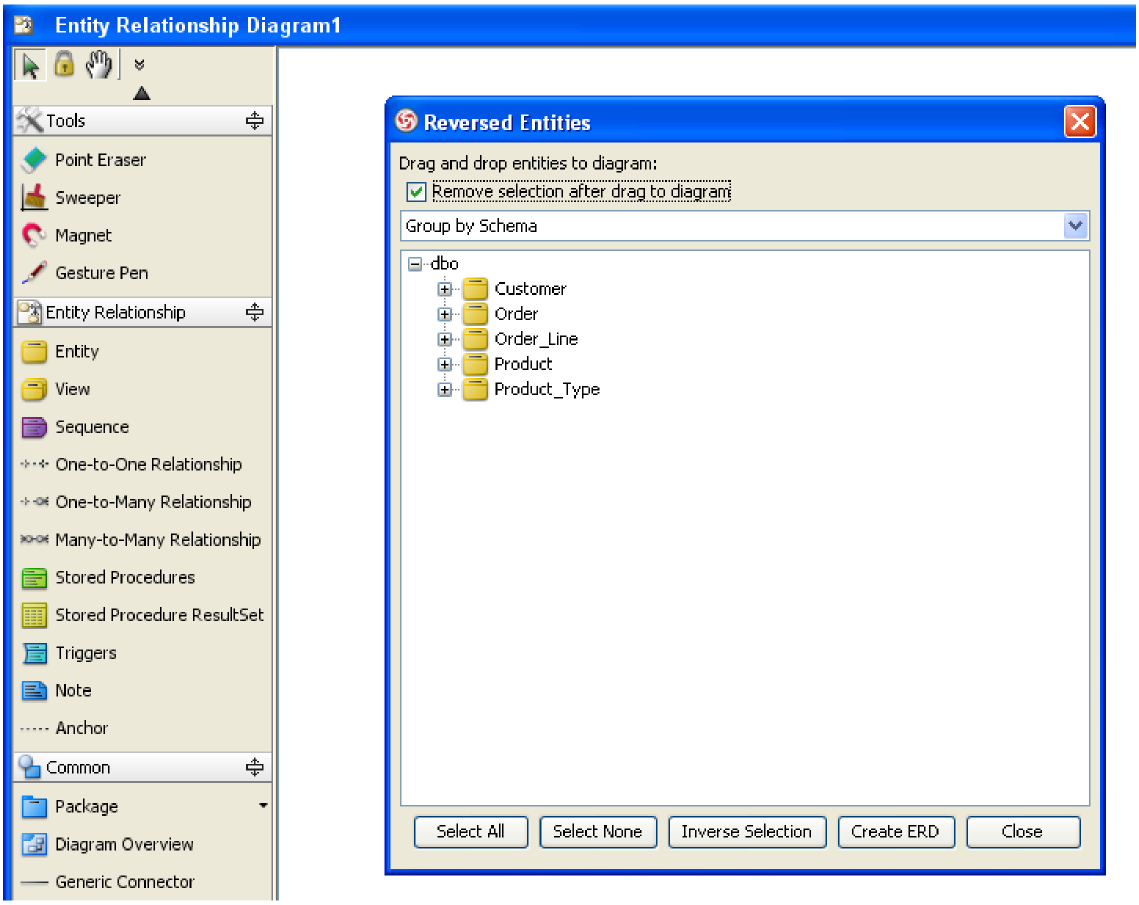
Visual Paradigm Database Reverse
Selecting Tables

No. of table(s) found: 5
Available Tables:

☑ Customer
☑ Order
☑ Order_Line
☑ Product
☑ Product_Type

Figure 3-5 *Select tables*

7. After reverse, a new **Entity Relationship Diagram** is created automatically, with a **Reversed Entities** dialog.



Drag and drop entities to diagram:

☑ Remove selection after drag to diagram

Group by Schema

- dbo
  - Customer
  - Order
  - Order_Line
  - Product
  - Product_Type

__Figure 3-6__ Reversed entities

8. Select the tables and drag on the diagram.

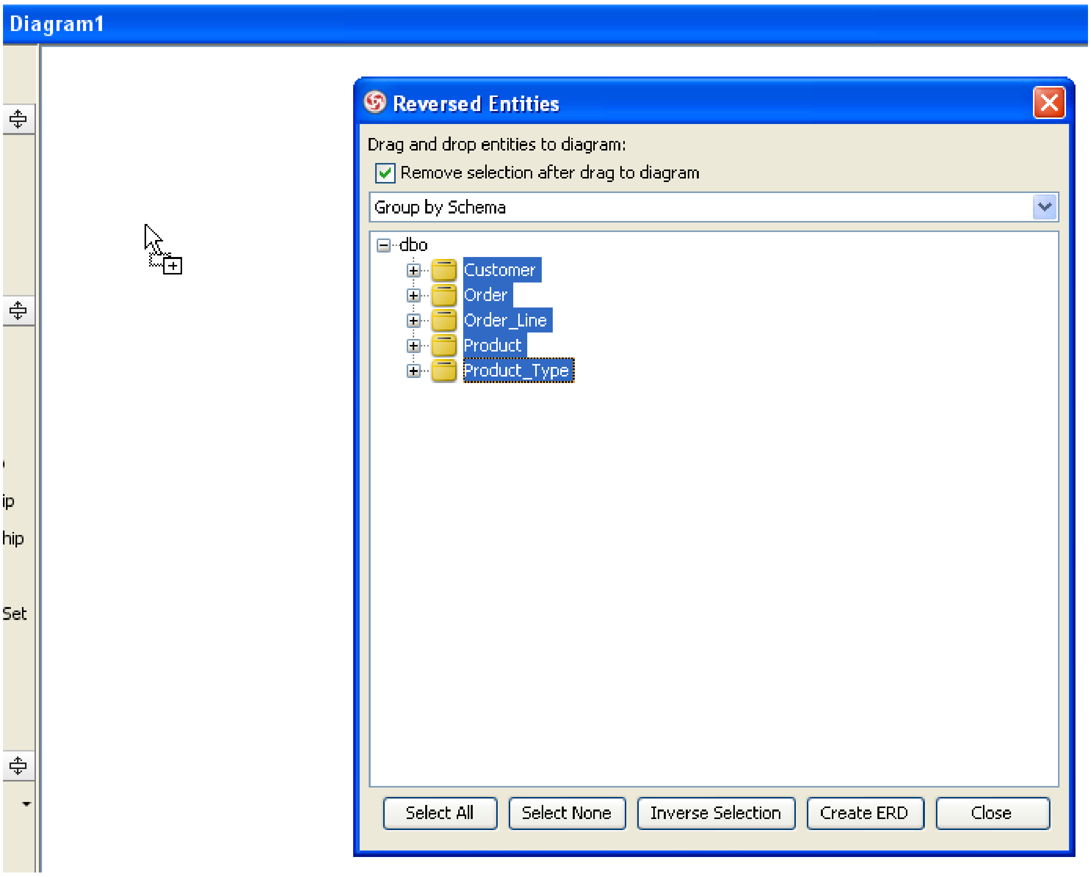


Diagram1

Reversed Entities

Drag and drop entities to diagram:

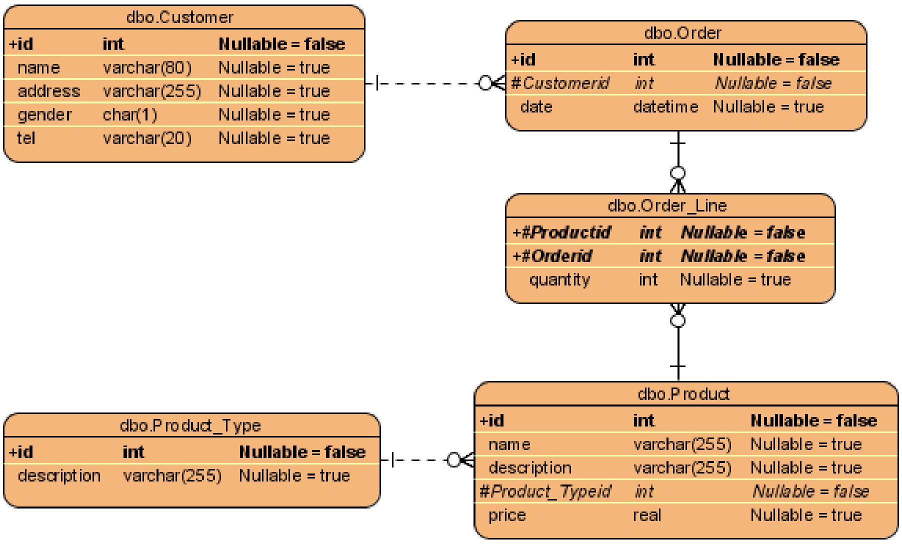
☑ Remove selection after drag to diagram

Group by Schema

- dbo
  - Customer
  - Order
  - Order_Line
  - Product
  - Product_Type

Figure 3-7 Drag and drop tables to diagram

9. The tables were created on the diagram, click **Close** button to finish reversing database.



dbo.Customer		
+id	**int**	**Nullable = false**
name	varchar(80)	Nullable = true
address	varchar(255)	Nullable = true
gender	char(1)	Nullable = true
tel	varchar(20)	Nullable = true

dbo.Order		
+id	**int**	**Nullable = false**
#Customerid	*int*	*Nullable = false*
date	datetime	Nullable = true

dbo.Order_Line		
+#Productid	***int***	***Nullable = false***
+#Orderid	***int***	***Nullable = false***
quantity	int	Nullable = true

dbo.Product_Type		
+id	**int**	**Nullable = false**
description	varchar(255)	Nullable = true

dbo.Product		
+id	**int**	**Nullable = false**
name	varchar(255)	Nullable = true
description	varchar(255)	Nullable = true
#Product_Typeid	*int*	*Nullable = false*
price	real	Nullable = true

Figure 3-8 Reversed ERD

Reversing DDL

1. Create a new project.
2. Select **Tools > Reverse DDL...** from the main menu.

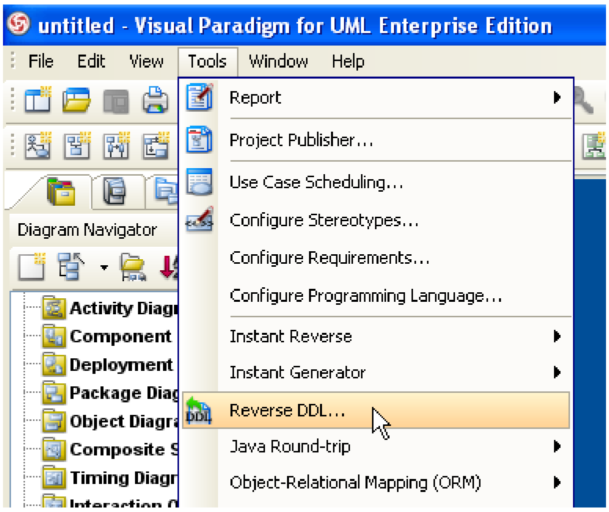
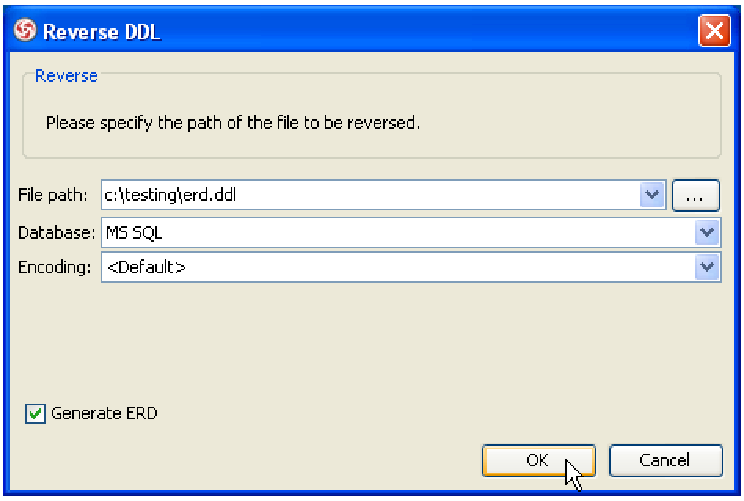


Figure 3-9 *Reverse DDL*

3. Fill in the filename of DDL and select database in **Reverse DDL** dialog. Click **OK** button to continue.



Reverse DDL

Reverse

Please specify the path of the file to be reversed.

File path: c:\testing\erd.ddl

Database: MS SQL

Encoding: <Default>

☑ Generate ERD

Figure 3-10 *Reverse DDL dialog*

Generate Class Diagram from ERD

1. Open an ERD.

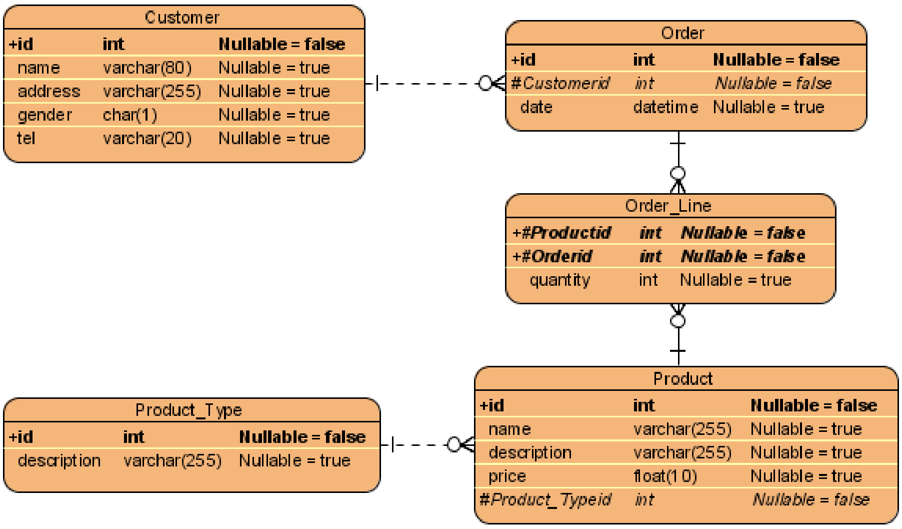

Customer
+id int Nullable = false
name varchar(80) Nullable = true
address varchar(255) Nullable = true
gender char(1) Nullable = true
tel varchar(20) Nullable = true
Order
+id int Nullable = false
#Customerid int Nullable = false
date datetime Nullable = true
Order_Line
+#Productid int Nullable = false
+#Orderid int Nullable = false
quantity int Nullable = true
Product_Type
+id int Nullable = false
description varchar(255) Nullable = true
Product
+id int Nullable = false
name varchar(255) Nullable = true
description varchar(255) Nullable = true
price float(10) Nullable = true
#Product_Typeid int Nullable = false

***Figure 4-1** ERD*

2. Right click on diagram, select **Synchronize to Class Diagram** from the popup menu.

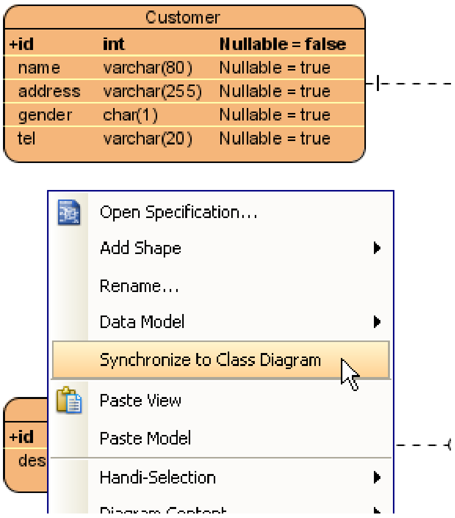
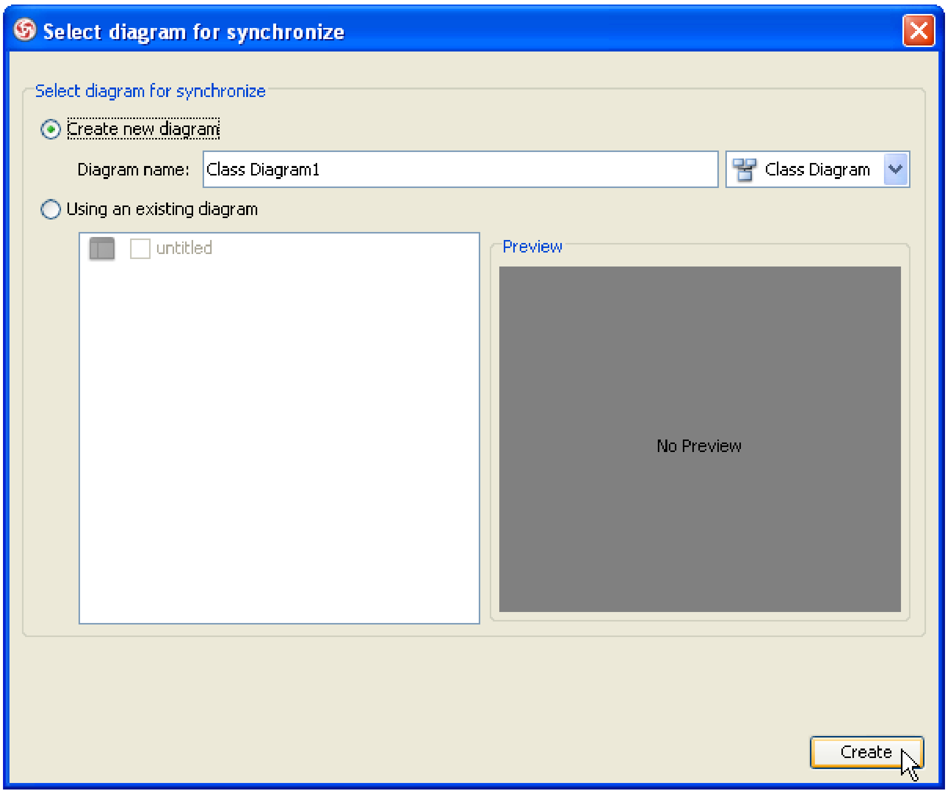

Customer
+id int Nullable = false
name varchar(80) Nullable = true
address varchar(255) Nullable = true
gender char(1) Nullable = true
tel varchar(20) Nullable = true
Open Specification...
Add Shape
Rename...
Data Model
Synchronize to Class Diagram
Paste View
Paste Model
Handi-Selection
Diagram Content
+id
des

***Figure 4-2** Synchronize to class diagram*

3. For the first time synchronize ERD to class diagram, it will show a **Select diagram for synchronize** dialog. Either select **Create new diagram** and specify diagram name, or select **Using an existing diagram** and select an existing diagram from the list. Click **Create** button to continue.



Select diagram for synchronize

Select diagram for synchronize

● Create new diagram

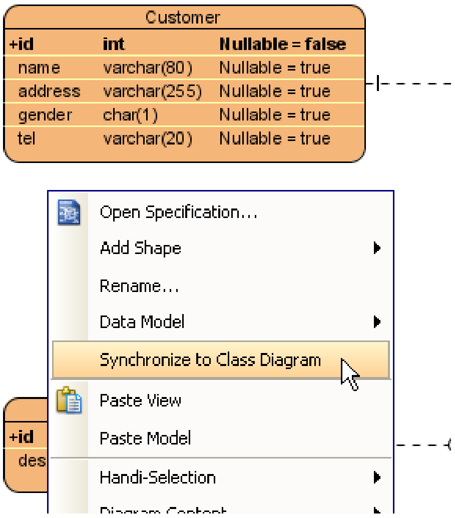
Diagram name: Class Diagram1 | Class Diagram

○ Using an existing diagram

□ untitled

Figure 4-3 *Select diagram for synchronize*

4. If there are new ERD model elements created since last synchronize, a **Synchronize to Class Diagram** dialog will show for you to rename the generated model elements. Click **OK** button after finish renaming.



Customer

+id	int	Nullable = false
name	varchar(80)	Nullable = true
address	varchar(255)	Nullable = true
gender	char(1)	Nullable = true
tel	varchar(20)	Nullable = true

Figure 4-4 *Synchronize to class diagram*

5. A class diagram with generated classes, associations created.

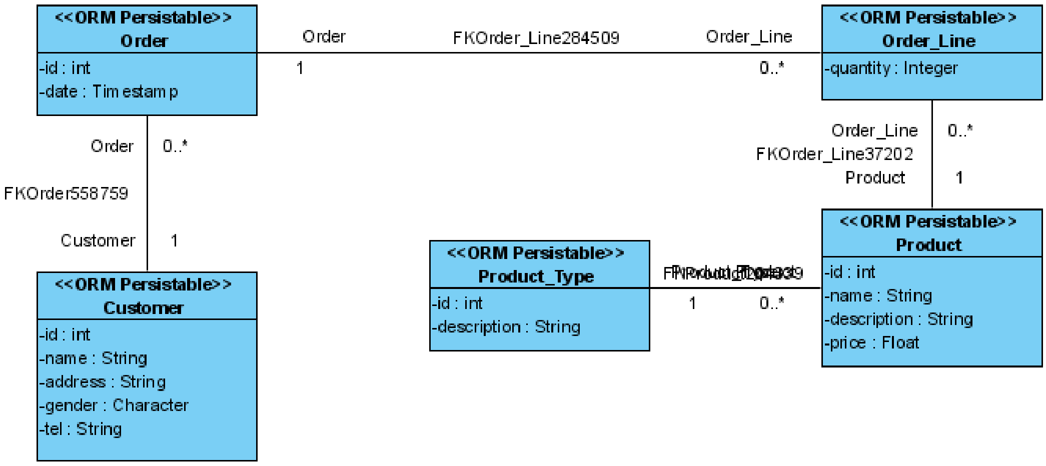


Figure 4-5 *Generated class diagram*

Synchronize from Class Diagram to ERD

1. Select ORM-Persistable Class from the diagram palette.

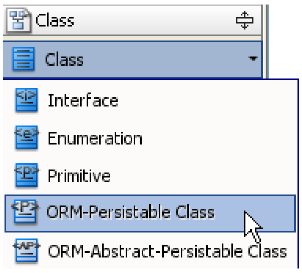


Figure 4-6 Create ORM-Persistable class

2. Click on the diagram to name it as *Supplier*, create attributes and association as follow.

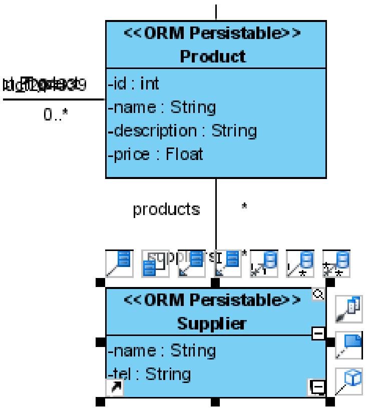
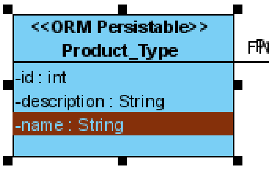


Figure 4-7 Create Supplier class

3. Add *name* attribute to Product_Type class.

Figure 4-8 Create attribute



Configure Key naming pattern

1. Select Tools > Options... from the main menu.

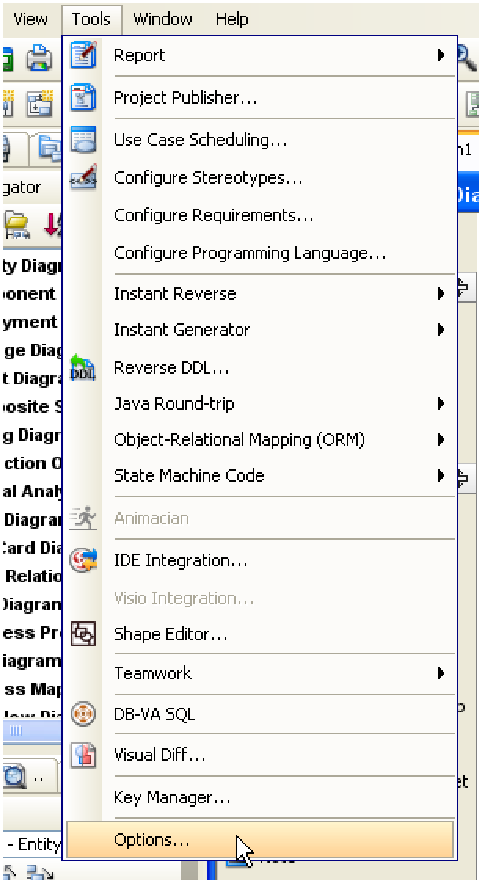


Figure 4-12 Options

2. Select **Diagramming** from the list on the left, then select **ERD & ORM** tab.

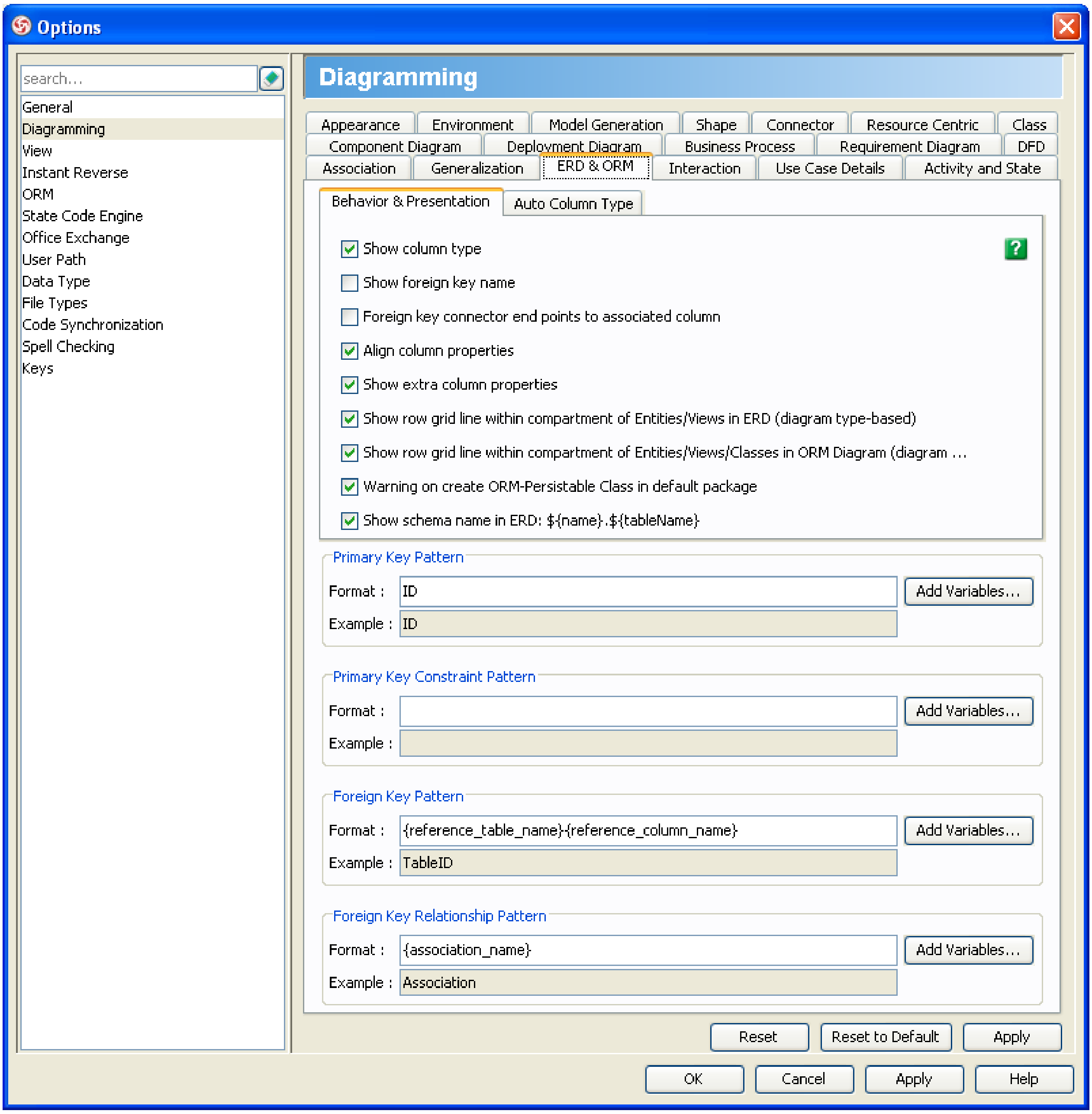


Figure 4-13 ERD & ORM option

☑ Show column type

□ Show foreign key name

□ Foreign key connector end points to associated column

☑ Align column properties

☑ Show extra column properties

☑ Show row grid line within compartment of Entities/Views in ERD (diagram type-based)

☑ Show row grid line within compartment of Entities/Views/Classes in ORM Diagram (diagram ...

☑ Warning on create ORM-Persistable Class in default package

☑ Show schema name in ERD: ${name}.${tableName}

Primary Key Pattern

Format : ID

Example : ID

Primary Key Constraint Pattern

Format :

Example :

Foreign Key Pattern

Format : {reference_table_name}{reference_column_name}

Example : TableID

Foreign Key Relationship Pattern

Format : {association_name}

Example : Association

Generating Code and Database

1. Select **Tools > Object-Relational Mapping (ORM) > Generate Code...** from the main menu.

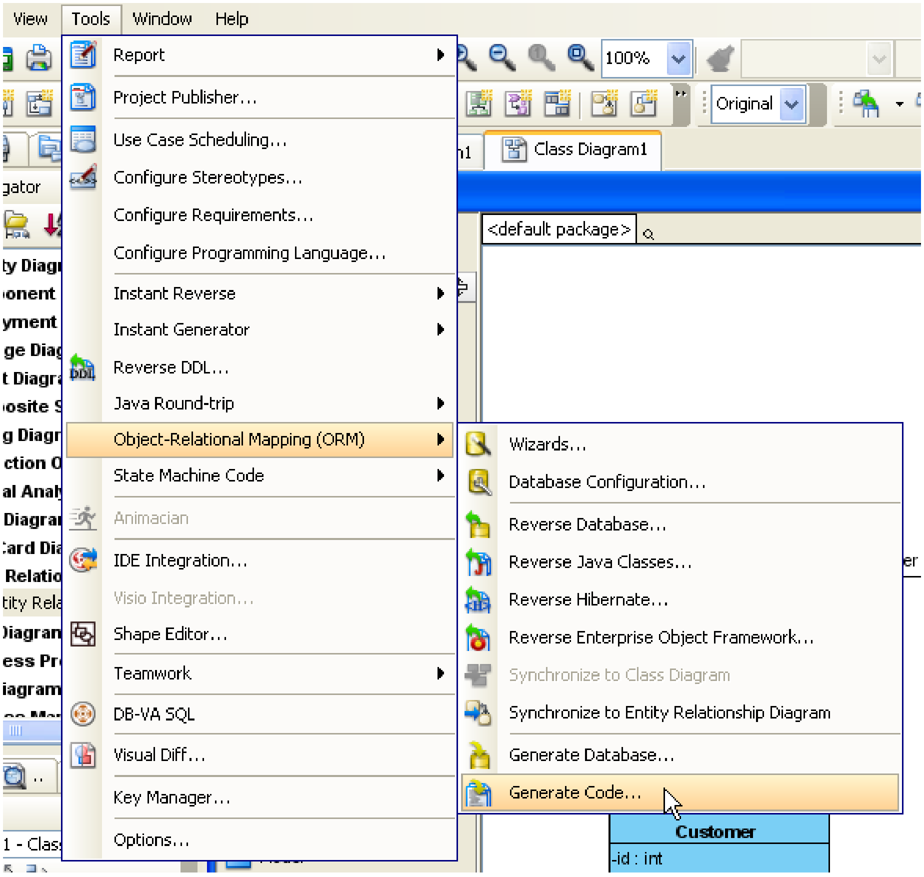


***Figure 5-1** Generate code*

2. Fill in the **Output Path** and select options:

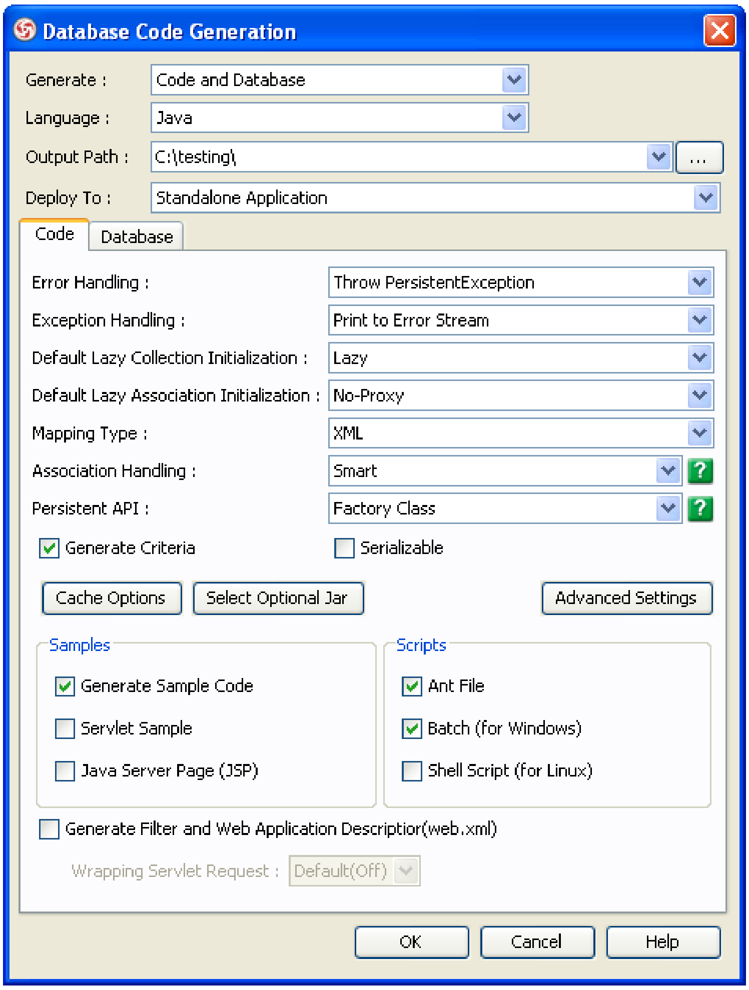


Figure 5-2 Database code generation

Database Code Generation

Generate : Code and Database

Language : Java

Output Path : C:\testing\

Deploy To : Standalone Application

Code | Database

Error Handling : Throw PersistentException

Exception Handling : Print to Error Stream

Default Lazy Collection Initialization : Lazy

Default Lazy Association Initialization : No-Proxy

Mapping Type : XML

Association Handling : Smart

Persistent API : Factory Class

☑ Generate Criteria

□ Serializable

Samples

☑ Generate Sample Code

□ Servlet Sample

□ Java Server Page (JSP)

Scripts

☑ Ant File

☑ Batch (for Windows)

□ Shell Script (for Linux)

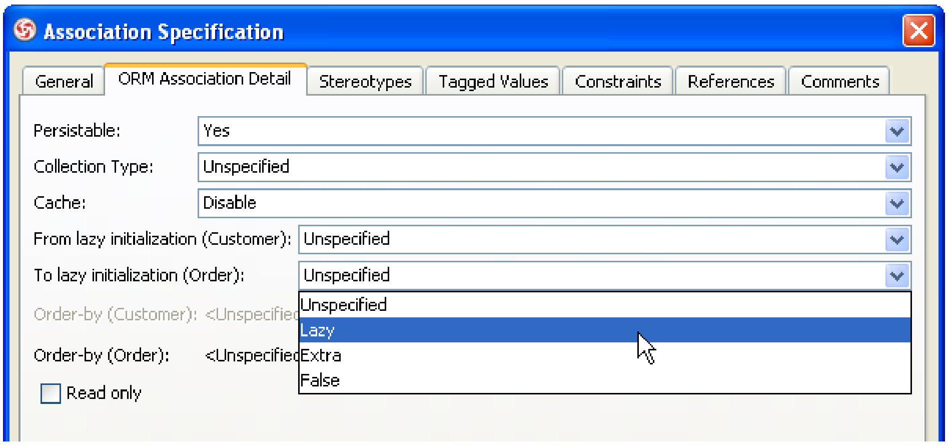
□ Generate Filter and Web Application Descriptior(web.xml)

Wrapping Servlet Request : Default(Off)

Lazy Collection Setting

Setting Lazy Collection for association

1. Open specification dialog of association.
2. Switch to ORM Association Detail tab, select **Lazy** or **Extra** for **From lazy initialization** or **To lazy initialization**, depending on which side multiplicity is *. Lazy collection is fetched when the application invokes an operation upon that collection. Extra lazy supports individual elements of the collection are accessed from the database as needed, rather than fetch the whole collection. If the value is **Unspecified**, it will follow the default lazy collection setting described below.



Persistable: Yes

Collection Type: Unspecified

Cache: Disable

From lazy initialization (Customer): Unspecified

To lazy initialization (Order): Unspecified

Order-by (Customer): <Unspecified

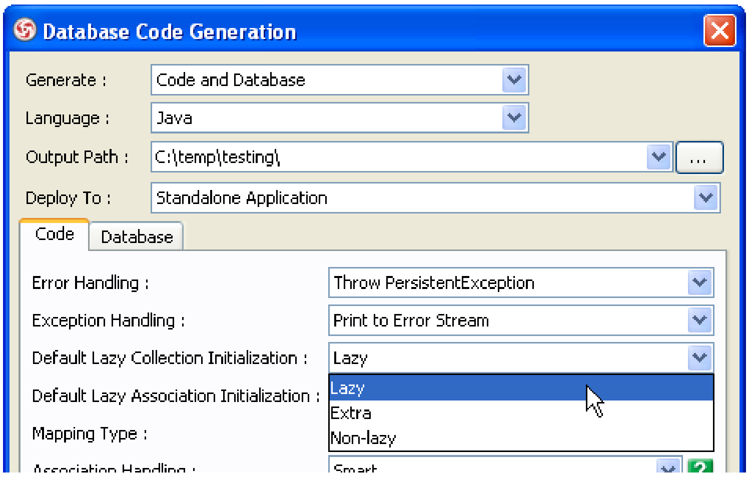
Order-by (Order): <Unspecified

☐ Read only

Figure 5-3 *Lazy collection setting*

Setting Default Lazy Collection when generating ORM

1. Open **Database Code Generation** dialog.
2. Specify a value for **Default Lazy Collection Initialization**.



Generate : Code and Database

Language : Java

Output Path : C:\temp\testing\

Deploy To : Standalone Application

Error Handling : Throw PersistentException

Exception Handling : Print to Error Stream

Default Lazy Collection Initialization : Lazy

Default Lazy Association Initialization :

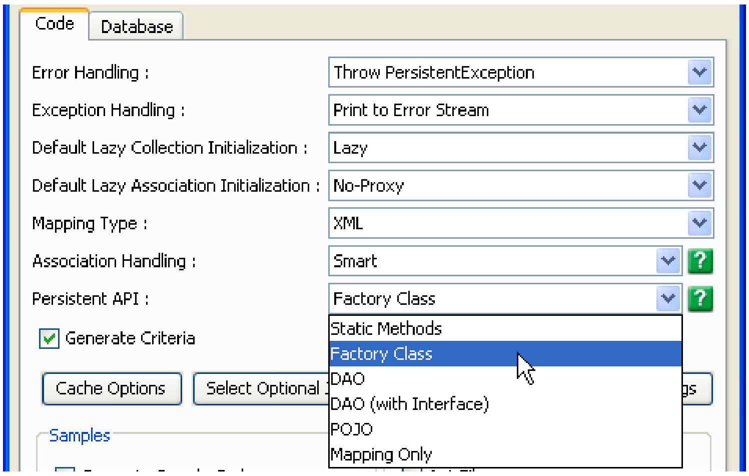
Mapping Type :

Association Handling : Smart

Figure 5-4 *Default lazy collection setting*

Persistent API

The persistent API setting supports various styles for generated code. It can be configurated in **Database Code Generation** dialog.



Code | Database

Error Handling : Throw PersistentException

Exception Handling : Print to Error Stream

Default Lazy Collection Initialization : Lazy

Default Lazy Association Initialization : No-Proxy

Mapping Type : XML

Association Handling : Smart

Persistent API : Factory Class

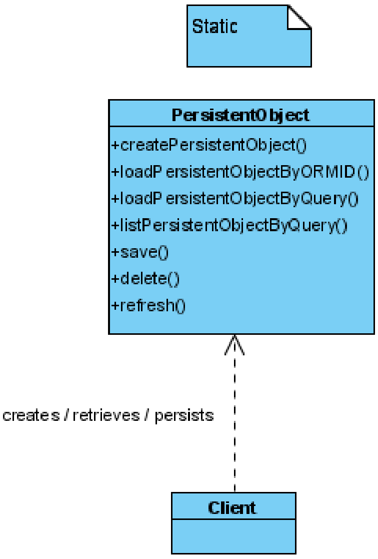
- Static Methods
- Factory Class
- DAO
- DAO (with Interface)
- POJO
- Mapping Only

☑ Generate Criteria

__Figure 5-5__ Persistent API setting

Static Methods

Static methods generate all persistent methods in the persistent class, client can access the methods in the same persistent object.



Static
PersistentObject
+createPersistentObject()
+loadPersistentObjectByORMID()
+loadPersistentObjectByQuery()
+listPersistentObjectByQuery()
+save()
+delete()
+refresh()
creates / retrieves / persists
Client

__Figure 5-6__ Static methods

Factory Class

Factory class generate save/delete/refresh methods in persistent class, other persistent methods that return persistent object are generated in factory class.

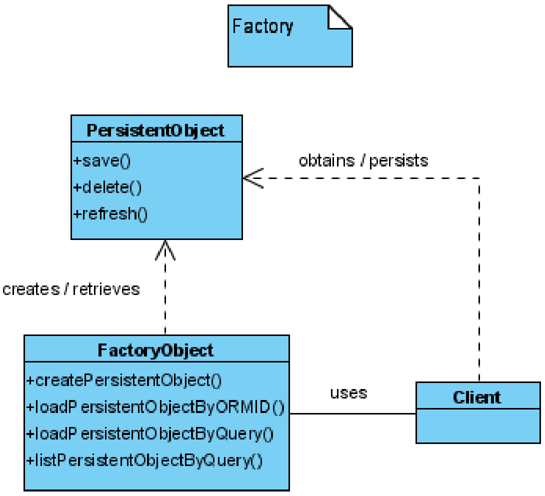

Factory
PersistentObject
+save()
+delete()
+refresh()
obtains / persists
creates / retrieves
FactoryObject
+createPersistentObject()
+loadPersistentObjectByORMID()
+loadPersistentObjectByQuery()
+listPersistentObjectByQuery()
uses
Client

Figure 5-7 *Factory class*

DAO

DAO generate all persistent methods in DAO class, a DAO class is generate for each persistent class.

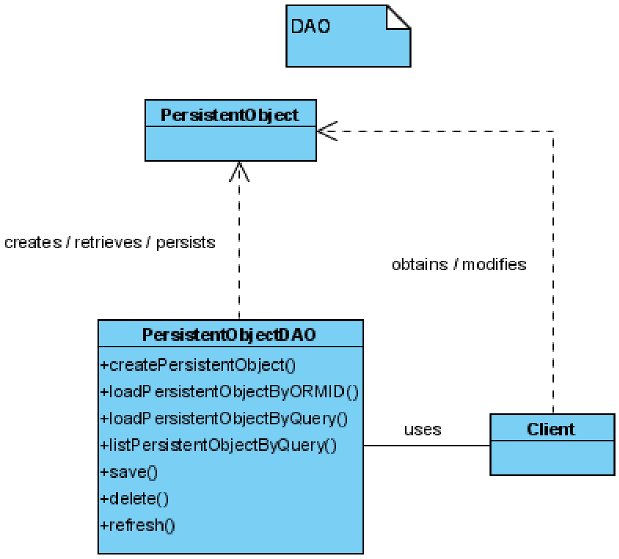

DAO
PersistentObject
creates / retrieves / persists
obtains / modifies
PersistentObjectDAO
+createPersistentObject()
+loadPersistentObjectByORMID()
+loadPersistentObjectByQuery()
+listPersistentObjectByQuery()
+save()
+delete()
+refresh()
uses
Client

Figure 5-8 *DAO*

DAO (with Interface)

DAO (with Interface) generate all persistent methods signature in DAO interface. A DAO interface is generate for each persistent class, and a corresponding DAO implementation class is generated with default persistent implement.

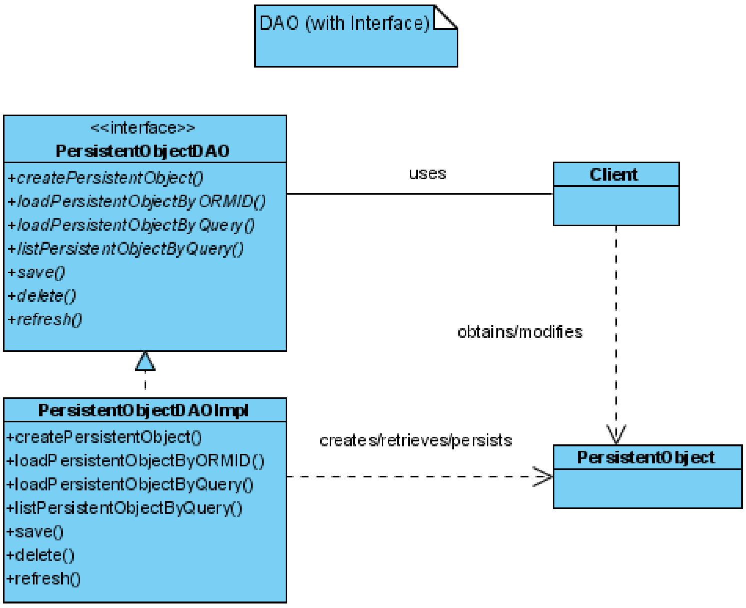

DAO (with Interface)
<<interface>>
PersistentObjectDAO
+createPersistentObject()
+loadPersistentObjectByORMID()
+loadPersistentObjectByQuery()
+listPersistentObjectByQuery()
+save()
+delete()
+refresh()
uses
Client
obtains/modifies
PersistentObjectDAOImpl
+createPersistentObject()
+loadPersistentObjectByORMID()
+loadPersistentObjectByQuery()
+listPersistentObjectByQuery()
+save()
+delete()
+refresh()
creates/retrieves/persists
PersistentObject

Figure 5-9 *DAO (with Interface)*

POJO

POJO generate persistent object in Plain Old Java Object style, without generating any persistent methods. Client can access persistent methods in PersistentManager object.

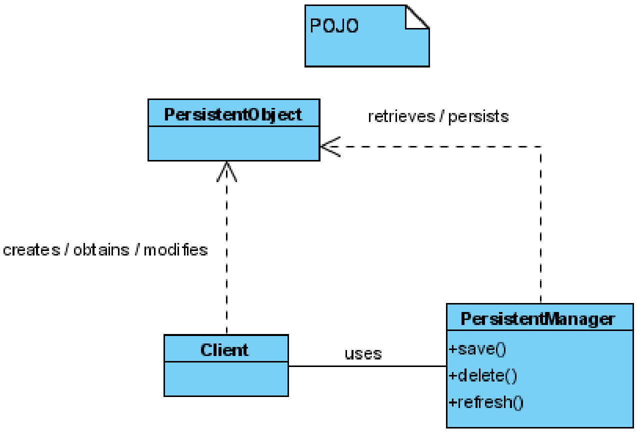


Figure 5-10 *POJO*

Mapping Only

Mapping Only does not generate any code, it only generate the XML mapping file required for ORM.

Using Generated Code

The following sections demonstrate how to use the generate ORM code with factory method persistent API.

Inserting Records

1. Create persistent object with factory create method.
2. Save persistent object with save method.

The following code demonstrate how to insert a *Product* record:

Selecting Records

Factory method provides a convinent listByQuery method, accept condition and order by as parameter, and return array of persistent object.

The following code demonstrate how to select a list of *Product* records, null for condition parameter will select all records, null for order by parameter does not sort in any order:

Another useful method to select a persistent object by ID is loadByORMID. The follow code demonstrate how to select a l *Prouct* record by ID.

Updating Records

1. Select a persistent object from database.
2. Update the persistent object.
3. Save persistent object with save method.

The following code demonstrate how to update a *Product* record:

Deleting Records

1. Select a persistent object from database.
2. Delete persistent object with delete method.

The following code demonstrate how to delete a *Product* record:

Using Model to manage project structure

Visual Paradigm for UML provided a very handy utility, Model Explorer for managing model elements, diagrams and different artifact of the VP Project.

1. Select **View > Panes > Model Explorer** from the main menu to open **Model Explorer**.

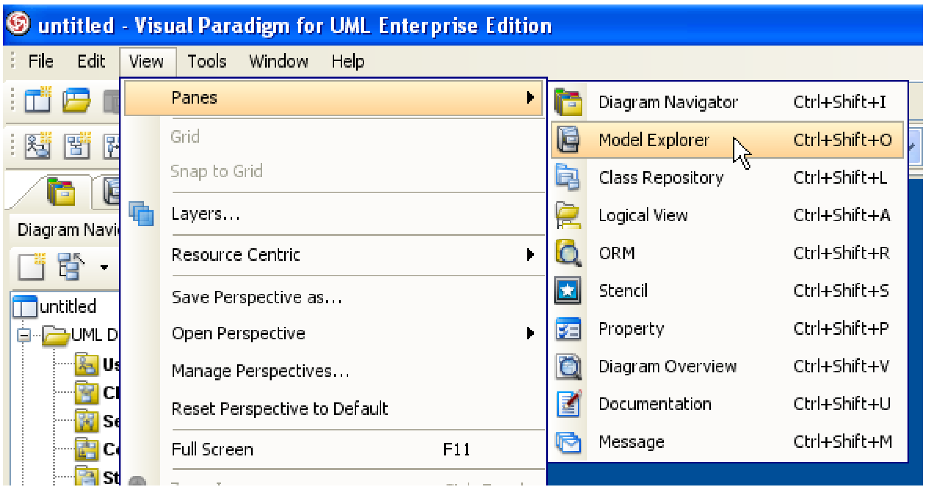


Figure 1-1 *Show model explorer*

2. Right click on the project node, select **Model > Analysis Model** from the popup to create an analysis model.

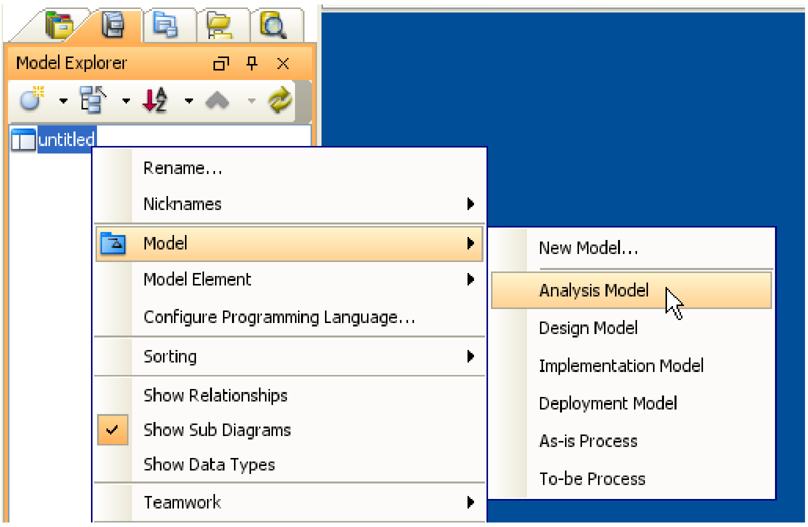


Figure 1-2 *Create analysis model*

3. Analysis Model appears on the **Model Explorer**.

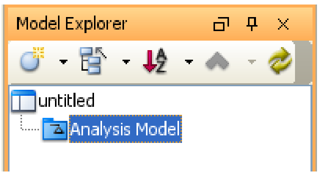


Figure 1-3 *Analysis model on mode explorer*

4. Right click on the Analysis Model node, select **Diagram > UML Diagrams > Use Case Diagram**.

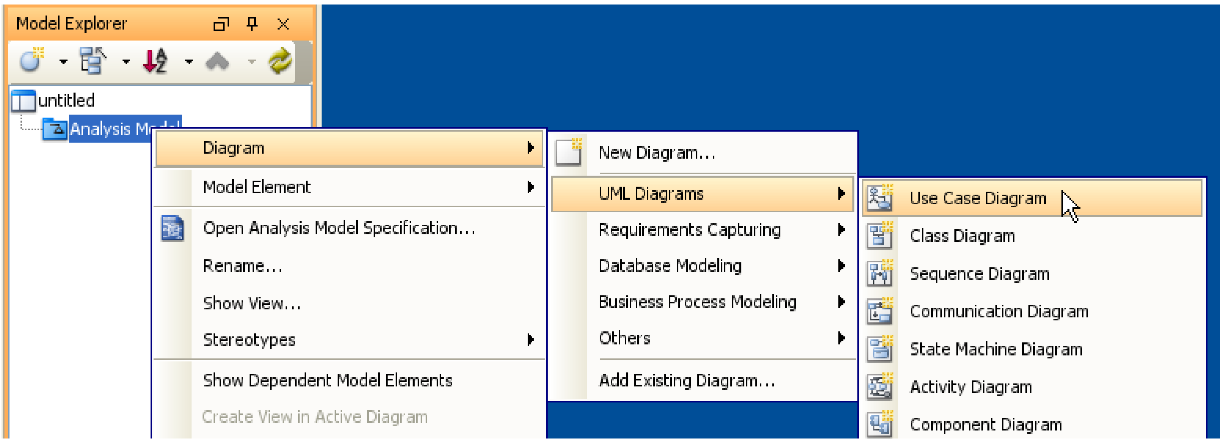


Figure 1-4 *Create Use Case Diagram*

5. Input the diagram name *ATM Use Case Diagram* in the text box.

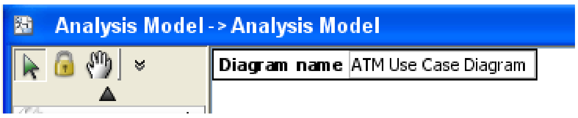


Diagram name ATM Use Case Diagram

Figure 1-5 *Input diagram name*

6. Draw actors and use cases on the diagram.

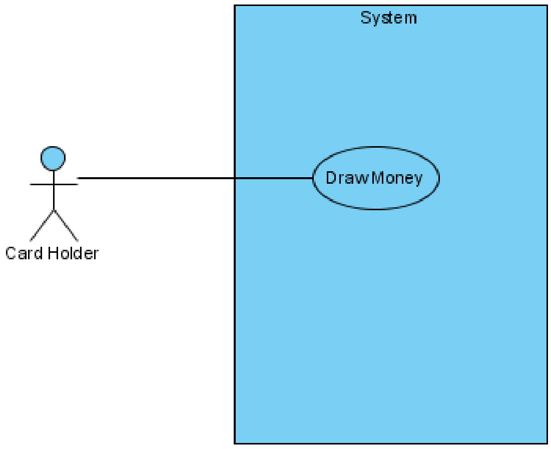


Figure 1-6 *Draw use case diagram*

7. The model elements created on diagram was added under Analysis Model.

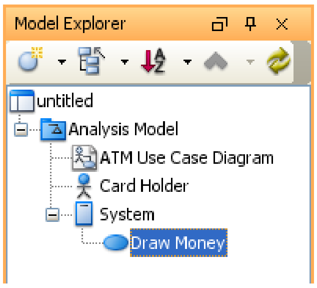


Figure 1-7 *Model elements under analysis model*

8. Right click on project node, select **Model > New Model...** from the popup menu.

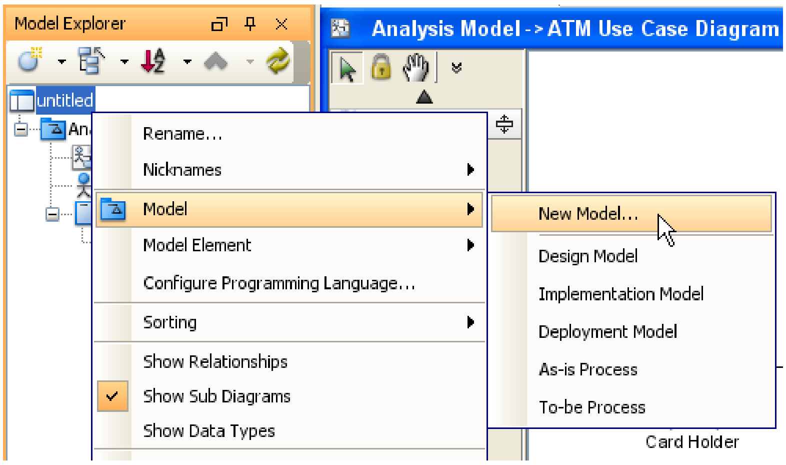


Figure 1-8 *New model*

9. Set model name *Use Case Model* and click **OK** button to confirm.
10. Right click on *Use Case Model* node, select **Diagram > Add Existing Diagram...** from the popup menu.

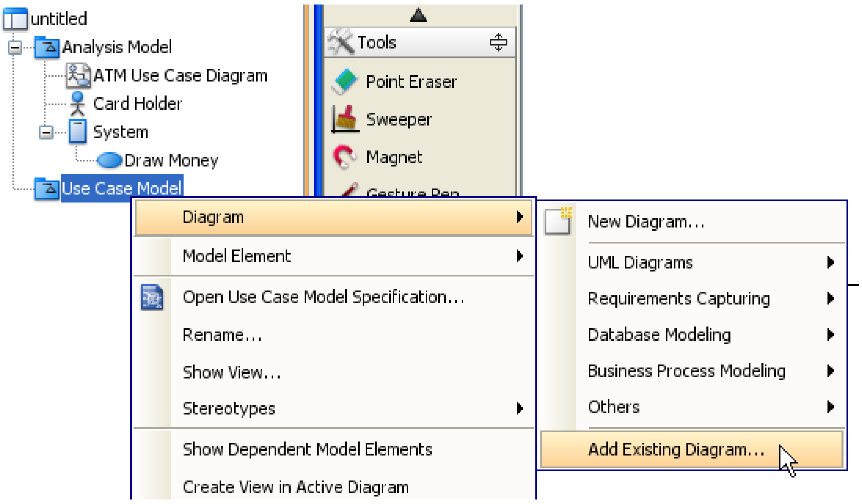
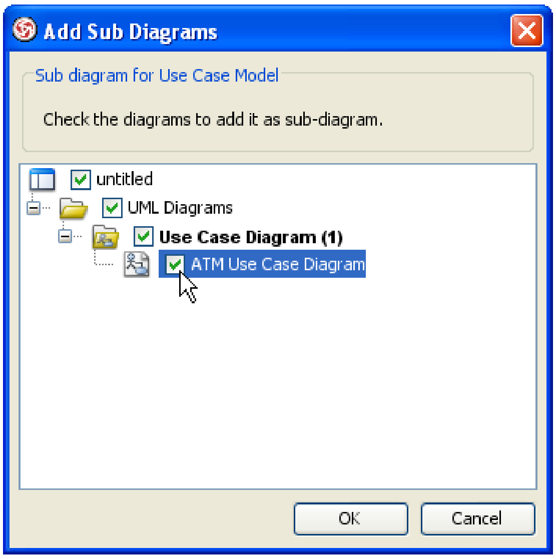


Figure 1-9 *Add existing diagram*

11. Click on the check box of *ATM Use Case Diagram*, click **OK** button.


Add Sub Diagrams

Sub diagram for Use Case Model

Check the diagrams to add it as sub-diagram.

Figure 1-10 *Select sub-diagram*

12. The actor and use case were moved to *Use Case Model*.

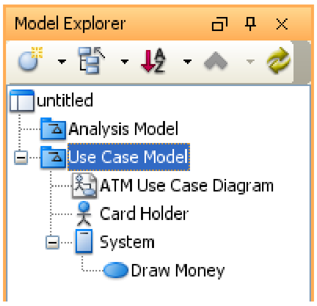


Figure 1-11 *Model elements moved to Use Case Model*

Developing context base diagram with multiple view

1. Open a class diagram *Safety Inspection*.
2. Select a class *Inspector*.
3. Select **Edit > Copy > Copy within VP-UML EE** from the main menu.

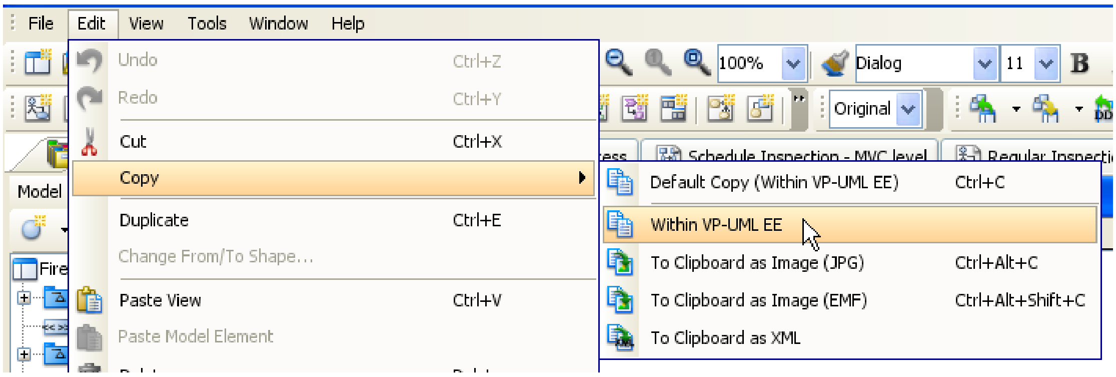


Figure 1-12 *Copy within VP-UML*

4. Create a new **Class Diagram** named as *Security*.
5. Select **Edit > Paste View** from the main menu.

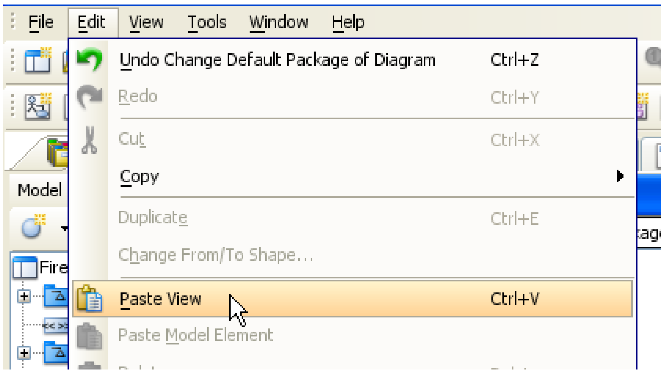


Figure 1-13 *Paste view*

6. The class added to *Security* diagram.

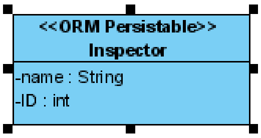


Figure 1-14 *Pasted view*

7. Draw more classes on the diagram.

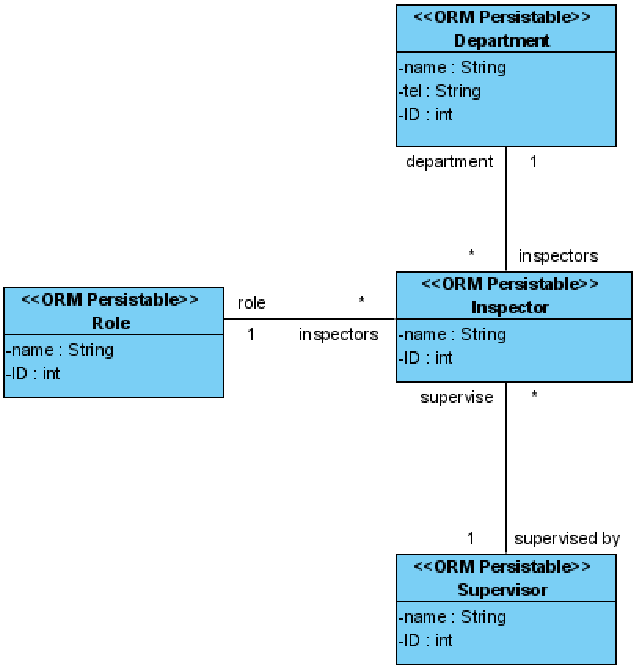


Figure 1-15 *Draw classes on diagram*

8. Open the *Safety Inspection* class diagram again, rename *Inspector* to *SafetyInspector*.

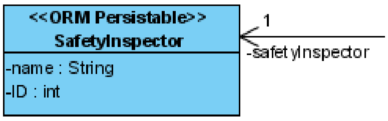


Figure 1-16 *Rename class*

9. Open *Security* class diagram again, the *Inspector* class also renamed to Safety *Inspector*. The *Inspector* class in both diagram are referencing the same model element, so edit either one will affect the other one.

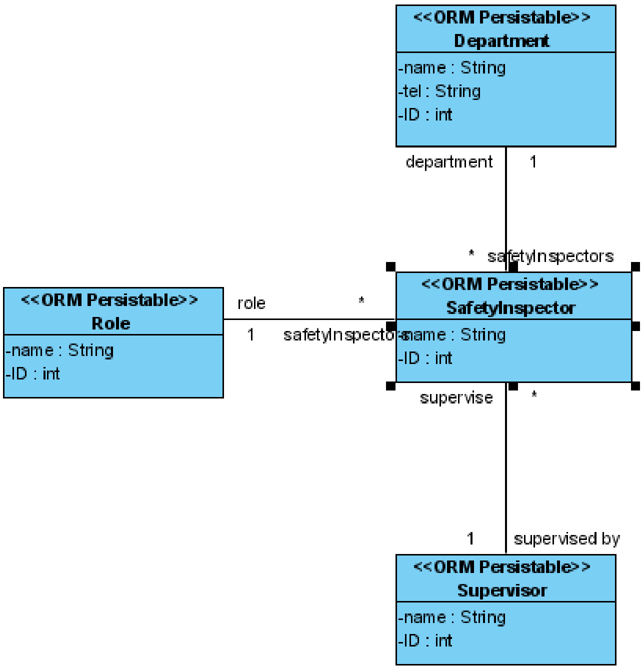


Figure 1-17 *Other views also renamed*

10. Create another class diagram *Login History*.

11. Locate *SafetyInspector* class on **Model Explorer**, drag it to the class diagram.

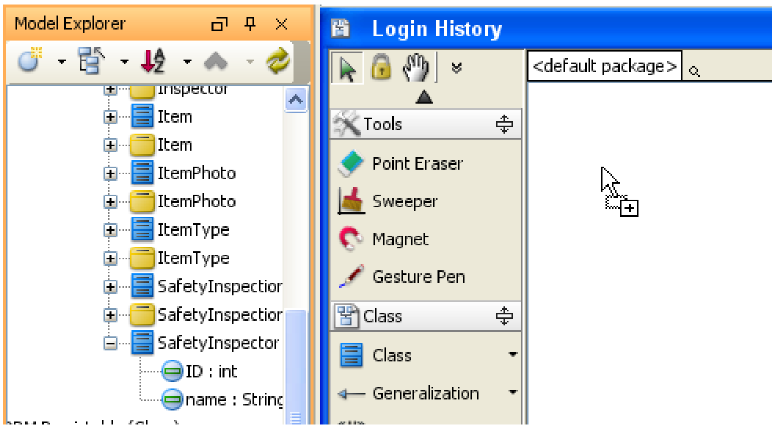


Figure 1-18 *Drag model elements from tree to diagram*

12. The class created on the diagram.

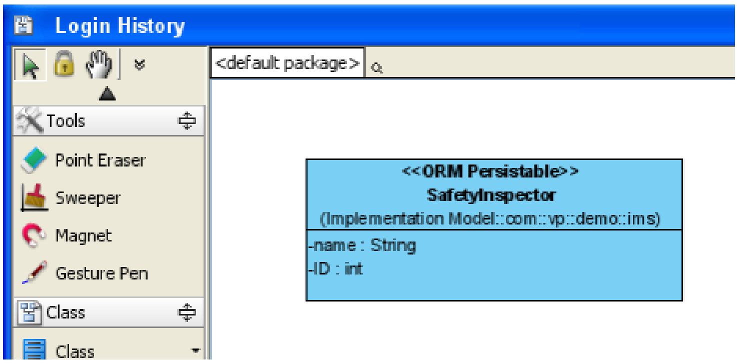


Figure 1-19 *Dropped model element from tree*

Understanding Master View

Master View is used to control the structure of the model element. One model element can have multiple views but only master view affect the structure.

1. Follow the steps on **02. Developing context base diagram with multiple view**.
2. Open *Security* class diagram.
3. Right click on diagram, select **Presentation Options > Configure Class Presentation Options...** from the popup menu.

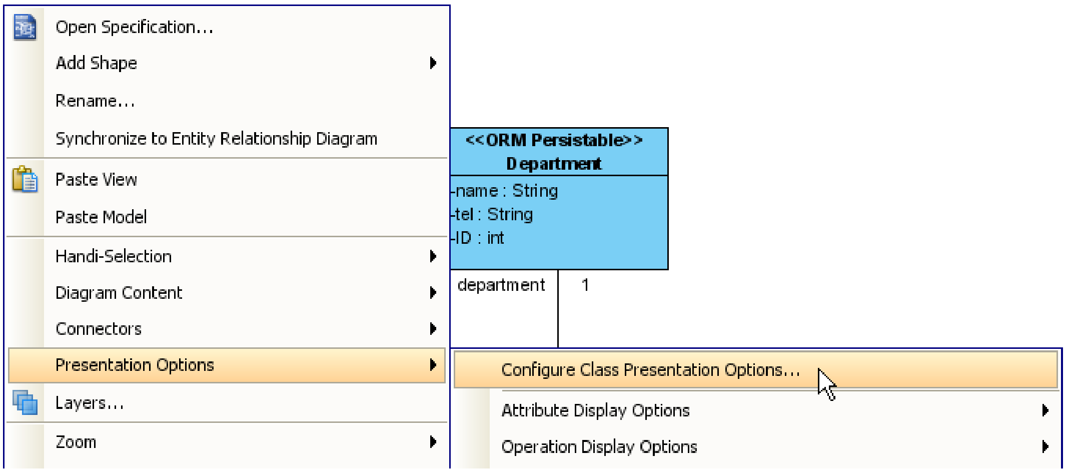
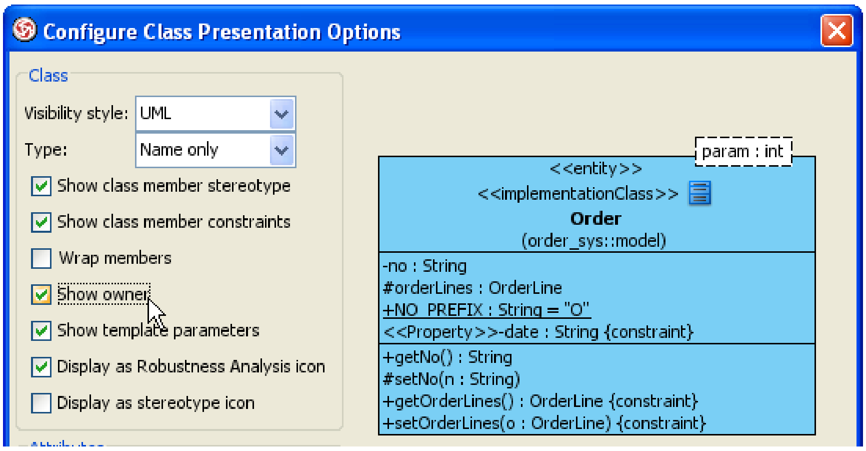


Figure 1-20 *Configure class presentation options*

4. Check the check box of Show owner.



Configure Class Presentation Options

Class

Visibility style: UML

Type: Name only

☑ Show class member stereotype

☑ Show class member constraints

☐ Wrap members

☑ Show owner

☑ Show template parameters

☑ Display as Robustness Analysis icon

☐ Display as stereotype icon

param : int
<<entity>>
<<implementationClass>>
Order
(order_sys::model)
-no : String
#orderLines : OrderLine
+NO_PREFIX : String = "O"
<<Property>>-date : String {constraint}
+getNo() : String
#setNo(n : String)
+getOrderLines() : OrderLine {constraint}
+setOrderLines(o : OrderLine) {constraint}

Figure 1-21 *Configure show owner option*

Showing different views

Show other views on diagram

1. Open a diagram
2. Select a shape and right click, select **Related Elements > Show Other Views...** from the popup menu.

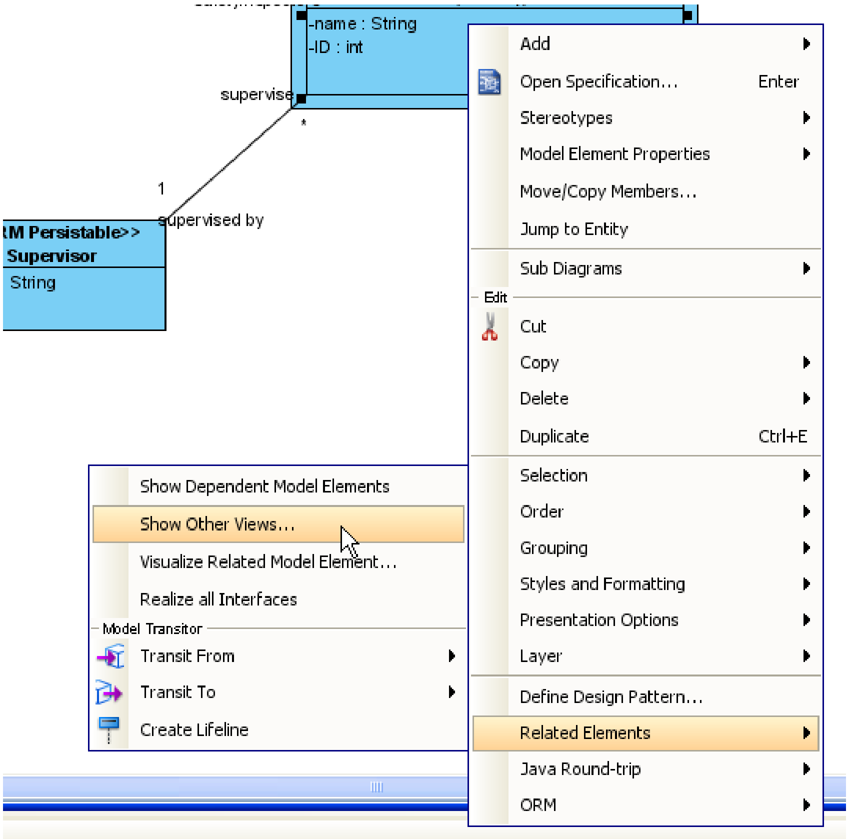


Figure 1-26 *Show other views*

3. The **Show View** dialog show a list of views (except the currently selected one) referencing the same model elements. Select a view from the list will show a preview image for the selected view.

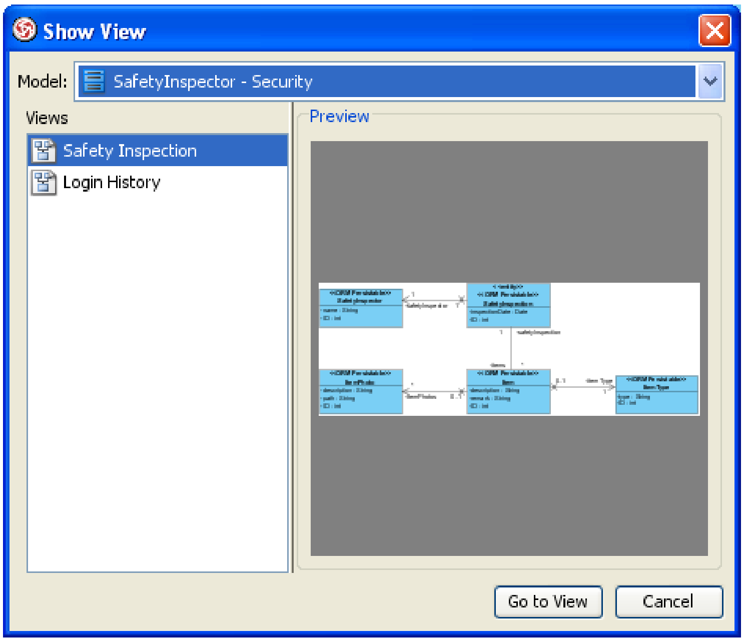


Figure 1-27 *Show view dialog*

4. Select a view and click **Go to View** button.

5. The diagram of select view will be opened, and the shape will be selected.

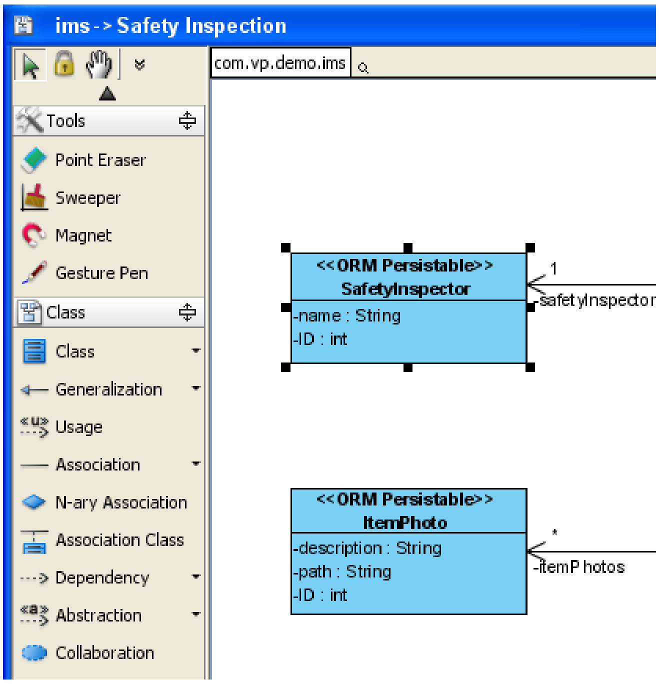


***Figure 1-28** Go to the selected view*

Show views on Model Explorer

1. Right click mode element on **Model Explorer**, select **Show View...** from the popup menu.

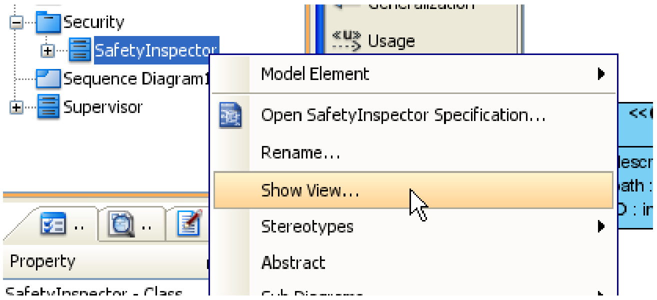


***Figure 1-29** Show view*

2. **Show View** dialog shows all views for the selected model element. The rest are the same as **Show other views on diagram**.

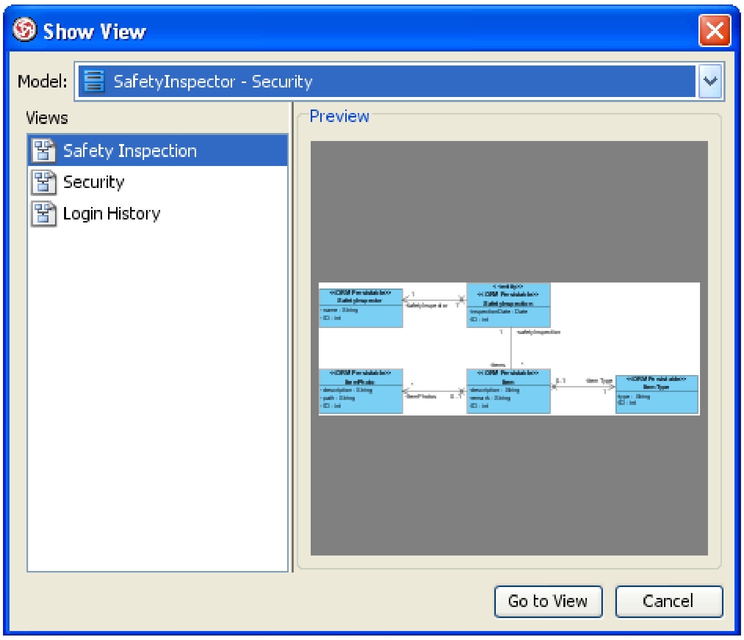


***Figure 1-30** Show view dialog*

Configure Stereotypes

1. Create a model element to configure stereotype.
2. Open the specification dialog for the model element.
3. Switch to Stereotypes tab and click **Edit Stereotypes...** button.

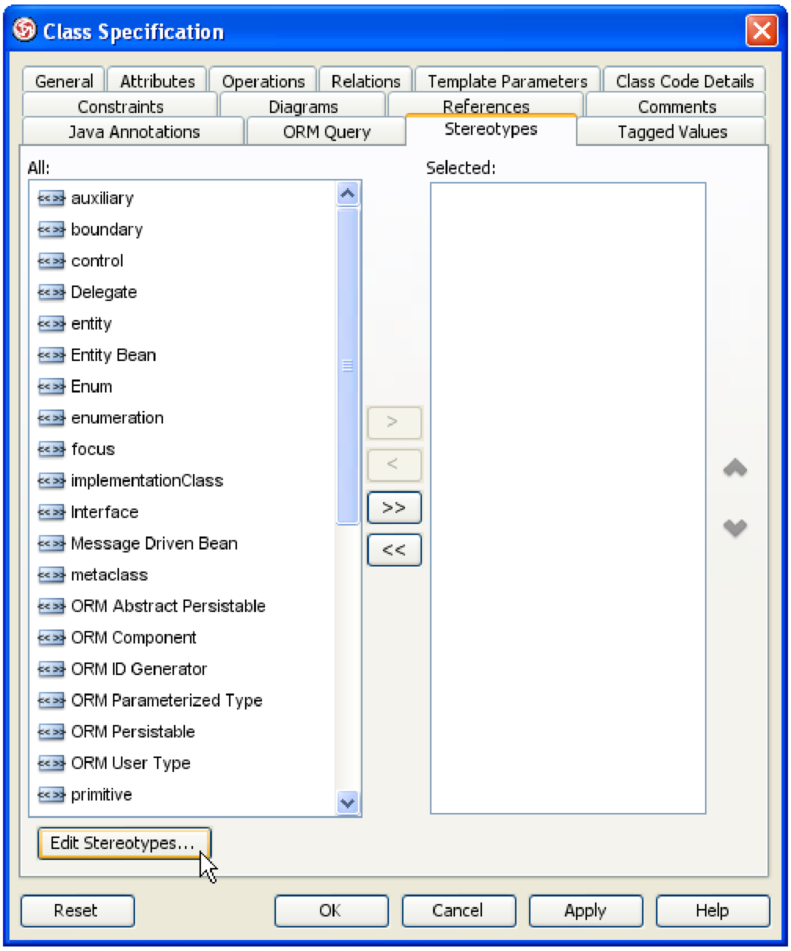
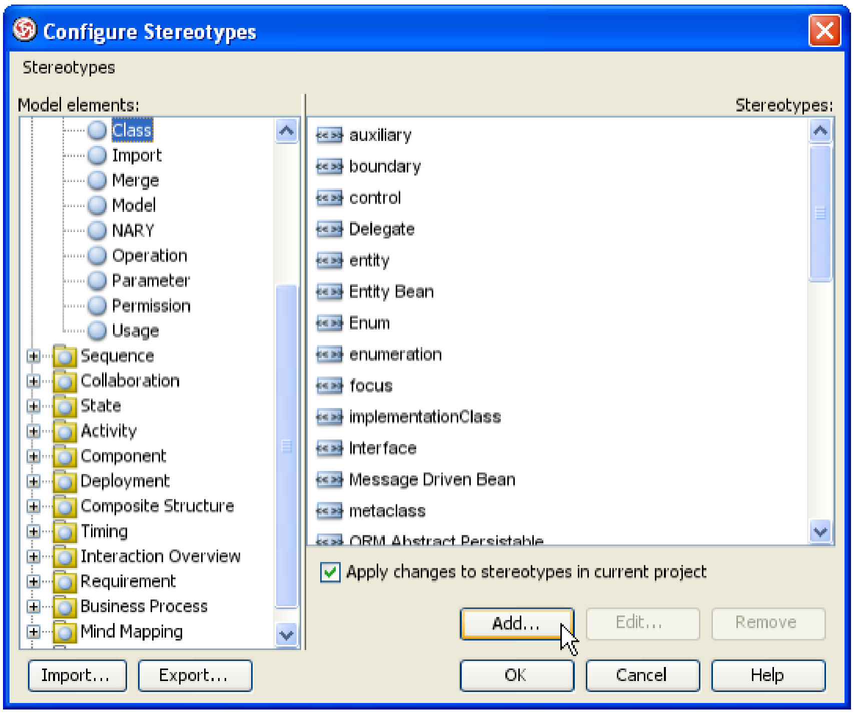


Figure 2-1 *Edit stereotypes*

4. The model element will be selected automatically, and the available stereotypes for the selected model element is show on the right. Click **Add...** button to create new stereotype.



☑ Apply changes to stereotypes in current project

Figure 2-2 *Add stereotype*

Apply Stereotype to Model Element

1. Create a shape.

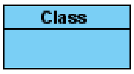


Figure 1-5 Create shape

2. Right click on the shape, select **Stereotype** from the popup menu.
3. A list of stereotypes will show on popup.
   - If the stereotype appears on the list, click on it to apply the stereotype.

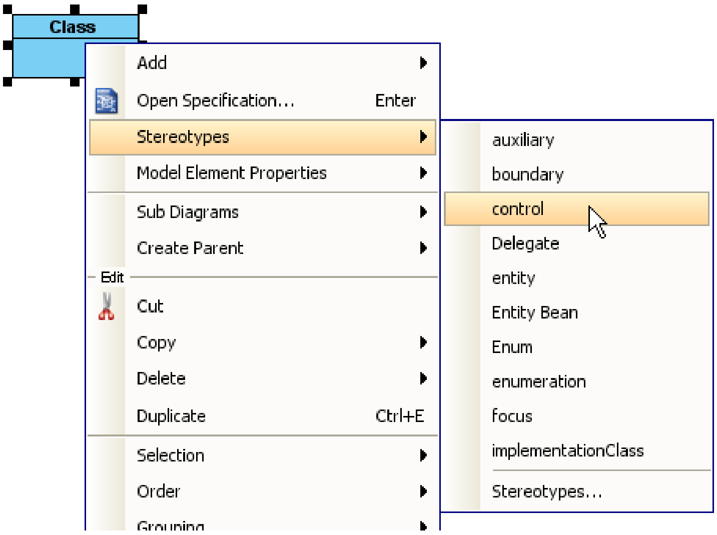


Figure 1-6 Apply stereotype from popup

Showing Stereotyped Model Element in Diagram Palette

1. Open a dialog.
2. Click ¥ button on diagram palette.

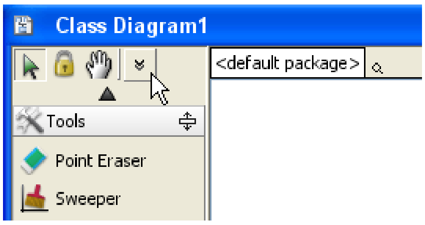


Figure 1-10 Diagram palette popup button

3. Select **Configure Buttons...** on the popup.

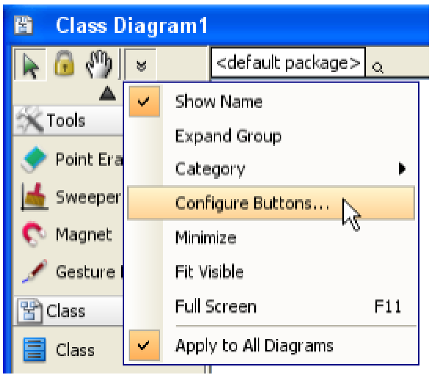


Figure 1-11 Configure buttons

Adding Dependent Project

Adding dependent project allows you to reference other project file's model elements.

1. Create a new project.
2. Open and right click on **Model Explorer**, select **Manage Depending Project...** from the popup menu.

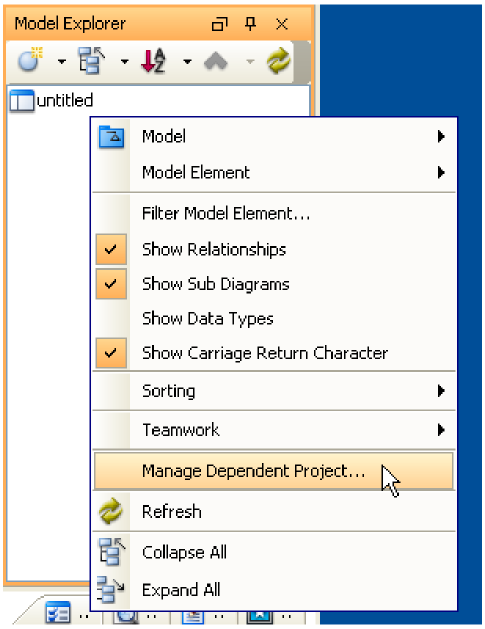
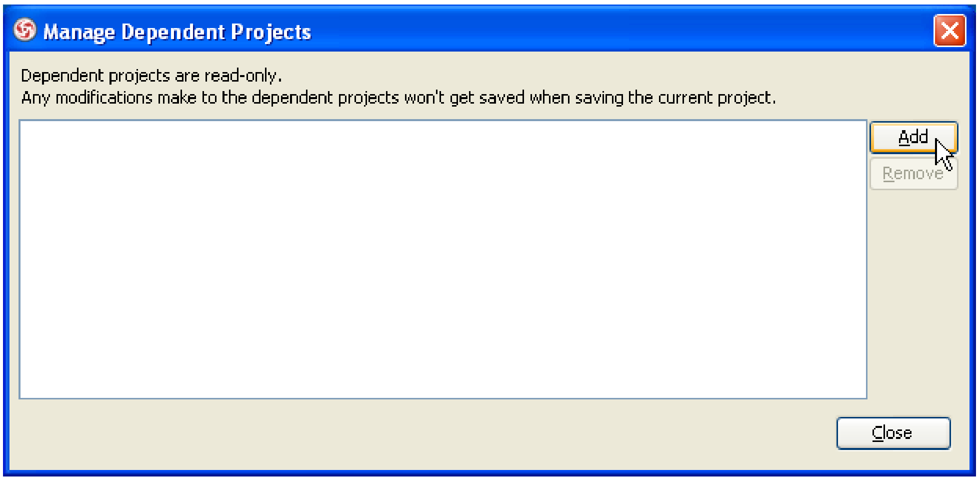


Figure 3-1 *Manage dependent project*

3. Click **Add** button on **Manage Depending Projects** dialog.



Dependent projects are read-only.
Any modifications make to the dependent projects won't get saved when saving the current project.

Figure 3-2 *Manage dependent projects dialog*

Using Dependent Project Model Elements

1. Create a new class diagram.
2. Open **Model Explorer**.
3. If you have added dependent projects, you see a combo box on top of **Model Explorer**. This combo box allow you to switch **Model Explorer** between current project and dependent projects.

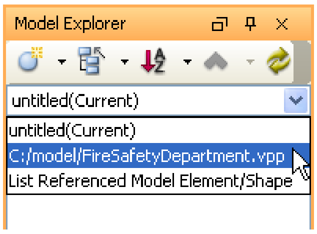

Model Explorer
untitled(Current)
untitled(Current)
C:/model/FireSafetyDepartment.vpp
List Referenced Model Element/Shape

***Figure 3-4** Select dependent project*

4. Select the dependent project from the combo box.
5. The **Model Explorer** will show the model elements of the dependent project as if you're working in the dependent project. Expand the tree and select some classes, drag and drop to the class diagram.

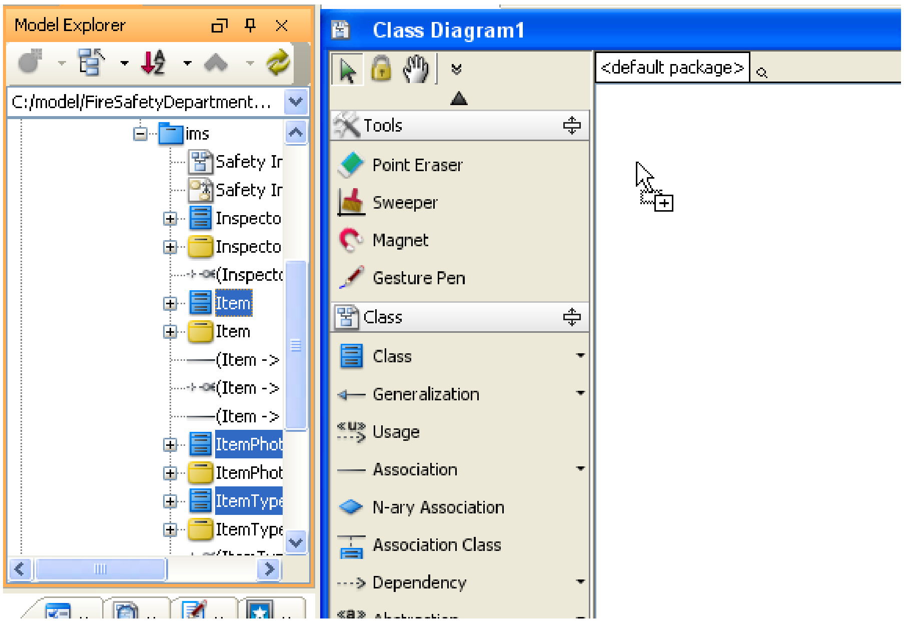


***Figure 3-5** Drag and drop from model element to diagram*

6. The dependent project's model elements were created on the diagram.

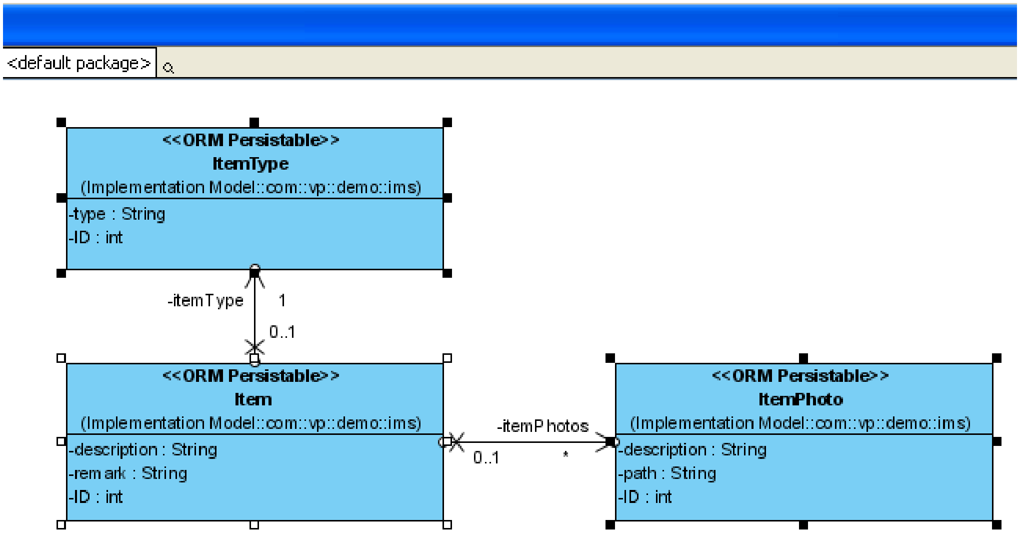


***Figure 3-6** Dependent model element on diagram*

Importing Projects to VP Teamwork Server

1. Start **VP-UML**.
2. Select **Tools > Teamwork > Open Teamwork Client...** from the main menu, or **Open Teamwork Client** icon from toolbar to open teamwork client.

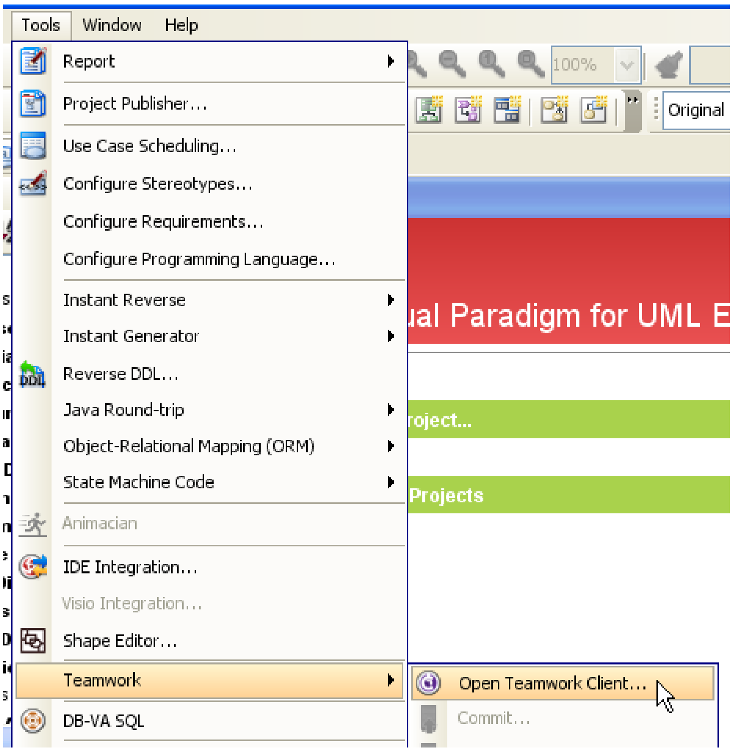
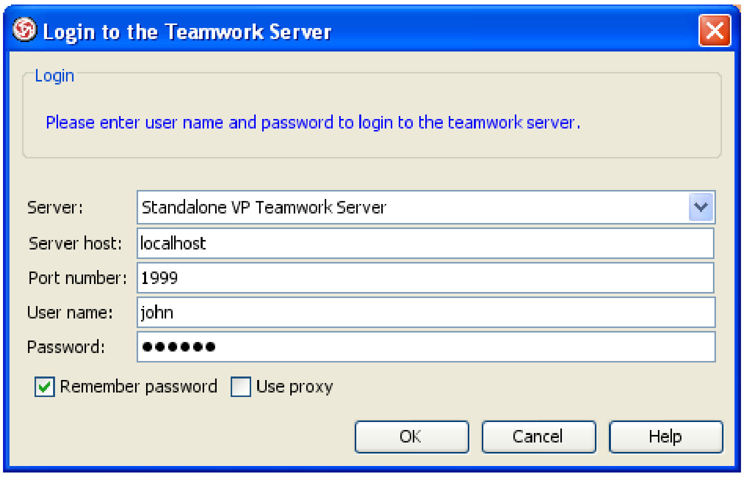


***Figure 1-1** Open Teamwork Client from main menu*

3. Fill in the server and user information to login teamwork server.



Login to the Teamwork Server

Login

Please enter user name and password to login to the teamwork server.

Server: Standalone VP Teamwork Server

Server host: localhost

Port number: 1999

User name: john

Password: ●●●●●●

☑ Remember password ☐ Use proxy

***Figure 1-2** Login teamwork server*

4. Select **Project > Import Project to Repository** from the menu.

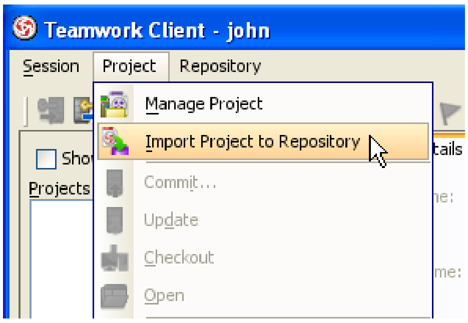


Figure 1-3 *Import project from menu*

Or click the **Import Project to Repository** button on toolbar.

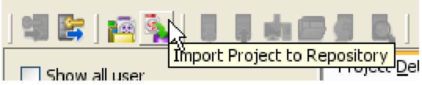
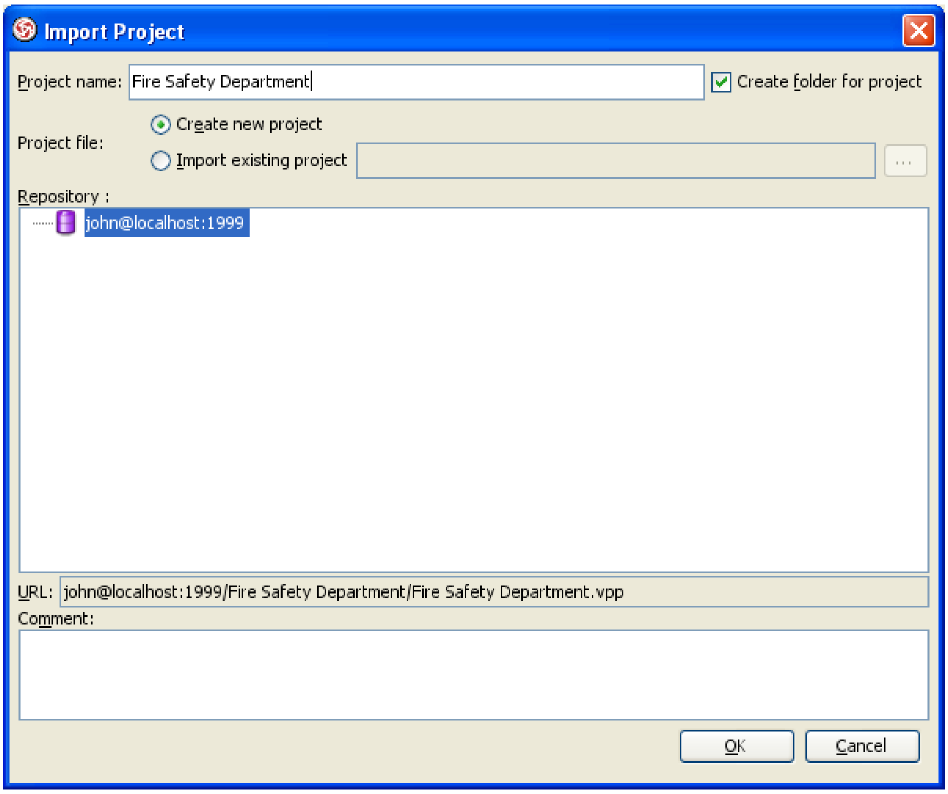


Figure 1-4 *Import project from toolbar*

5. In the **Import Project** dialog, fill in the **Project name**.



Project name: Fire Safety Department ☑ Create folder for project

Project file: ● Create new project ○ Import existing project

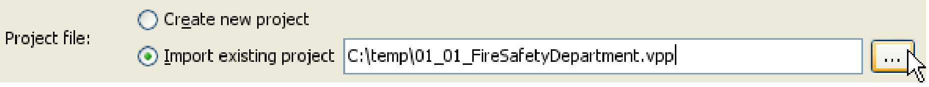
Repository : john@localhost:1999

URL: john@localhost:1999/Fire Safety Department/Fire Safety Department.vpp

Comment:

Figure 1-5 *Fill project name in Import Project dialog*

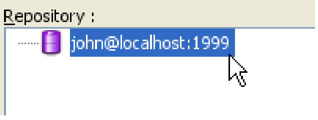
6. Choose **Create new project** or **Import existing project** from **Project file**. **Create new project** will import a blank project into teamwork server, **Import existing project** will import an existing VP-UML Project file (*.vpp) into teamwork server as first revision.



Project file: ○ Create new project ● Import existing project C:\temp\01_01_FireSafetyDepartment.vpp

Figure 1-6 *Specify project file*

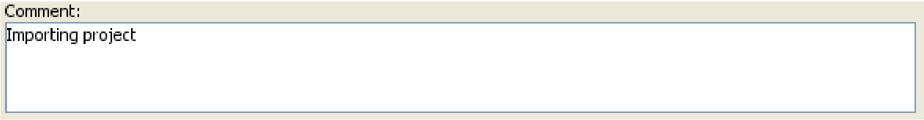
7. Select the repository for import the project.



Repository : john@localhost:1999

Figure 1-7 *Select repository*

8. Fill in the comment.



Comment:

Importing project

Figure 1-8 Input comment

9. Click **OK** button to import project.

NOTE: Please make sure the user login to **Teamwork Client** has **Create Project** permission.

Checkout Project from VP Teamwork Server

Checkout and open VP Teamwork Server Projects

1. Open **Teamwork Client** dialog.
2. Select **Project > Manage Project** from the menu.

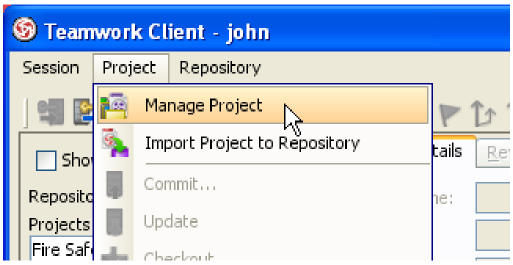


Figure 1-9 *Manage project from main menu*

Or click **Manage Project** icon on the toolbar.

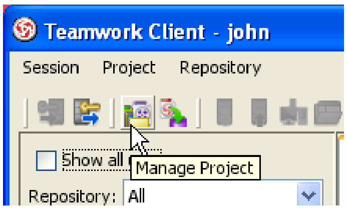


Figure 1-10 *Manage project from toolbar*

3. In **Manage Project** dialog, expand the repository node and select the project to checkout, click **>** button to add project. Click **OK** button to close the dialog.

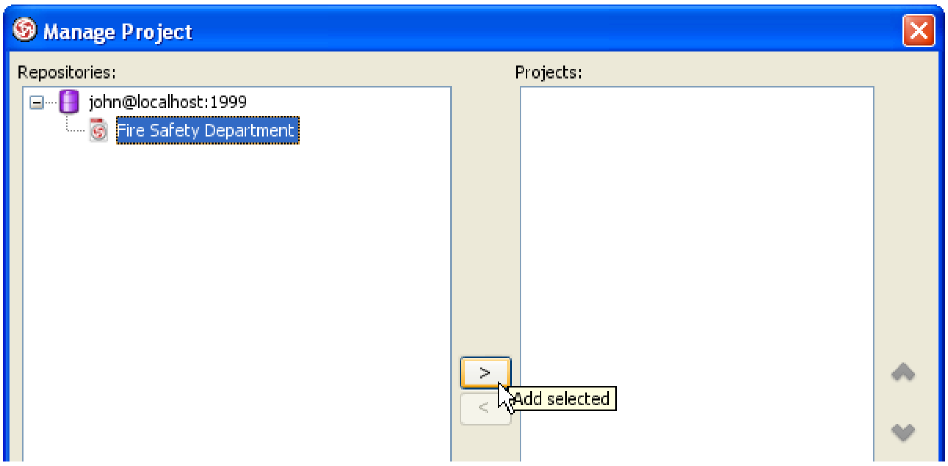


Figure 1-11 *Manage projects*

4. Select the project in project list.

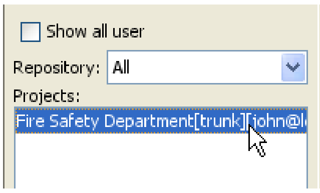


Figure 1-12 *Select project from the list*

5. Click **Open Project** button to checkout and open the project.

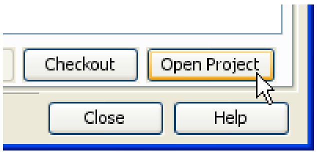
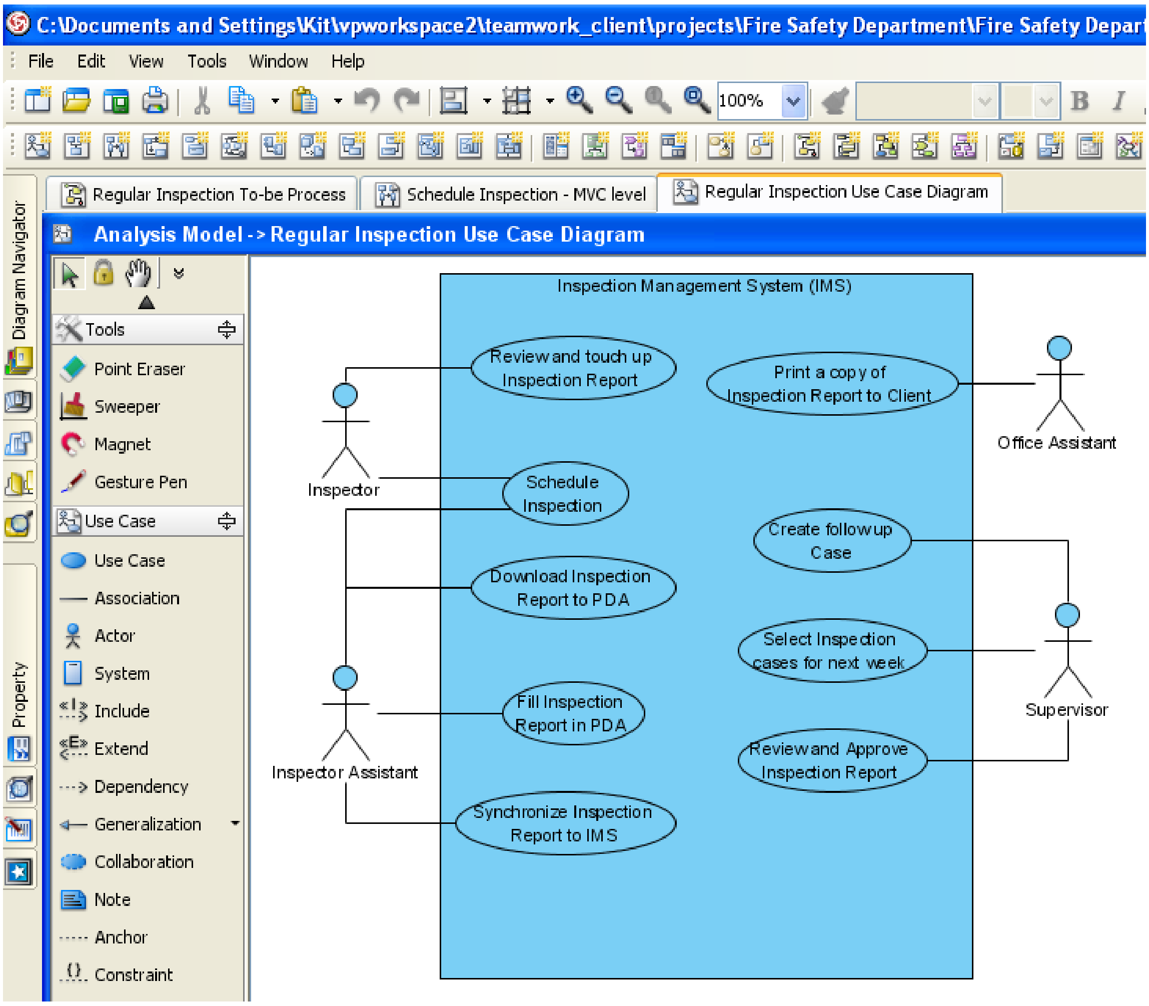


Figure 1-13 *Open project*

6. The project opened in **VP-UML**.



C:\Documents and Settings\Kit\vpworkspace2\teamwork_client\projects\Fire Safety Department\Fire Safety Depart
File
Edit
View
Tools
Window
Help
100%
Diagram Navigator
Property
Regular Inspection To-be Process
Schedule Inspection - MVC level
Regular Inspection Use Case Diagram
Analysis Model -> Regular Inspection Use Case Diagram
Tools
Point Eraser
Sweeper
Magnet
Gesture Pen
Use Case
Use Case
Association
Actor
System
Include
Extend
Dependency
Generalization
Collaboration
Note
Anchor
Constraint
Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Review and Approve Inspection Report
Inspector Assistant
Synchronize Inspection Report to IMS

***Figure 1-14** Project opened*

Reset Password

1. Open **Teamwork Client** dialog.

2. Select **Project > Reset Password...** from the menu.

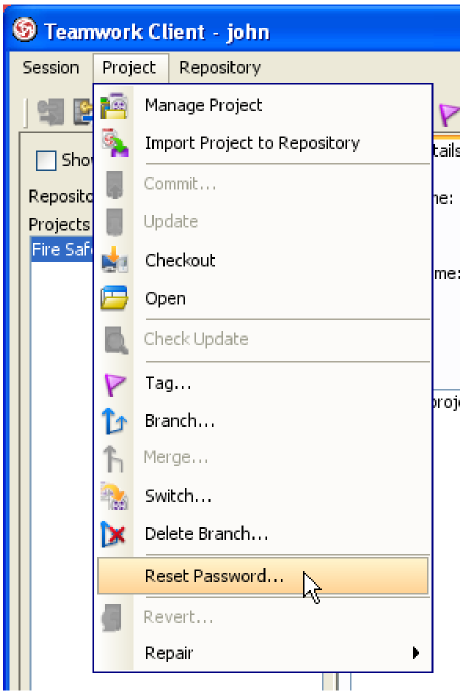
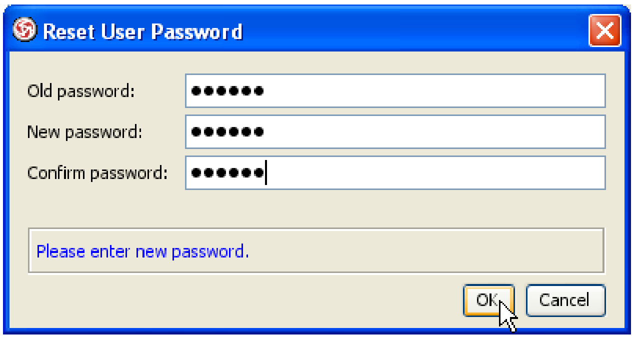


Figure 1-15 *Reset password from menu*

3. Fill in the old password, new password, and fill the new password in confirm password again. Click **OK** button to confirm.



Old password: ●●●●●●

New password: ●●●●●●

Confirm password: ●●●●●●

Please enter new password.

Figure 1-16 *Reset password dialog*

Committing local modification to VP Teamwork Server

1. Modify the project.

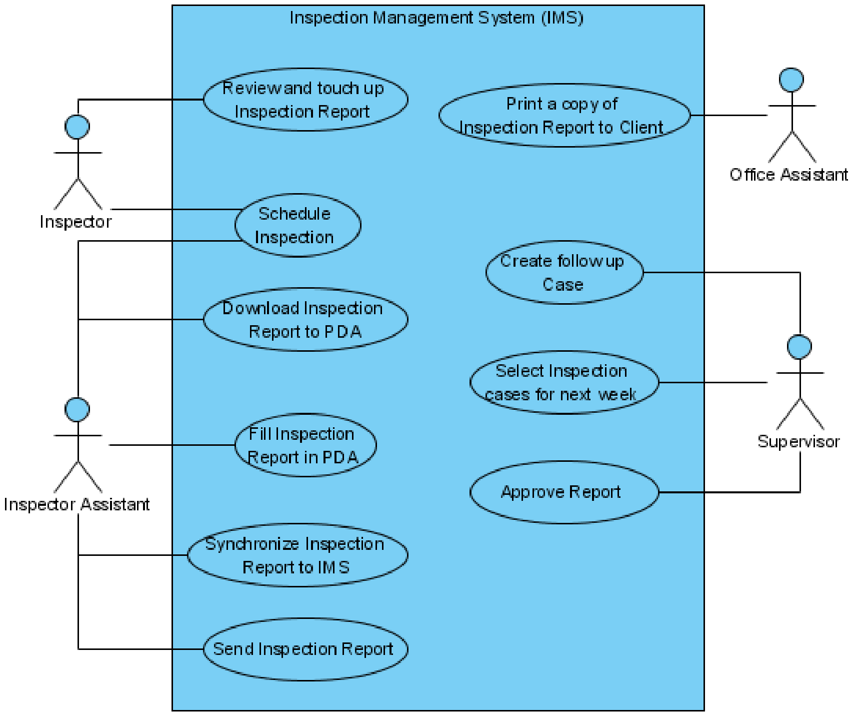

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Approve Report
Synchronize Inspection Report to IMS
Send Inspection Report

__Figure 1-17__ Modified project

2. Show **Teamwork Toolbar** by right click on the Toolbar, select **Teamwork** if not selected already.

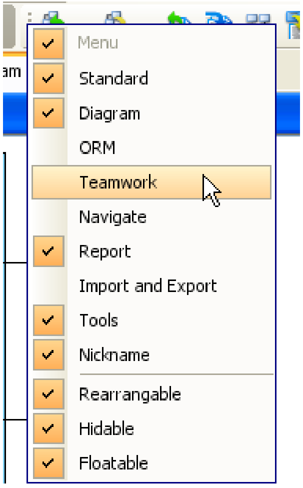


__Figure 1-18__ Show teamwork toolbar

3. Click the **Commit** button on **Teamwork Toolbar**.

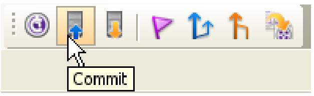
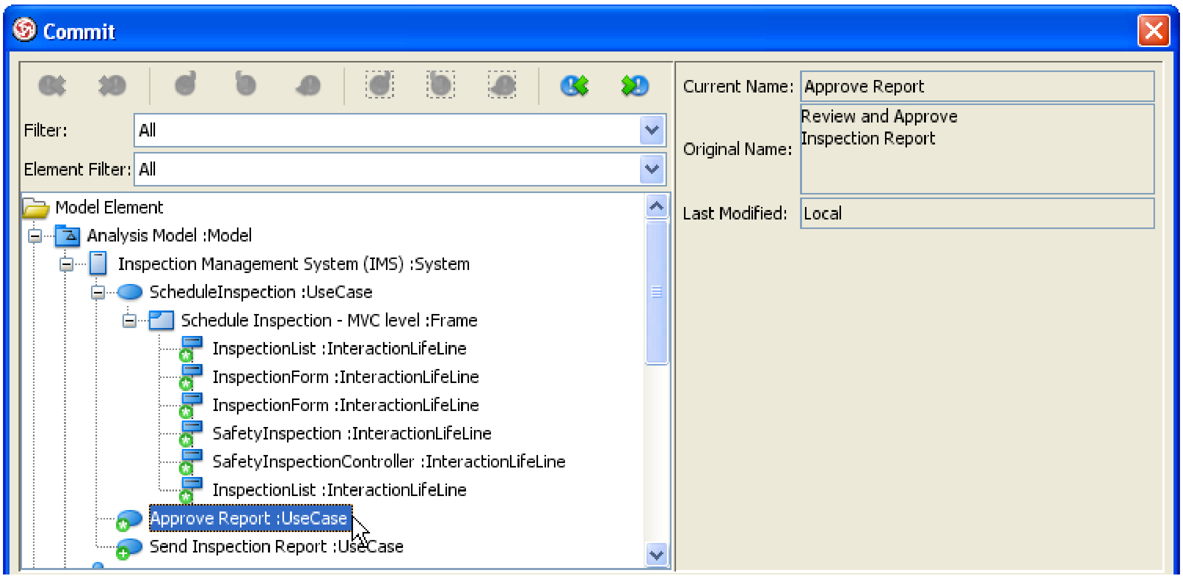


__Figure 1-19__ Commit project

4. In **Commit** dialog, you can review the changes for commit. On the left of dialog, you can see a list of changes shapes and model elements, click on it to view the detail changes on the right.



Commit

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionList :InteractionLifeLine
          - InspectionForm :InteractionLifeLine
          - InspectionForm :InteractionLifeLine
          - SafetyInspection :InteractionLifeLine
          - SafetyInspectionController :InteractionLifeLine
          - InspectionList :InteractionLifeLine
      - Approve Report :UseCase
      - Send Inspection Report :UseCase

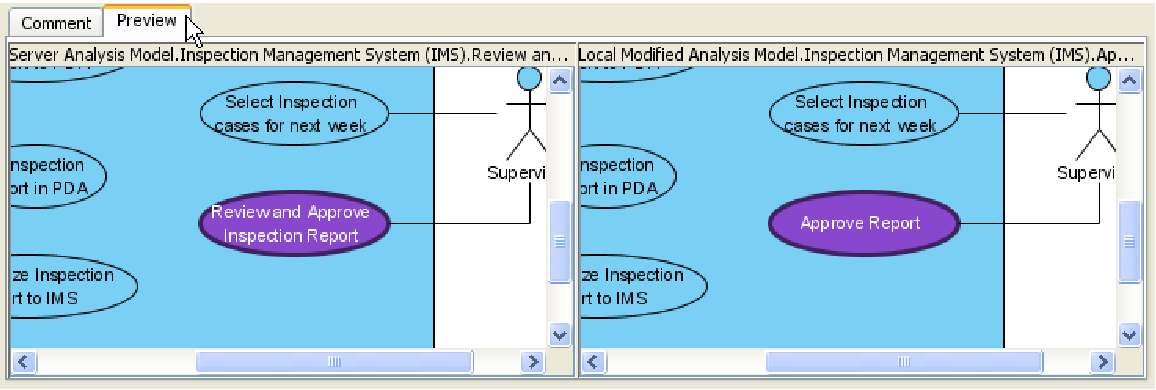
Current Name: Approve Report

Original Name: Review and Approve Inspection Report

Last Modified: Local

Figure 1-20 Review commit changes

5. On the bottom of dialog, click the **Preview** tab to visually preview the changes in diagram. The selected shape is highlighed in purple.



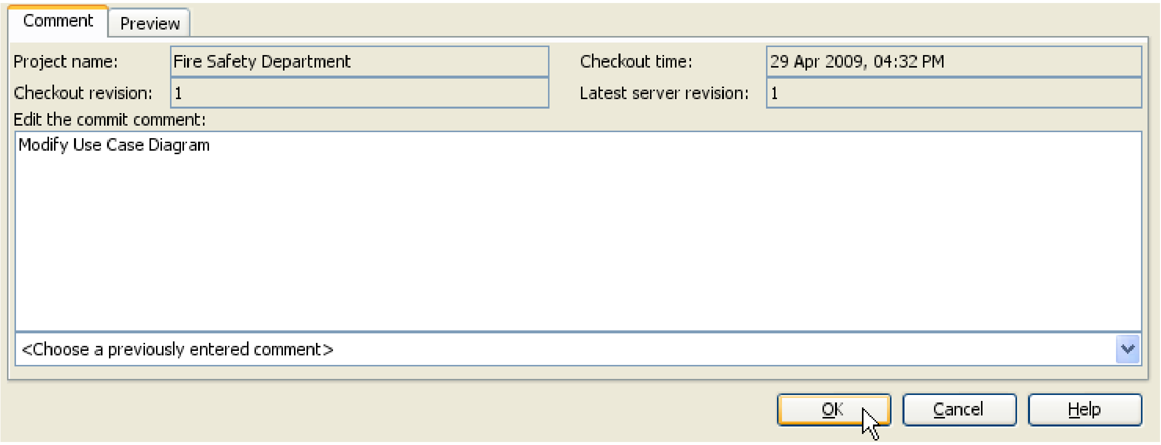
Comment | Preview

Server Analysis Model.Inspection Management System (IMS).Review an...

Local Modified Analysis Model.Inspection Management System (IMS).Ap...

Figure 1-21 Preview commit changes

6. After review the changes, click the **Comment** tab and input the comment for commit. Click **OK** button to start commit.



Comment | Preview

Project name: Fire Safety Department

Checkout time: 29 Apr 2009, 04:32 PM

Checkout revision: 1

Latest server revision: 1

Edit the commit comment:

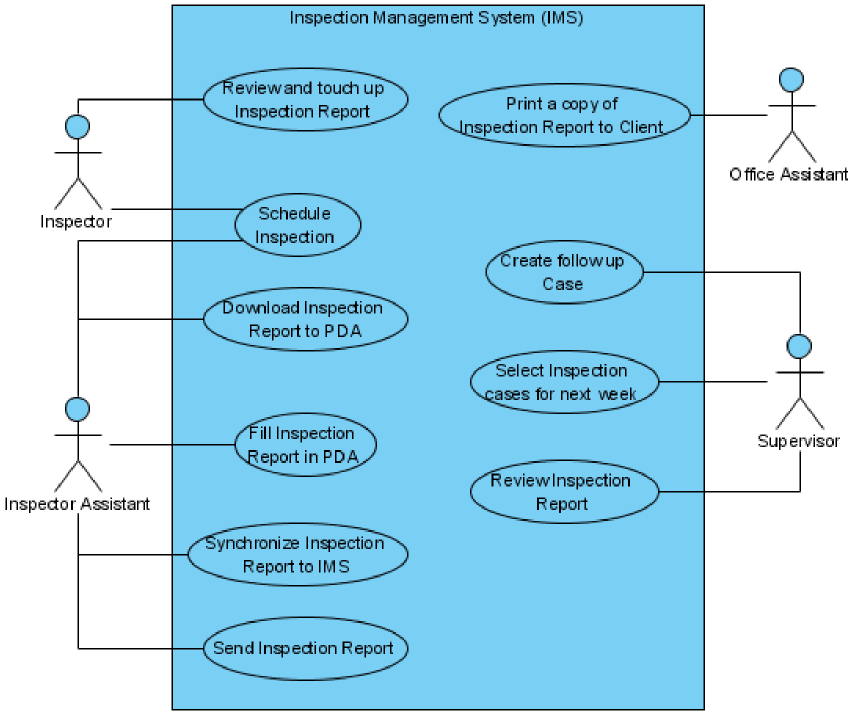
Modify Use Case Diagram

<Choose a previously entered comment>

Figure 1-22 Input commit comment

Resolving conflicts

1. Modify the project.

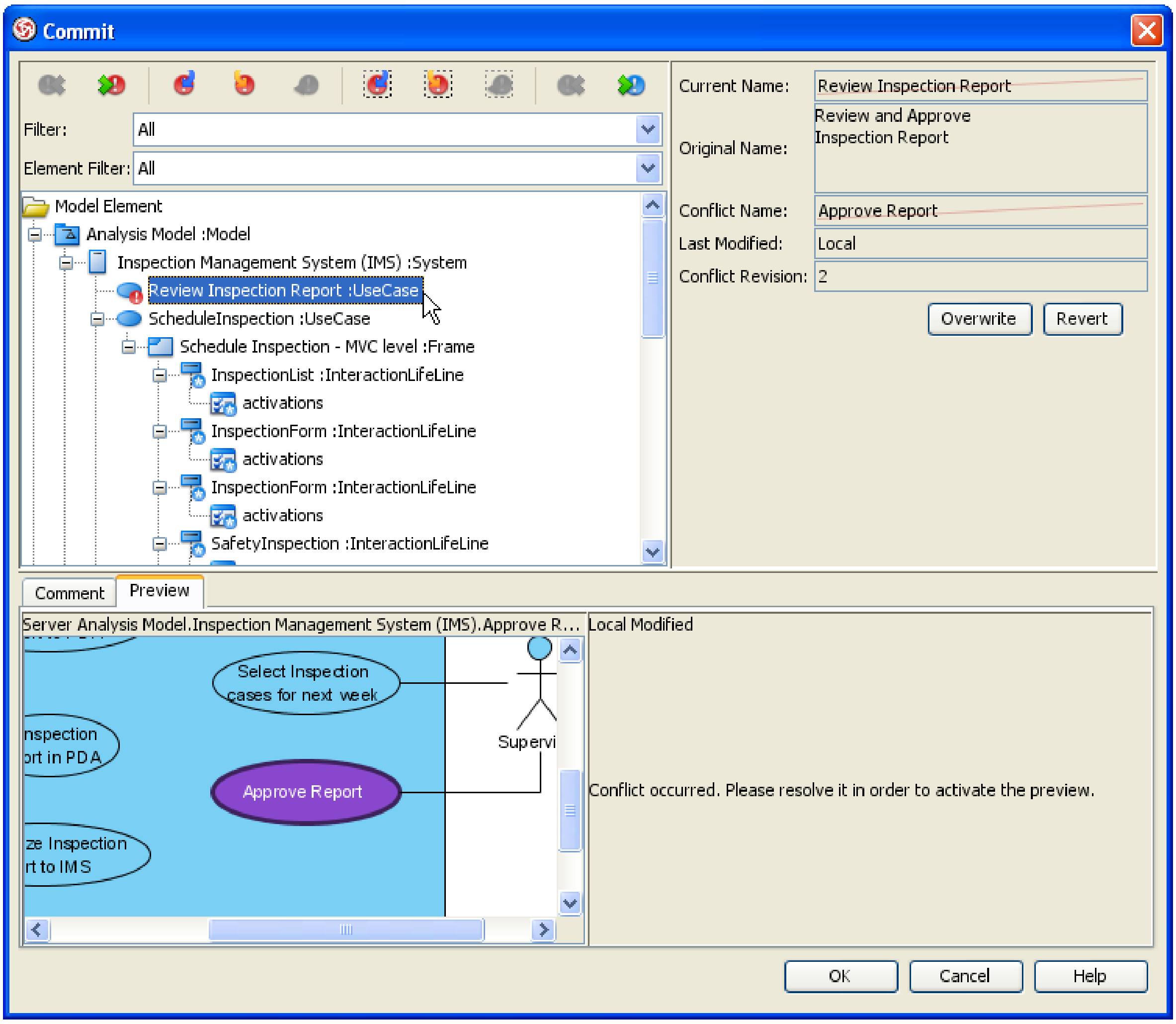


Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Review Inspection Report
Inspector Assistant
Synchronize Inspection Report to IMS
Send Inspection Report

__Figure 1-23__ Modified project

2. Click **Commit** button on the **Teamwork Toolbar**.

3. In **Commit** dialog, you may found some shapes or model elements show with red icon. This indicate there is conflict when commit, it is caused by someone modified the same content and commit after you checkout. You can review the current value, original value (the value when you checkout), and conflict value (the value changed by other users). And dialog preview is disabled until you resolve the conflict.



Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Review Inspection Report :UseCase
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionList :InteractionLifeLine
            - activations
          - InspectionForm :InteractionLifeLine
            - activations
          - InspectionForm :InteractionLifeLine
            - activations
          - SafetyInspection :InteractionLifeLine

Current Name: Review Inspection Report

Original Name: Review and Approve Inspection Report

Conflict Name: Approve Report

Last Modified: Local

Conflict Revision: 2

Server Analysis Model.Inspection Management System (IMS).Approve R...

Local Modified

Conflict occurred. Please resolve it in order to activate the preview.

Figure 1-24 Commit with conflicts

4. Select the conflict element, click **Overwrite selected conflicts** button on the toolbar to overwrite other user's change.

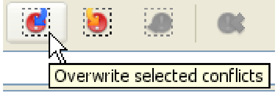


Figure 1-25 Overwrite selected conflicts

Or click **Revert selected conflicts** to revert your own changes.

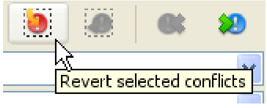


Figure 1-26 Revert selected conflicts

You can also click **Overwrite all conflicts** or **Revert all conflicts** to overwrite or revert all conflicts at once.

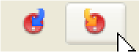
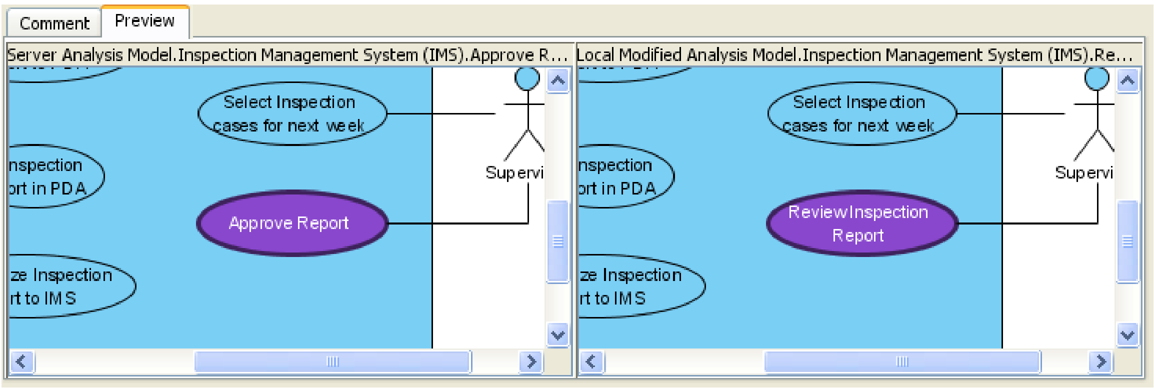


Figure 1-27 Overwrite/Revert all conflicts

5. After resolve conflict, the preview will be enabled to visualize the final result.



Comment
Preview
Server Analysis Model.Inspection Management System (IMS).Approve R...
Local Modified Analysis Model.Inspection Management System (IMS).Re...
Select Inspection cases for next week
nspection ort in PDA
Approve Report
ze Inspection rt to IMS
Supervi
Select Inspection cases for next week
nspection ort in PDA
ReviewInspection Report
ze Inspection rt to IMS
Supervi

Figure 1-28 *Preview resolved conflict*

6. You can change your mind by click **Reset selected conflicts** or **Reset all conflicts** button. Then overwrite or revert the conflict again.

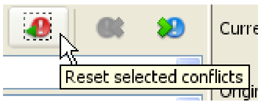


Figure 1-29 *Reset conflicts*

7. After all conflicts was resolved, you can now input comment and click **OK** button to commit.

Updating latest revision from VP Teamwork Server

Updating modification from VP Teamwork Server

1. Click **Update** button on **Teamwork Toolbar**.

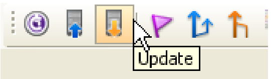


Figure 1-30 *Update from toolbar*

2. Similar to commit, you can review the change and preview the diagram in **Update** dialog.

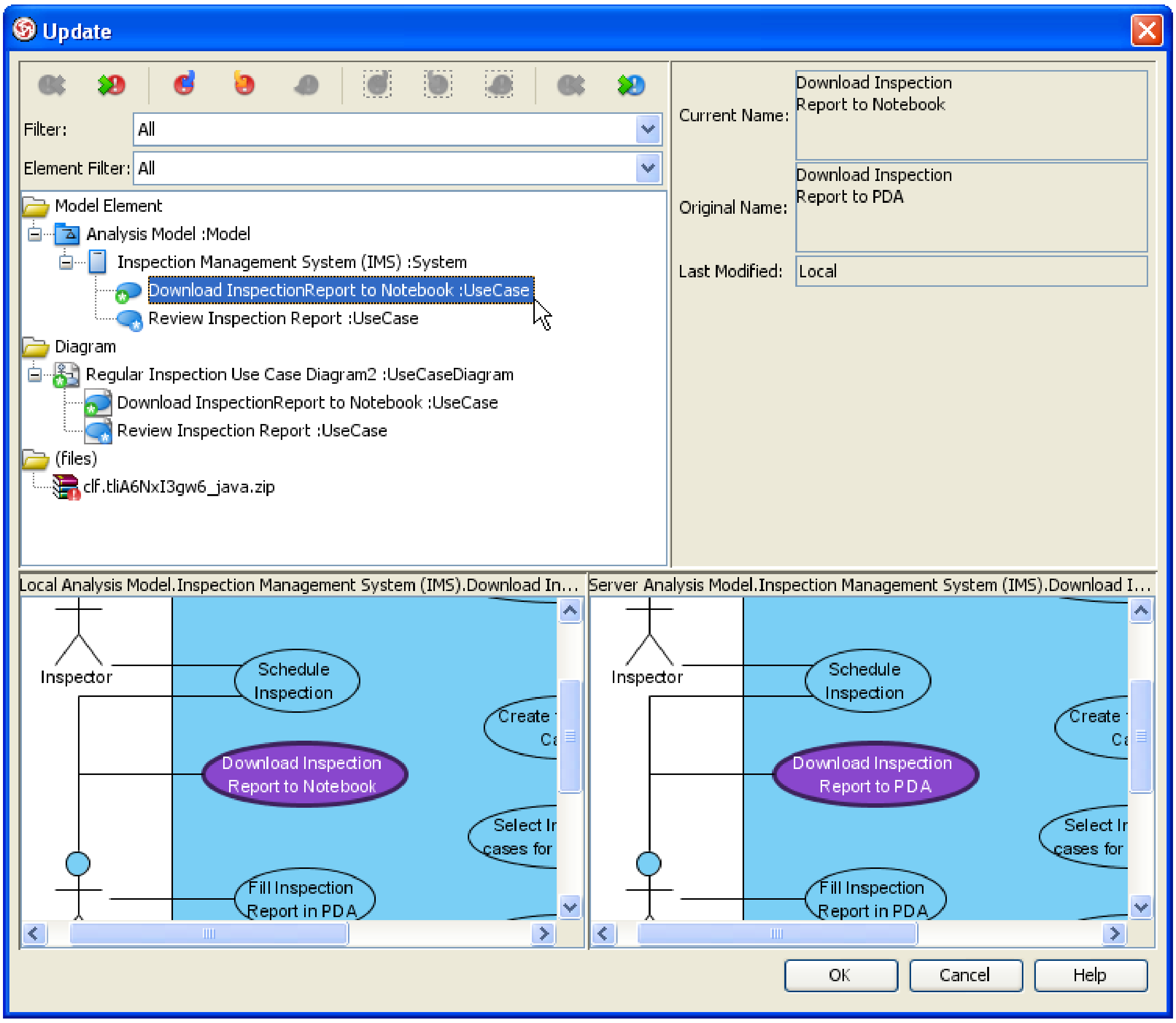

Update
Filter: All
Element Filter: All
Model Element
Analysis Model :Model
Inspection Management System (IMS) :System
Download InspectionReport to Notebook :UseCase
Review Inspection Report :UseCase
Diagram
Regular Inspection Use Case Diagram2 :UseCaseDiagram
Download InspectionReport to Notebook :UseCase
Review Inspection Report :UseCase
(files)
clf.tliA6NxI3gw6_java.zip
Current Name: Download Inspection Report to Notebook
Original Name: Download Inspection Report to PDA
Last Modified: Local
Local Analysis Model.Inspection Management System (IMS).Download In...
Server Analysis Model.Inspection Management System (IMS).Download I...
Inspector
Schedule Inspection
Create
Download Inspection Report to Notebook
Download Inspection Report to PDA
Select In cases for
Fill Inspection Report in PDA
OK
Cancel
Help

Figure 1-31 *Update dialog*

3. If there are conflicts, resolve it similar to commit, but the changes will apply to local project instead of commit to server immediately.

4. Click **OK** button to update. **VP-UML** will open the updated project.

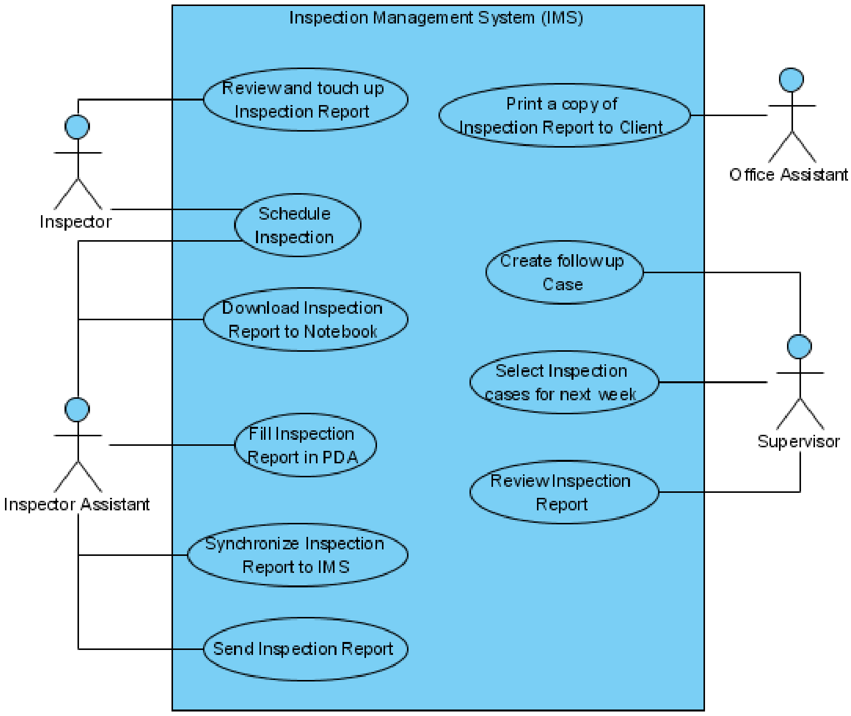

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to Notebook
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 1-32 Updated project

Checking status

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Check Update** from menu.

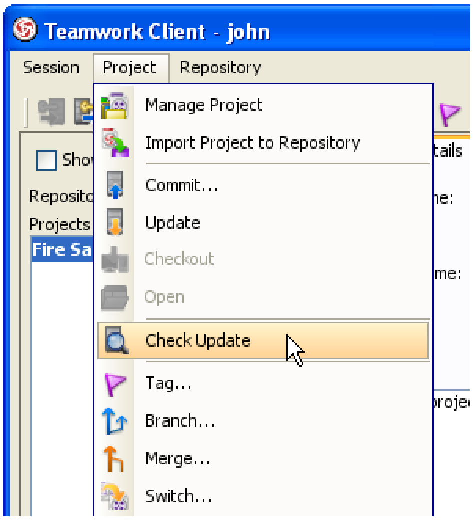


Figure 1-33 Update project from menu

Or click **Check Update** button from toolbar.

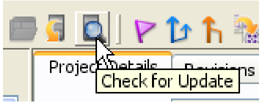


Figure 1-34 Update project from toolbar

Reverting Changes

Revert local modifications

1. Modify the project.

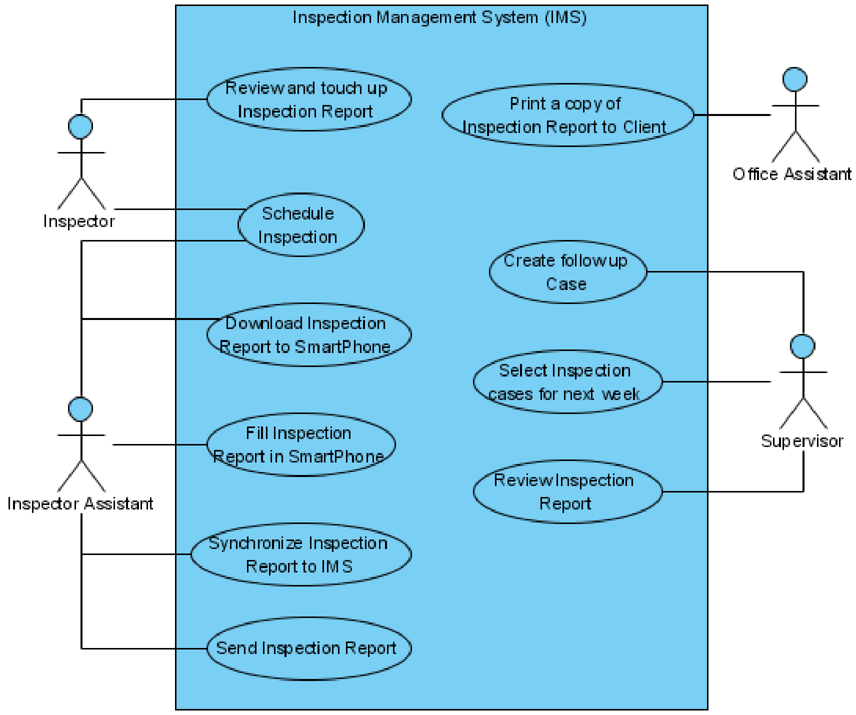

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to SmartPhone
Select Inspection cases for next week
Supervisor
Fill Inspection Report in SmartPhone
Review Inspection Report
Inspector Assistant
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 1-36 *Modify project*

2. Open **Teamwork Client** dialog.
3. Select the teamwork project in the list.

4. Select **Project > Revert...** from menu

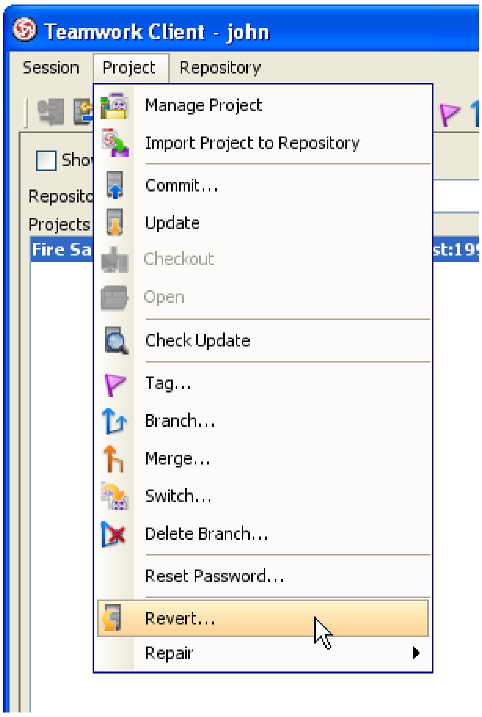


***Figure 1-37** Revert from menu*

Or click **Revert...** button from toolbar.

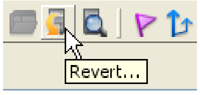


***Figure 1-38** Revert from toolbar*

5. Click **Yes** button from the confirmation dialog.
6. The reverted project opened in **VP-UML**.

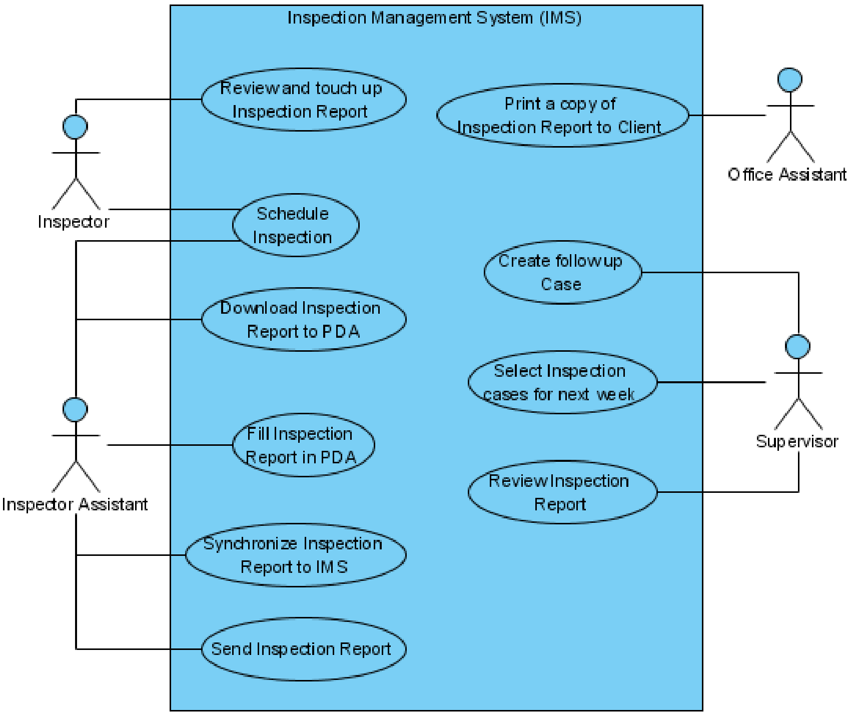

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Inspector Assistant
Fill Inspection Report in PDA
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

***Figure 1-39** Reverted project*

Revert server modifications

1. Open project.

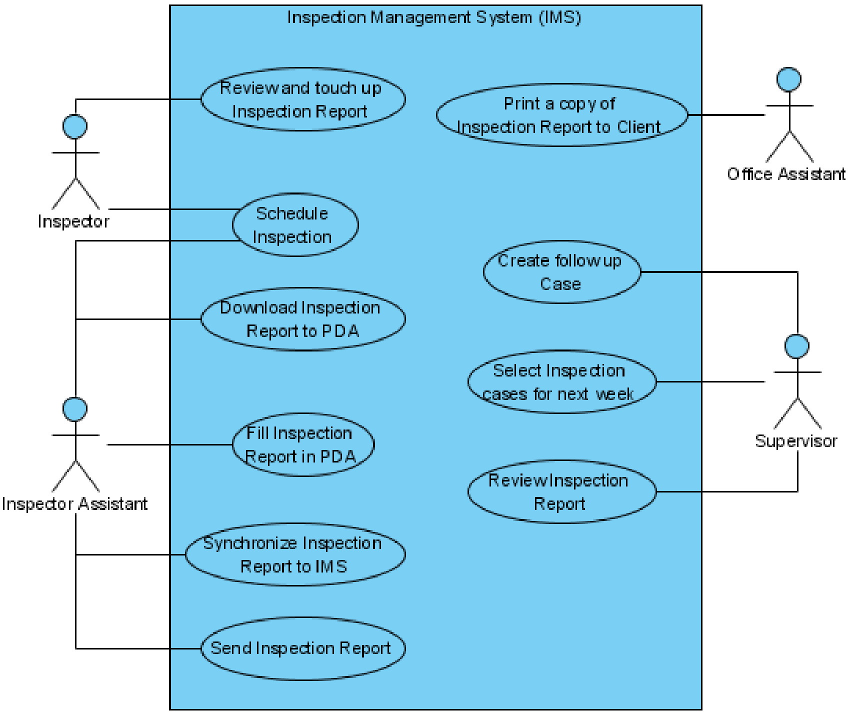
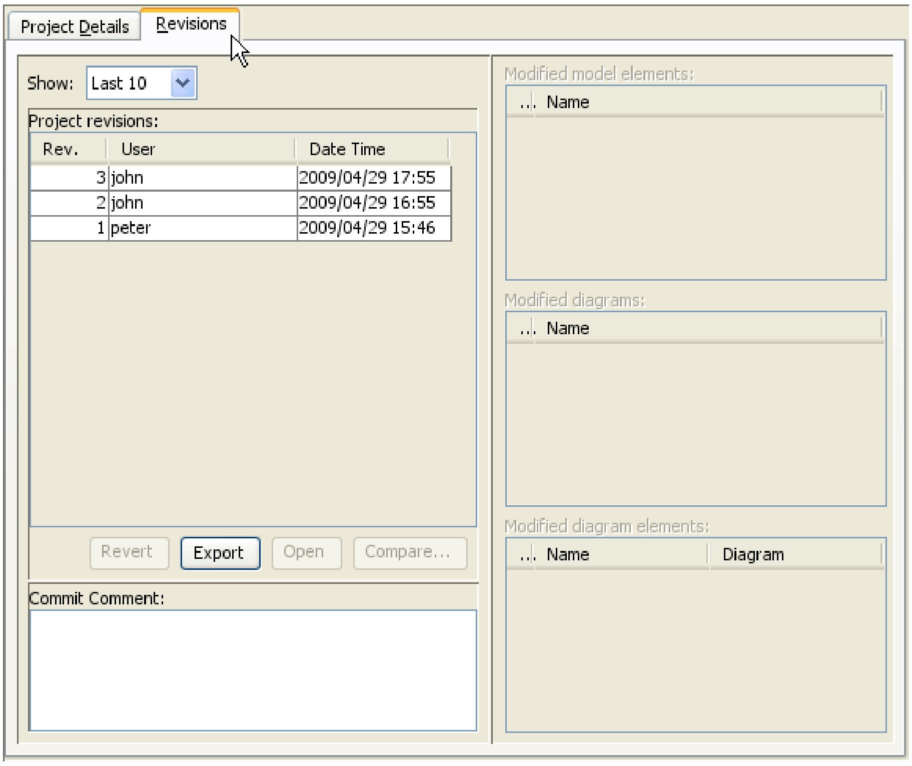

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Review Inspection Report
Inspector Assistant
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 1-40 *Open project*

2. Open **Teamwork Client** dialog.
3. Select the teamwork project in the list.
4. Click the **Revisions** tab on the right of the project list.



Show: Last 10

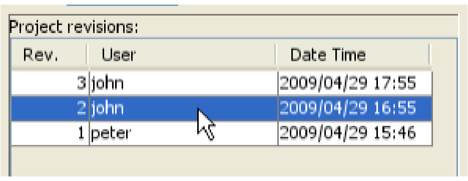
Project revisions:

Rev.	User	Date Time
3	john	2009/04/29 17:55
2	john	2009/04/29 16:55
1	peter	2009/04/29 15:46

Commit Comment:

Figure 1-41 *Revision tab*

5. Select the revision to revert.



Project revisions:

Rev.	User	Date Time
3	john	2009/04/29 17:55
2	john	2009/04/29 16:55
1	peter	2009/04/29 15:46

Figure 1-42 *Select revision*

Browsing change histories (old revisions)

Checkout old revisions

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select the revision to open.
5. Click Open button.

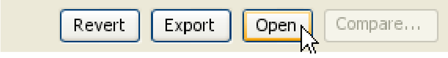


Figure 1-46 Open button

6. Selected revision opened in **VP-UML**.

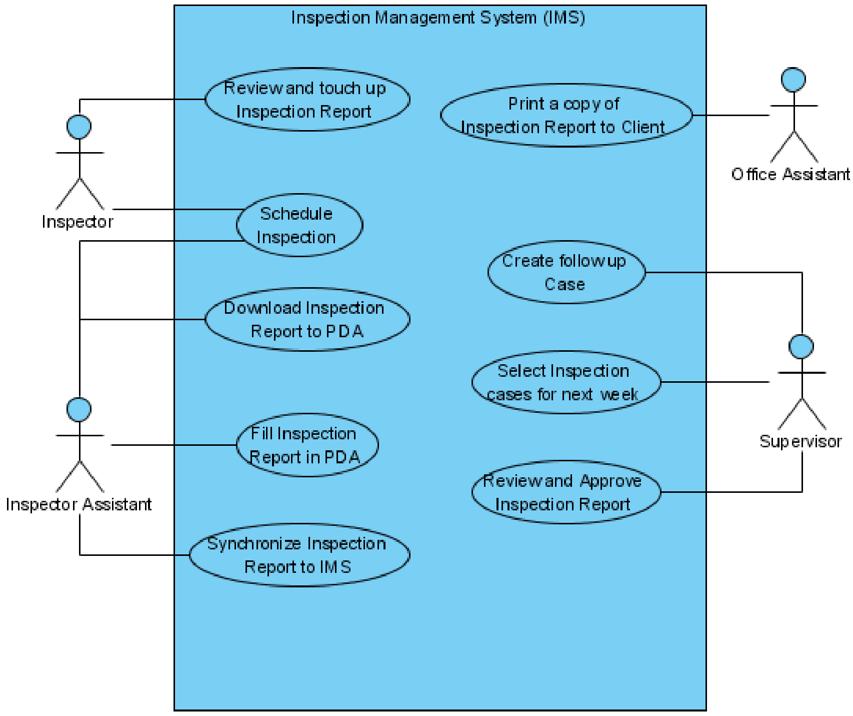
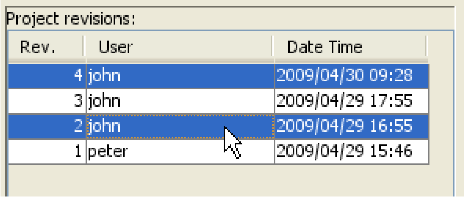

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS

Figure 1-47 Open selected revision

Showing differences between revisions visually

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select a revision for compare.
5. Select another revision by **Ctrl+Click**.



Project revisions:

Rev.	User	Date Time
4	john	2009/04/30 09:28
3	john	2009/04/29 17:55
2	john	2009/04/29 16:55
1	peter	2009/04/29 15:46

Figure 1-48 Select two revisions

6. Click the **Compare...** button.

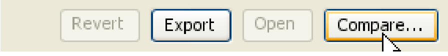
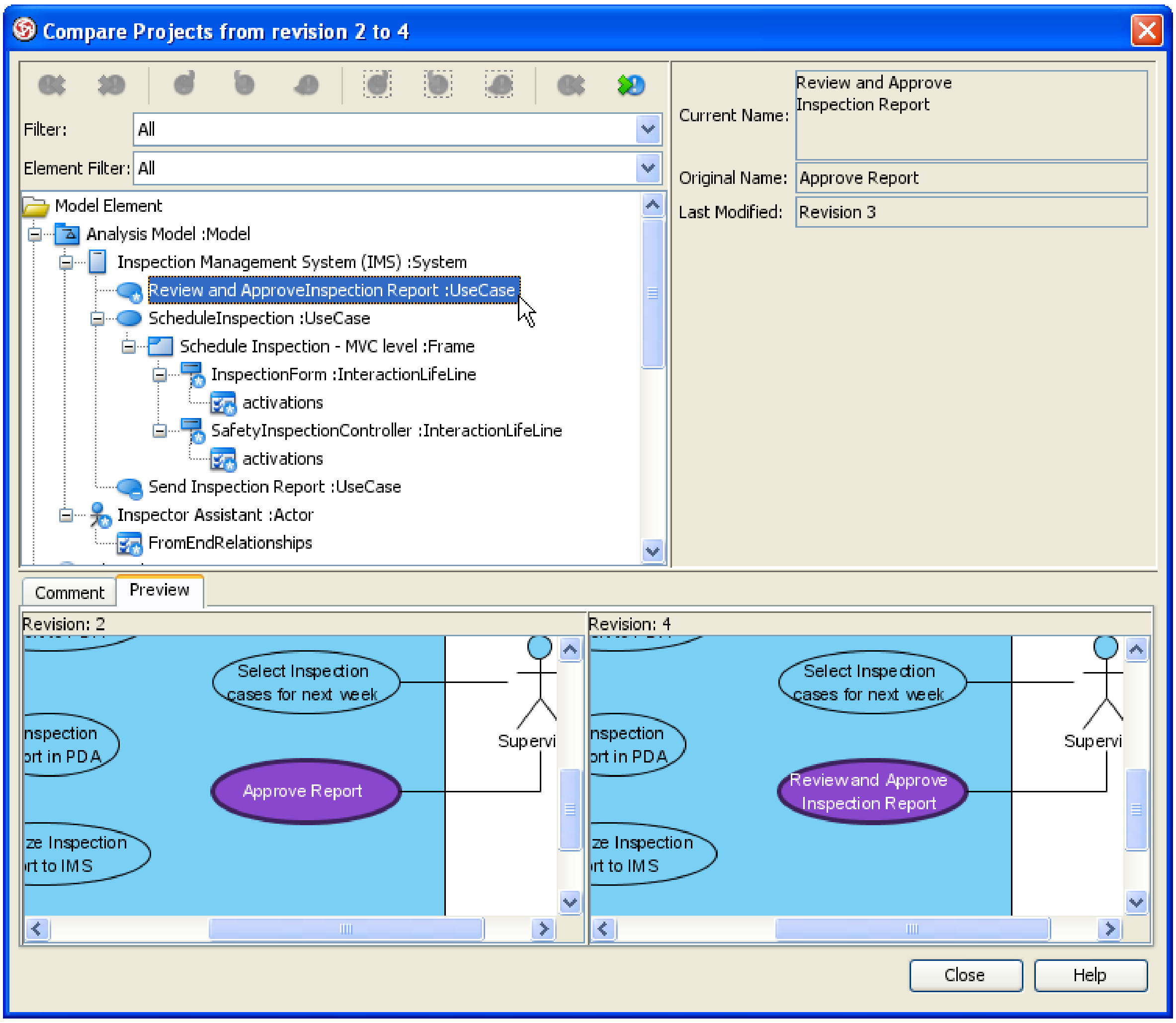


Figure 1-49 Compare button

7. Similar to **Commit** and **Update** dialog, the **Compare Projects** dialog show a list of differences between the selected revisions. You can also view the differences visually in diagram on the preview tab.



Compare Projects from revision 2 to 4

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Review and ApproveInspection Report :UseCase
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionForm :InteractionLifeLine
            - activations
          - SafetyInspectionController :InteractionLifeLine
            - activations
      - Send Inspection Report :UseCase
    - Inspector Assistant :Actor
      - FromEndRelationships

Current Name: Review and Approve Inspection Report

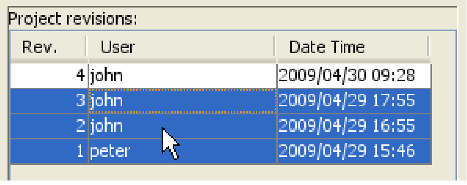
Original Name: Approve Report

Last Modified: Revision 3

Figure 1-50 *Compare differences*

Export multiple revisions to local

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select multiple revisions for export, by **Ctrl+Click** or **Shift+Click**.



Project revisions:

Rev.	User	Date Time
4	john	2009/04/30 09:28
3	john	2009/04/29 17:55
2	john	2009/04/29 16:55
1	peter	2009/04/29 15:46

Figure 1-51 *Select multiple revisions*

5. Click the **Export** button.

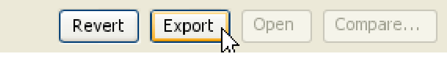
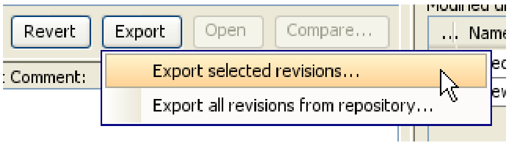


Figure 1-52 *Export buttons*

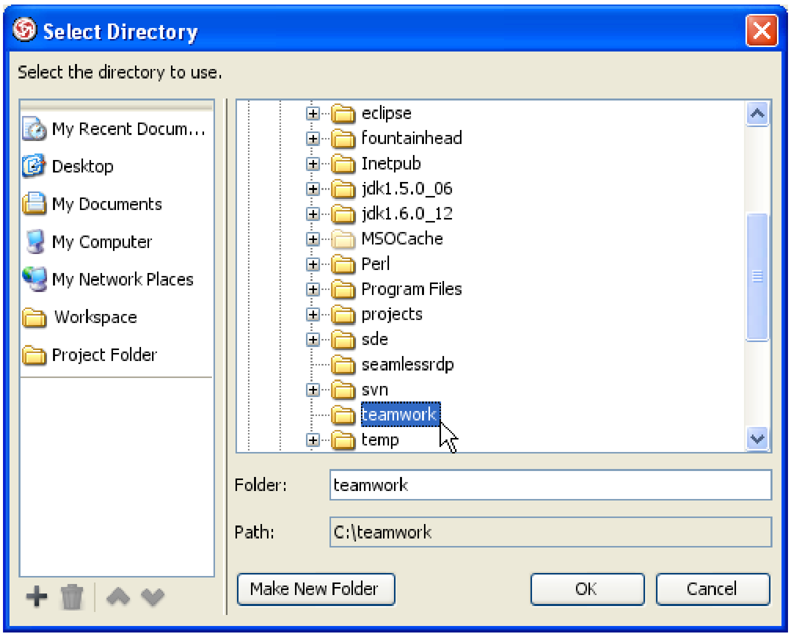
6. Select **Export selected revisions...** from the popup.



Revert
Export
Open
Compare...
Comment:
... Name
Export selected revisions...
Export all revisions from repository...

Figure 1-53 *Export selected revisions*

7. Select the directory to save the exported project revisions.



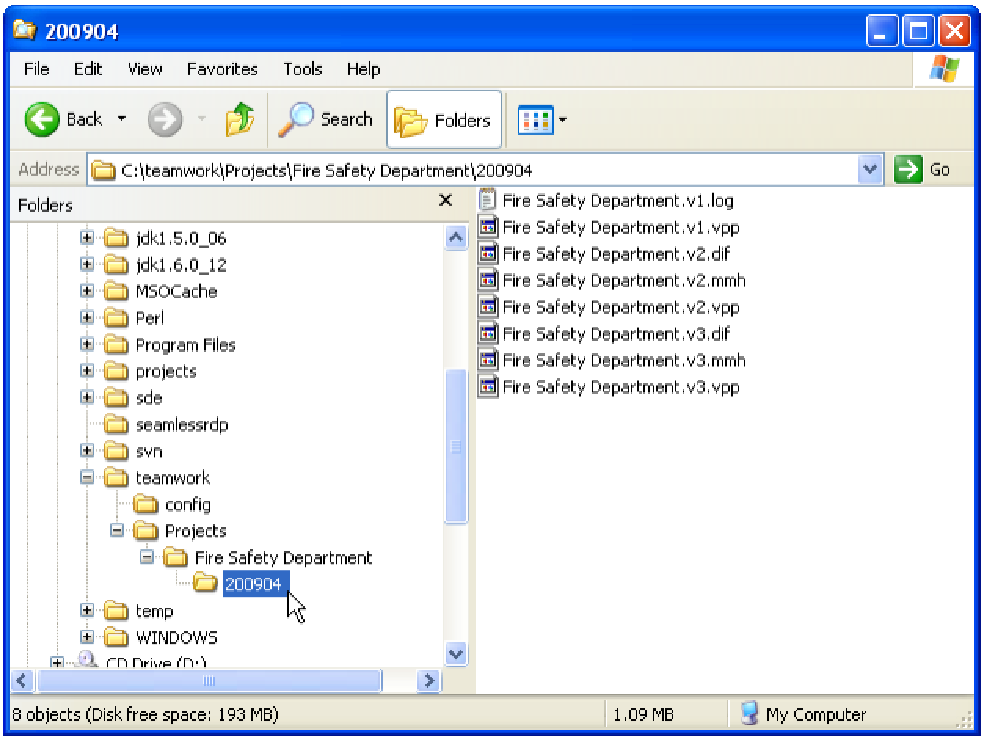
Select the directory to use.

Folder: teamwork

Path: C:\teamwork

Figure 1-54 *Select directory*

8. You'll found the selected revisions was exported to the select directory.



Address: C:\teamwork\Projects\Fire Safety Department\200904

Fire Safety Department.v1.log
Fire Safety Department.v1.vpp
Fire Safety Department.v2.dif
Fire Safety Department.v2.mmh
Fire Safety Department.v2.vpp
Fire Safety Department.v3.dif
Fire Safety Department.v3.mmh
Fire Safety Department.v3.vpp

8 objects (Disk free space: 193 MB) 1.09 MB

Figure 1-55 *Exported revisions*

Isolating last long modifications with Branches

Creating Branch

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Branch...** from menu

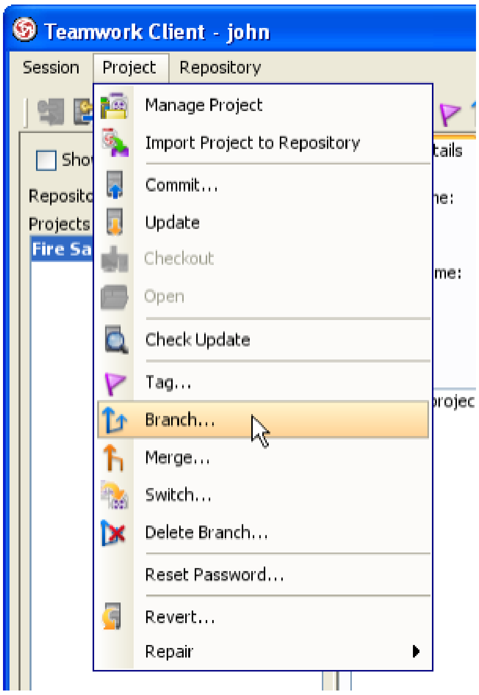


Figure 1-56 *Branch from menu*

Or click **Branch...** button from toolbar.

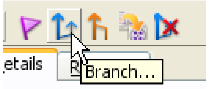
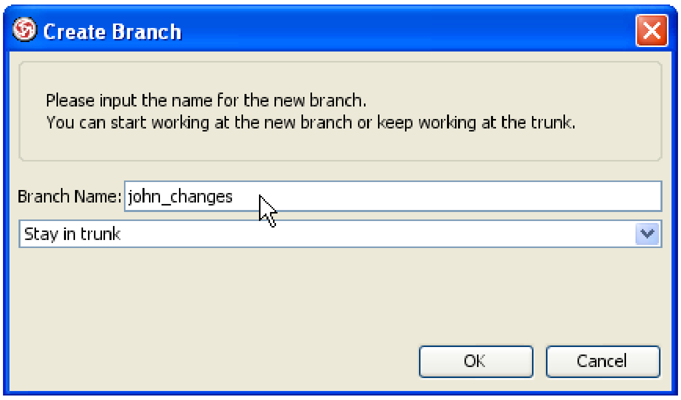


Figure 1-57 *Branch from toolbar*

4. Fill in the **Branch Name** in **Create Branch** dialog.



Create Branch

Please input the name for the new branch.
You can start working at the new branch or keep working at the trunk.

Branch Name: john_changes

Stay in trunk

Figure 1-58 *Create Branch dialog*

5. Select **Stay in trunk** in the list.

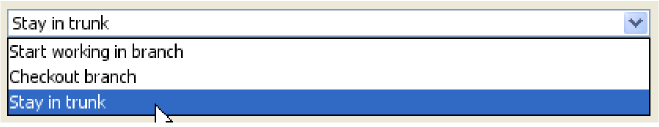


Figure 1-59 *Stay in trunk*

Switch local copy between Branches

1. Select **Project > Switch...** from menu

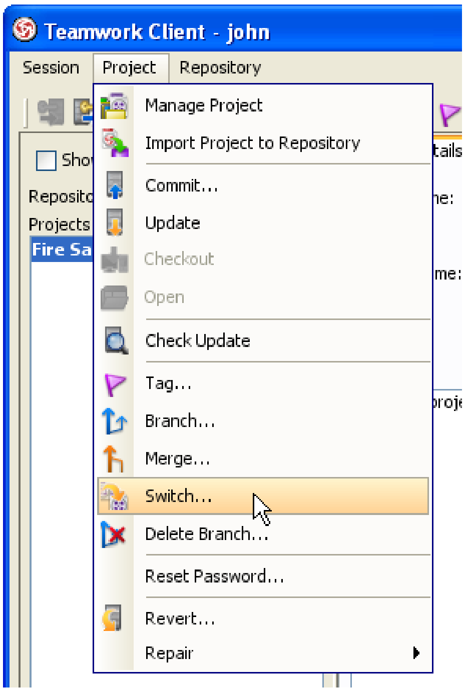


Figure 1-60 *Switch from menu*

Or click **Switch...** button from toolbar.

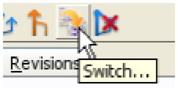
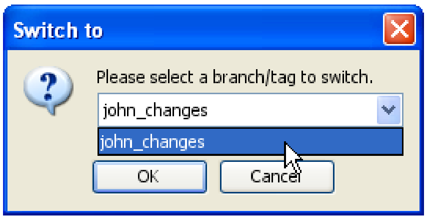


Figure 1-61 *Switch from toolbar*

2. Select the branch to switch.


Switch to

Please select a branch/tag to switch.

john_changes

Figure 1-62 *Select branch*

3. The branch is opened in **VP-UML**.

Merging from Trunk To Branch

1. Open, modify and commit trunk project.

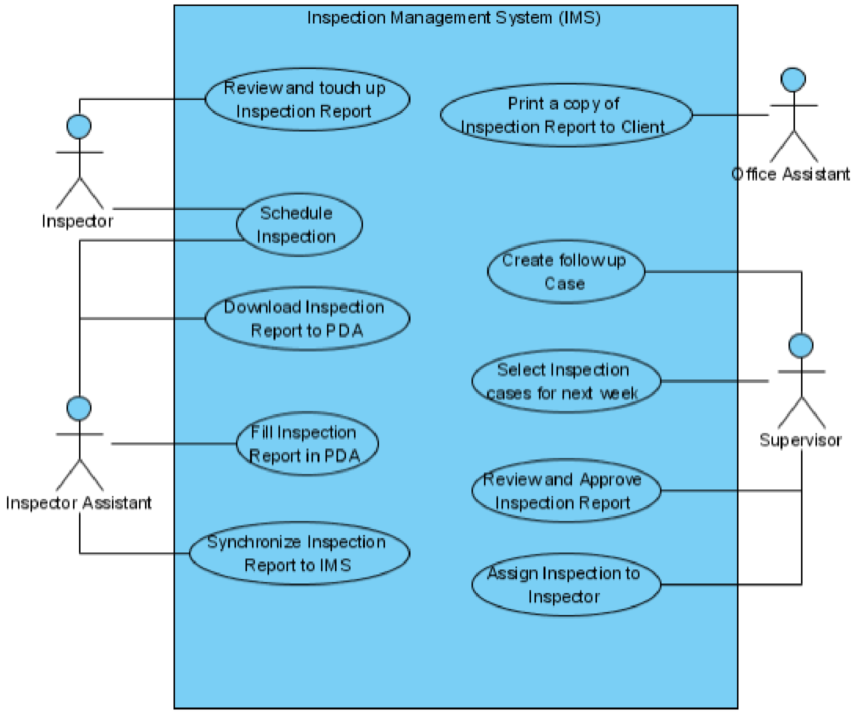

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 1-63 *Modified trunk project*

2. Open branch project.

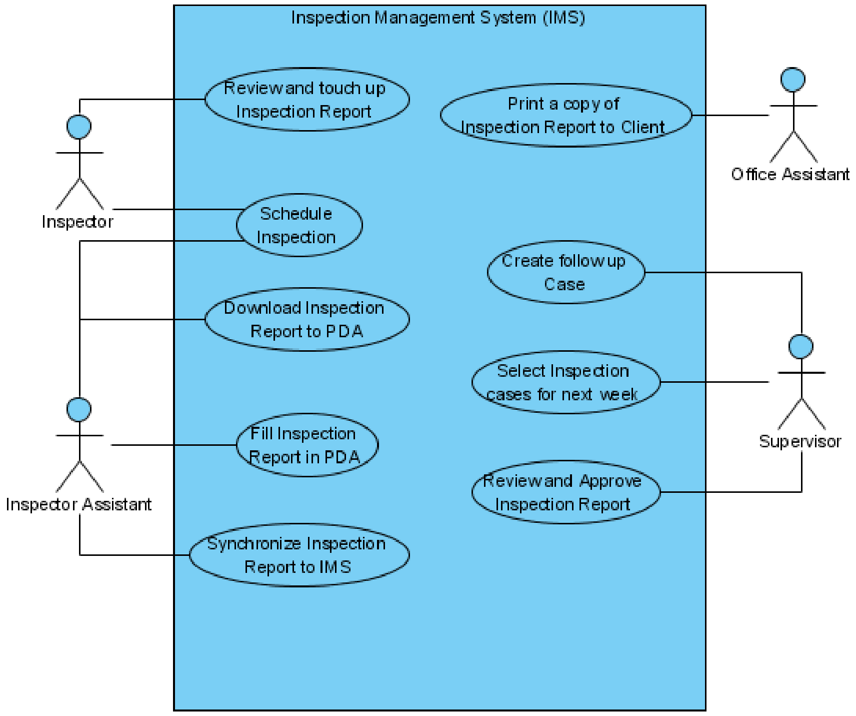

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS

Figure 1-64 *Branch project*

3. Select **Project > Merge...** from menu.

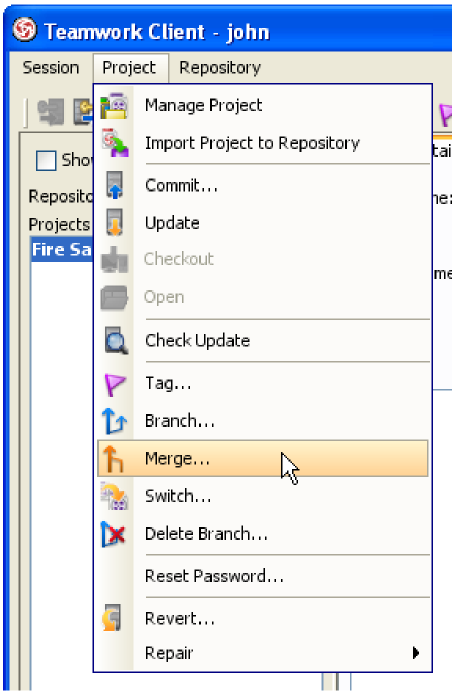


Figure 1-65 *Merge from menu*

Or click **Merge...** button from toolbar.

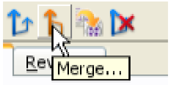
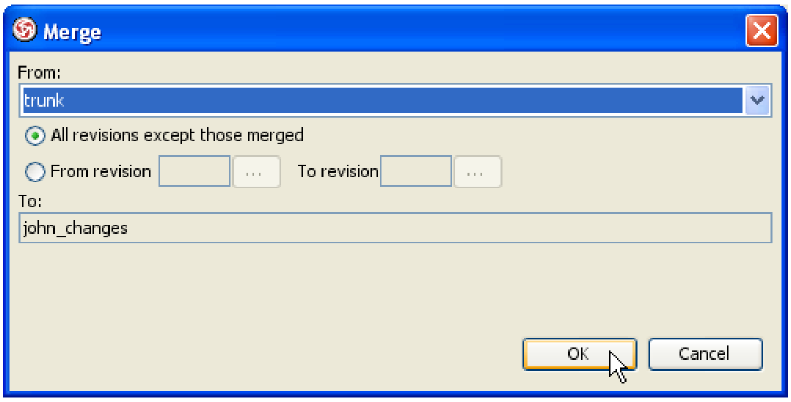


Figure 1-66 *Merge from toolbar*

4. Select from trunk in the **Merge** dialog.



Merge

From: trunk

● All revisions except those merged

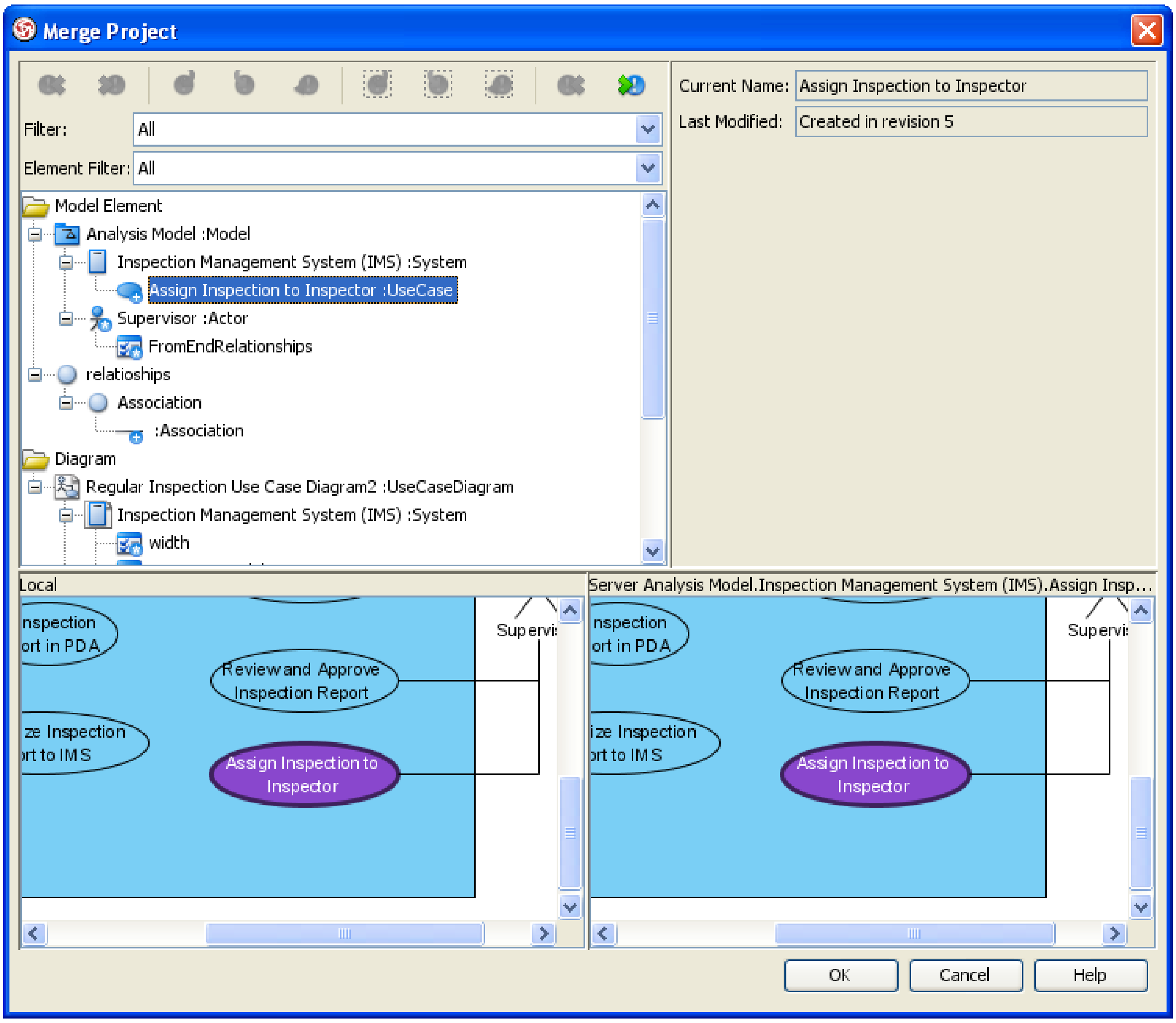
○ From revision ... To revision ...

To: john_changes

OK Cancel

Figure 1-67 *Merge dialog*

5. The **Merge Project** dialog show a list of changes similar to **Commit** and **Update** dialog.



Merge Project

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Assign Inspection to Inspector :UseCase
    - Supervisor :Actor
      - FromEndRelationships
  - relatioships
    - Association
      - :Association
- Diagram
  - Regular Inspection Use Case Diagram2 :UseCaseDiagram
    - Inspection Management System (IMS) :System
      - width

Current Name: Assign Inspection to Inspector

Last Modified: Created in revision 5

Local

Server Analysis Model.Inspection Management System (IMS).Assign Insp...

nspection
ort in PDA
Review and Approve Inspection Report
ze Inspection
rt to IMS
Assign Inspection to Inspector
Supervis
OK
Cancel
Help

Figure 1-68 Merge project dialog

6. Finally, review the changes in **Commit** dialog. Input comment and click **OK** to commit.

7. The branch project now contains the changes from trunk.

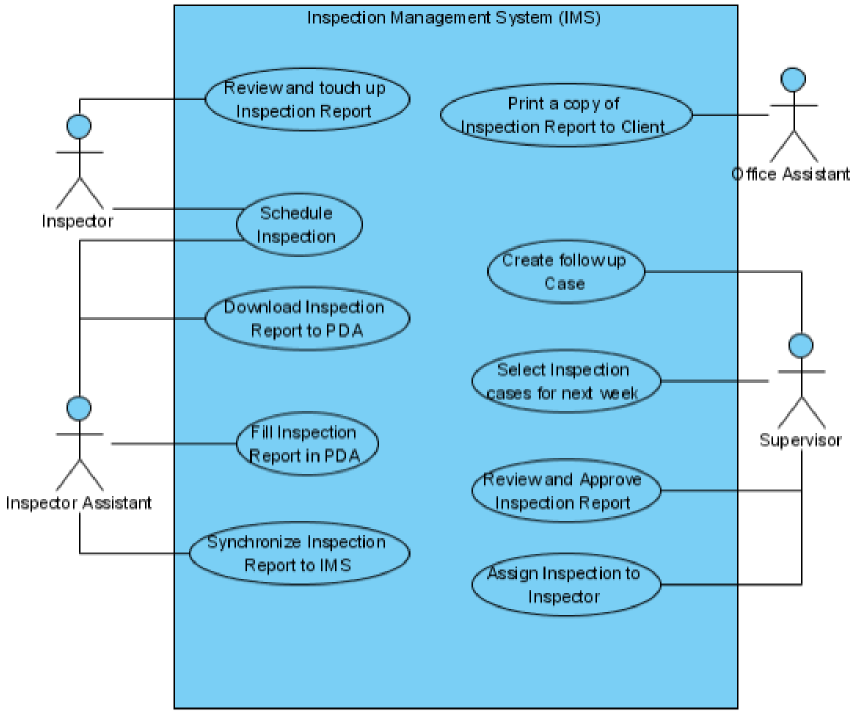

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Inspector Assistant
Fill Inspection Report in PDA
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 1-69 Branch project merged from trunk

Merging from Branch to Trunk

1. Open, modify and commit branch project.

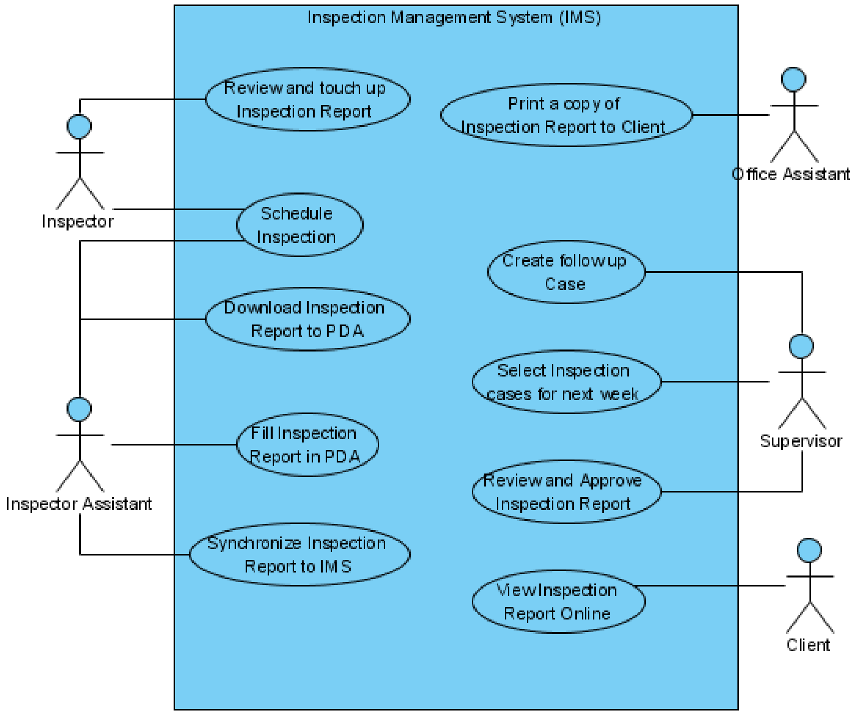

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Inspector Assistant
Fill Inspection Report in PDA
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
View Inspection Report Online
Client

Figure 1-70 Modified branch project

2. Open trunk project.

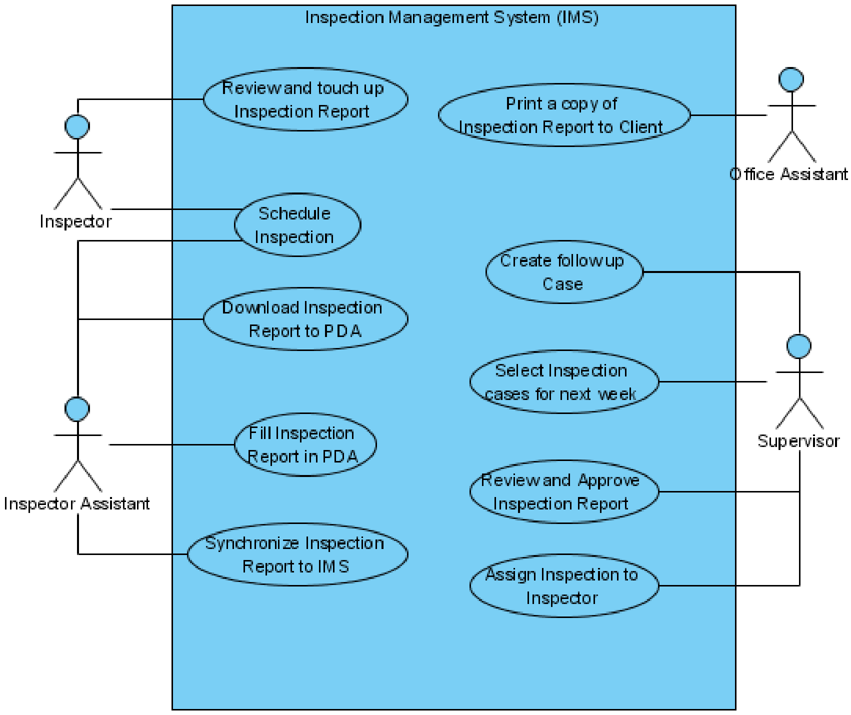
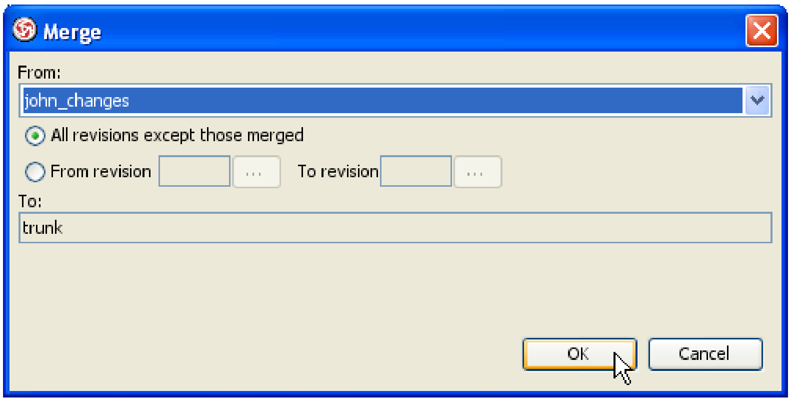

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 1-71 *Trunk project*

3. Select **Project > Merge...** from menu, or click **Merge...** button from toolbar.
4. Select a branch name in **From** combo box.



From: john_changes

● All revisions except those merged

○ From revision ... To revision ...

To: trunk

Figure 1-72 *Merge dialog*

5. The **Merge Project** dialog show a list of changes similar to **Commit** and **Update** dialog.
6. Finally, review the changes in **Commit** dialog. Input comment and click **OK** to commit.

7. The trunk project now contains the changes from branch.

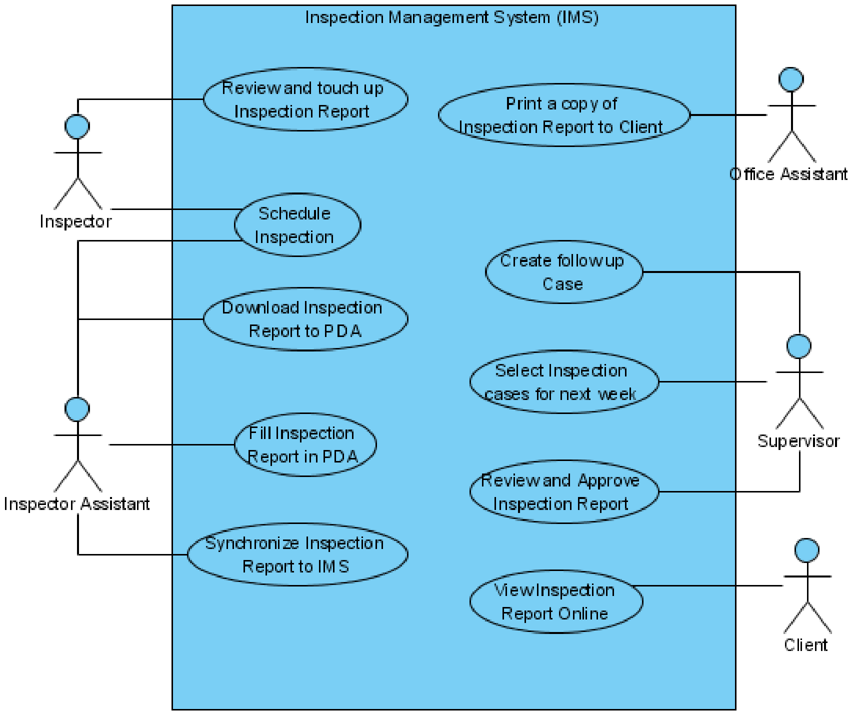

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Inspector Assistant
Fill Inspection Report in PDA
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
View Inspection Report Online
Client

Figure 1-73 Trunk project merged from branch

Delete Branch

1. Select **Project > Delete Branch...** from menu

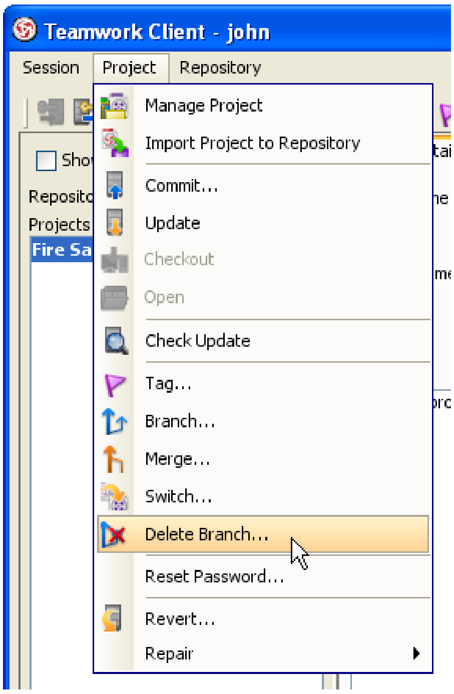


Figure 1-74 Delete Branch from menu

Or click **Delete Branch ...** button from toolbar.

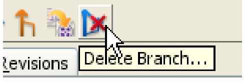
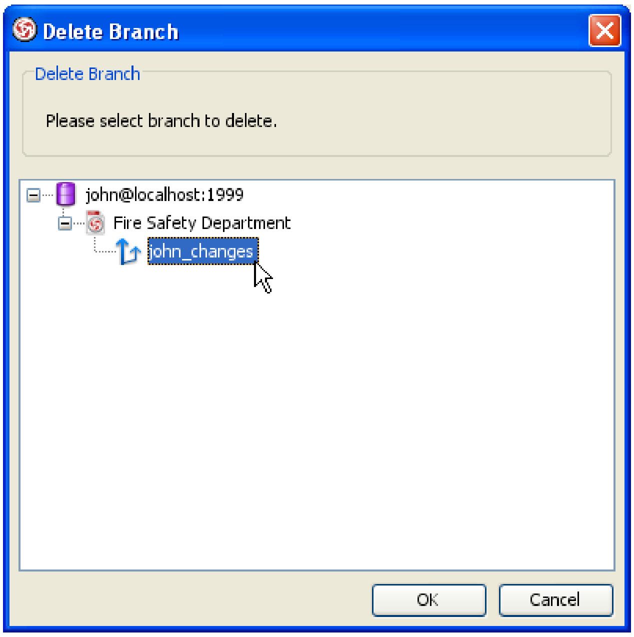


Figure 1-75 Delete Branch from toolbar

2. Expand the repository and folder node, select the branch to delete. Click **OK** button to continue.



Delete Branch

Delete Branch

Please select branch to delete.

Figure 1-76 *Delete branch dialog*

3. Click **Yes** button on confirmation dialog.

Marking release or milestone with Tags

Tags and branches are almost the same, with the only difference - tags are read only. Creating read only tags ensure you are able open the release version project later.

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Tag...** from menu

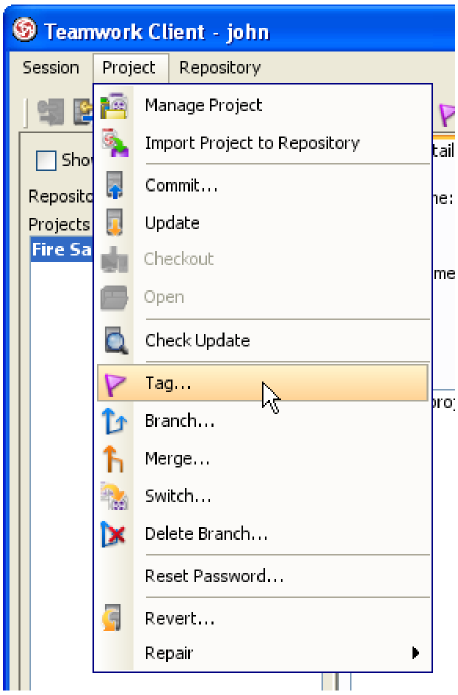


Figure 1-77 *Tag from menu*

Or click **Branch...** button from toolbar.

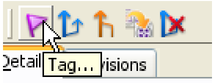
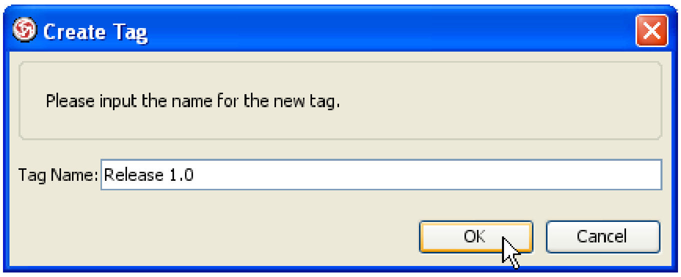


Figure 1-78 *Tag from toolbar*

4. Fill in the **Tag Name** in **Create Tag** dialog.



Create Tag

Please input the name for the new tag.

Tag Name: Release 1.0

Figure 1-79 *Create Tag dialog*

5. Click **OK** button, the tag will be created.

Importing Projects to Subversion

1. Start **VP-UML**.
2. Select **Tools > Teamwork > Open Teamwork Client...** from the main menu, or **Open Teamwork Client** icon from toolbar to open teamwork client.

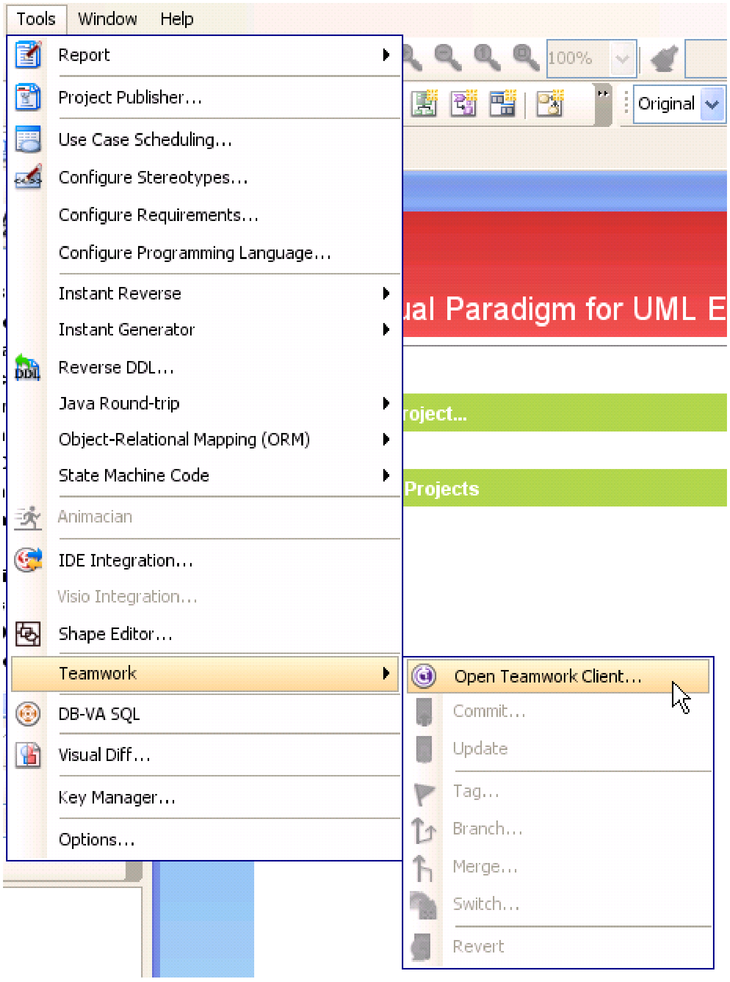
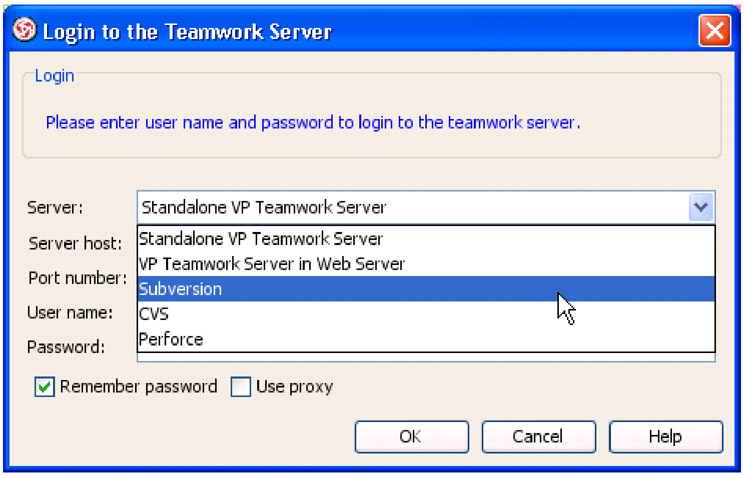


Figure 2-1 *Open Teamwork Client from main menu*

3. Select **Subversion** in **Server** field.


Login to the Teamwork Server

Login

Please enter user name and password to login to the teamwork server.

Server: Standalone VP Teamwork Server

Server host:

Port number:

User name:

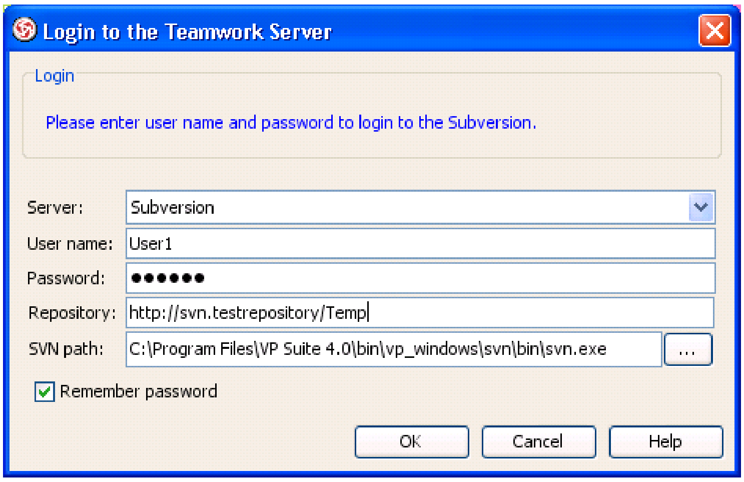
Password:

Standalone VP Teamwork Server
VP Teamwork Server in Web Server
Subversion
CVS
Perforce

☑ Remember password □ Use proxy

Figure 2-2 *subversion teamwork server*

4. Fill in the server and user information to login teamwork server.



Login to the Teamwork Server

Login

Please enter user name and password to login to the Subversion.

Server: Subversion

User name: User1

Password: ●●●●●●

Repository: http://svn.testrepository/Temp

SVN path: C:\Program Files\VP Suite 4.0\bin\vp_windows\svn\bin\svn.exe

☑ Remember password

Figure 2-3 Login teamwork server

5. Select **Project > Import Project to Repository** from the menu.

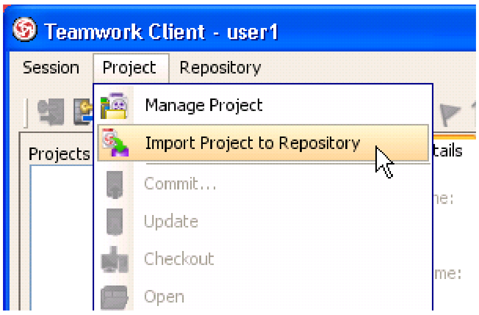


Figure 2-4 Import project from menu

Or click the **Import Project to Repository** button on toolbar.

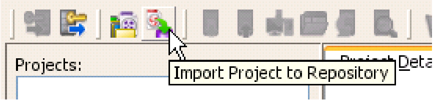
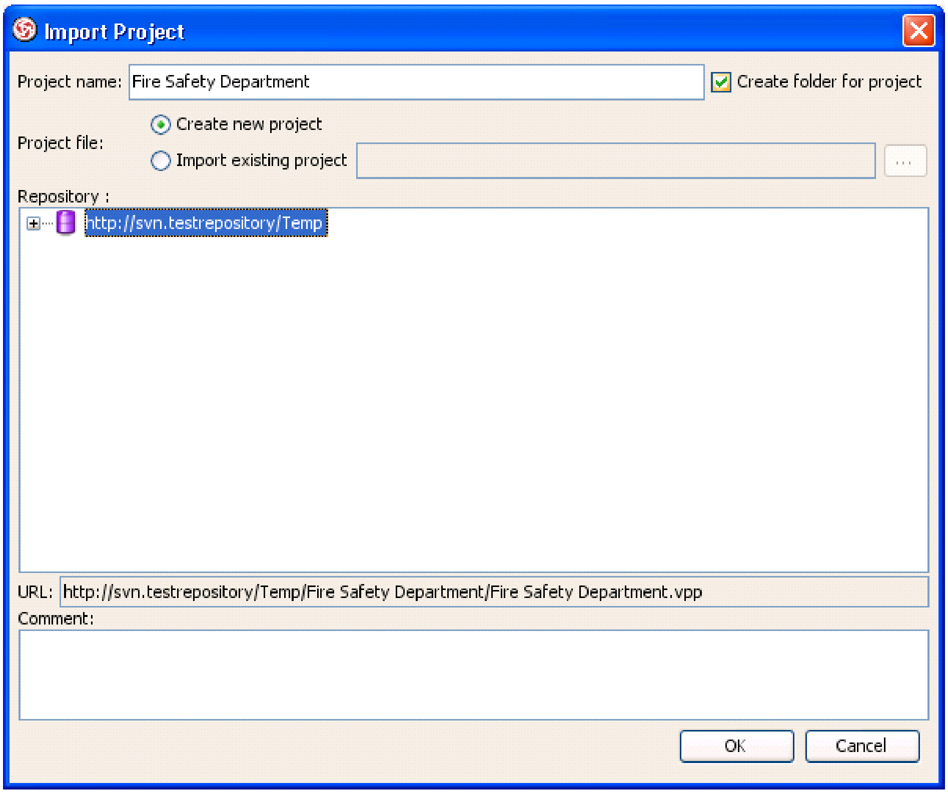


Figure 2-5 Import project from toolbar

6. In the **Import Project** dialog, fill in the **Project name**.



Import Project

Project name: Fire Safety Department ☑ Create folder for project

Project file: ● Create new project ○ Import existing project

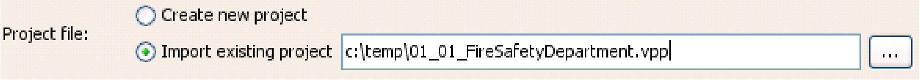
Repository : http://svn.testrepository/Temp

URL: http://svn.testrepository/Temp/Fire Safety Department/Fire Safety Department.vpp

Comment:

Figure 2-6 *Fill project name in Import Project dialog*

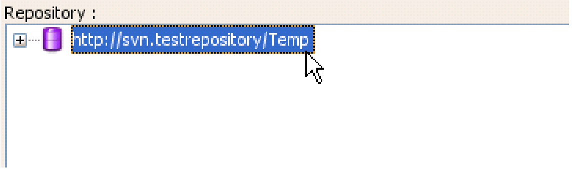
7. Choose **Create new project** or **Import existing project** from **Project file**. **Create new project** will import a blank project into teamwork server, **Import existing project** will import an existing VP-UML Project file (*.vpp) into teamwork server as first revision.



Project file: ○ Create new project ● Import existing project c:\temp\01_01_FireSafetyDepartment.vpp

Figure 2-7 *Specify project file*

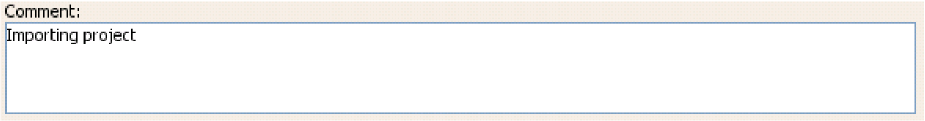
8. Select the repository for import the project.



Repository : http://svn.testrepository/Temp

Figure 2-8 *Select repository*

9. Fill in the comment.



Comment: Importing project

Figure 2-9 *Input comment*

10. Click **OK** button to import project.

NOTE: Please make sure the user login to **Teamwork Client** has **Create Project** permission.

Checkout Project from Subversion

Checkout and open Subversion Projects

1. Open **Teamwork Client** dialog.
2. Select **Project > Manage Project** from the menu.

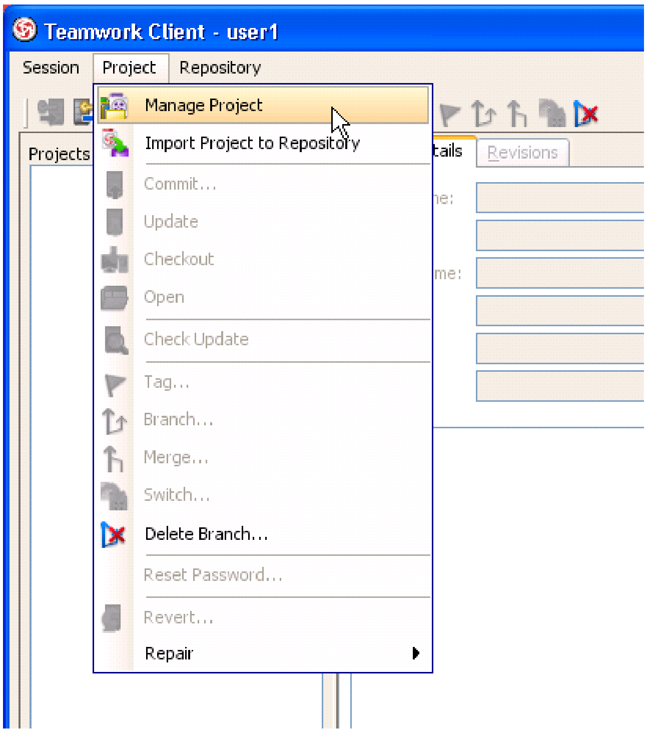


Figure 2-10 *Manage project from main menu*

Or click **Manage Project** icon on the toolbar.

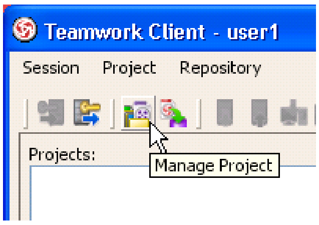


Figure 2-11 *Manage project from toolbar*

3. In **Manage Project** dialog, expand the repository node and select the project to checkout, click **>** button to add project. Click **OK** button to close the dialog.

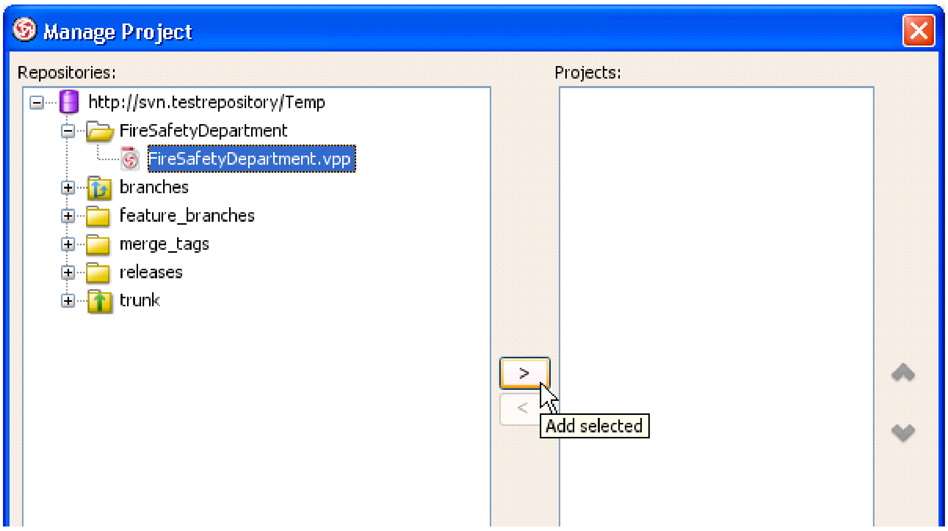
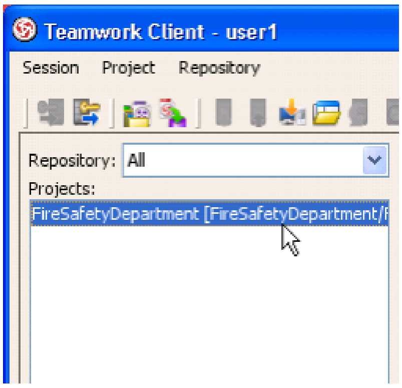


Figure 2-12 *Manage projects*

4. Select the project in project list.



Repository: All

Projects:

FireSafetyDepartment [FireSafetyDepartment/F

Figure 2-13 Select project from the list

5. Click **Open Project** button to checkout and open the project.

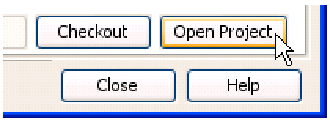


Figure 2-14 Open project

6. The project opened in **VP-UML**.

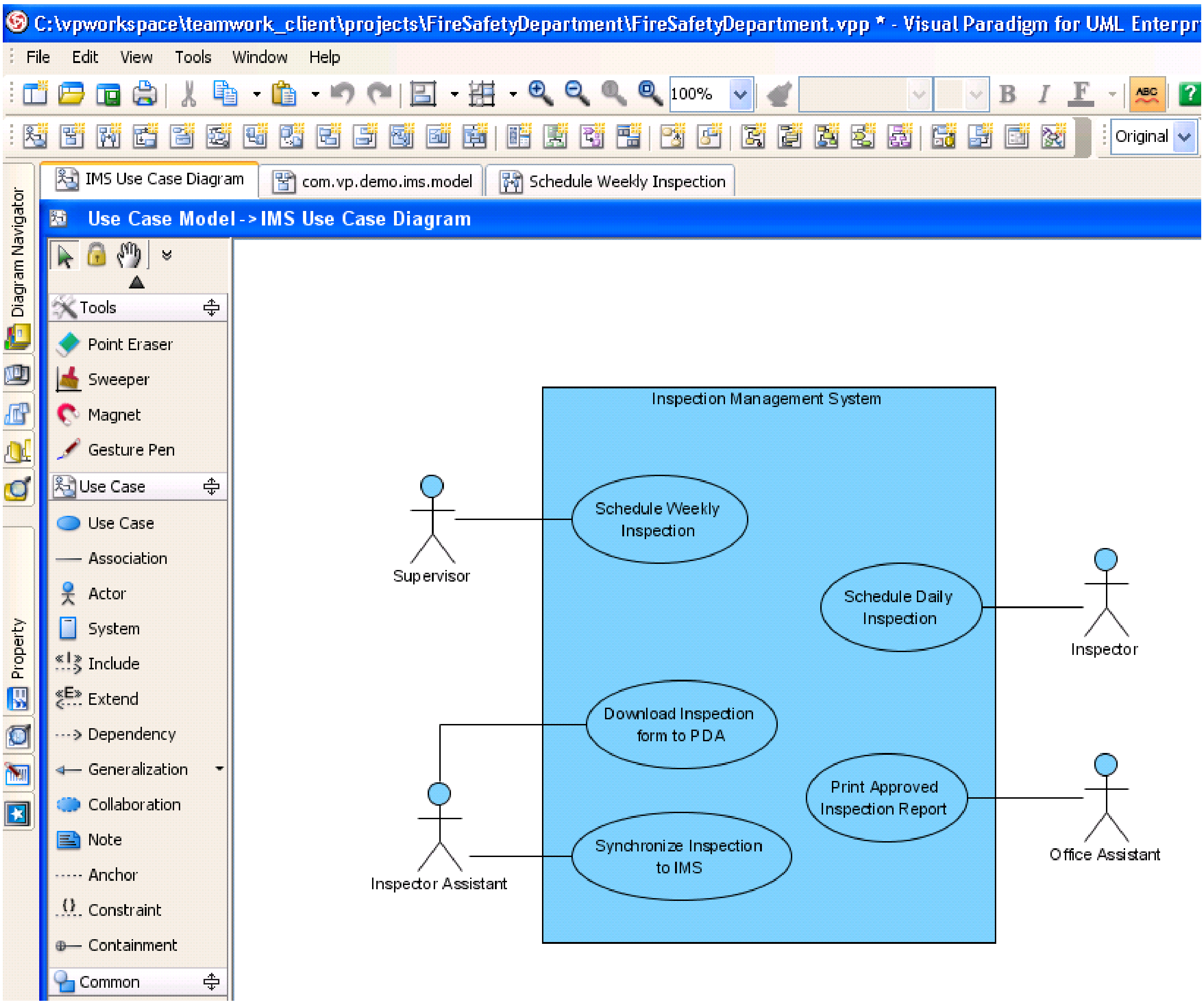

C:\vpworkspace\teamwork_client\projects\FireSafetyDepartment\FireSafetyDepartment.vpp * - Visual Paradigm for UML Enterpr
File
Edit
View
Tools
Window
Help
100%
Original
IMS Use Case Diagram
com.vp.demo.ims.model
Schedule Weekly Inspection
Use Case Model -> IMS Use Case Diagram
Diagram Navigator
Property
Tools
Point Eraser
Sweeper
Magnet
Gesture Pen
Use Case
Use Case
Association
Actor
System
Include
Extend
Dependency
Generalization
Collaboration
Note
Anchor
Constraint
Containment
Common
Inspection Management System
Supervisor
Schedule Weekly Inspection
Schedule Daily Inspection
Inspector
Download Inspection form to PDA
Print Approved Inspection Report
Office Assistant
Synchronize Inspection to IMS
Inspector Assistant

Figure 2-15 Project opened

Committing local modification to Subversion

1. Modify the project.

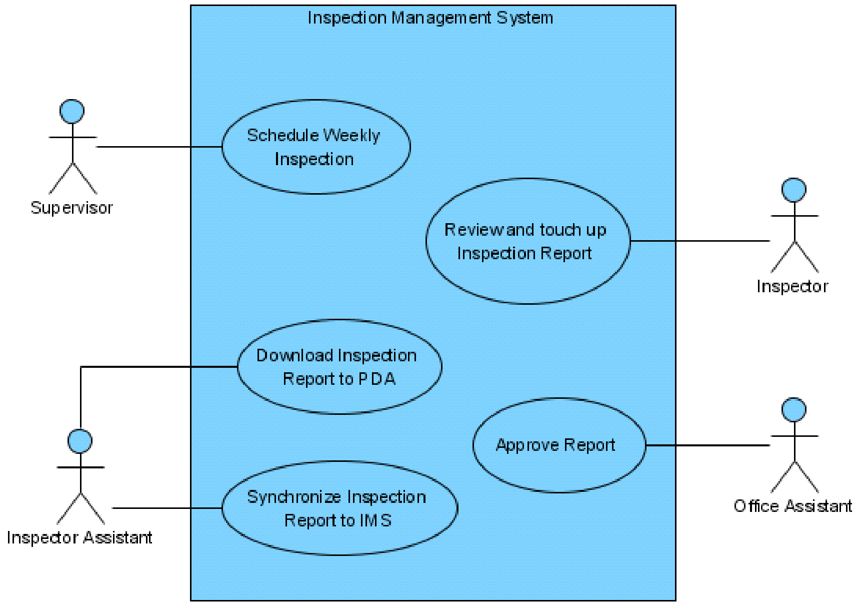

Inspection Management System
Schedule Weekly Inspection
Review and touch up Inspection Report
Download Inspection Report to PDA
Approve Report
Synchronize Inspection Report to IMS
Supervisor
Inspector
Inspector Assistant
Office Assistant

Figure 2-16 *Modified project*

2. Show **Teamwork Toolbar** by right click on the Toolbar, select **Teamwork** if not selected already.

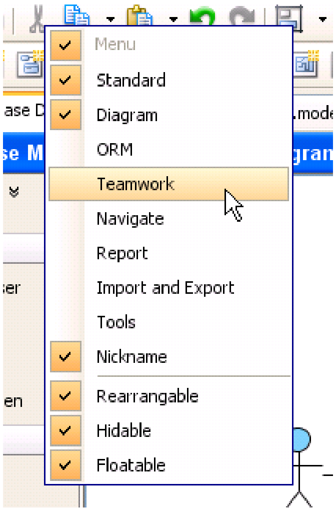


Figure 2-17 *Show teamwork toolbar*

3. Click the **Commit** button on **Teamwork Toolbar**.

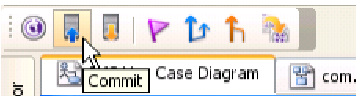
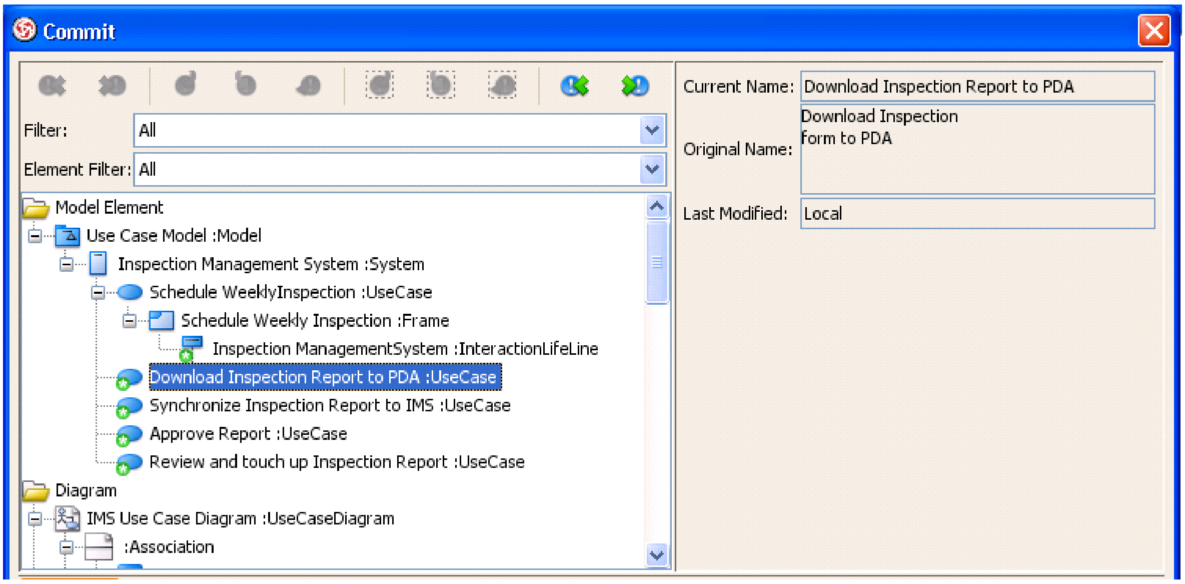


Figure 2-18 *Commit project*

4. In **Commit** dialog, you can review the changes for commit. On the left of dialog, you can see a list of changes shapes and model elements, click on it to view the detail changes on the right.



Filter: All

Element Filter: All

- Model Element
  - Use Case Model :Model
    - Inspection Management System :System
      - Schedule WeeklyInspection :UseCase
        - Schedule Weekly Inspection :Frame
          - Inspection ManagementSystem :InteractionLifeLine
      - Download Inspection Report to PDA :UseCase
      - Synchronize Inspection Report to IMS :UseCase
      - Approve Report :UseCase
      - Review and touch up Inspection Report :UseCase
- Diagram
  - IMS Use Case Diagram :UseCaseDiagram
    - :Association

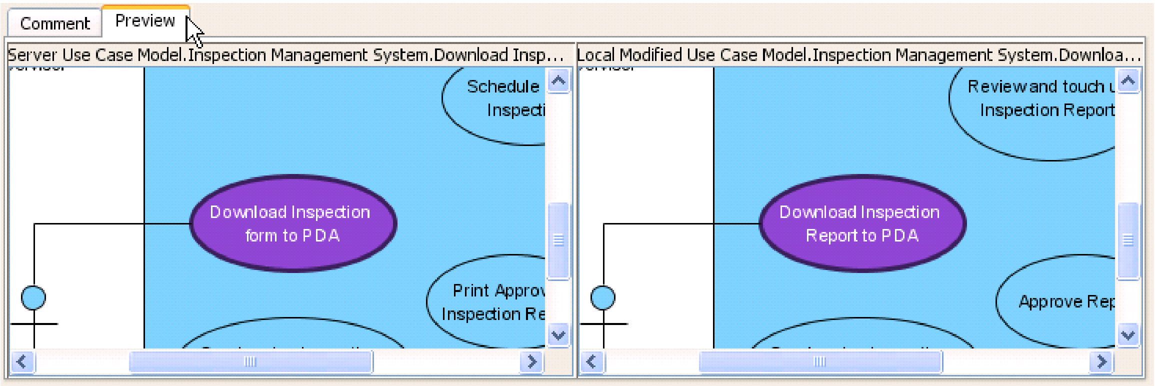
Current Name: Download Inspection Report to PDA

Original Name: Download Inspection form to PDA

Last Modified: Local

Figure 2-19 *Review commit changes*

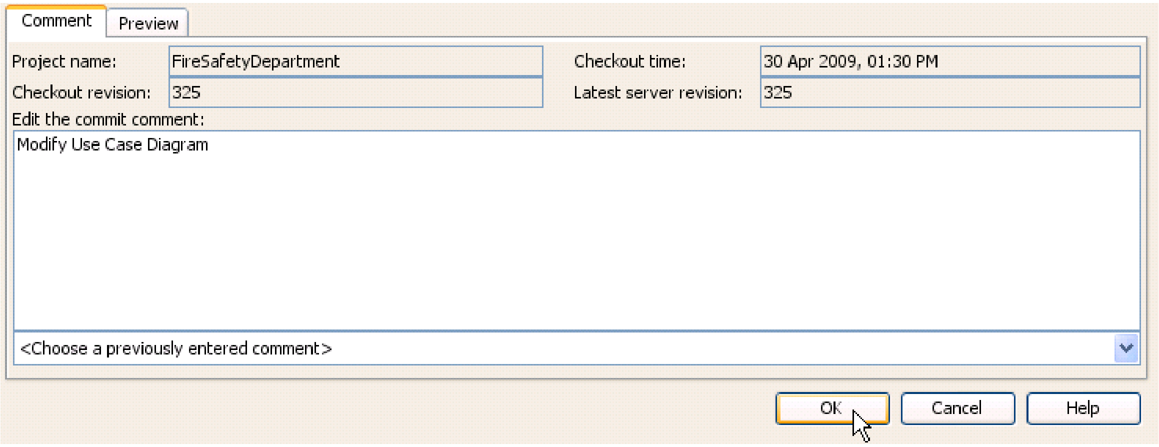
5. On the bottom of dialog, click the **Preview** tab to visually preview the changes in diagram. The selected shape is highlighed in purple.



Comment
Preview
Server Use Case Model.Inspection Management System.Download Insp...
Schedule Inspecti
Download Inspection form to PDA
Print Approv Inspection Re
Local Modified Use Case Model.Inspection Management System.Downloa...
Review and touch u Inspection Report
Download Inspection Report to PDA
Approve Rep

Figure 2-20 *Preview commit changes*

6. After review the changes, click the **Comment** tab and input the comment for commit. Click **OK** button to start commit.



Project name: FireSafetyDepartment

Checkout time: 30 Apr 2009, 01:30 PM

Checkout revision: 325

Latest server revision: 325

Edit the commit comment:

Modify Use Case Diagram

<Choose a previously entered comment>

Figure 2-21 *Input commit comment*

Resolving conflicts

1. Modify the project.

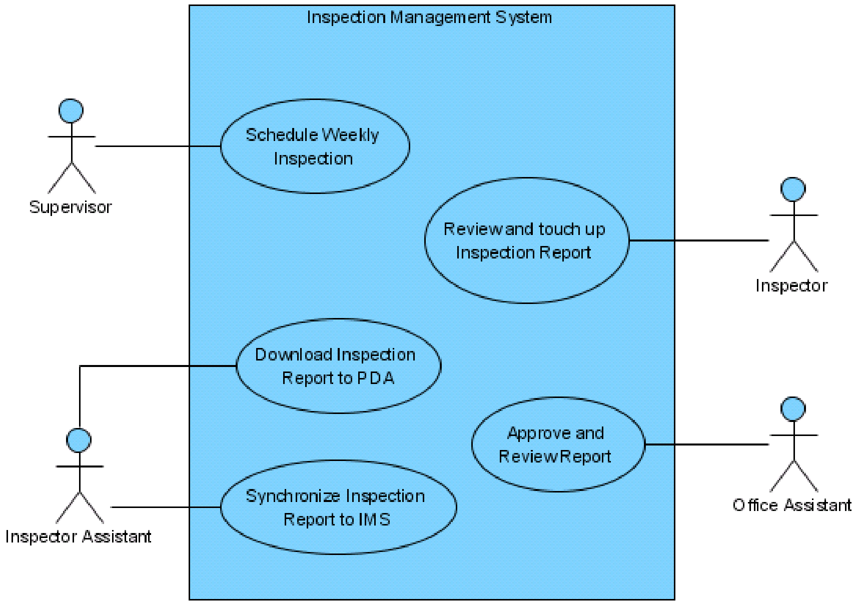
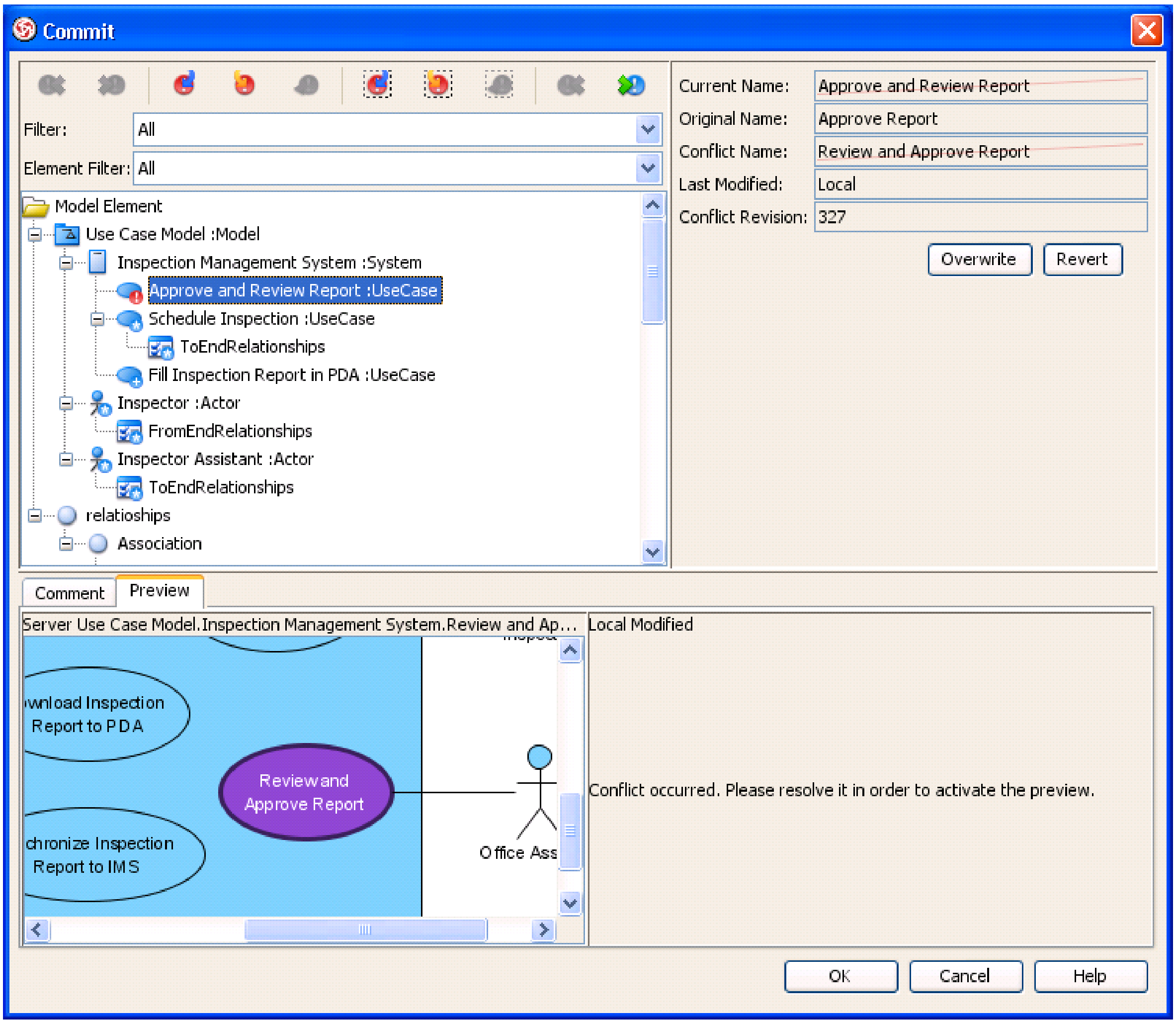

Inspection Management System
Supervisor
Schedule Weekly Inspection
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Approve and Review Report
Office Assistant
Synchronize Inspection Report to IMS
Inspector Assistant

***Figure 2-22** Modified project*

2. Click **Commit** button on the **Teamwork Toolbar**.

3. In **Commit** dialog, you may found some shapes or model elements show with red icon. This indicate there is conflict when commit, it is caused by someone modified the same content and commit after you checkout. You can review the current value, original value (the value when you checkout), and conflict value (the value changed by other users). And dialog preview is disabled until you resolve the conflict.



Filter: All

Element Filter: All

- Model Element
  - Use Case Model :Model
    - Inspection Management System :System
      - Approve and Review Report :UseCase
      - Schedule Inspection :UseCase
        - ToEndRelationships
      - Fill Inspection Report in PDA :UseCase
    - Inspector :Actor
      - FromEndRelationships
    - Inspector Assistant :Actor
      - ToEndRelationships
  - relatioships
    - Association

Current Name: Approve and Review Report

Original Name: Approve Report

Conflict Name: Review and Approve Report

Last Modified: Local

Conflict Revision: 327

Server Use Case Model.Inspection Management System.Review and Ap...

wnload Inspection Report to PDA
Review and Approve Report
chronize Inspection Report to IMS
Office Ass

Local Modified

Conflict occurred. Please resolve it in order to activate the preview.

Figure 2-23 *Commit with conflicts*

4. Select the conflict element, click **Overwrite selected conflicts** button on the toolbar to overwrite other user's change.

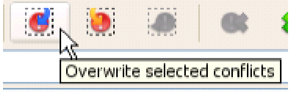


Figure 2-24 *Overwrite selected conflicts*

Or click **Revert selected conflicts** to revert your own changes.

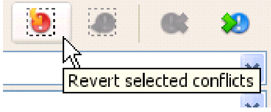


Figure 2-25 *Revert selected conflicts*

You can also click **Overwrite all conflicts** or **Revert all conflicts** to overwrite or revert all conflicts at once.

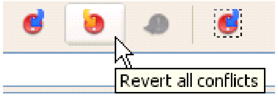


Figure 2-26 *Overwrite/Revert all conflicts*

5. After resolve conflict, the preview will be enabled to visualize the final result.

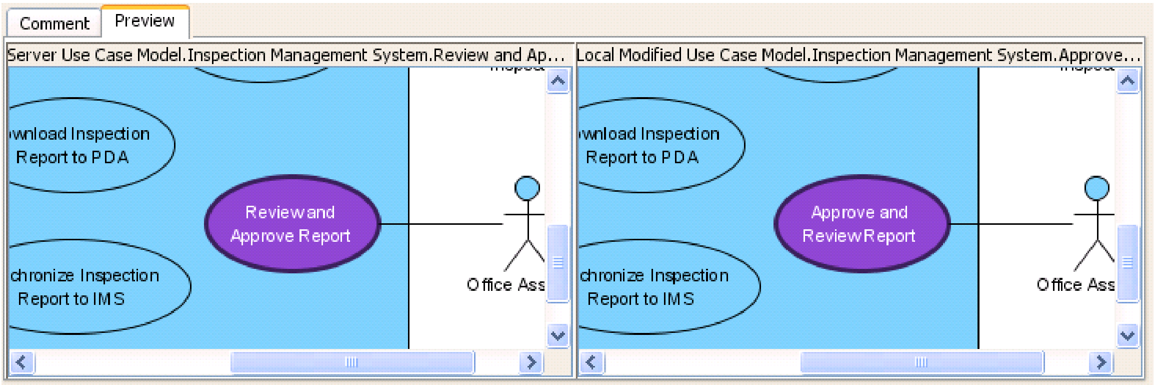

Comment
Preview
Server Use Case Model.Inspection Management System.Review and Ap...
Local Modified Use Case Model.Inspection Management System.Approve...
wnload Inspection Report to PDA
Review and Approve Report
chronize Inspection Report to IMS
Office Ass
wnload Inspection Report to PDA
Approve and Review Report
chronize Inspection Report to IMS
Office Ass

Figure 2-27 Preview resolved conflict

6. You can change your mind by click **Reset selected conflicts** or **Reset all conflicts** button. Then overwrite or revert the conflict again.

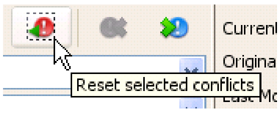


Figure 2-28 Reset conflicts

7. After all conflicts was resolved, you can now input comment and click **OK** button to commit.

Updating latest revision from Subversion

Updating modification from Subversion

1. Click **Update** button on **Teamwork Toolbar**.

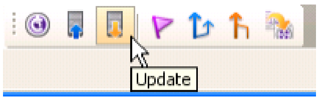
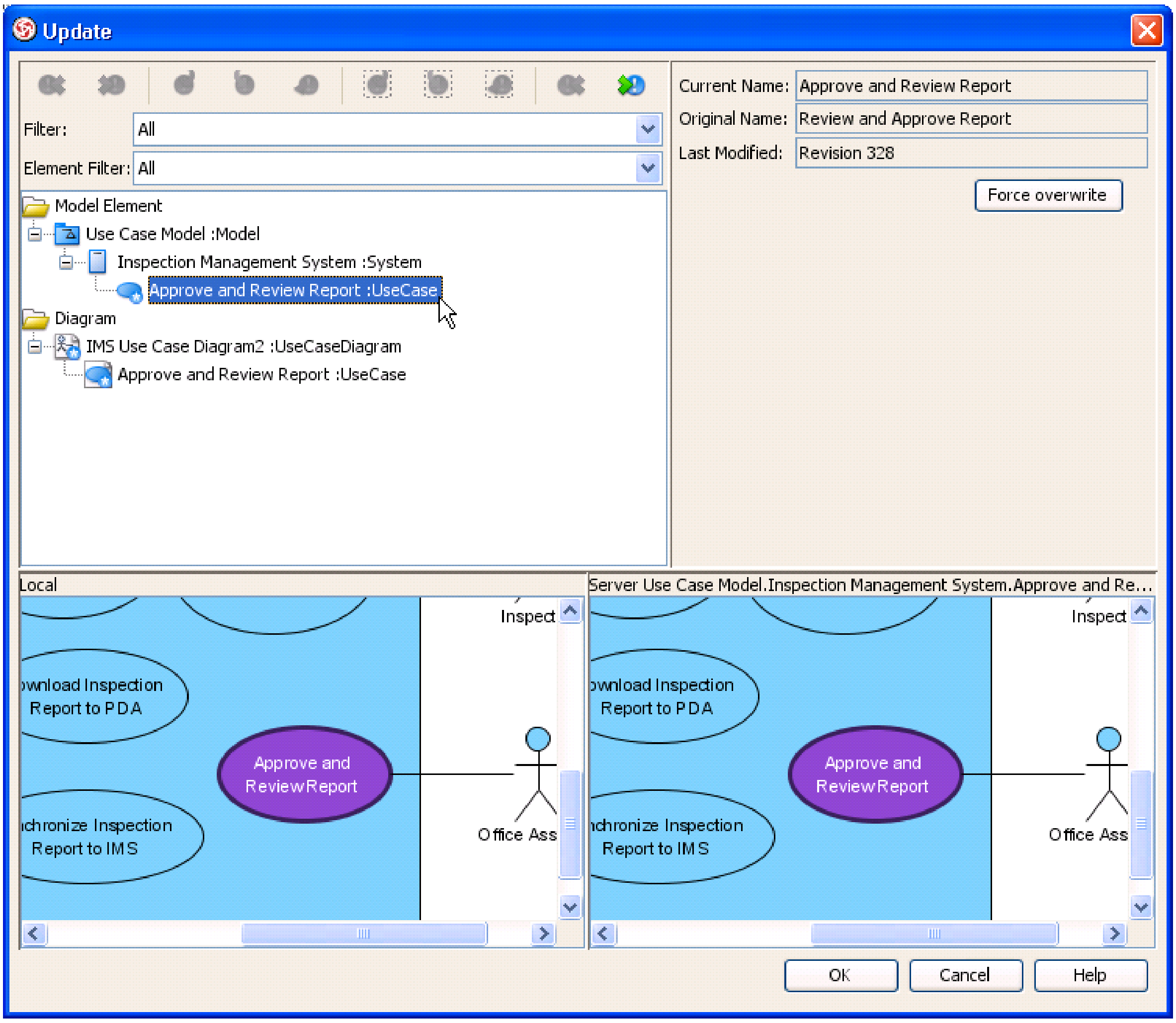


Figure 2-29 *Update from toolbar*

2. Similar to commit, you can review the change and preview the diagram in **Update** dialog.


Update

Filter: All

Element Filter: All

Model Element
- Use Case Model :Model
  - Inspection Management System :System
    - Approve and Review Report :UseCase

Diagram
- IMS Use Case Diagram2 :UseCaseDiagram
  - Approve and Review Report :UseCase

Current Name: Approve and Review Report

Original Name: Review and Approve Report

Last Modified: Revision 328

Force overwrite
Local
Server Use Case Model.Inspection Management System.Approve and Re...
ownload Inspection Report to PDA
Approve and Review Report
nchronize Inspection Report to IMS
Inspect
Office Ass
OK
Cancel
Help

Figure 2-30 *Update dialog*

3. If there are conflicts, resolve it similar to commit, but the changes will apply to local project instead of commit to server immediately.

4. Click **OK** button to update. **VP-UML** will open the updated project.

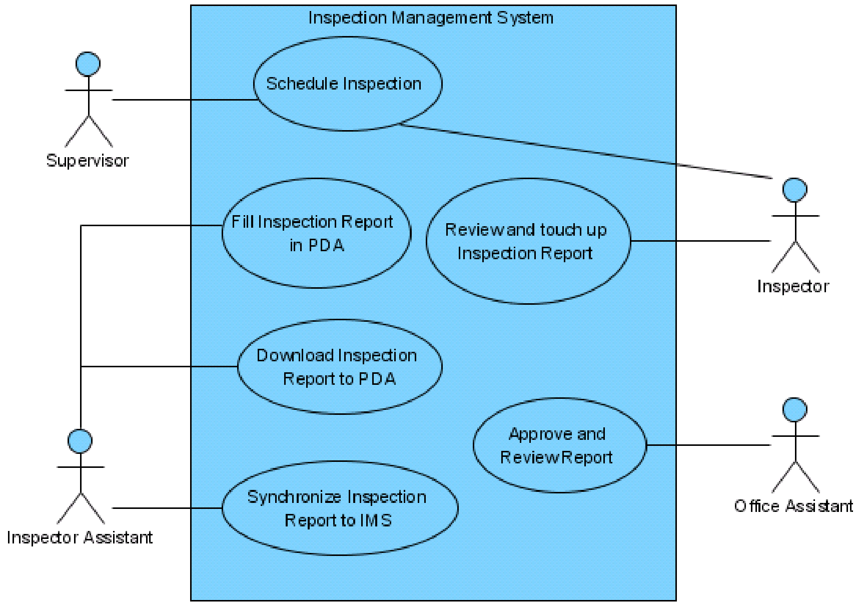

Inspection Management System
Schedule Inspection
Supervisor
Fill Inspection Report in PDA
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Approve and Review Report
Office Assistant
Synchronize Inspection Report to IMS
Inspector Assistant

Figure 2-31 Updated project

Checking status

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Check Update** from menu.

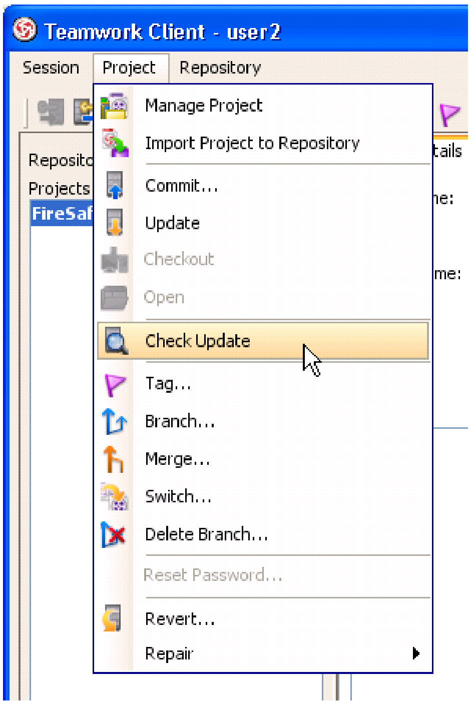


Figure 2-32 Update project from menu

Or click **Check Update** button from toolbar.

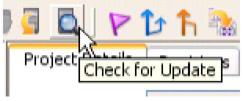
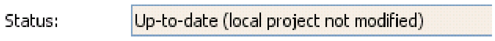


Figure 2-33 Update project from toolbar

4. You can see the status showing **Has update** or **Up-to-date**, it also indicate local project status **(local project not modified)** or **(local project modified)**.



Status: Up-to-date (local project not modified)

Figure 2-34 *Project status*

Reverting Changes

Revert local modifications

1. Modify the project.

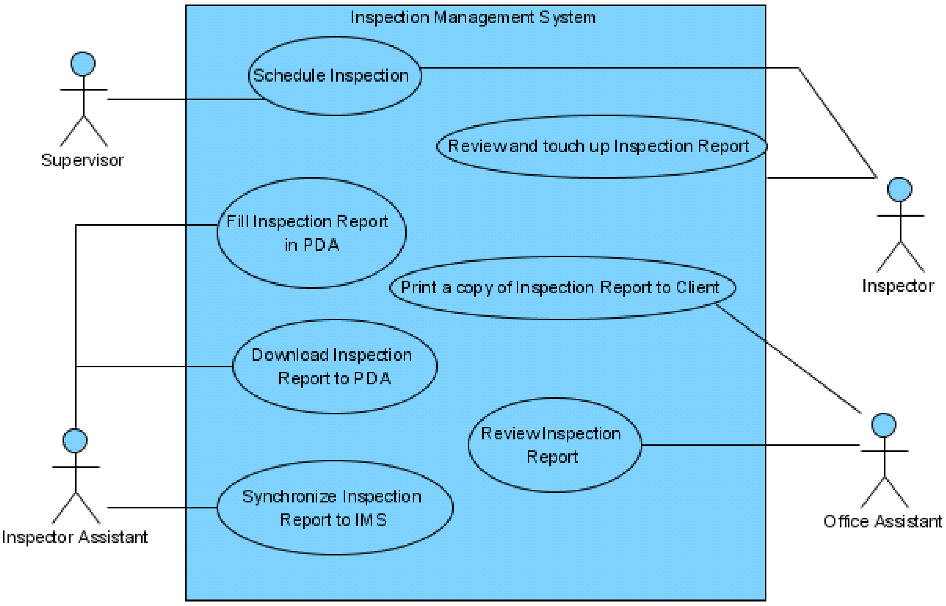

Inspection Management System
Schedule Inspection
Review and touch up Inspection Report
Fill Inspection Report in PDA
Print a copy of Inspection Report to Client
Download Inspection Report to PDA
Review Inspection Report
Synchronize Inspection Report to IMS
Supervisor
Inspector
Inspector Assistant
Office Assistant

Figure 2-35 *Modify project*

2. Open **Teamwork Client** dialog.
3. Select the teamwork project in the list.
4. Select **Project > Revert...** from menu

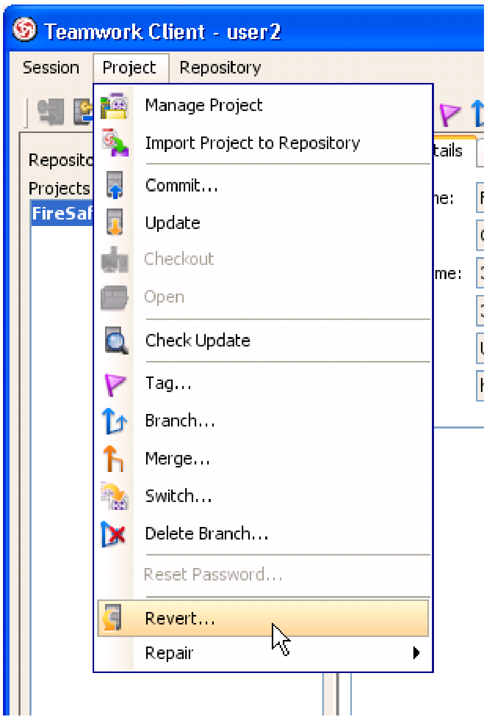


Figure 2-36 *Revert from menu*

Or click **Revert...** button from toolbar.

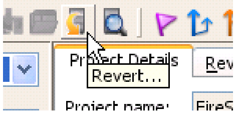


Figure 2-37 *Revert from toolbar*

5. Click **Yes** button from the confirmation dialog.

6. The reverted project opened in **VP-UML**.

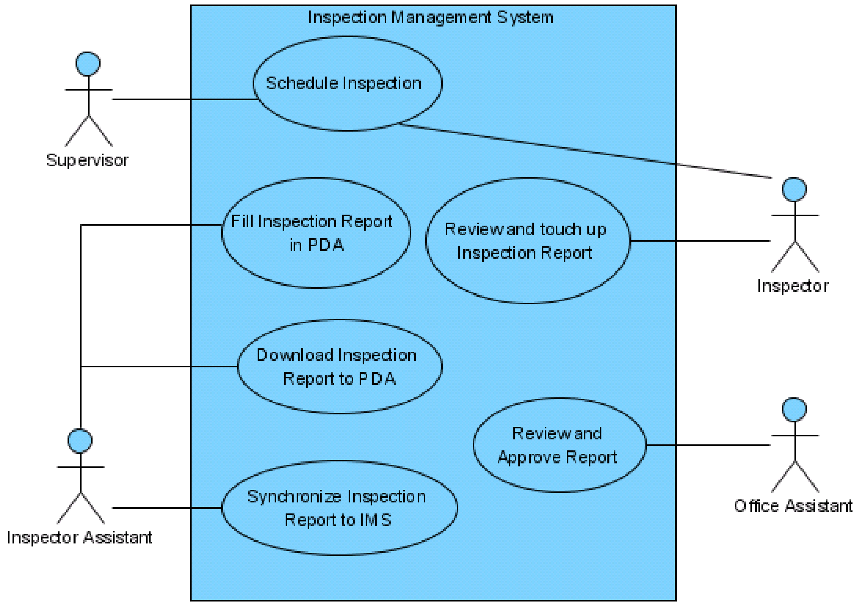

Inspection Management System
Schedule Inspection
Supervisor
Fill Inspection Report in PDA
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Review and Approve Report
Office Assistant
Synchronize Inspection Report to IMS
Inspector Assistant

Figure 2-38 *Reverted project*

Revert server modifications

1. Open project.

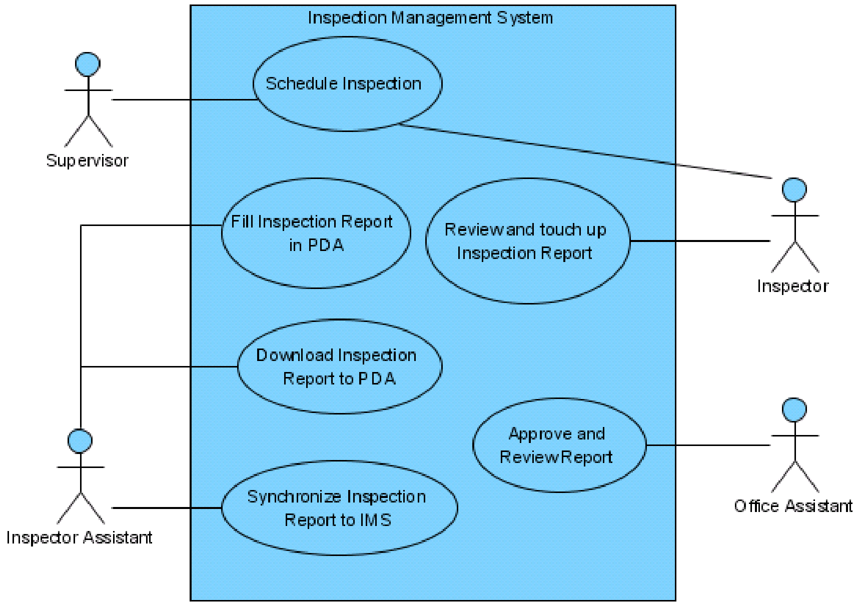
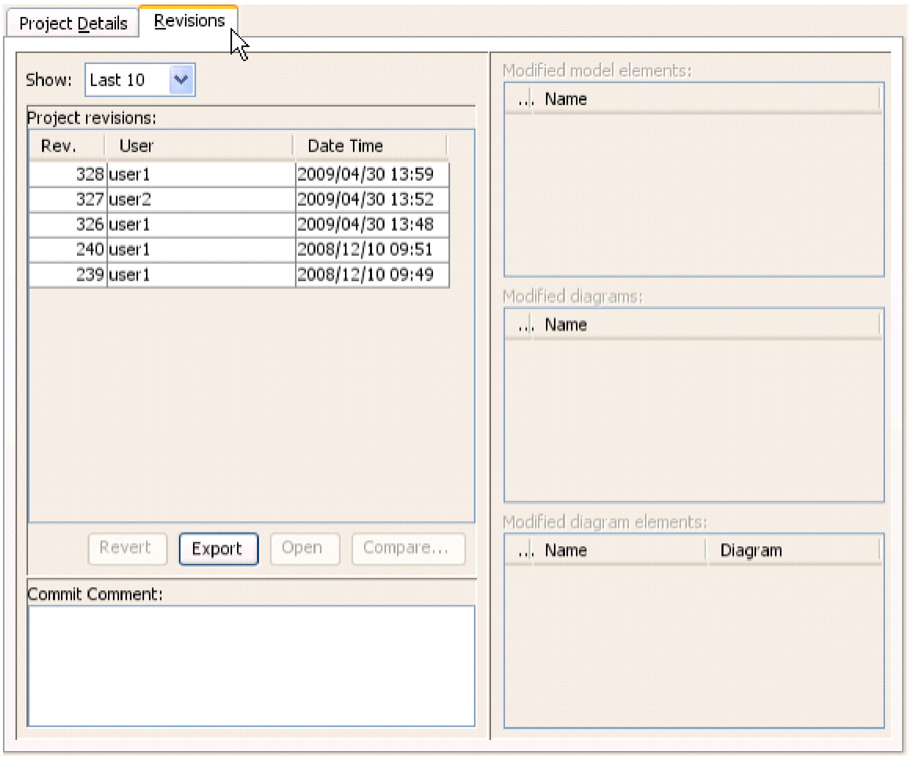

Inspection Management System
Schedule Inspection
Supervisor
Fill Inspection Report in PDA
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Approve and Review Report
Office Assistant
Synchronize Inspection Report to IMS
Inspector Assistant

Figure 2-39 *Open project*

2. Open **Teamwork Client** dialog.
3. Select the teamwork project in the list.

4. Click the **Revisions** tab on the right of the project list.



Show: Last 10

Project revisions:

Rev.	User	Date Time
328	user1	2009/04/30 13:59
327	user2	2009/04/30 13:52
326	user1	2009/04/30 13:48
240	user1	2008/12/10 09:51
239	user1	2008/12/10 09:49

Commit Comment:

Modified model elements:

...	Name

Modified diagrams:

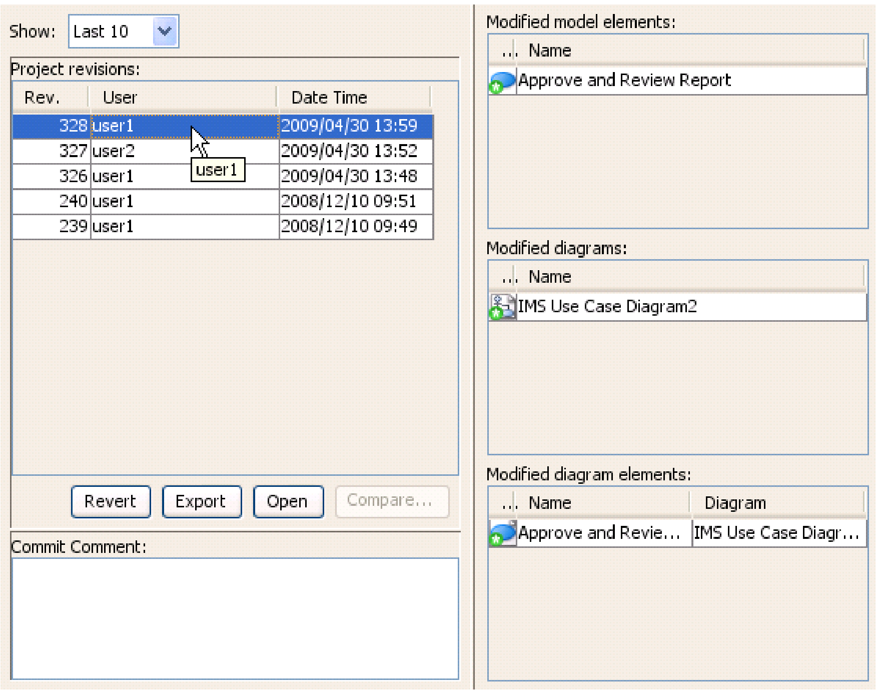
...	Name

Modified diagram elements:

...	Name	Diagram

Figure 2-40 Revision tab

5. Select the revision to revert.



Show: Last 10

Project revisions:

Rev.	User	Date Time
328	user1	2009/04/30 13:59
327	user2	2009/04/30 13:52
326	user1	2009/04/30 13:48
240	user1	2008/12/10 09:51
239	user1	2008/12/10 09:49

Commit Comment:

Modified model elements:

...	Name
	Approve and Review Report

Modified diagrams:

...	Name
	IMS Use Case Diagram2

Modified diagram elements:

...	Name	Diagram
	Approve and Revie...	IMS Use Case Diagr...

Figure 2-41 Select revision

6. Click **Revert** button.

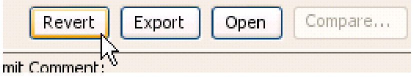
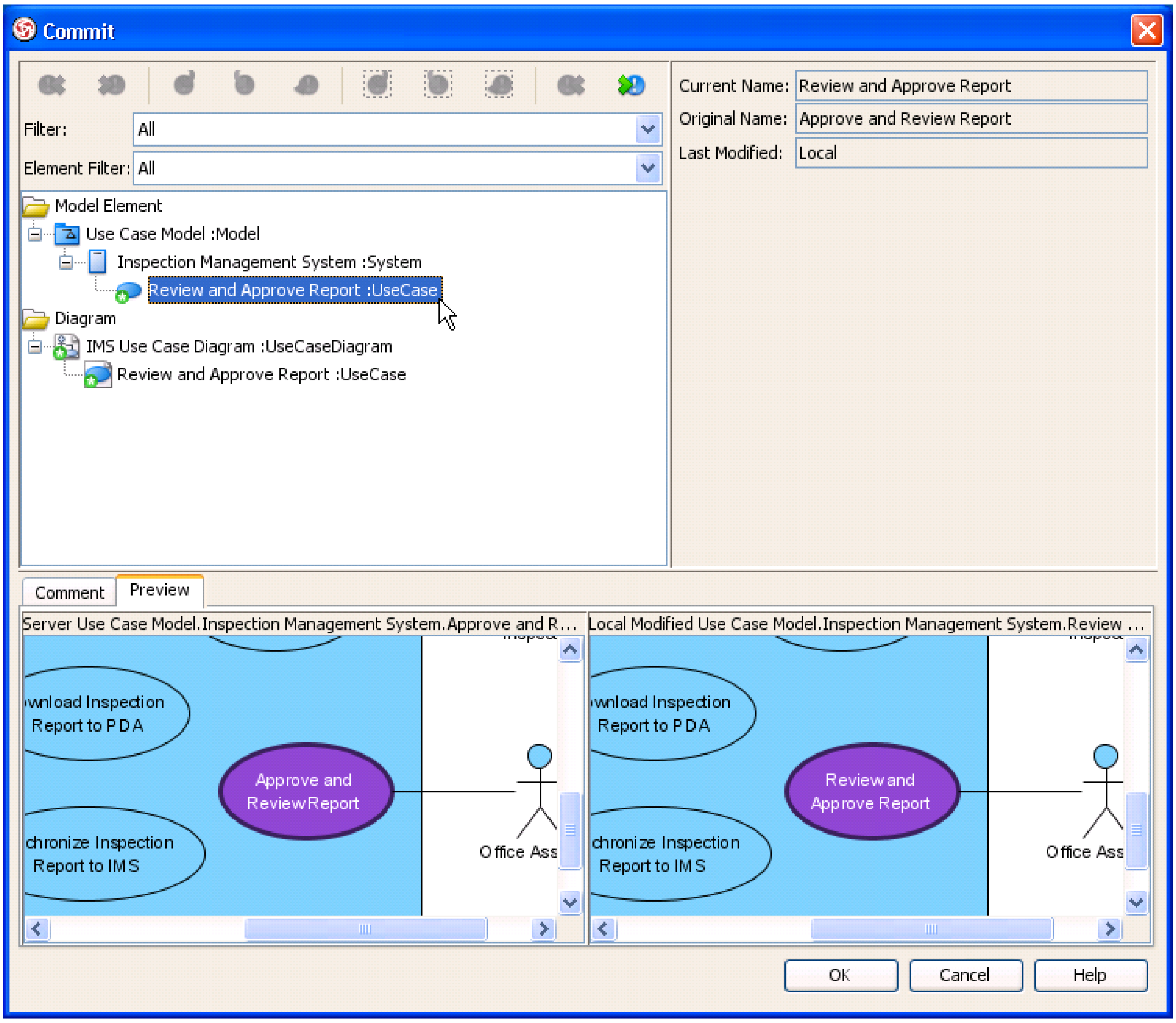


Figure 2-42 Revert button

7. The **Commit** dialog show a list of shapes and model elements reverted for review the changes and preview diagram, input the comment and click **OK** button to commit the revert.



Commit

Filter: All

Element Filter: All

Model Element
- Use Case Model :Model
  - Inspection Management System :System
    - Review and Approve Report :UseCase

Diagram
- IMS Use Case Diagram :UseCaseDiagram
  - Review and Approve Report :UseCase

Current Name: Review and Approve Report

Original Name: Approve and Review Report

Last Modified: Local

Comment | Preview

Server Use Case Model.Inspection Management System.Approve and R...

Local Modified Use Case Model.Inspection Management System.Review ...

Figure 2-43 Commit dialog with reverted changes

8. A new revision with the reverted changes was created in server, and opened in **VP-UML**.

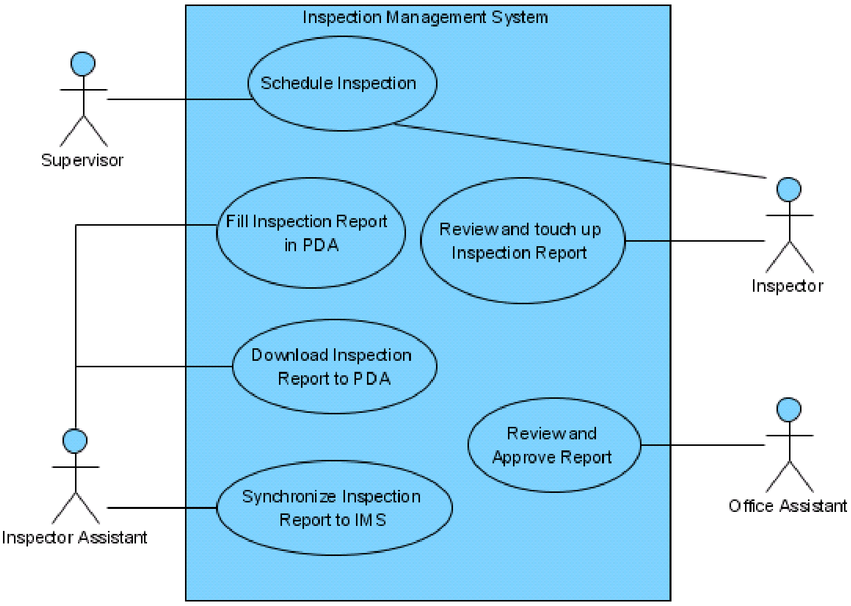
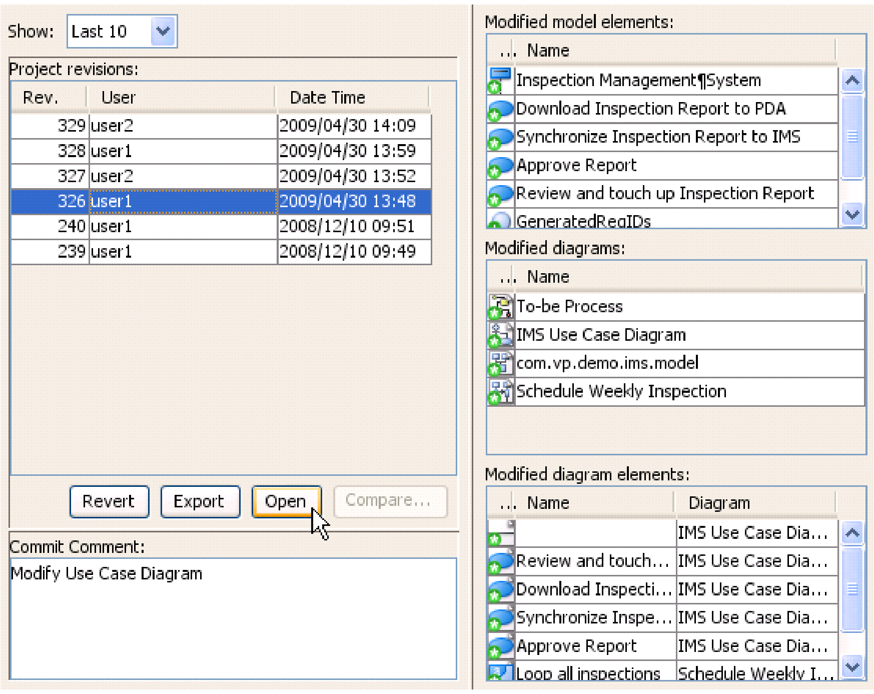


Figure 2-44 Reverted project

Browsing change histories (old revisions)

Checkout old revisions

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select the revision to open.
5. Click Open button.



Show: Last 10

Project revisions:

Rev.	User	Date Time
329	user2	2009/04/30 14:09
328	user1	2009/04/30 13:59
327	user2	2009/04/30 13:52
326	user1	2009/04/30 13:48
240	user1	2008/12/10 09:51
239	user1	2008/12/10 09:49

Commit Comment:

Modify Use Case Diagram

Modified model elements:

...	Name
	Inspection Management¶System
	Download Inspection Report to PDA
	Synchronize Inspection Report to IMS
	Approve Report
	Review and touch up Inspection Report
	GeneratedReqIDs

Modified diagrams:

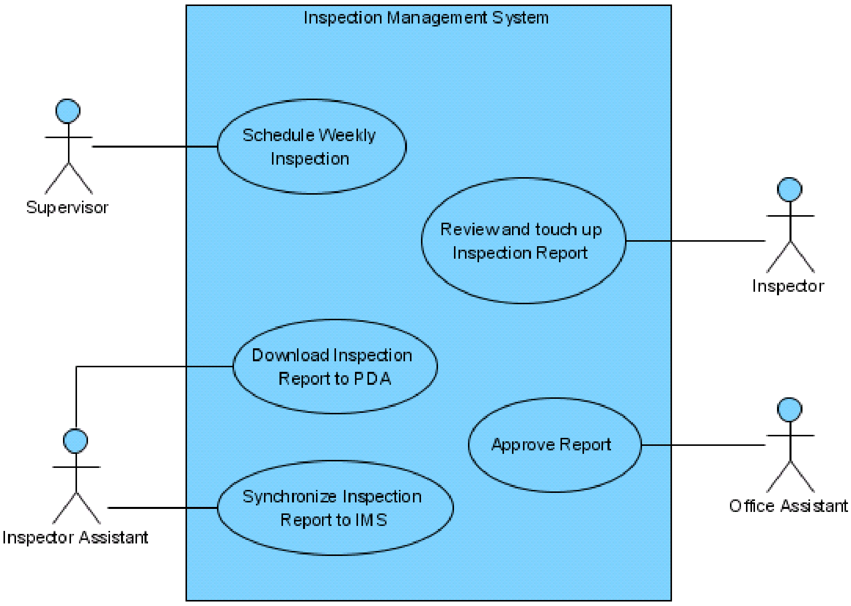
...	Name
	To-be Process
	IMS Use Case Diagram
	com.vp.demo.ims.model
	Schedule Weekly Inspection

Modified diagram elements:

...	Name	Diagram
		IMS Use Case Dia...
	Review and touch...	IMS Use Case Dia...
	Download Inspecti...	IMS Use Case Dia...
	Synchronize Inspe...	IMS Use Case Dia...
	Approve Report	IMS Use Case Dia...
	Loop all inspections	Schedule Weekly I...

Figure 2-45 *Open button*

6. Selected revision opened in **VP-UML**.



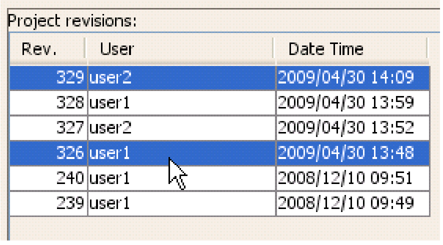
Inspection Management System
Schedule Weekly Inspection
Supervisor
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Approve Report
Office Assistant
Inspector Assistant
Synchronize Inspection Report to IMS

Figure 2-46 *Open selected revision*

Showing differences between revisions visually

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select a revision for compare.

5. Select another revision by **Ctrl+Click**.



Project revisions:

Rev.	User	Date Time
329	user2	2009/04/30 14:09
328	user1	2009/04/30 13:59
327	user2	2009/04/30 13:52
326	user1	2009/04/30 13:48
240	user1	2008/12/10 09:51
239	user1	2008/12/10 09:49

Figure 2-47 *Select two revisions*

6. Click the **Compare...** button.

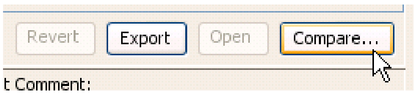
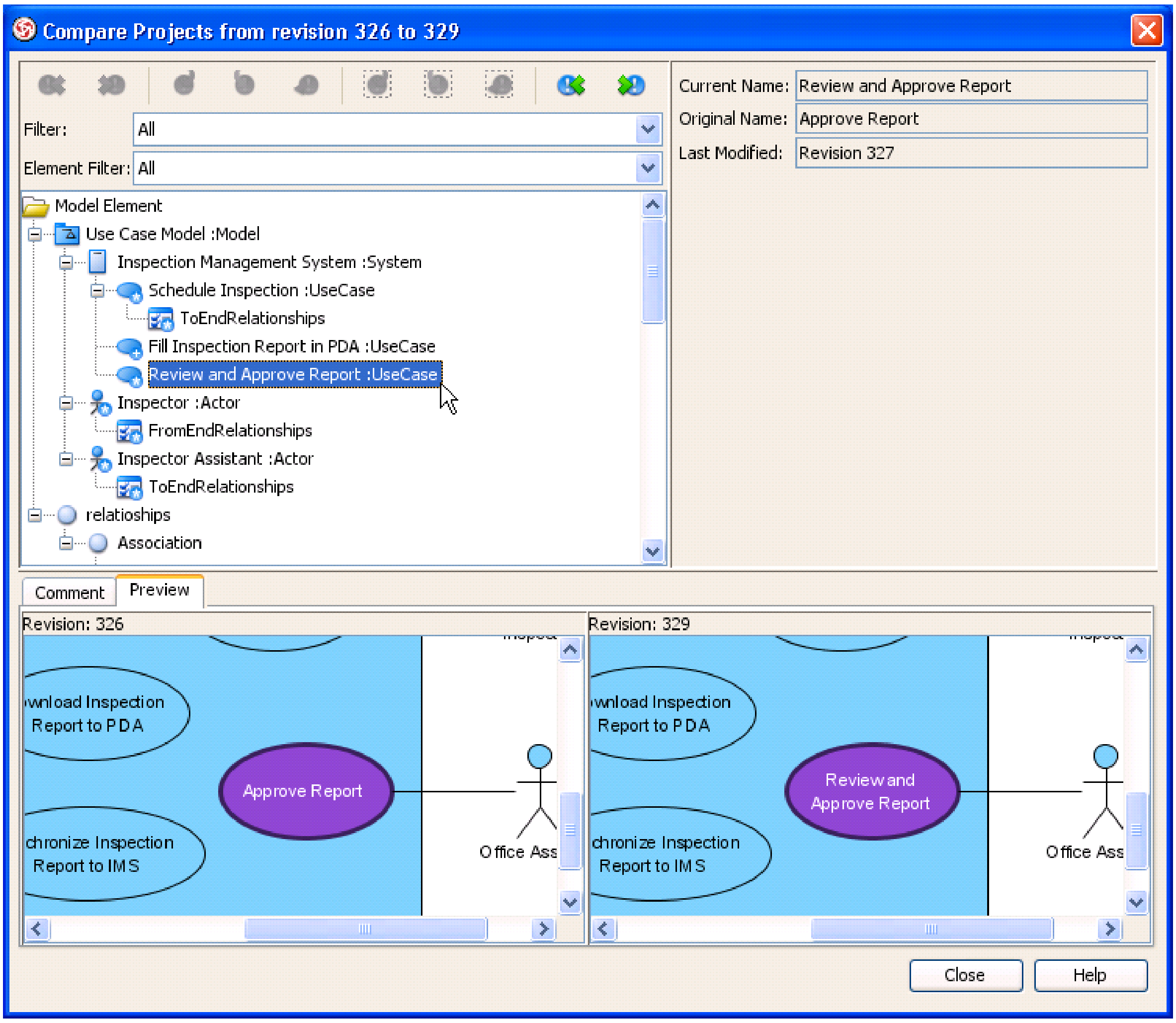


Figure 2-48 *Compare button*

7. Similar to **Commit** and **Update** dialog, the **Compare Projects** dialog show a list of differences between the selected revisions. You can also view the differences visually in diagram on the preview tab.



Compare Projects from revision 326 to 329

Filter: All

Element Filter: All

Current Name: Review and Approve Report

Original Name: Approve Report

Last Modified: Revision 327

Model Element
- Use Case Model :Model
  - Inspection Management System :System
    - Schedule Inspection :UseCase
      - ToEndRelationships
    - Fill Inspection Report in PDA :UseCase
    - Review and Approve Report :UseCase
  - Inspector :Actor
    - FromEndRelationships
  - Inspector Assistant :Actor
    - ToEndRelationships
- relatioships
  - Association

Revision: 326

Revision: 329

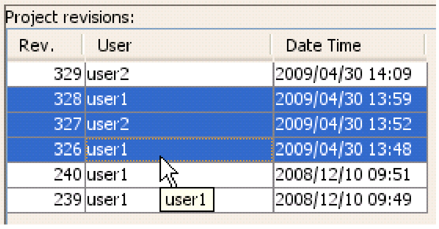
Comment
Preview
wnload Inspection Report to PDA
Approve Report
chronize Inspection Report to IMS
Office Ass
wnload Inspection Report to PDA
Review and Approve Report
chronize Inspection Report to IMS
Office Ass
Close
Help

Figure 2-49 *Compare differences*

Export multiple revisions to local

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.

4. Select multiple revisions for export, by **Ctrl+Click** or **Shift+Click**.



Project revisions:

Rev.	User	Date Time
329	user2	2009/04/30 14:09
328	user1	2009/04/30 13:59
327	user2	2009/04/30 13:52
326	user1	2009/04/30 13:48
240	user1	2008/12/10 09:51
239	user1	2008/12/10 09:49

Figure 2-50 *Select multiple revisions*

5. Click the **Export** button.

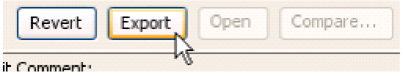
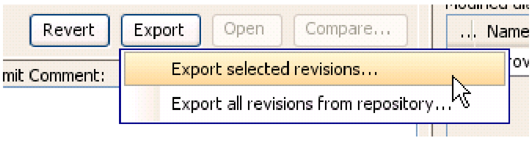


Figure 2-51 *Export buttons*

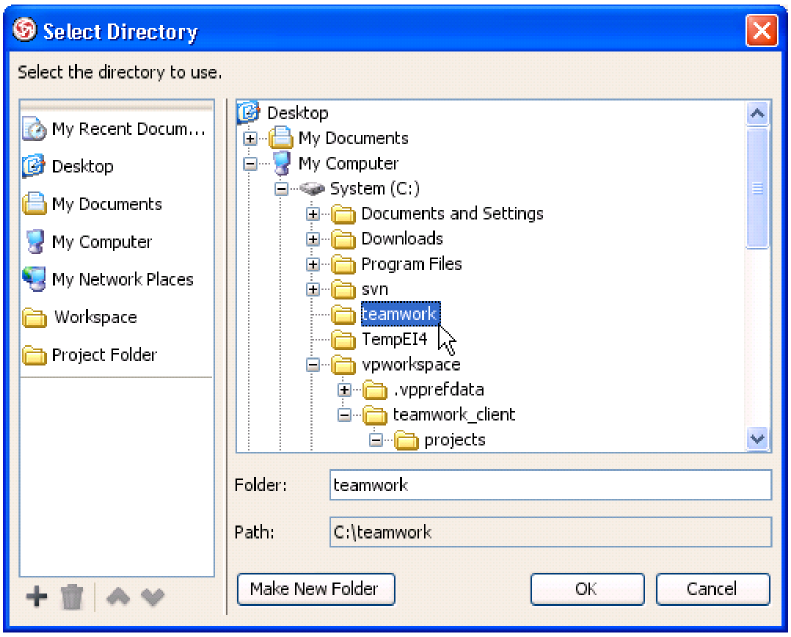
6. Select **Export selected revisions...** from the popup.



Revert
Export
Open
Compare...
... Name
mit Comment:
Export selected revisions...
Export all revisions from repository...

Figure 2-52 *Export selected revisions*

7. Select the directory to save the exported project revisions.

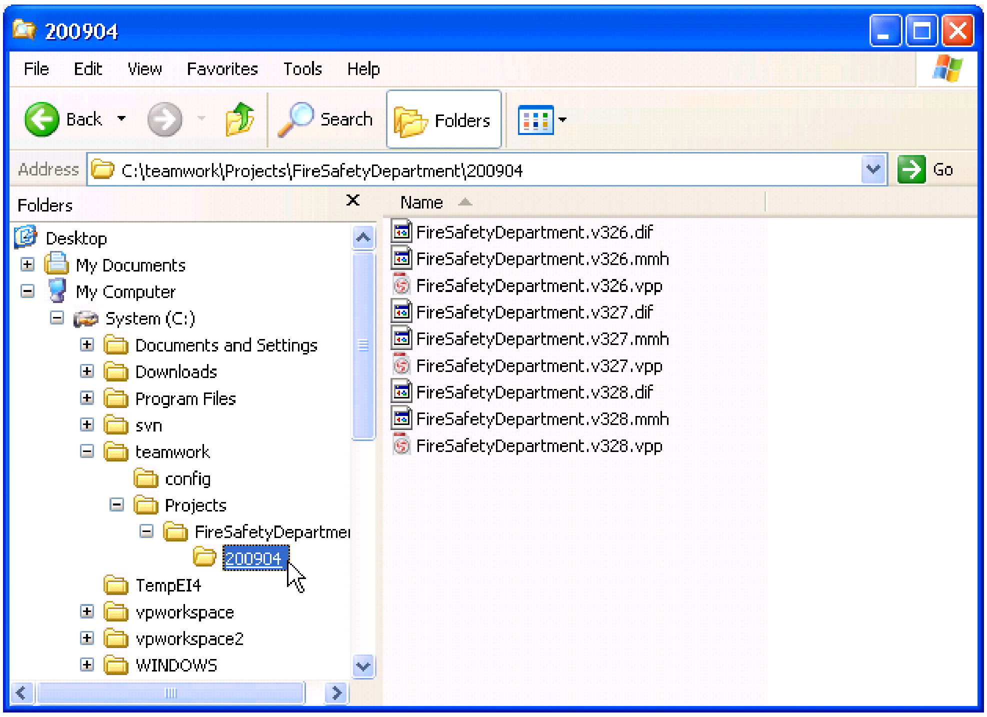


Folder: teamwork

Path: C:\teamwork

Figure 2-53 *Select directory*

8. You'll found the selected revisions was exported to the select directory.



FireSafetyDepartment.v326.dif
FireSafetyDepartment.v326.mmh
FireSafetyDepartment.v326.vpp
FireSafetyDepartment.v327.dif
FireSafetyDepartment.v327.mmh
FireSafetyDepartment.v327.vpp
FireSafetyDepartment.v328.dif
FireSafetyDepartment.v328.mmh
FireSafetyDepartment.v328.vpp

Figure 2-54 *Exported revisions*

Isolating last long modifications with Branches

Creating Branch

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Branch...** from menu

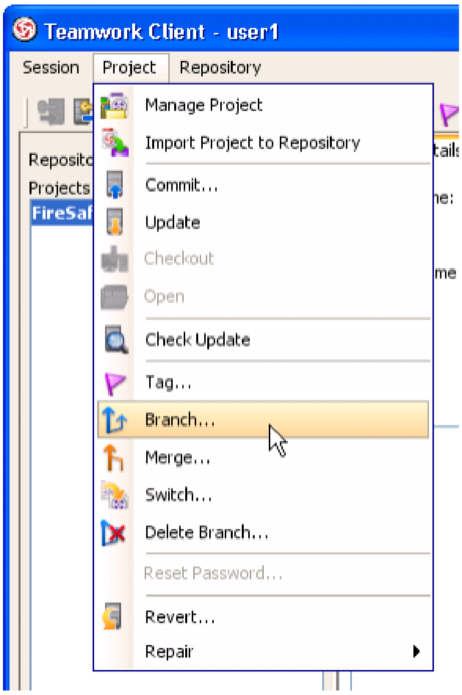


Figure 2-55 Branch from menu

Or click **Branch...** button from toolbar.

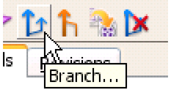
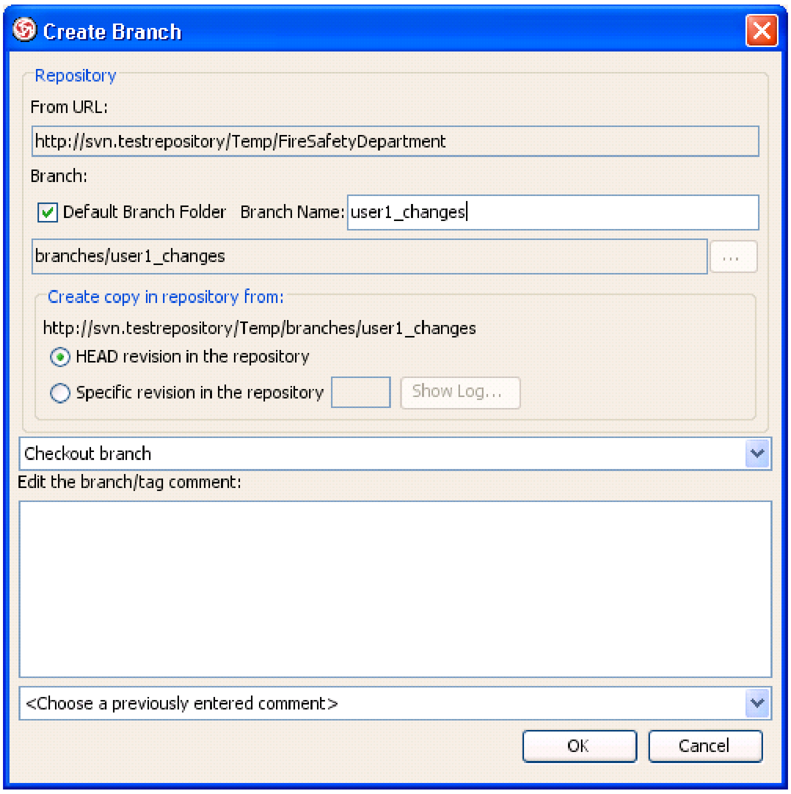


Figure 2-56 Branch from toolbar

4. Fill in the **Branch Name** in **Create Branch** dialog.



Create Branch

Repository

From URL:

http://svn.testrepository/Temp/FireSafetyDepartment

Branch:

☑ Default Branch Folder Branch Name: user1_changes

branches/user1_changes

Create copy in repository from:

http://svn.testrepository/Temp/branches/user1_changes

● HEAD revision in the repository

○ Specific revision in the repository

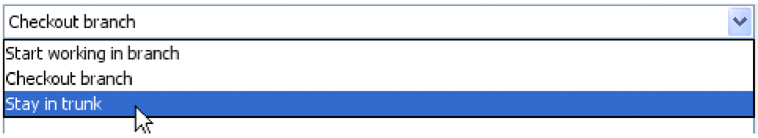
Checkout branch

Edit the branch/tag comment:

<Choose a previously entered comment>

Figure 2-57 *Create Branch dialog*

5. Select **Stay in trunk** in the list.

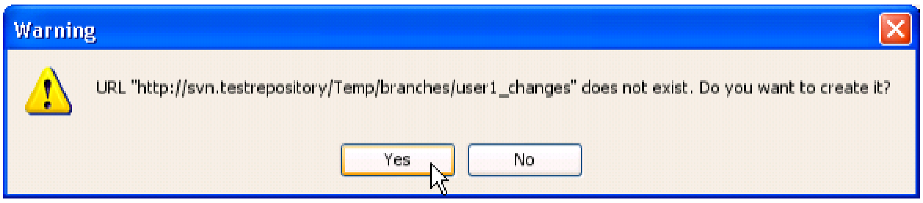


Checkout branch

Start working in branch
Checkout branch
Stay in trunk

Figure 2-58 *Stay in trunk*

6. If the folder is not exists in the server, it will display a warning dialog for asking create or not.



Warning

URL "http://svn.testrepository/Temp/branches/user1_changes" does not exist. Do you want to create it?

Figure 2-59 *Create folder warning*

Switch local copy between Branches

1. Select **Project > Switch...** from menu

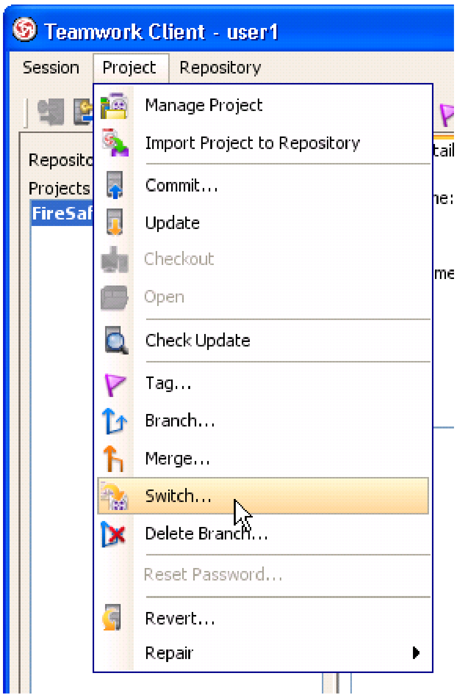


Figure 2-60 *Switch from menu*

Or click **Switch...** button from toolbar.

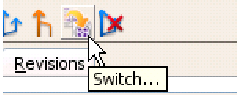
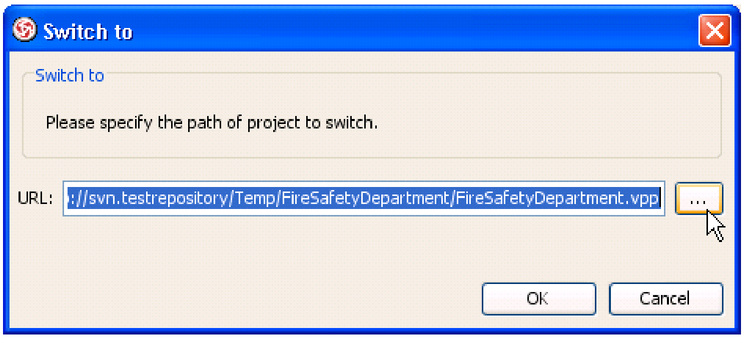


Figure 2-61 *Switch from toolbar*

2. Click **...** to show the server hierarchy.



Please specify the path of project to switch.

URL: ://svn.testrepository/Temp/FireSafetyDepartment/FireSafetyDepartment.vpp

Figure 2-62 *Select branch*

3. Select branch path **branches/user1_changes/FireSafetyDepartment/FireSafetyDepartment.vpp**.

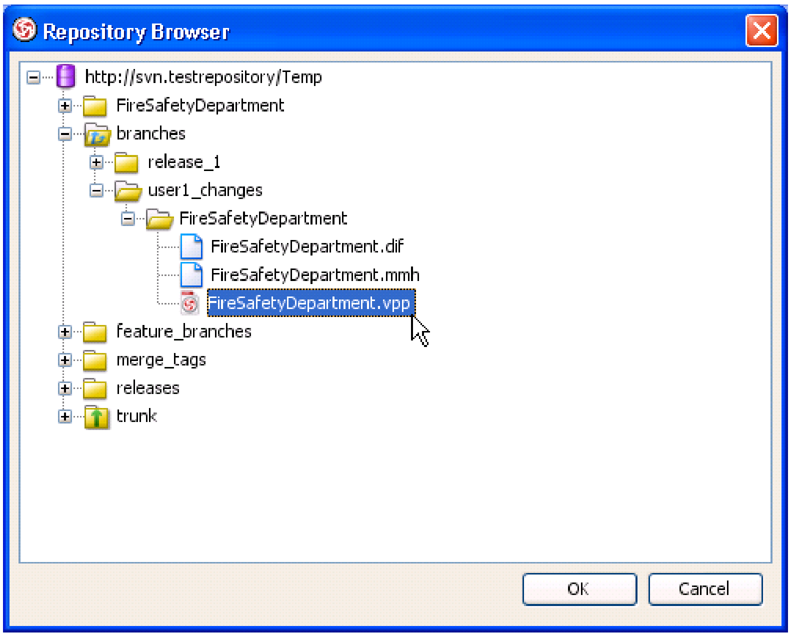
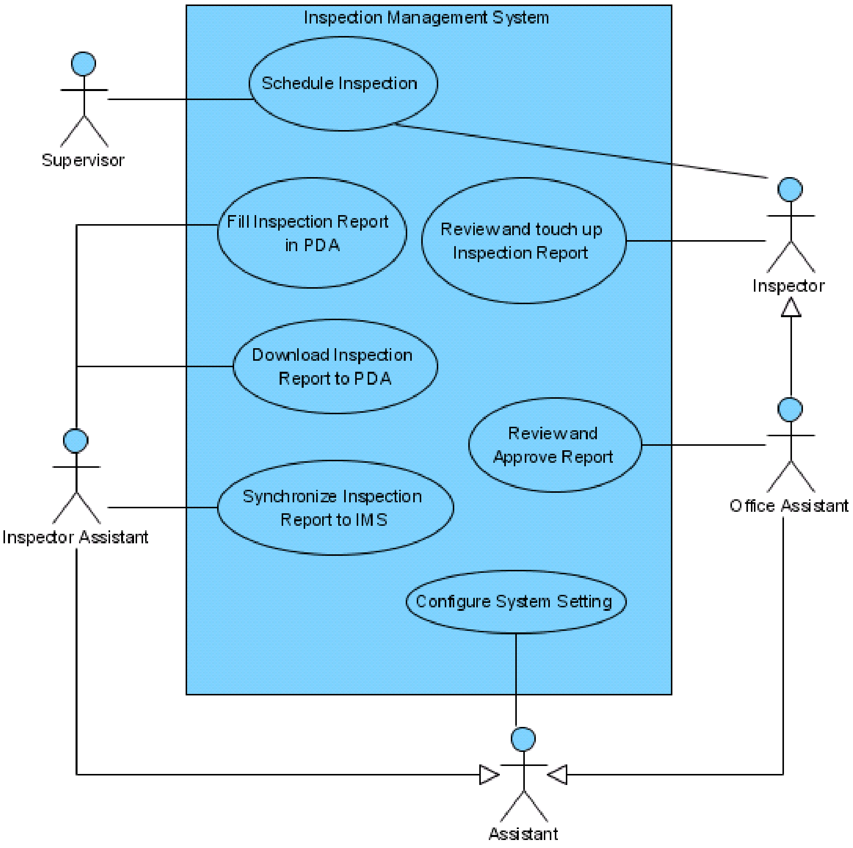


Figure 2-63 *Select branch path*

4. Press **OK** to open.
5. The branch is opened in **VP-UML**.

Merging from Trunk To Branch

1. Open, modify and commit trunk project.



Inspection Management System
Schedule Inspection
Supervisor
Fill Inspection Report in PDA
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Review and Approve Report
Office Assistant
Inspector Assistant
Synchronize Inspection Report to IMS
Configure System Setting
Assistant

Figure 2-64 *Modified trunk project*

2. Open branch project.

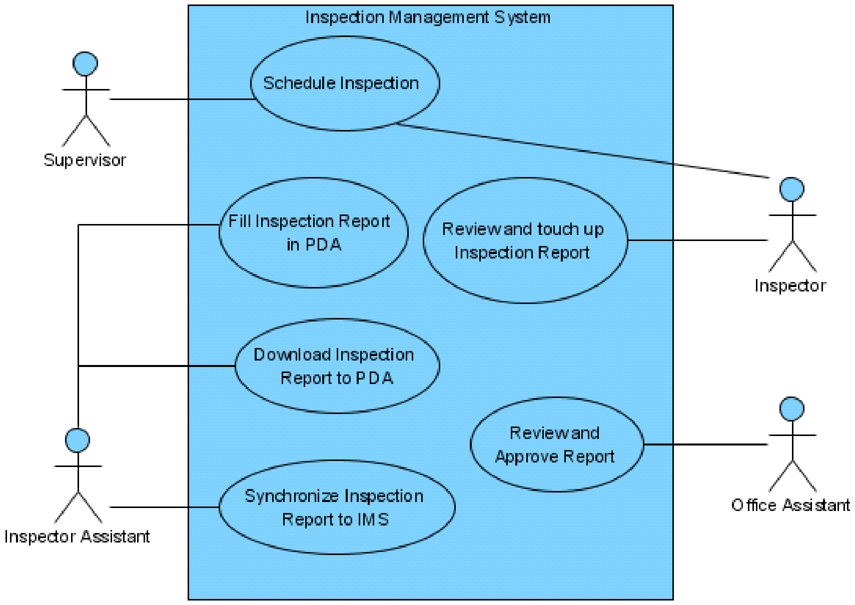

Inspection Management System
Schedule Inspection
Supervisor
Fill Inspection Report in PDA
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Review and Approve Report
Office Assistant
Synchronize Inspection Report to IMS
Inspector Assistant

***Figure 2-65** Branch project*

3. Select **Project > Merge...** from menu.

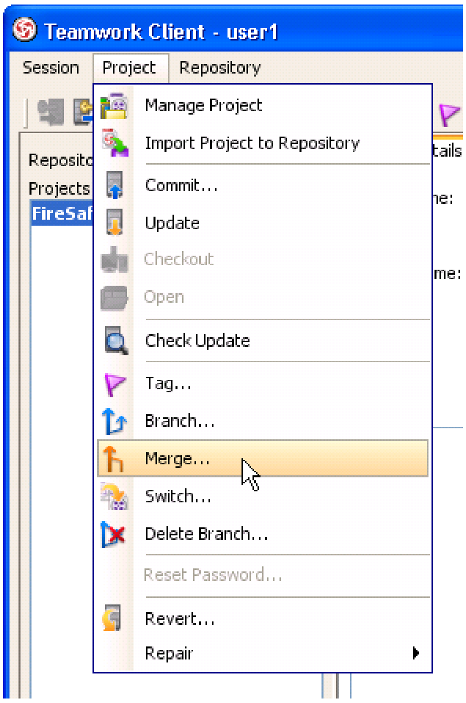


***Figure 2-66** Merge from menu*

Or click **Merge...** button from toolbar.

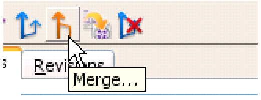
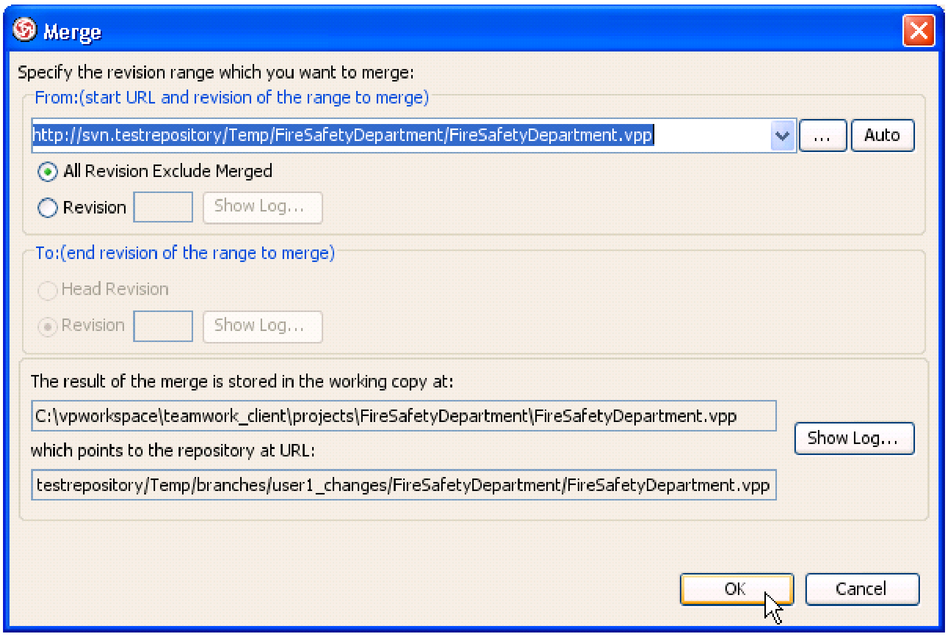


***Figure 2-67** Merge from toolbar*

4. Select the trunk path(*http://svn.testrepository/Temp/FireSafetyDepartment/FireSafetyDepartment.vpp*) as **From** in the **Merge** dialog, and press **OK**.



Merge

Specify the revision range which you want to merge:

From:(start URL and revision of the range to merge)

http://svn.testrepository/Temp/FireSafetyDepartment/FireSafetyDepartment.vpp

● All Revision Exclude Merged

○ Revision

To:(end revision of the range to merge)

○ Head Revision

● Revision

The result of the merge is stored in the working copy at:

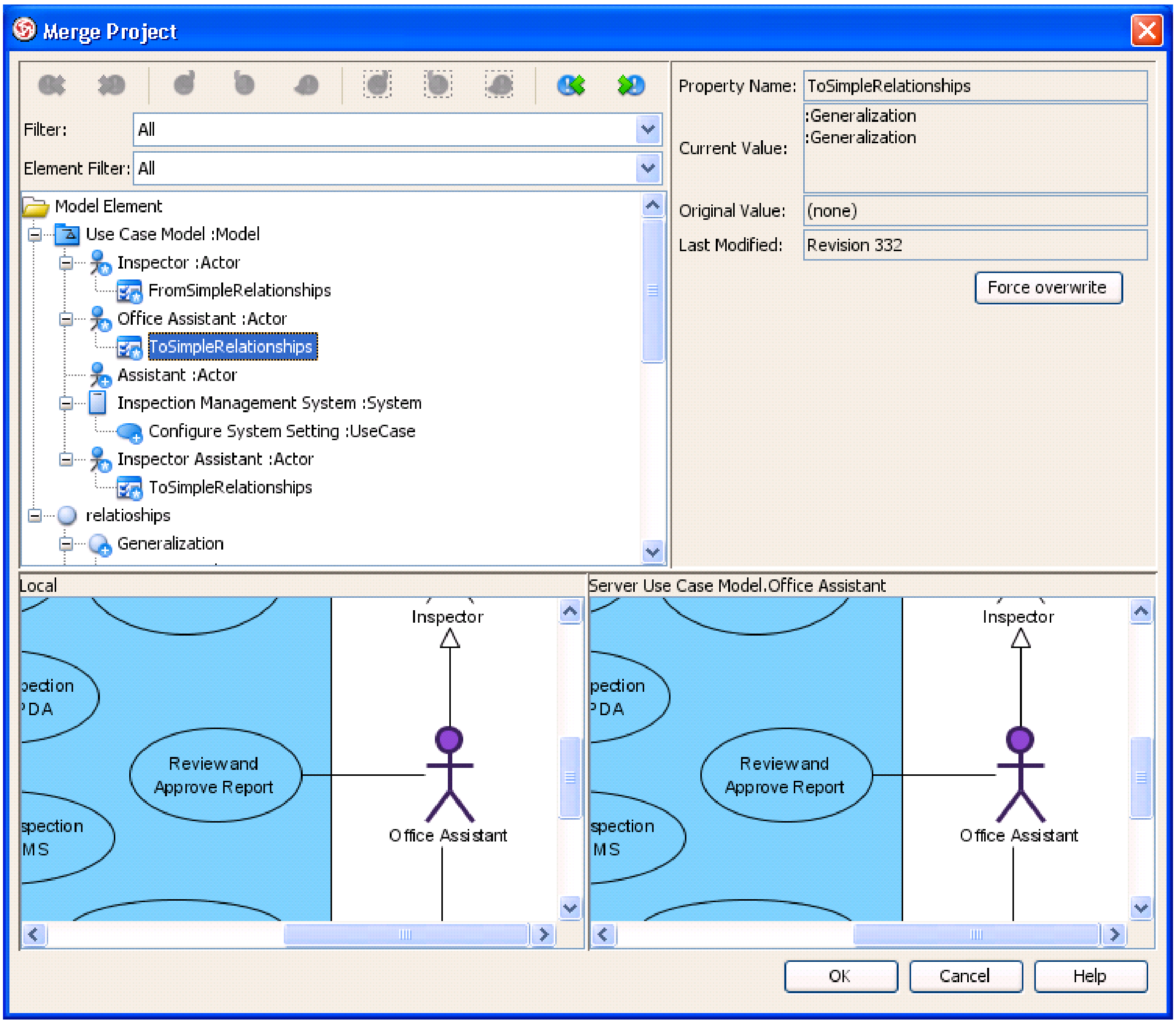
C:\vpworkspace\teamwork_client\projects\FireSafetyDepartment\FireSafetyDepartment.vpp

which points to the repository at URL:

testrepository/Temp/branches/user1_changes/FireSafetyDepartment/FireSafetyDepartment.vpp

Figure 2-68 *Merge dialog*

5. The **Merge Project** dialog show a list of changes similar to **Commit** and **Update** dialog.



Merge Project

Filter: All

Element Filter: All

Property Name: ToSimpleRelationships

Current Value: :Generalization
:Generalization

Original Value: (none)

Last Modified: Revision 332

- Model Element
  - Use Case Model :Model
    - Inspector :Actor
      - FromSimpleRelationships
    - Office Assistant :Actor
      - ToSimpleRelationships
    - Assistant :Actor
    - Inspection Management System :System
      - Configure System Setting :UseCase
    - Inspector Assistant :Actor
      - ToSimpleRelationships
  - relatioships
    - Generalization

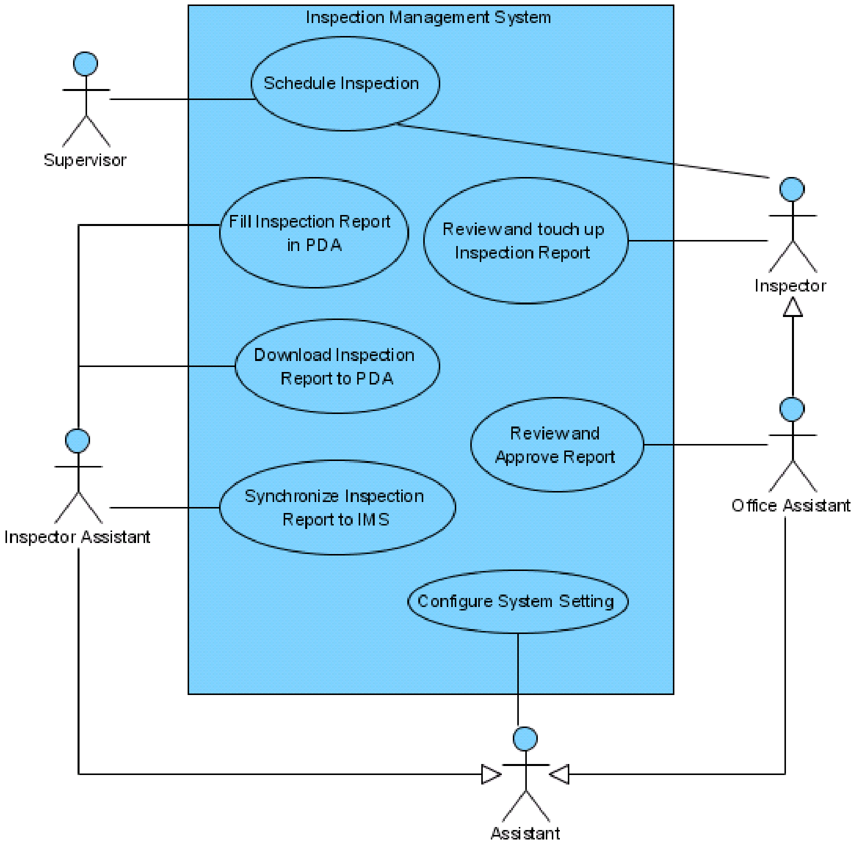
Local

Server Use Case Model.Office Assistant

Figure 2-69 *Merge project dialog*

6. Finally, review the changes in **Commit** dialog. Input comment and click **OK** to commit.

7. The branch project now contains the changes from trunk.



Inspection Management System
Schedule Inspection
Supervisor
Fill Inspection Report in PDA
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Review and Approve Report
Office Assistant
Synchronize Inspection Report to IMS
Inspector Assistant
Configure System Setting
Assistant

Figure 2-70 *Branch project merged from trunk*

Merging from Branch to Trunk

1. Open, modify and commit branch project.

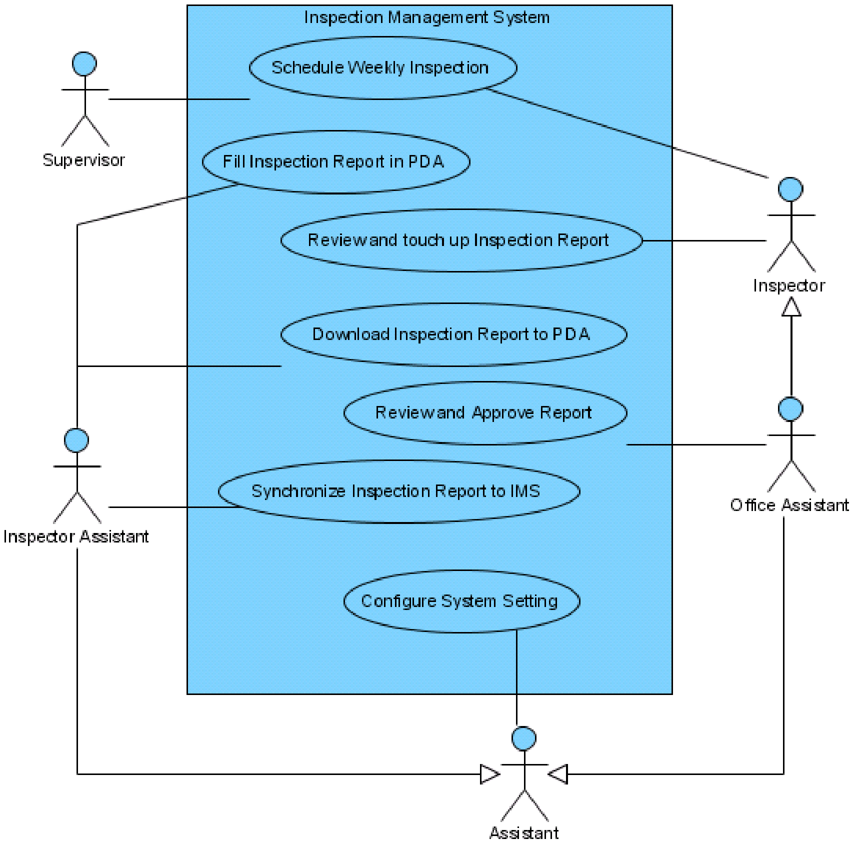

Inspection Management System
Schedule Weekly Inspection
Fill Inspection Report in PDA
Review and touch up Inspection Report
Download Inspection Report to PDA
Review and Approve Report
Synchronize Inspection Report to IMS
Configure System Setting
Supervisor
Inspector
Office Assistant
Inspector Assistant
Assistant

Figure 2-71 *Modified branch project*

2. Open trunk project.

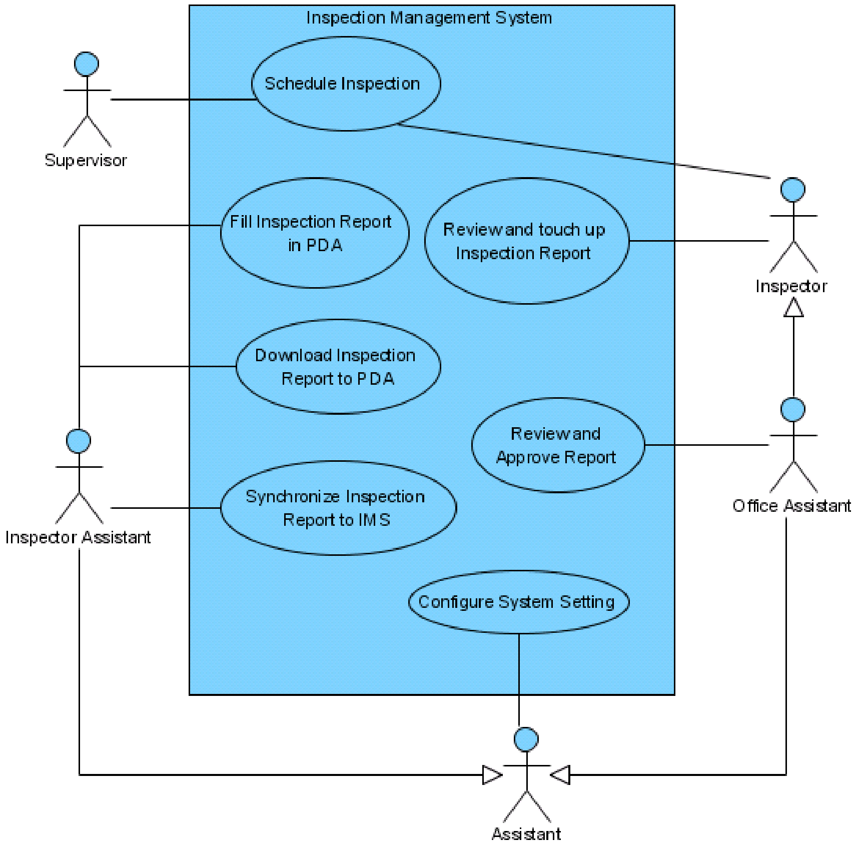

Inspection Management System
Schedule Inspection
Supervisor
Fill Inspection Report in PDA
Review and touch up Inspection Report
Inspector
Download Inspection Report to PDA
Review and Approve Report
Office Assistant
Synchronize Inspection Report to IMS
Inspector Assistant
Configure System Setting
Assistant

Figure 2-72 *Trunk project*

3. Select **Project > Merge...** from menu, or click **Merge...** button from toolbar.
4. Select the branch path(*http://svn.testrepository/Temp/branches/user1_changes/FireSafetyDepartment/FireSafetyDepartment.vpp*) as **From** and press **OK**.
5. The **Merge Project** dialog show a list of changes similar to **Commit** and **Update** dialog.
6. Finally, review the changes in **Commit** dialog. Input comment and click **OK** to commit.

7. The trunk project now contains the changes from branch.

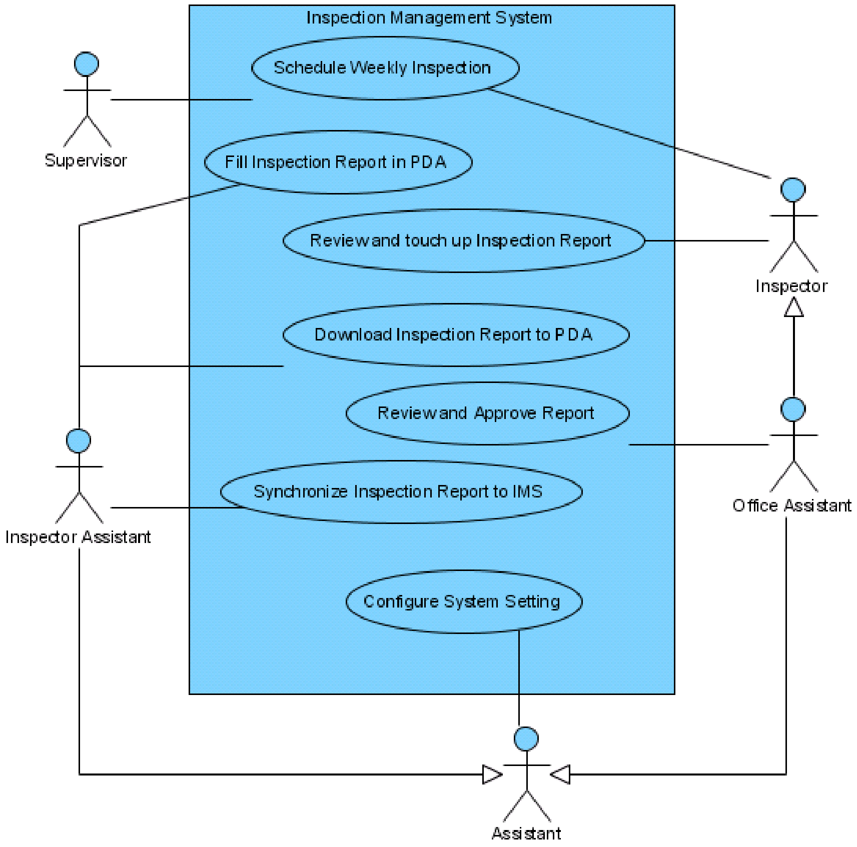

Inspection Management System
Schedule Weekly Inspection
Fill Inspection Report in PDA
Reviewand touch up Inspection Report
Download Inspection Report to PDA
Reviewand Approve Report
Synchronize Inspection Report to IMS
Configure System Setting
Supervisor
Inspector
Office Assistant
Inspector Assistant
Assistant

Figure 2-73 *Trunk project merged from branch*

Delete Branch

1. Select **Project > Delete Branch...** from menu

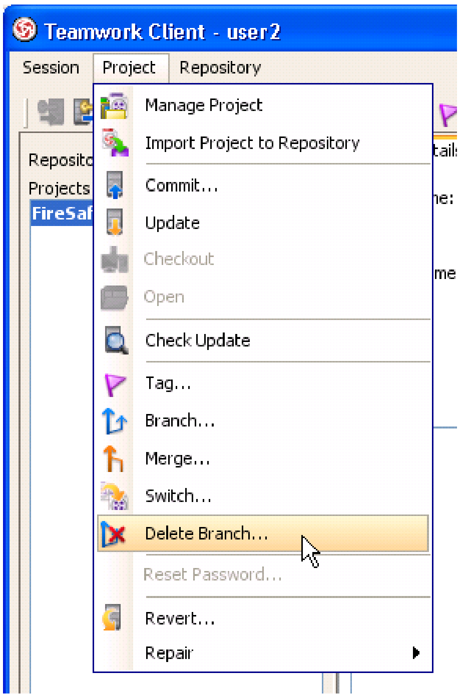


Figure 2-74 *Delete Branch from menu*

Or click **Delete Branch ...** button from toolbar.

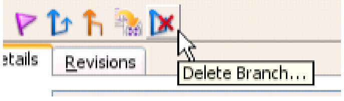
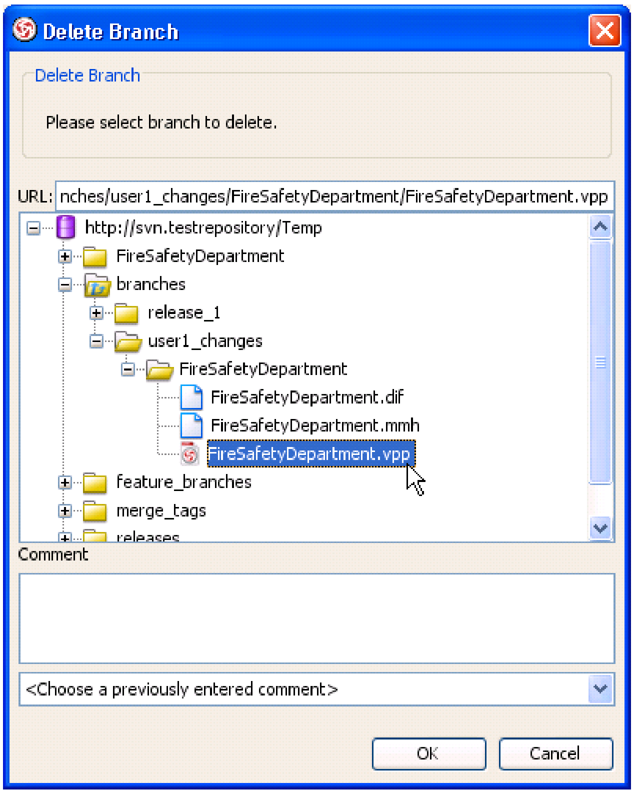


Figure 1-75 *Delete Branch from toolbar*

2. Expand the repository and folder node, select the branch to delete. Click **OK** button to continue.


Delete Branch

Delete Branch

Please select branch to delete.

URL: nches/user1_changes/FireSafetyDepartment/FireSafetyDepartment.vpp

Figure 2-76 *Delete branch dialog*

3. Click **Yes** button on confirmation dialog.

Marking release or milestone with Tags

Tags and branches are almost the same, with the only difference - tags are read only. Creating read only tags ensure you are able open the release version project later.

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Tag...** from menu

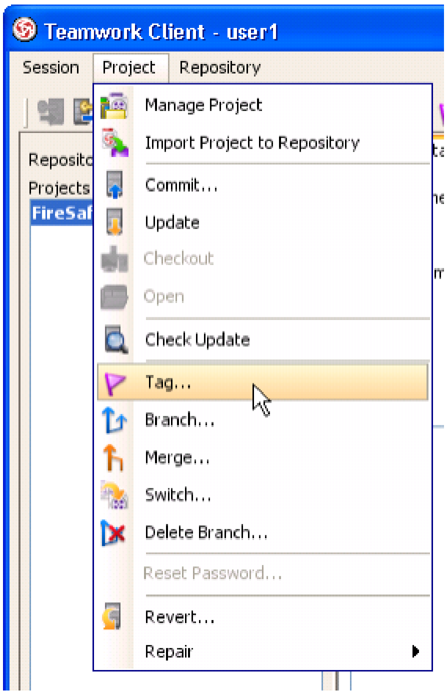


***Figure 2-77** Tag from menu*

Or click **Branch...** button from toolbar.

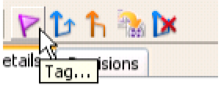
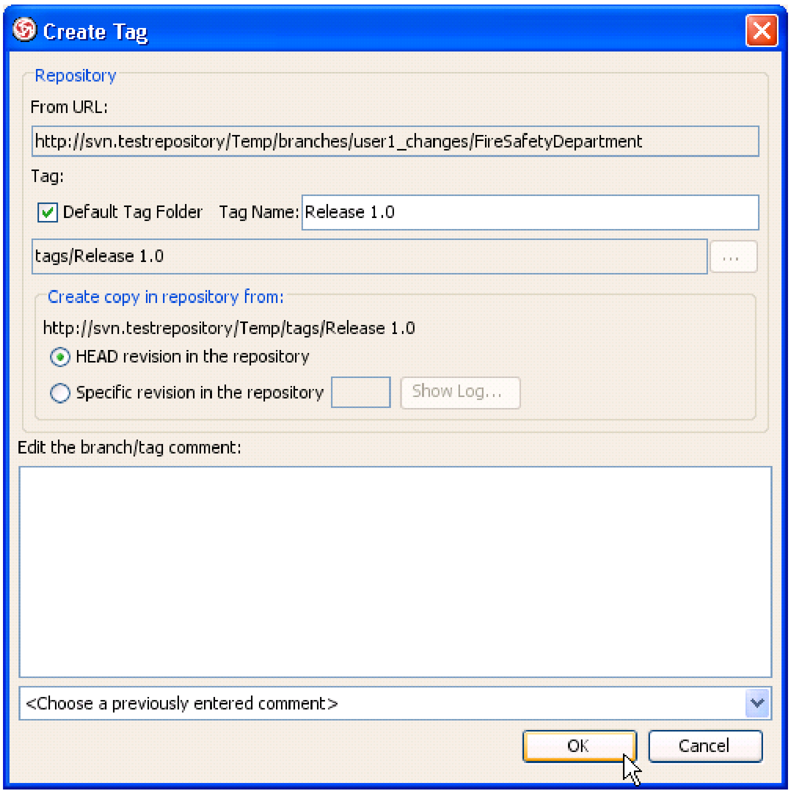


***Figure 2-78** Tag from toolbar*

4. Fill in the **Tag Name** in **Create Tag** dialog.



Create Tag

Repository

From URL:

http://svn.testrepository/Temp/branches/user1_changes/FireSafetyDepartment

Tag:

☑ Default Tag Folder Tag Name: Release 1.0

tags/Release 1.0

Create copy in repository from:

http://svn.testrepository/Temp/tags/Release 1.0

● HEAD revision in the repository

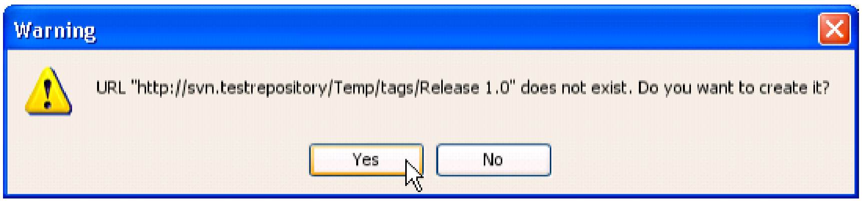
○ Specific revision in the repository

Edit the branch/tag comment:

<Choose a previously entered comment>

Figure 2-79 Create Tag dialog

5. Click **OK** button, the tag will be created.
6. If the path doesn't exists, it will prompt for ask create.



Warning

URL "http://svn.testrepository/Temp/tags/Release 1.0" does not exist. Do you want to create it?

Figure 2-80 Create folder warning

Importing Projects to CVS Server

1. Start **VP-UML**.
2. Select **Tools > Teamwork > Open Teamwork Client...** from the main menu, or **Open Teamwork Client** icon from toolbar to open teamwork client.

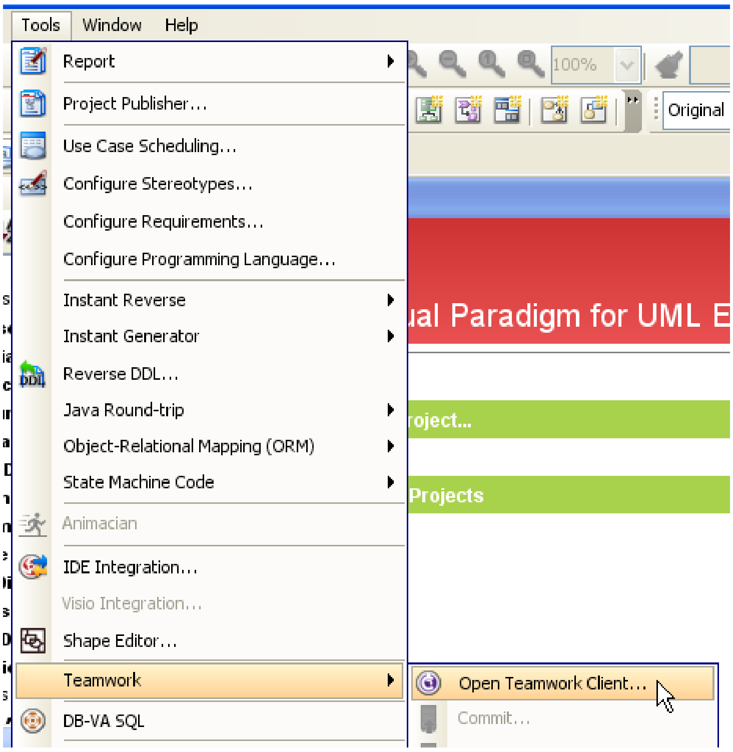
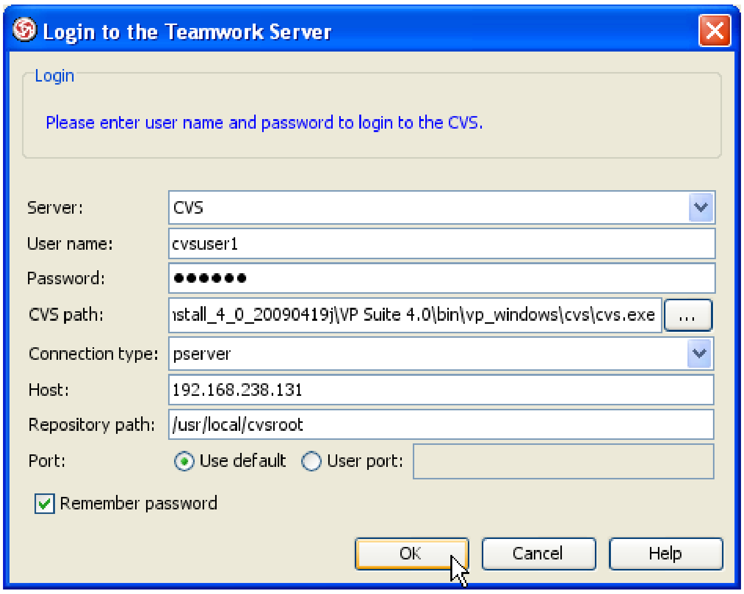


Figure 3-1 Open Teamwork Client from main menu

3. Select CVS Server, fill in the server and user information to login cvs server.



Login to the Teamwork Server

Login

Please enter user name and password to login to the CVS.

Server: CVS

User name: cvsuser1

Password: ●●●●●●

CVS path: ıstall_4_0_20090419j\VP Suite 4.0\bin\vp_windows\cvs\cvs.exe ...

Connection type: pserver

Host: 192.168.238.131

Repository path: /usr/local/cvsroot

Port: ● Use default ○ User port:

☑ Remember password

OK Cancel Help

Figure 3-2 Login cvs server

4. Select **Project > Import Project to Repository** from the menu.

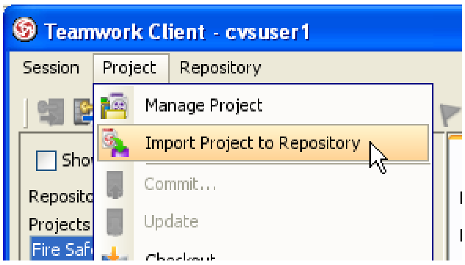


***Figure 3-3** Import project from menu*

Or click the **Import Project to Repository** button on toolbar.

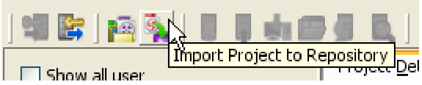
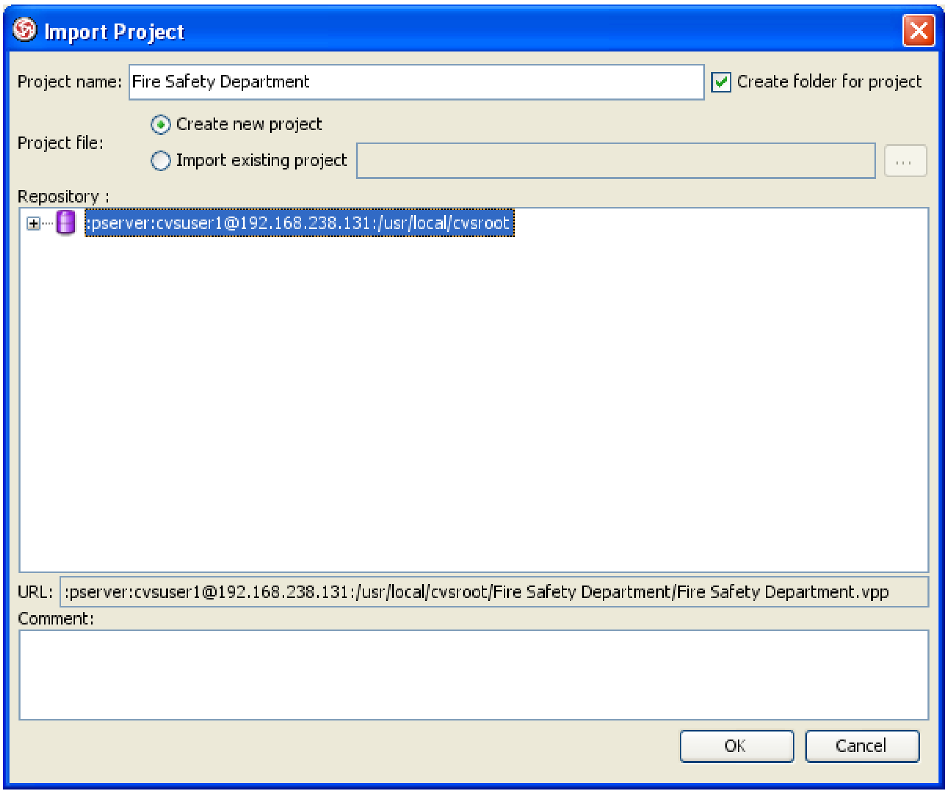


***Figure 3-4** Import project from toolbar*

5. In the **Import Project** dialog, fill in the **Project name**.



Import Project

Project name: Fire Safety Department ☑ Create folder for project

Project file: ● Create new project ○ Import existing project

Repository :

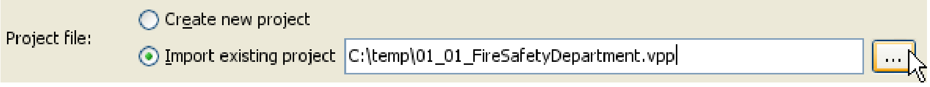
:pserver:cvsuser1@192.168.238.131:/usr/local/cvsroot

URL: :pserver:cvsuser1@192.168.238.131:/usr/local/cvsroot/Fire Safety Department/Fire Safety Department.vpp

Comment:

***Figure 3-5** Fill project name in Import Project dialog*

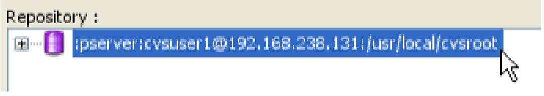
6. Choose **Create new project** or **Import existing project** from **Project file**. **Create new project** will import a blank project into teamwork server, **Import existing project** will import an existing VP-UML Project file (*.vpp) into teamwork server as first revision.



Project file: ○ Create new project ● Import existing project C:\temp\01_01_FireSafetyDepartment.vpp

***Figure 3-6** Specify project file*

7. Select the repository for import the project.

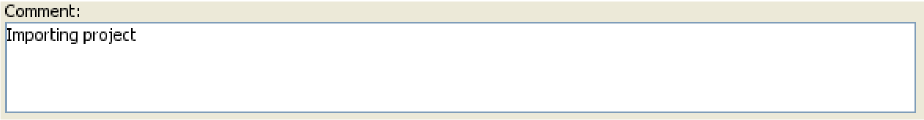


Repository :

:pserver:cvsuser1@192.168.238.131:/usr/local/cvsroot

***Figure 3-7** Select repository*

8. Fill in the comment.



Comment:

Importing project

__Figure 3-8__ Input comment

9. Click **OK** button to import project.

Checkout Project from CVS Server

Checkout and open CVS Server Projects

1. Open **Teamwork Client** dialog.
2. Select **Project > Manage Project** from the menu.

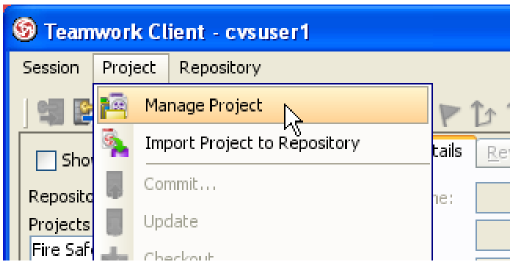


Figure 3-9 *Manage project from main menu*

Or click **Manage Project** icon on the toolbar.

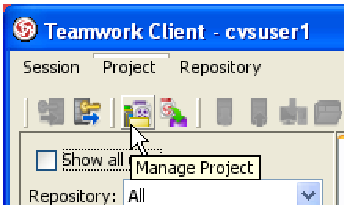


Figure 3-10 *Manage project from toolbar*

3. In **Manage Project** dialog, expand the repository node and select the project to checkout, click **>** button to add project. Click **OK** button to close the dialog.

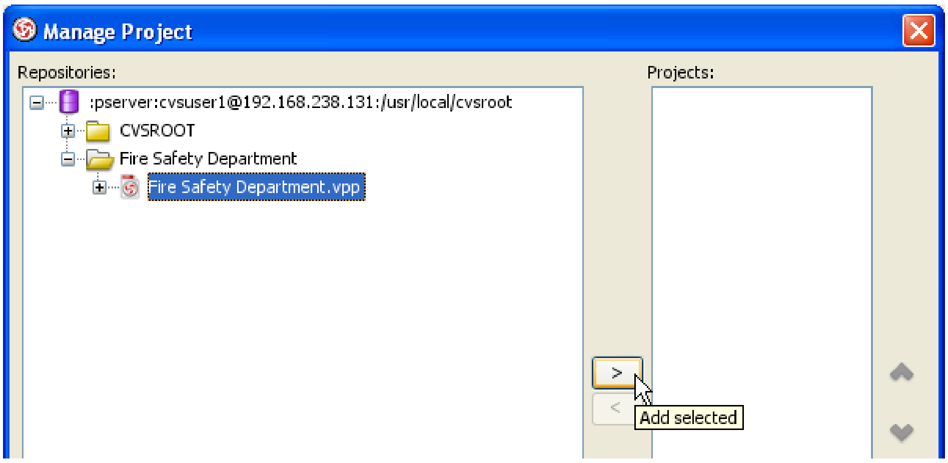


Figure 3-11 *Manage projects*

4. Select the project in project list.

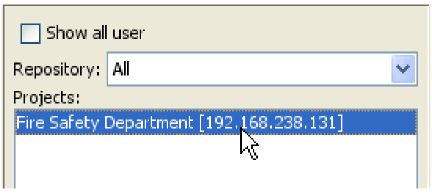


Figure 3-12 *Select project from the list*

5. Click **Open Project** button to checkout and open the project.

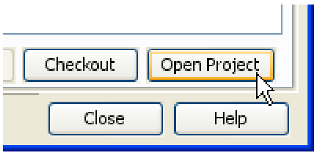


Figure 3-13 *Open project*

6. The project opened in **VP-UML**.

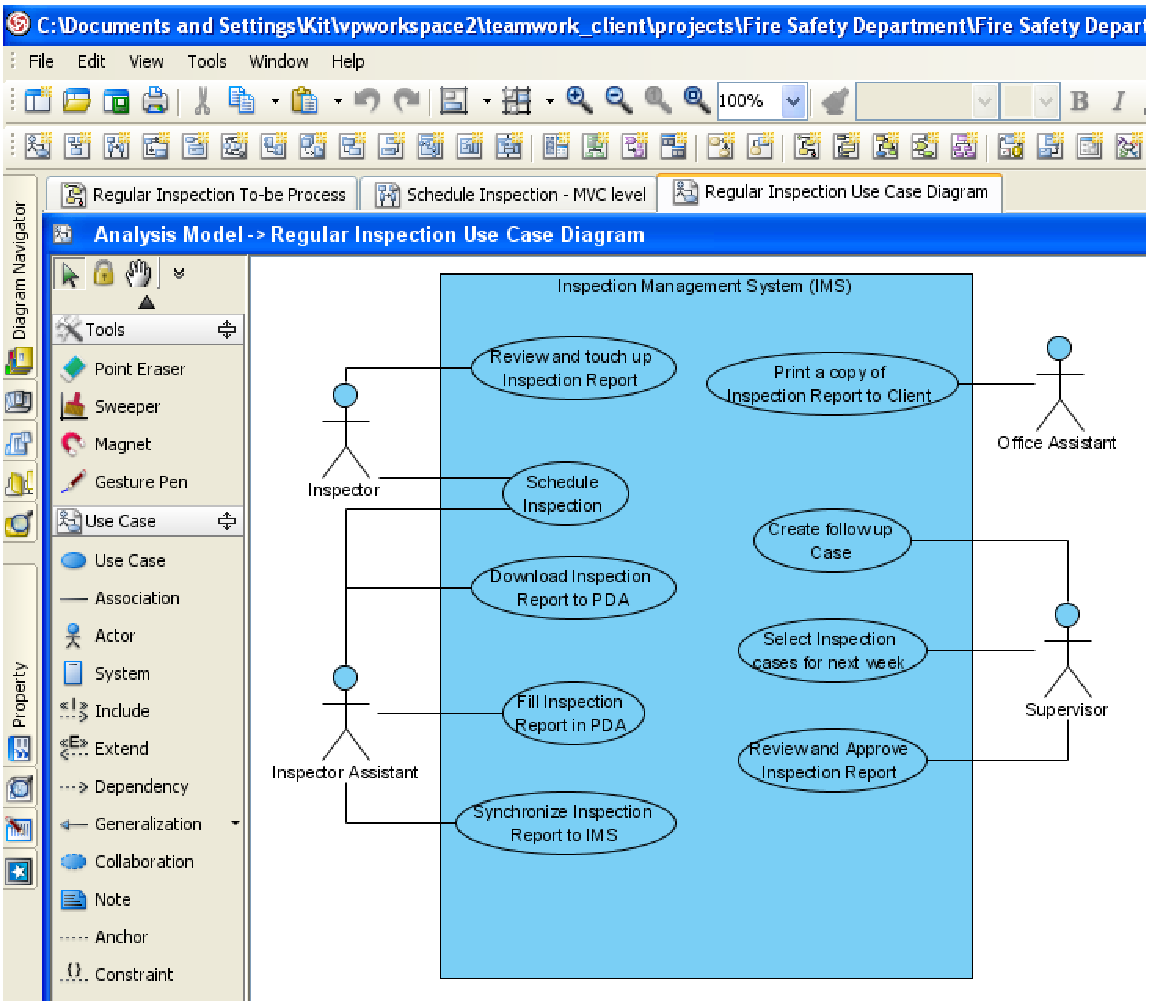

C:\Documents and Settings\Kit\vpworkspace2\teamwork_client\projects\Fire Safety Department\Fire Safety Depart
File
Edit
View
Tools
Window
Help
100%
Regular Inspection To-be Process
Schedule Inspection - MVC level
Regular Inspection Use Case Diagram
Analysis Model -> Regular Inspection Use Case Diagram
Diagram Navigator
Property
Tools
Point Eraser
Sweeper
Magnet
Gesture Pen
Use Case
Use Case
Association
Actor
System
Include
Extend
Dependency
Generalization
Collaboration
Note
Anchor
Constraint
Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS

Figure 3-14 Project opened

Committing local modification to CVS Server

1. Modify the project.

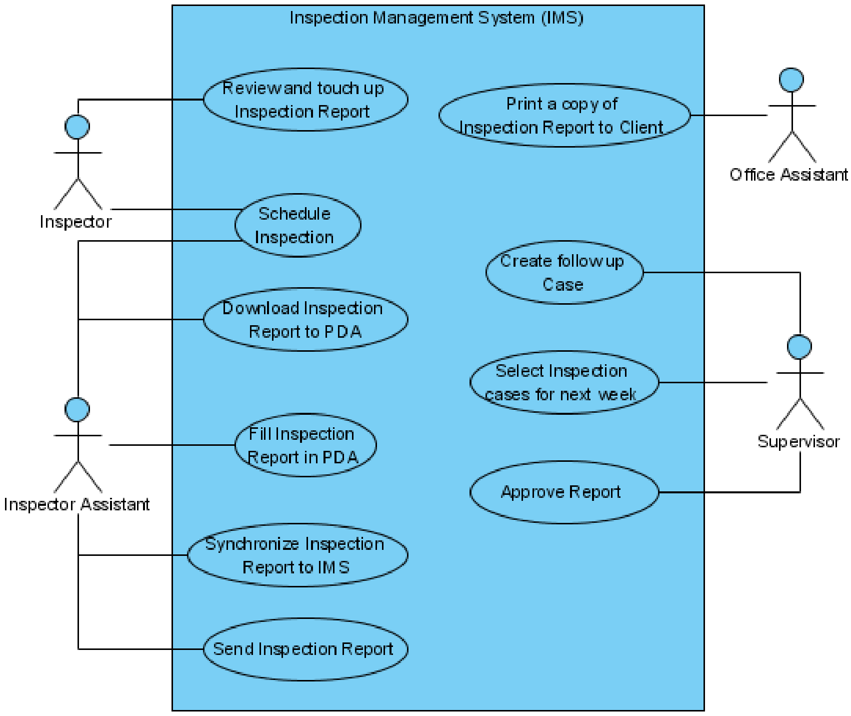

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Approve Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 3-15 *Modified project*

2. Show **Teamwork Toolbar** by right click on the Toolbar, select **Teamwork** if not selected already.

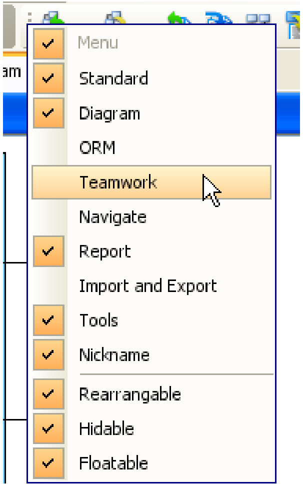


Figure 3-16 *Show teamwork toolbar*

3. Click the **Commit** button on **Teamwork Toolbar**.

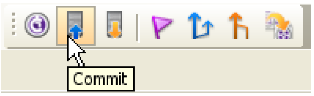
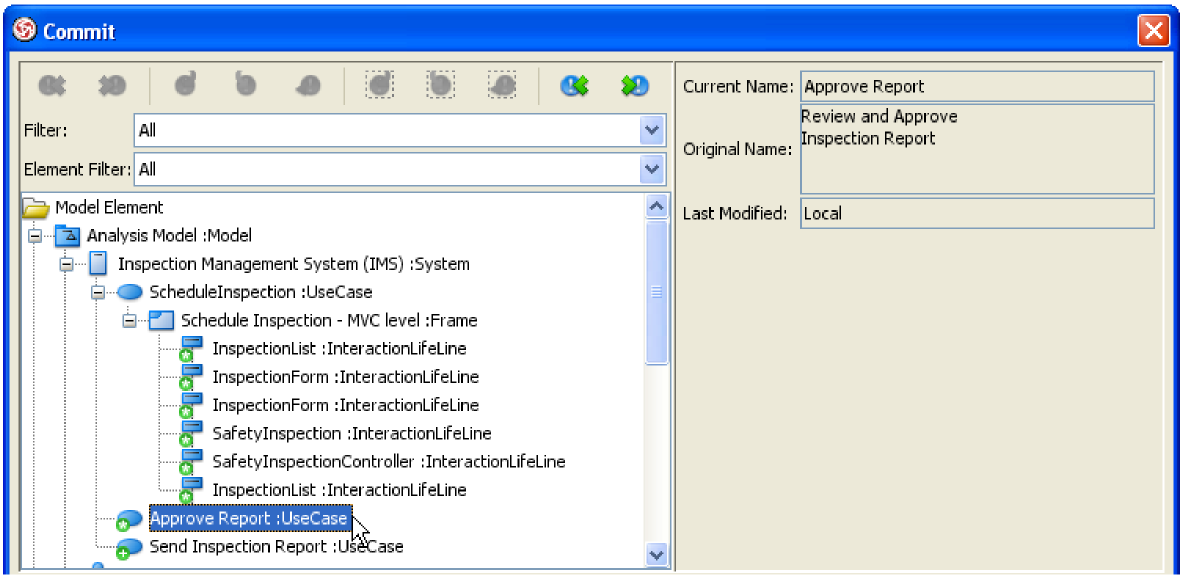


Figure 3-17 *Commit project*

4. In **Commit** dialog, you can review the changes for commit. On the left of dialog, you can see a list of changes shapes and model elements, click on it to view the detail changes on the right.



Commit

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionList :InteractionLifeLine
          - InspectionForm :InteractionLifeLine
          - InspectionForm :InteractionLifeLine
          - SafetyInspection :InteractionLifeLine
          - SafetyInspectionController :InteractionLifeLine
          - InspectionList :InteractionLifeLine
      - Approve Report :UseCase
      - Send Inspection Report :UseCase

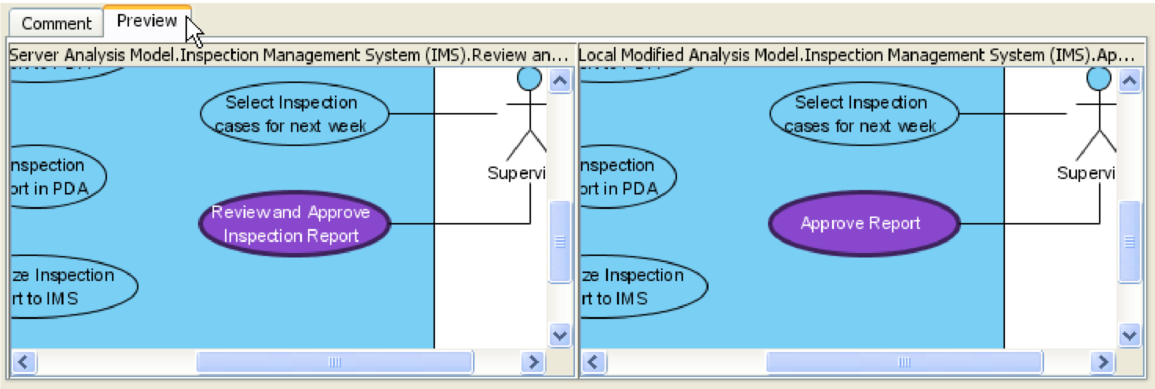
Current Name: Approve Report

Original Name: Review and Approve Inspection Report

Last Modified: Local

Figure 3-18 Review commit changes

5. On the bottom of dialog, click the **Preview** tab to visually preview the changes in diagram. The selected shape is highlighed in purple.



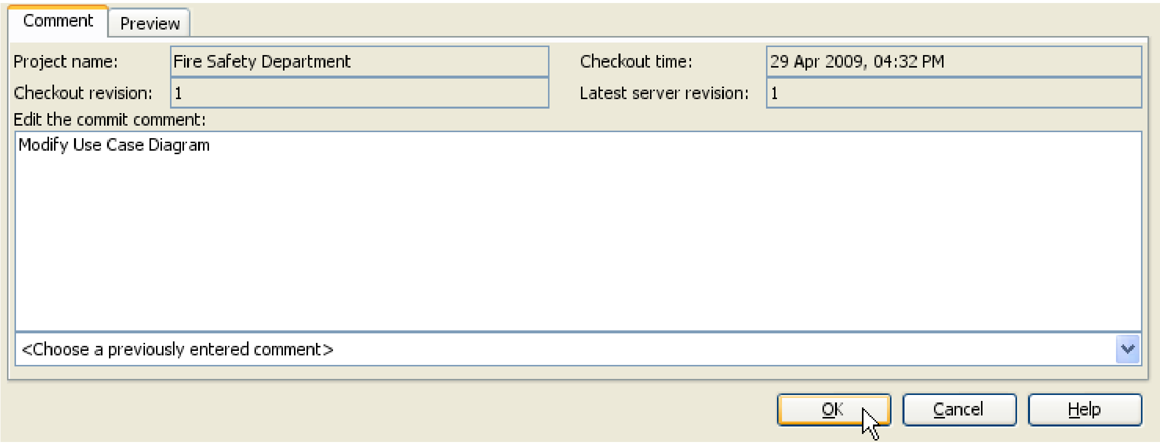
Comment | Preview

Server Analysis Model.Inspection Management System (IMS).Review an...

Local Modified Analysis Model.Inspection Management System (IMS).Ap...

Figure 3-19 Preview commit changes

6. After review the changes, click the **Comment** tab and input the comment for commit. Click **OK** button to start commit.



Comment | Preview

Project name: Fire Safety Department

Checkout time: 29 Apr 2009, 04:32 PM

Checkout revision: 1

Latest server revision: 1

Edit the commit comment:

Modify Use Case Diagram

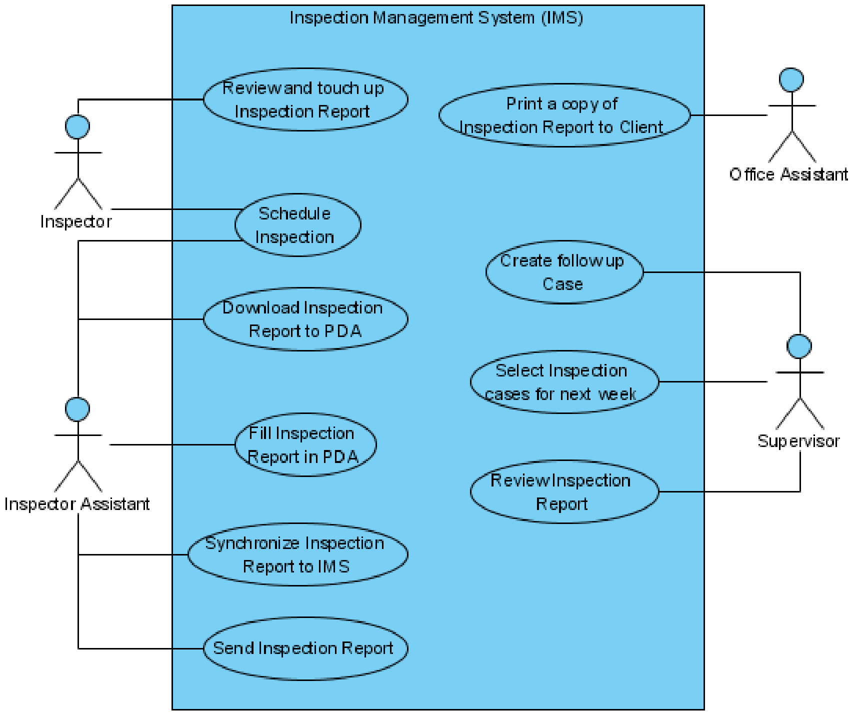
<Choose a previously entered comment>

OK | Cancel | Help

Figure 3-20 Input commit comment

Resolving conflicts

1. Modify the project.

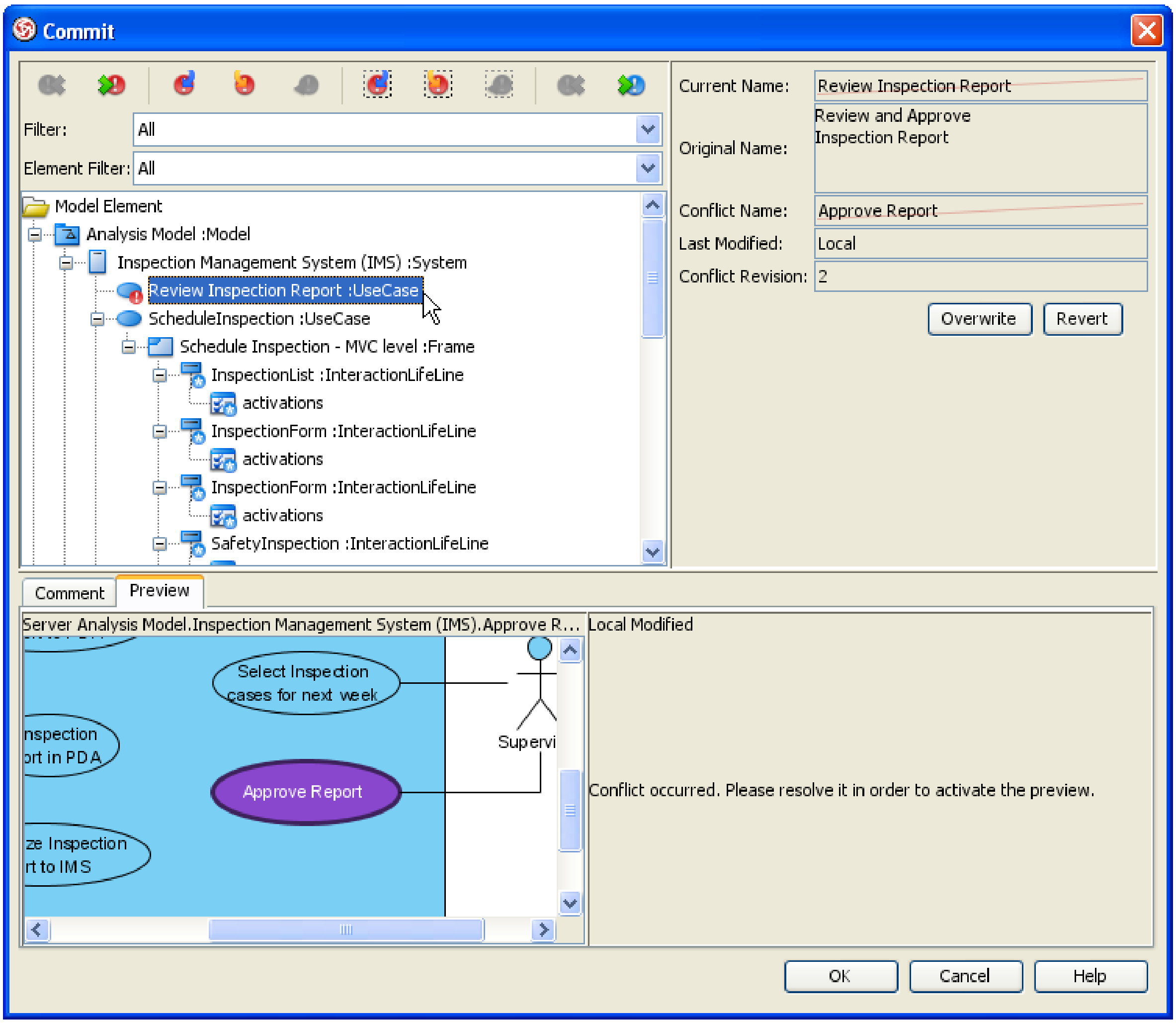


Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Review Inspection Report
Inspector Assistant
Synchronize Inspection Report to IMS
Send Inspection Report

__Figure 3-21__ Modified project

2. Click **Commit** button on the **Teamwork Toolbar**.

3. In **Commit** dialog, you may found some shapes or model elements show with red icon. This indicate there is conflict when commit, it is caused by someone modified the same content and commit after you checkout. You can review the current value, original value (the value when you checkout), and conflict value (the value changed by other users). And dialog preview is disabled until you resolve the conflict.



Commit

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Review Inspection Report :UseCase
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionList :InteractionLifeLine
            - activations
          - InspectionForm :InteractionLifeLine
            - activations
          - InspectionForm :InteractionLifeLine
            - activations
          - SafetyInspection :InteractionLifeLine

Current Name: ~~Review Inspection Report~~

Original Name: Review and Approve Inspection Report

Conflict Name: ~~Approve Report~~

Last Modified: Local

Conflict Revision: 2

Server Analysis Model.Inspection Management System (IMS).Approve R...

Local Modified

Conflict occurred. Please resolve it in order to activate the preview.

Figure 3-22 Commit with conflicts

4. Select the conflict element, click **Overwrite selected conflicts** button on the toolbar to overwrite other user's change.

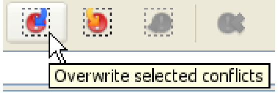


Figure 3-23 Overwrite selected conflicts

Or click **Revert selected conflicts** to revert your own changes.

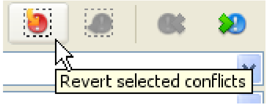


Figure 3-24 Revert selected conflicts

You can also click **Overwrite all conflicts** or **Revert all conflicts** to overwrite or revert all conflicts at once.

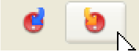


Figure 3-25 Overwrite/Revert all conflicts

5. After resolve conflict, the preview will be enabled to visualize the final result.

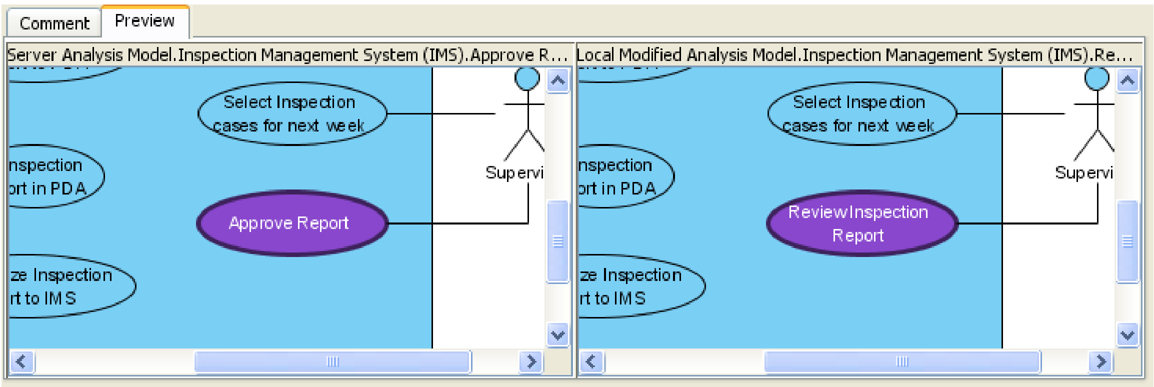

Comment
Preview
Server Analysis Model.Inspection Management System (IMS).Approve R...
Select Inspection cases for next week
nspection ort in PDA
Supervi
Approve Report
ze Inspection rt to IMS
Local Modified Analysis Model.Inspection Management System (IMS).Re...
Select Inspection cases for next week
nspection ort in PDA
Supervi
ReviewInspection Report
ze Inspection rt to IMS

Figure 3-26 *Preview resolved conflict*

6. You can change your mind by click **Reset selected conflicts** or **Reset all conflicts** button. Then overwrite or revert the conflict again.

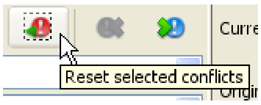


Figure 3-27 *Reset conflicts*

7. After all conflicts was resolved, you can now input comment and click **OK** button to commit.

Updating latest revision from CVS Server

Updating modification from CVS Server

1. Click **Update** button on **Teamwork Toolbar**.

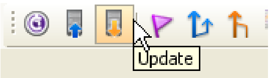
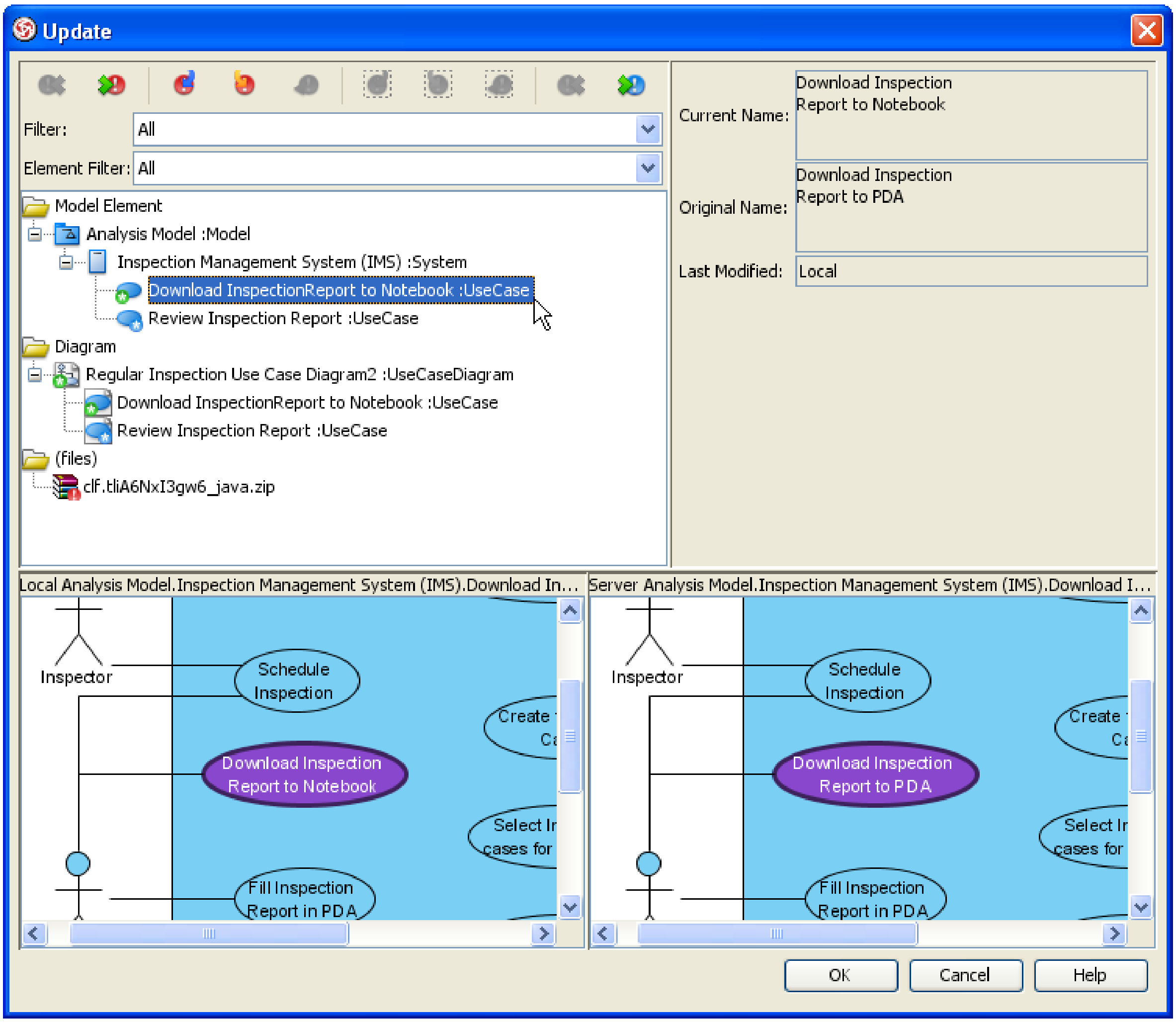


Figure 3-28 *Update from toolbar*

2. Similar to commit, you can review the change and preview the diagram in **Update** dialog.


Update

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Download InspectionReport to Notebook :UseCase
      - Review Inspection Report :UseCase
- Diagram
  - Regular Inspection Use Case Diagram2 :UseCaseDiagram
    - Download InspectionReport to Notebook :UseCase
    - Review Inspection Report :UseCase
- (files)
  - clf.tliA6NxI3gw6_java.zip

Current Name: Download Inspection Report to Notebook

Original Name: Download Inspection Report to PDA

Last Modified: Local

Local Analysis Model.Inspection Management System (IMS).Download In...

Server Analysis Model.Inspection Management System (IMS).Download I...

Inspector
Schedule Inspection
Create Ca
Download Inspection Report to Notebook
Select In cases for
Fill Inspection Report in PDA
Inspector
Schedule Inspection
Create Ca
Download Inspection Report to PDA
Select In cases for
Fill Inspection Report in PDA
OK
Cancel
Help

Figure 3-29 *Update dialog*

3. If there are conflicts, resolve it similar to commit, but the changes will apply to local project instead of commit to server immediately.

4. Click **OK** button to update. **VP-UML** will open the updated project.

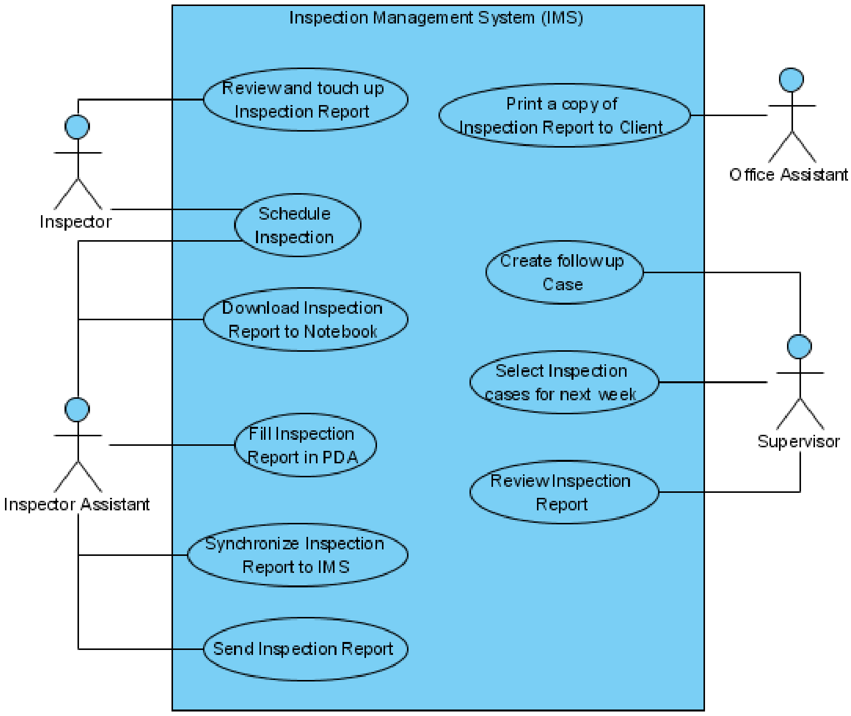

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to Notebook
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 3-30*Updated project*

Checking status

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Check Update** from menu.

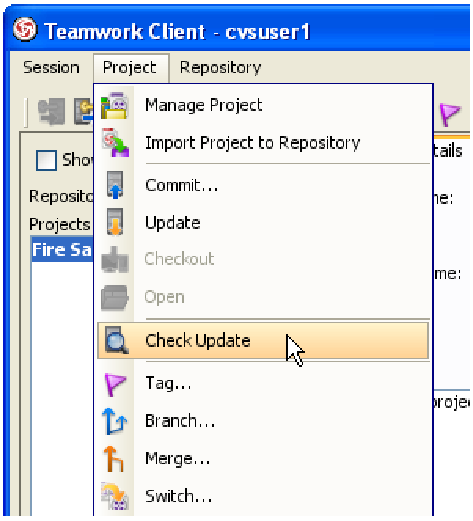


Figure 3-31 *Update project from menu*

Or click **Check Update** button from toolbar.

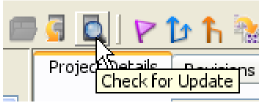
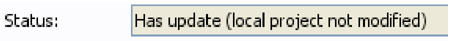


Figure 3-32 *Update project from toolbar*

4. You can see the status showing **Has update** or **Up-to-date**, it also indicate local project status **(local project not modified)** or **(local project modified)**.



Status: Has update (local project not modified)

Figure 3-33 *Project status*

Reverting Changes

Revert local modifications

1. Modify the project.

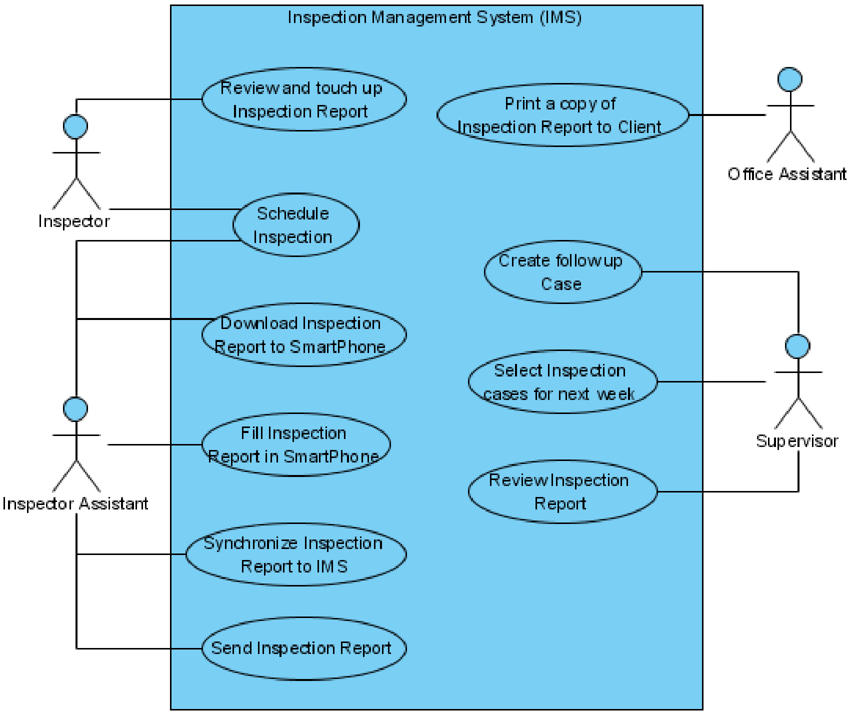

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to SmartPhone
Select Inspection cases for next week
Supervisor
Fill Inspection Report in SmartPhone
Review Inspection Report
Inspector Assistant
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 3-34 *Modify project*

2. Open **Teamwork Client** dialog.
3. Select the teamwork project in the list.

4. Select **Project > Revert...** from menu

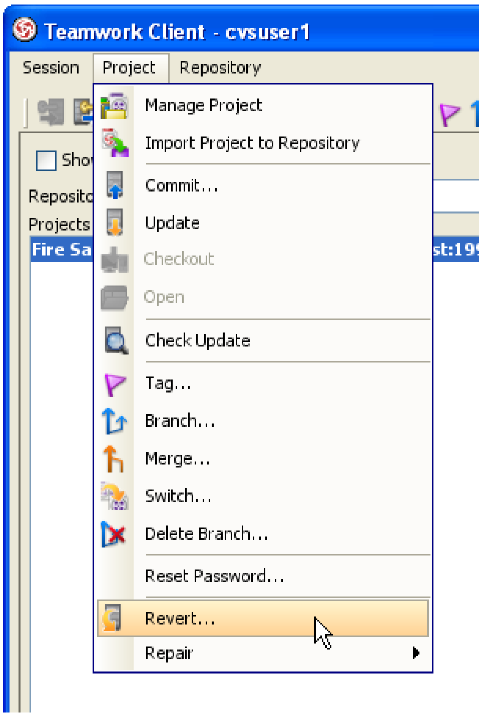


Figure 3-35 *Revert from menu*

Or click **Revert...** button from toolbar.

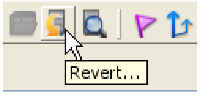


Figure 3-36 *Revert from toolbar*

5. Click **Yes** button from the confirmation dialog.
6. The reverted project opened in **VP-UML**.

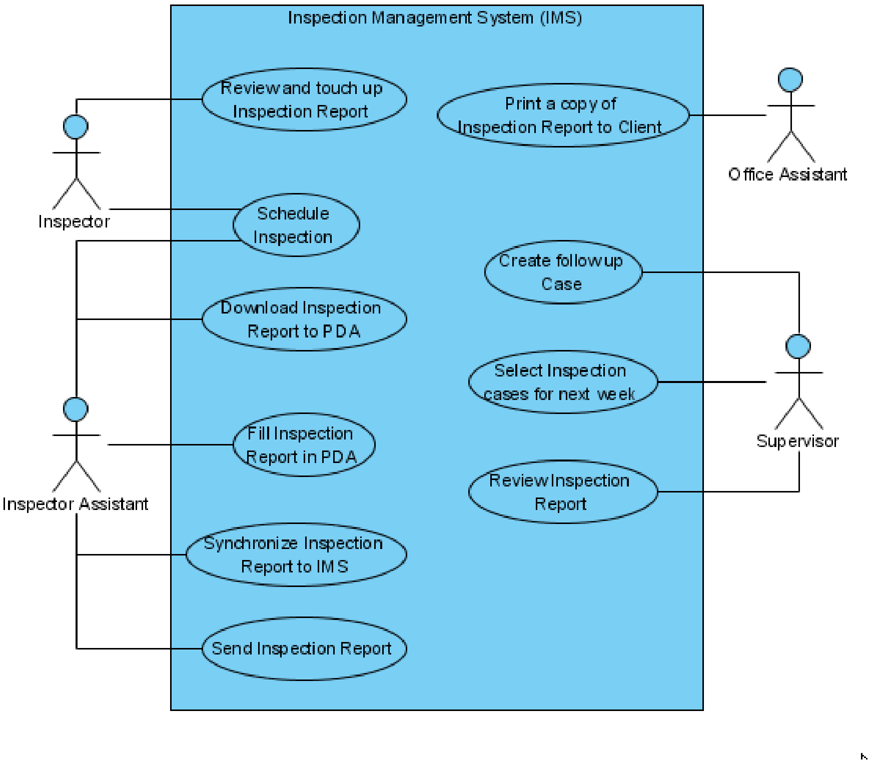

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Inspector Assistant
Fill Inspection Report in PDA
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 3-37 *Reverted project*

Revert server modifications

1. Open project.

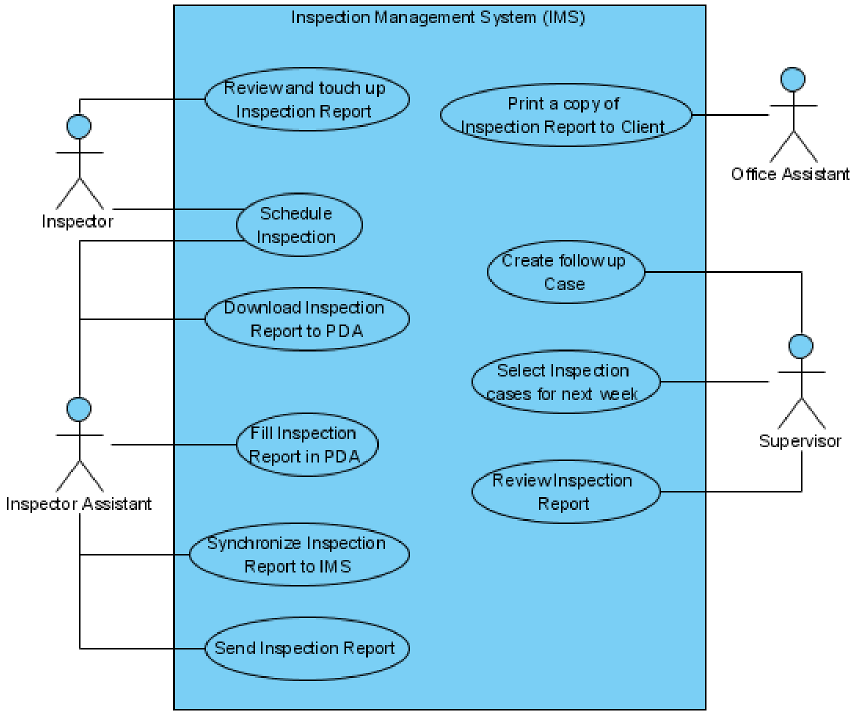
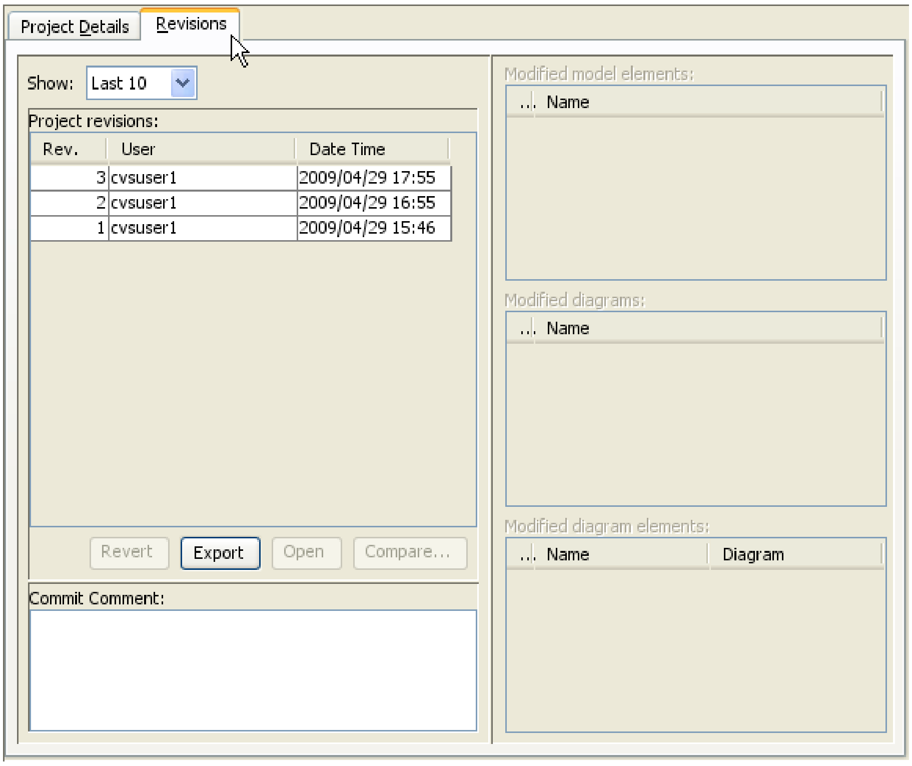

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 3-38 Open project

2. Open **Teamwork Client** dialog.
3. Select the teamwork project in the list.
4. Click the **Revisions** tab on the right of the project list.



Show: Last 10

Project revisions:

Rev.	User	Date Time
3	cvsuser1	2009/04/29 17:55
2	cvsuser1	2009/04/29 16:55
1	cvsuser1	2009/04/29 15:46

Commit Comment:

Modified model elements:

...	Name

Modified diagrams:

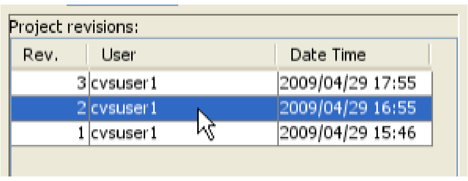
...	Name

Modified diagram elements:

...	Name	Diagram

Figure 3-39 Revision tab

5. Select the revision to revert.



Project revisions:

Rev.	User	Date Time
3	cvsuser1	2009/04/29 17:55
2	cvsuser1	2009/04/29 16:55
1	cvsuser1	2009/04/29 15:46

Figure 3-40 *Select revision*

6. Click **Revert** button.

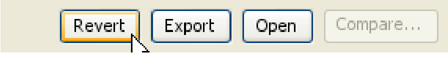
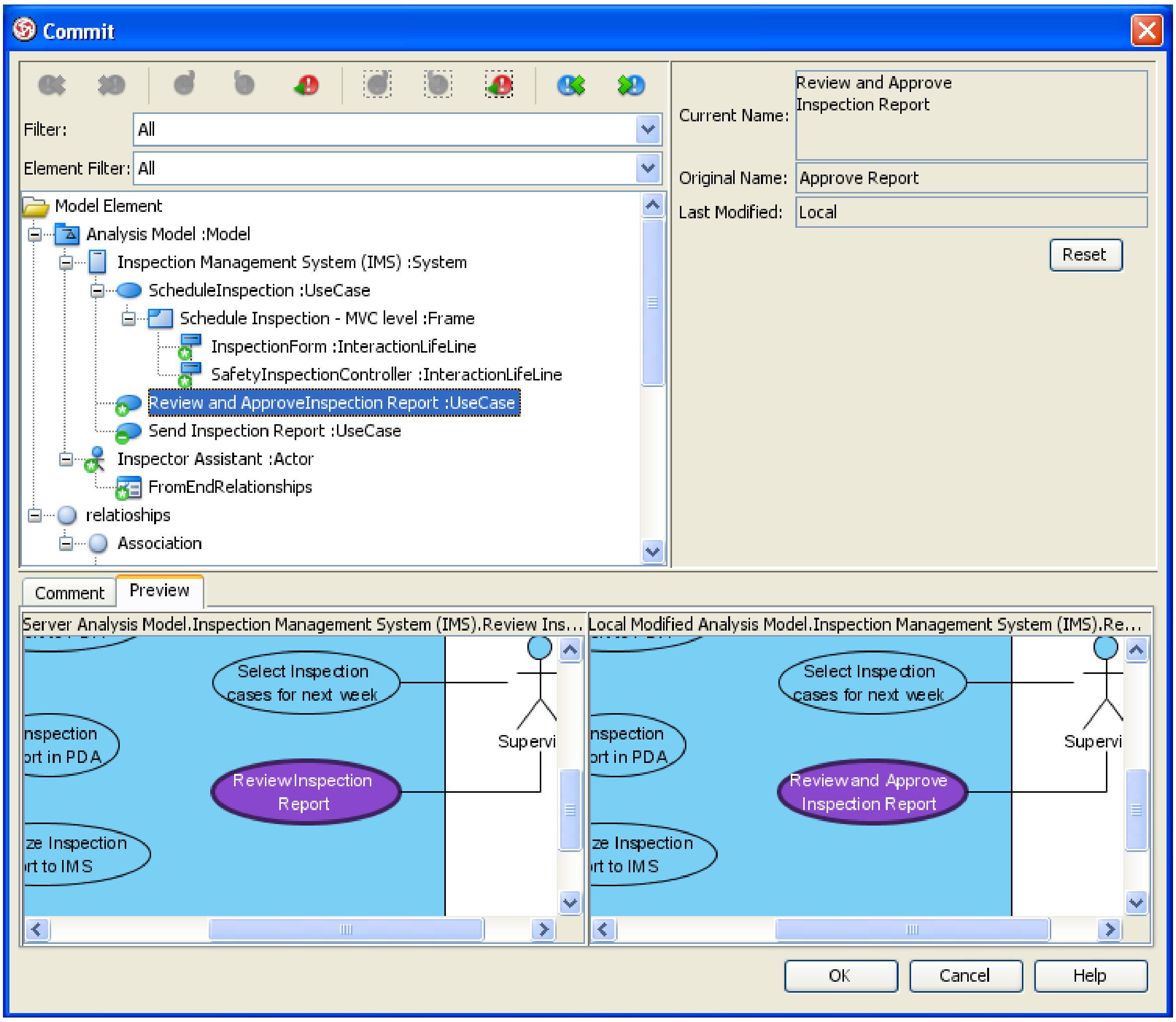


Figure 3-41 *Revert button*

7. The **Commit** dialog show a list of shapes and model elements reverted for review the changes and preview diagram, input the comment and click **OK** button to commit the revert.



Commit

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionForm :InteractionLifeLine
          - SafetyInspectionController :InteractionLifeLine
      - Review and ApproveInspection Report :UseCase
      - Send Inspection Report :UseCase
    - Inspector Assistant :Actor
      - FromEndRelationships
  - relatioships
    - Association

Current Name: Review and Approve Inspection Report

Original Name: Approve Report

Last Modified: Local

Server Analysis Model.Inspection Management System (IMS).Review Ins...

Local Modified Analysis Model.Inspection Management System (IMS).Re...

Figure 3-42 *Commit dialog with reverted changes*

8. A new revision with the reverted changes was created in server, and opened in **VP-UML**.

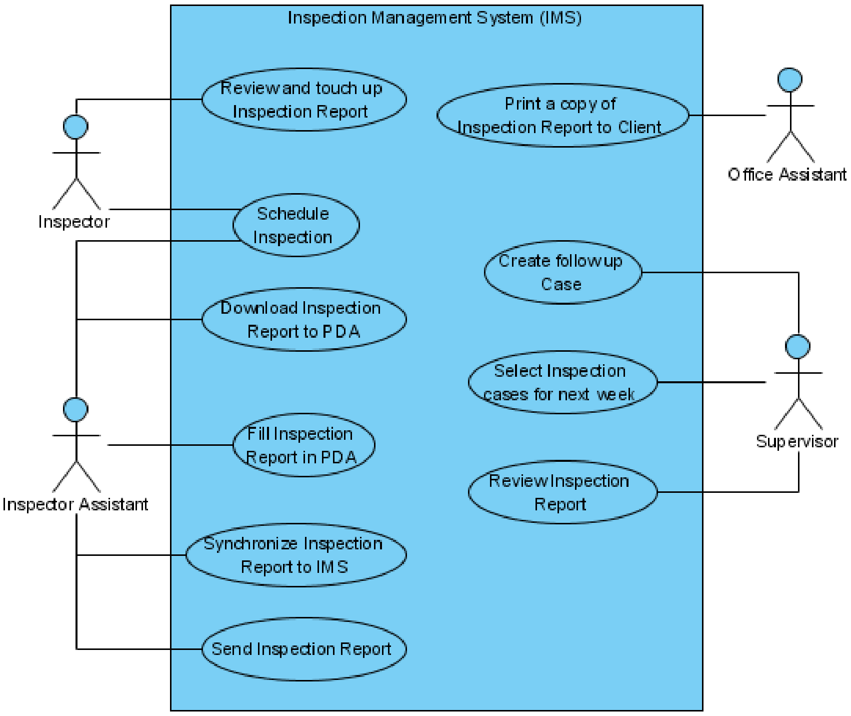

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 3-43 *Reverted project*

Browsing change histories (old revisions)

Checkout old revisions

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select the revision to open.
5. Click Open button.

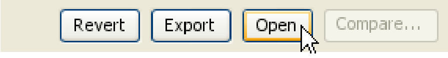


***Figure 3-44** Open button*

6. Selected revision opened in **VP-UML**.

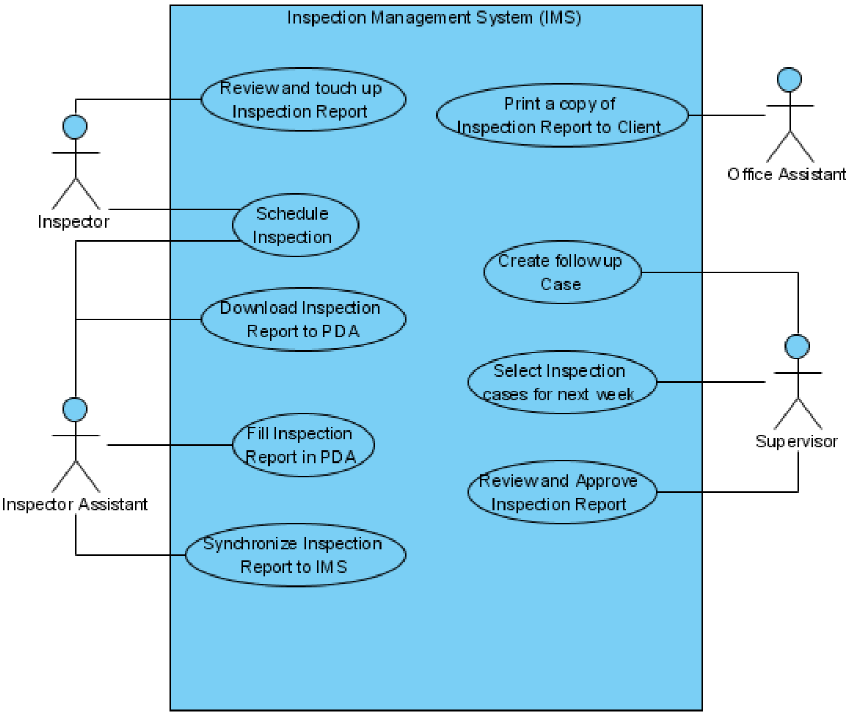
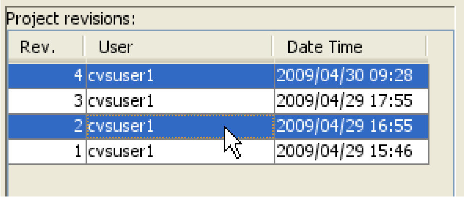

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS

***Figure 3-45** Open selected revision*

Showing differences between revisions visually

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select a revision for compare.
5. Select another revision by **Ctrl+Click**.



Project revisions:

Rev.	User	Date Time
4	cvsuser1	2009/04/30 09:28
3	cvsuser1	2009/04/29 17:55
2	cvsuser1	2009/04/29 16:55
1	cvsuser1	2009/04/29 15:46

***Figure 3-46** Select two revisions*

6. Click the **Compare...** button.

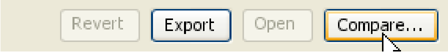
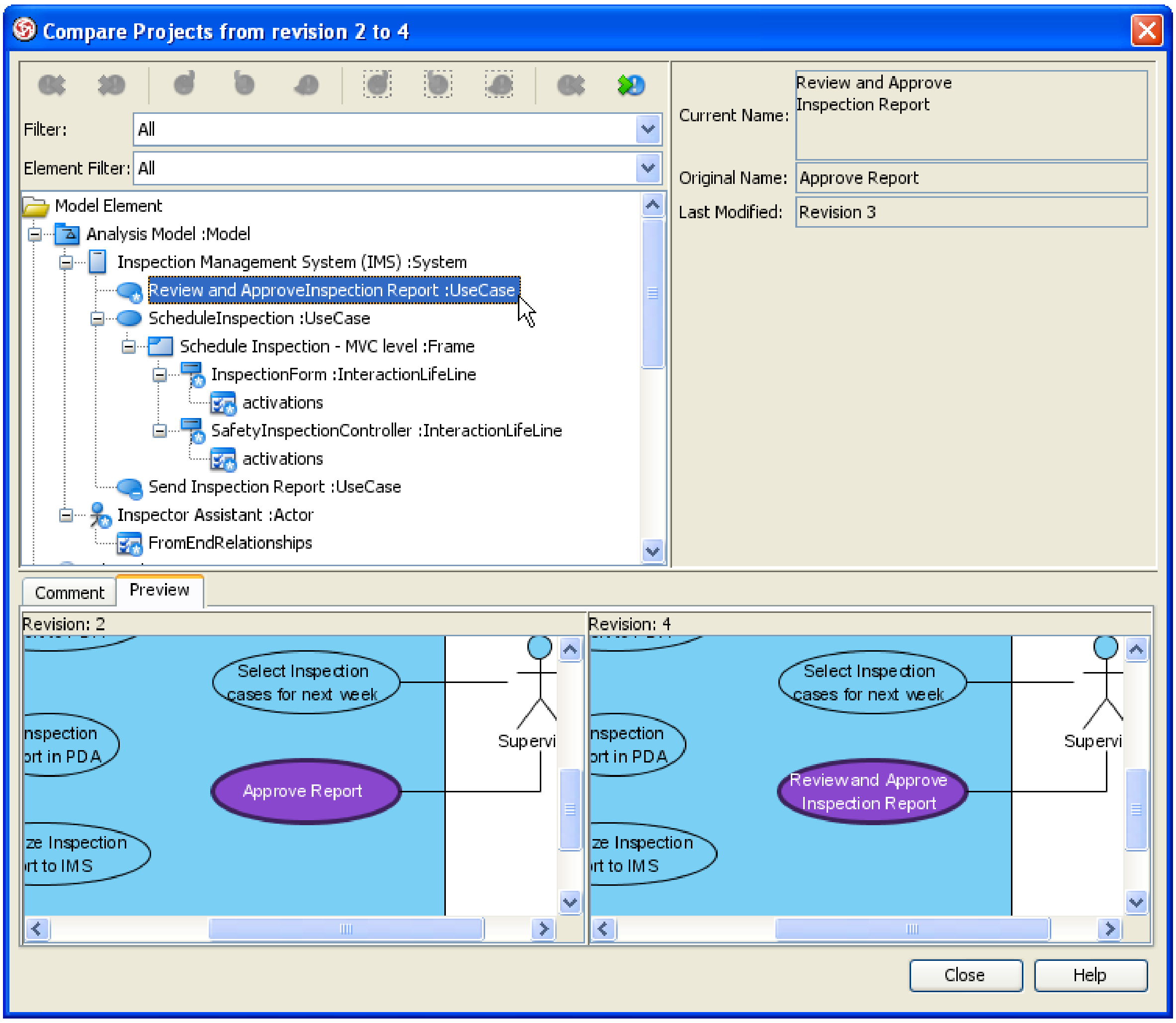


***Figure 3-47** Compare button*

7. Similar to **Commit** and **Update** dialog, the **Compare Projects** dialog show a list of differences between the selected revisions. You can also view the differences visually in diagram on the preview tab.



Compare Projects from revision 2 to 4

Filter: All

Element Filter: All

Current Name: Review and Approve Inspection Report

Original Name: Approve Report

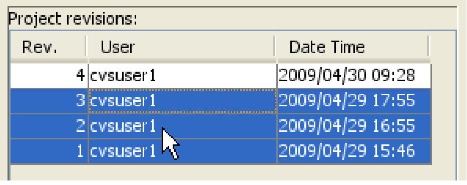
Last Modified: Revision 3

Model Element
Analysis Model :Model
Inspection Management System (IMS) :System
Review and ApproveInspection Report :UseCase
ScheduleInspection :UseCase
Schedule Inspection - MVC level :Frame
InspectionForm :InteractionLifeLine
activations
SafetyInspectionController :InteractionLifeLine
activations
Send Inspection Report :UseCase
Inspector Assistant :Actor
FromEndRelationships
Comment
Preview
Revision: 2
Select Inspection cases for next week
nspection ort in PDA
Approve Report
ze Inspection rt to IMS
Supervi
Revision: 4
Select Inspection cases for next week
nspection ort in PDA
Review and Approve Inspection Report
ze Inspection rt to IMS
Supervi
Close
Help

Figure 3-48 *Compare differences*

Export multiple revisions to local

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select multiple revisions for export, by **Ctrl+Click** or **Shift+Click**.



Project revisions:

Rev.	User	Date Time
4	cvsuser1	2009/04/30 09:28
3	cvsuser1	2009/04/29 17:55
2	cvsuser1	2009/04/29 16:55
1	cvsuser1	2009/04/29 15:46

Figure 3-49 *Select multiple revisions*

5. Click the **Export** button.

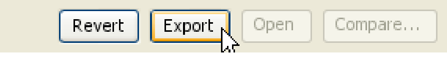
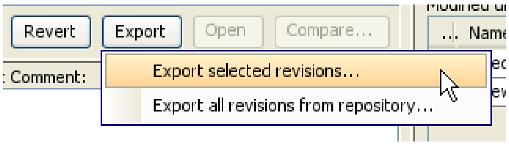


Figure 3-50 *Export buttons*

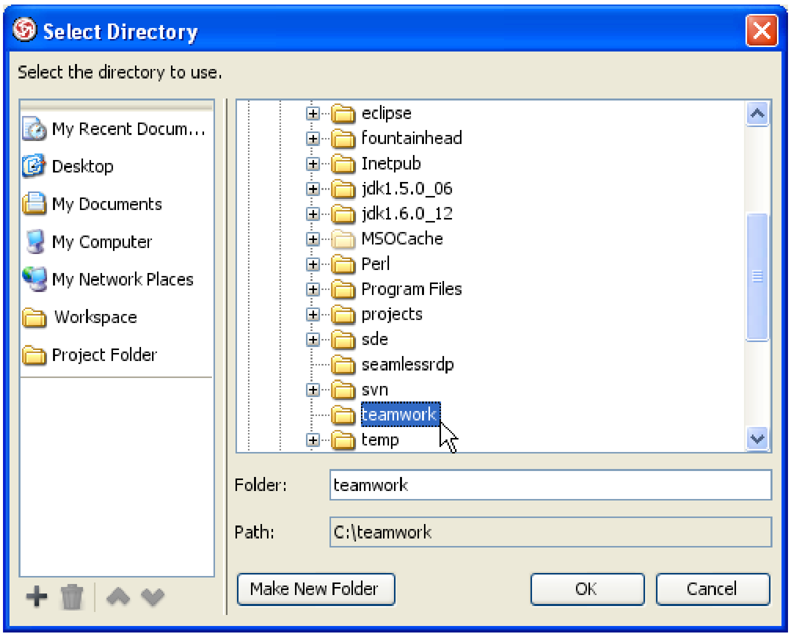
6. Select **Export selected revisions...** from the popup.



Revert
Export
Open
Compare...
Modified di
... Name
Comment:
Export selected revisions...
Export all revisions from repository...

Figure 3-51 *Export selected revisions*

7. Select the directory to save the exported project revisions.



Select Directory

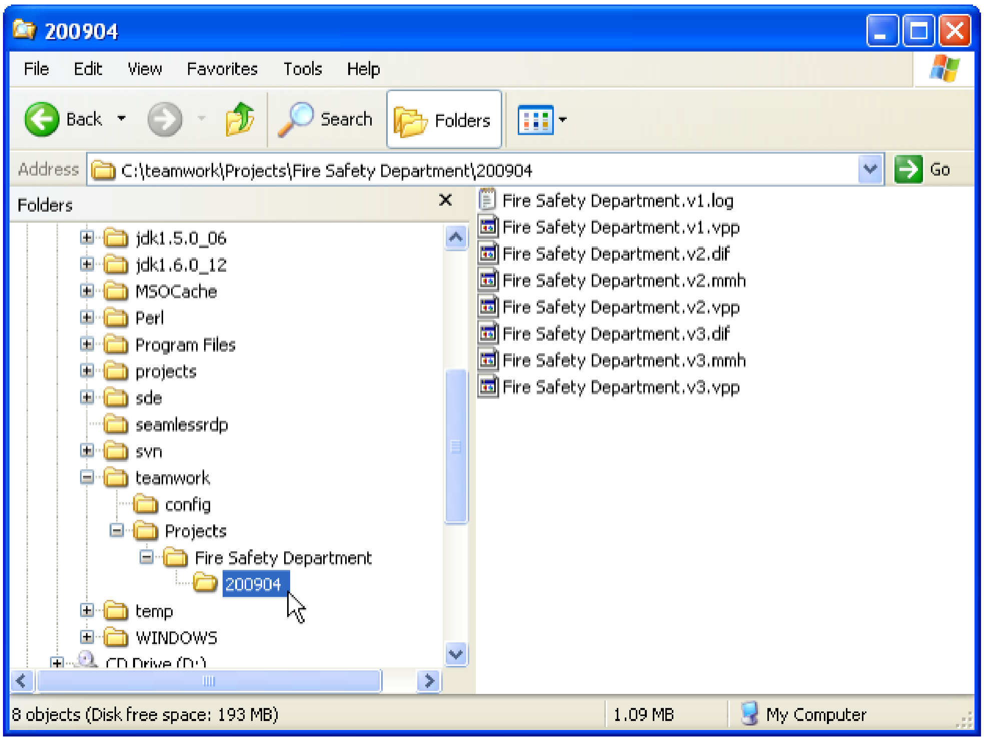
Select the directory to use.

Folder: teamwork

Path: C:\teamwork

Figure 3-52 *Select directory*

8. You'll found the selected revisions was exported to the select directory.



Address C:\teamwork\Projects\Fire Safety Department\200904

Fire Safety Department.v1.log
Fire Safety Department.v1.vpp
Fire Safety Department.v2.dif
Fire Safety Department.v2.mmh
Fire Safety Department.v2.vpp
Fire Safety Department.v3.dif
Fire Safety Department.v3.mmh
Fire Safety Department.v3.vpp

8 objects (Disk free space: 193 MB) 1.09 MB My Computer

Figure 3-53 *Exported revisions*

Isolating last long modifications with Branches

Creating Branch

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Branch...** from menu

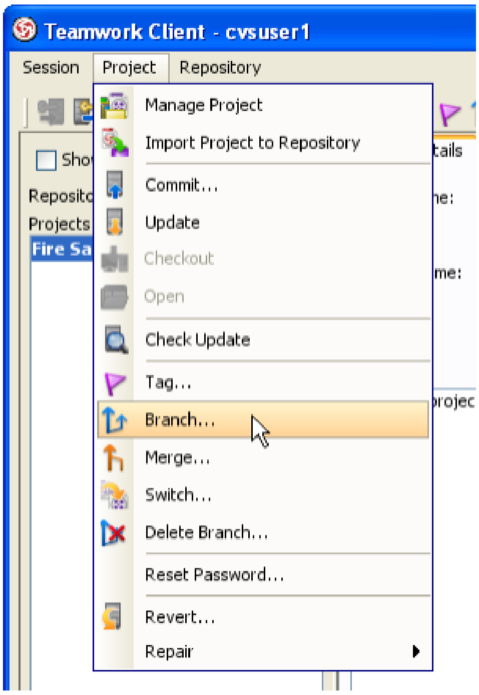


Figure 3-54 *Branch from menu*

Or click **Branch...** button from toolbar.

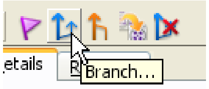
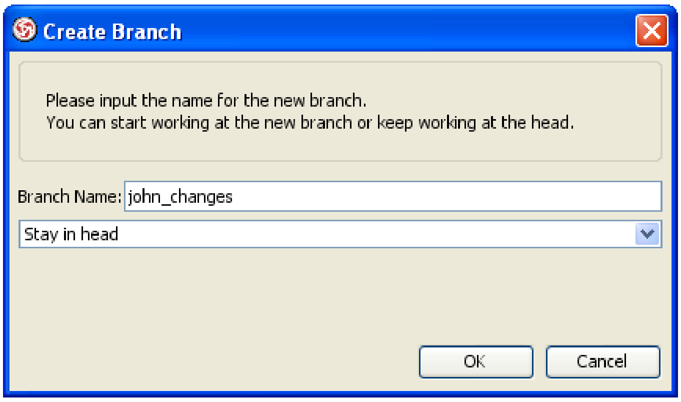


Figure 3-55 *Branch from toolbar*

4. Fill in the **Branch Name** in **Create Branch** dialog.



Please input the name for the new branch.
You can start working at the new branch or keep working at the head.

Branch Name: john_changes

Stay in head

Figure 3-56 *Create Branch dialog*

5. Select **Stay in head** in the list.

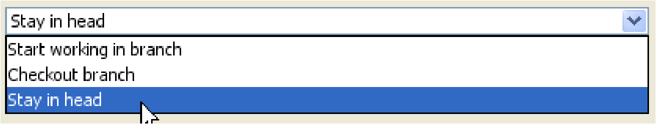


Figure 3-57 *Stay in head*

Switch local copy between Branches

1. Select **Project > Switch...** from menu

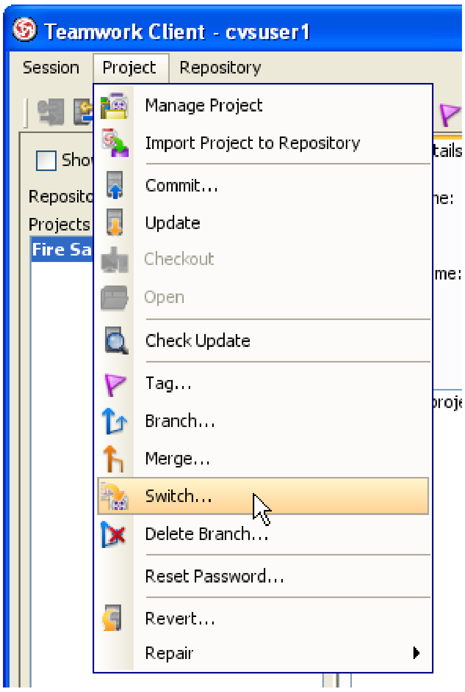


Figure 3-58 *Switch from menu*

Or click **Switch...** button from toolbar.

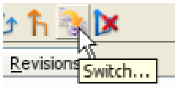
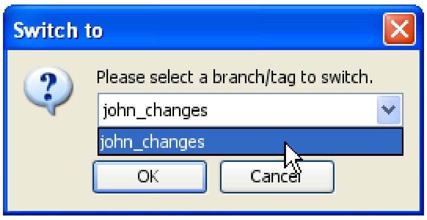


Figure 3-59 *Switch from toolbar*

2. Select the branch to switch.


Switch to

Please select a branch/tag to switch.

john_changes

Figure 3-60 *Select branch*

3. The branch is opened in **VP-UML**.

Merging from HEAD To Branch

1. Open, modify and commit HEAD project.

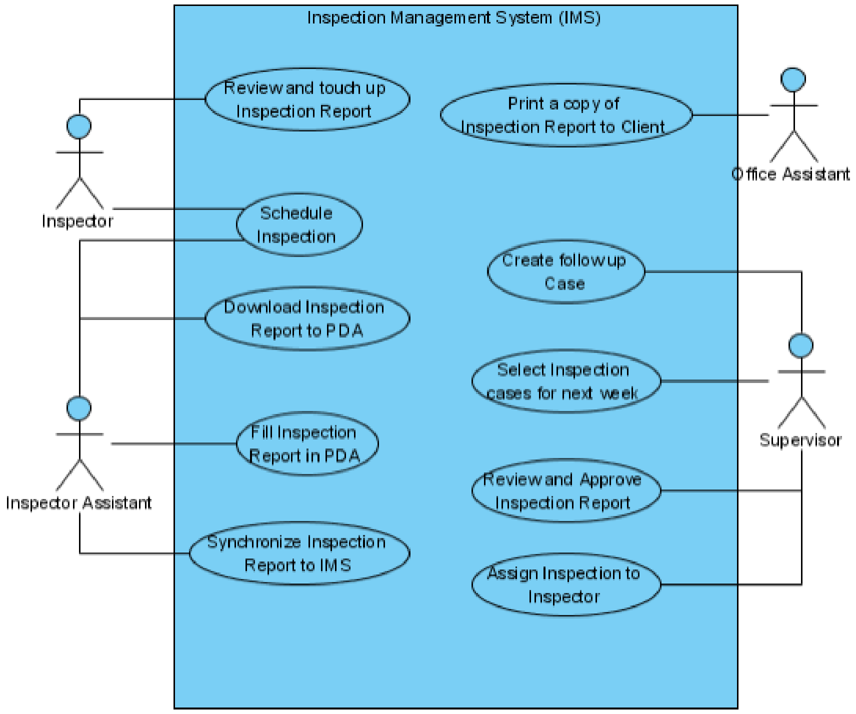

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 3-61 *Modified HEAD project*

2. Open branch project.

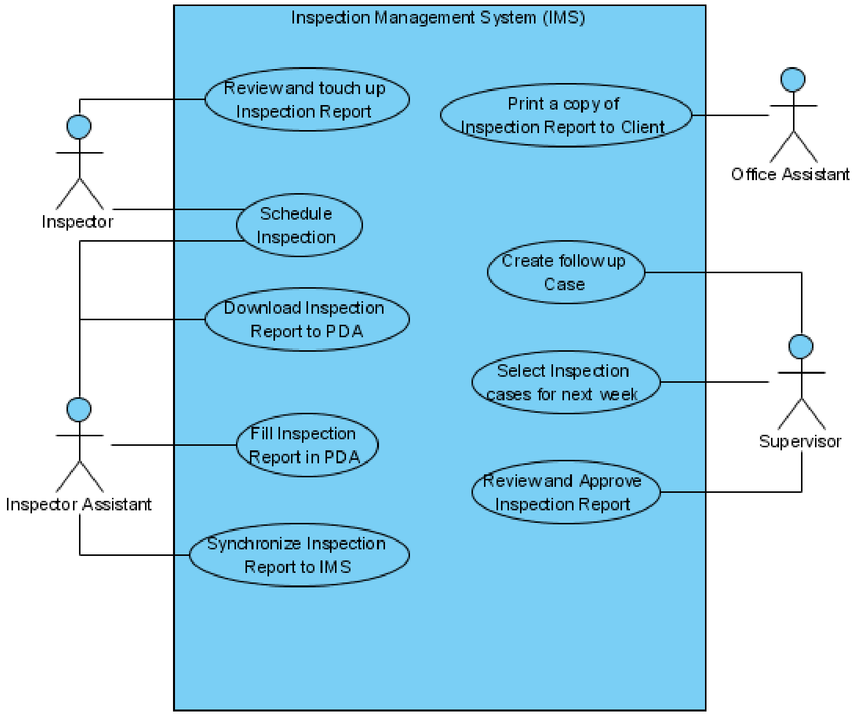

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS

Figure 3-62 *Branch project*

3. Select **Project > Merge...** from menu.

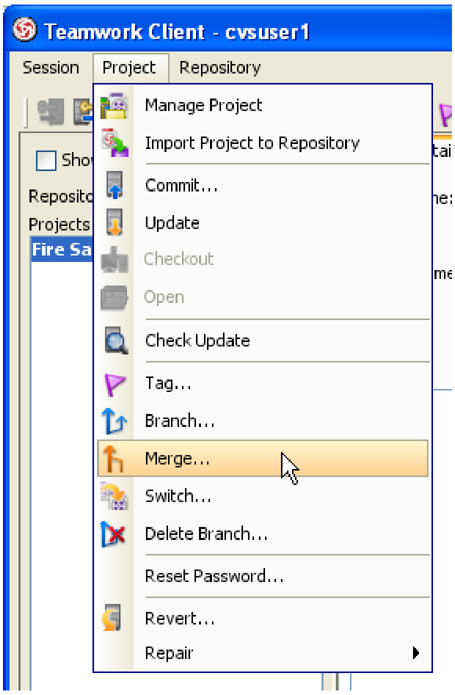


Figure 3-63 *Merge from menu*

Or click **Merge...** button from toolbar.

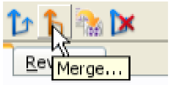
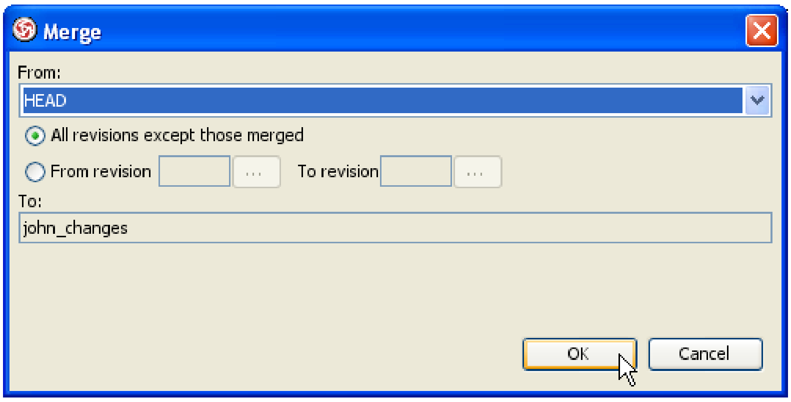


Figure 3-64 *Merge from toolbar*

4. Select from HEAD in the **Merge** dialog.



Merge

From: HEAD

● All revisions except those merged

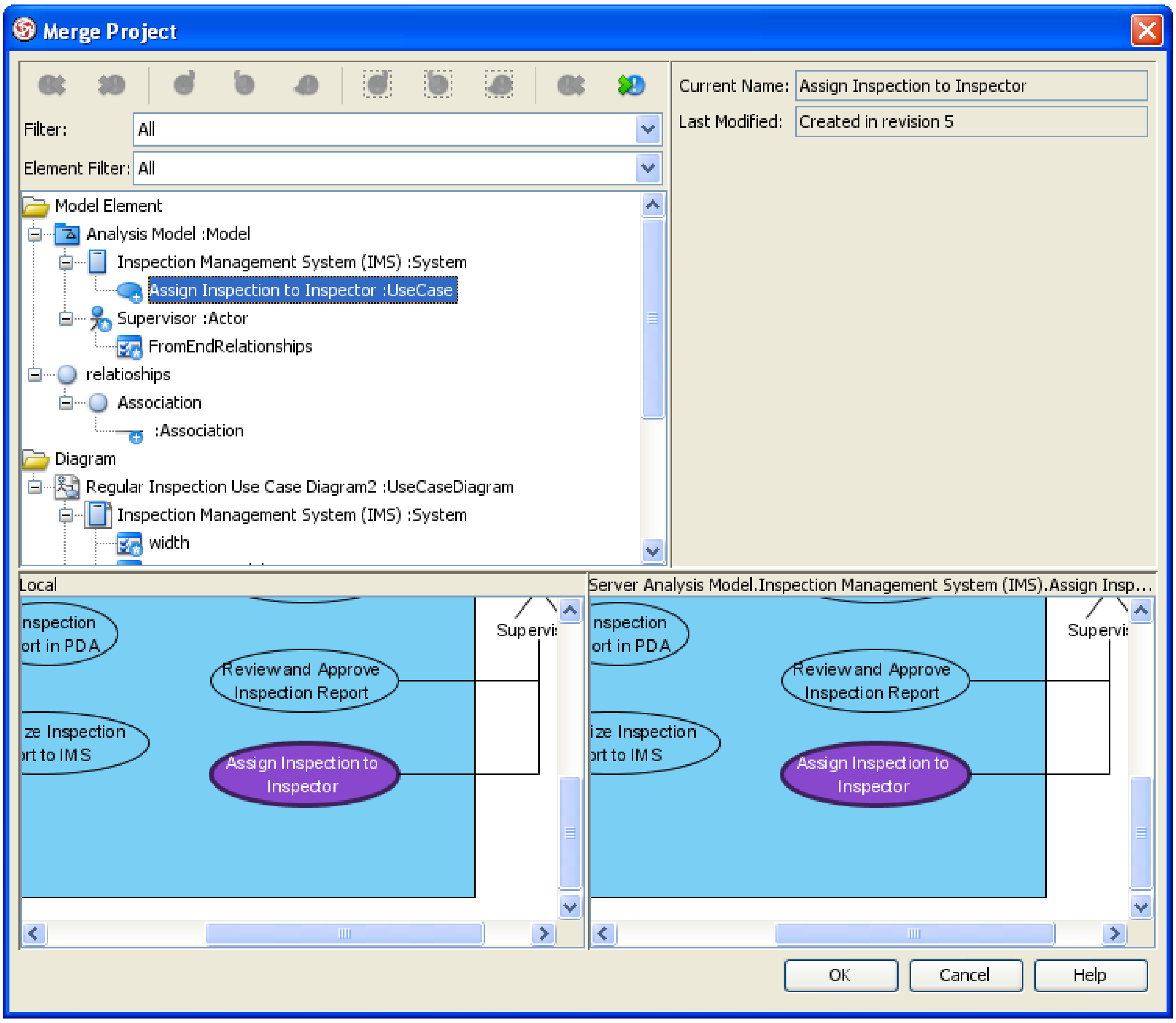
○ From revision ... To revision ...

To: john_changes

OK Cancel

Figure 3-65 *Merge dialog*

5. The **Merge Project** dialog show a list of changes similar to **Commit** and **Update** dialog.



Merge Project

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Assign Inspection to Inspector :UseCase
    - Supervisor :Actor
      - FromEndRelationships
  - relatioships
    - Association
      - :Association
- Diagram
  - Regular Inspection Use Case Diagram2 :UseCaseDiagram
    - Inspection Management System (IMS) :System
      - width

Current Name: Assign Inspection to Inspector

Last Modified: Created in revision 5

Local
nspection
ort in PDA
Review and Approve Inspection Report
ze Inspection
rt to IMS
Assign Inspection to Inspector
Supervi
Server Analysis Model.Inspection Management System (IMS).Assign Insp...
nspection
ort in PDA
Review and Approve Inspection Report
ize Inspection
rt to IMS
Assign Inspection to Inspector
Supervi
OK
Cancel
Help

Figure 3-66 *Merge project dialog*

6. Finally, review the changes in **Commit** dialog. Input comment and click **OK** to commit.

7. The branch project now contains the changes from HEAD.

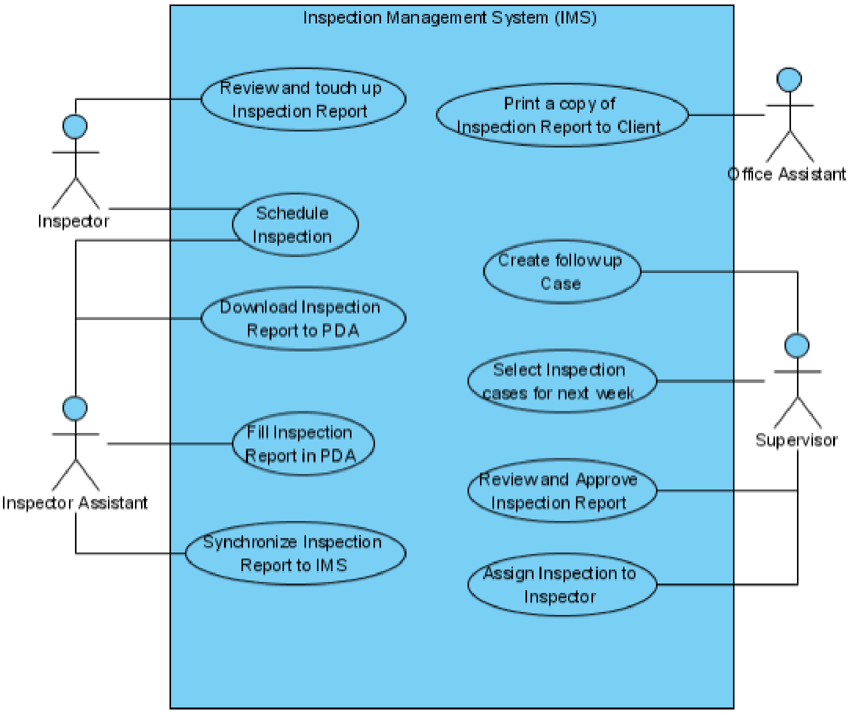

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 3-67 Branch project merged from HEAD

Merging from Branch to HEAD

1. Open, modify and commit branch project.

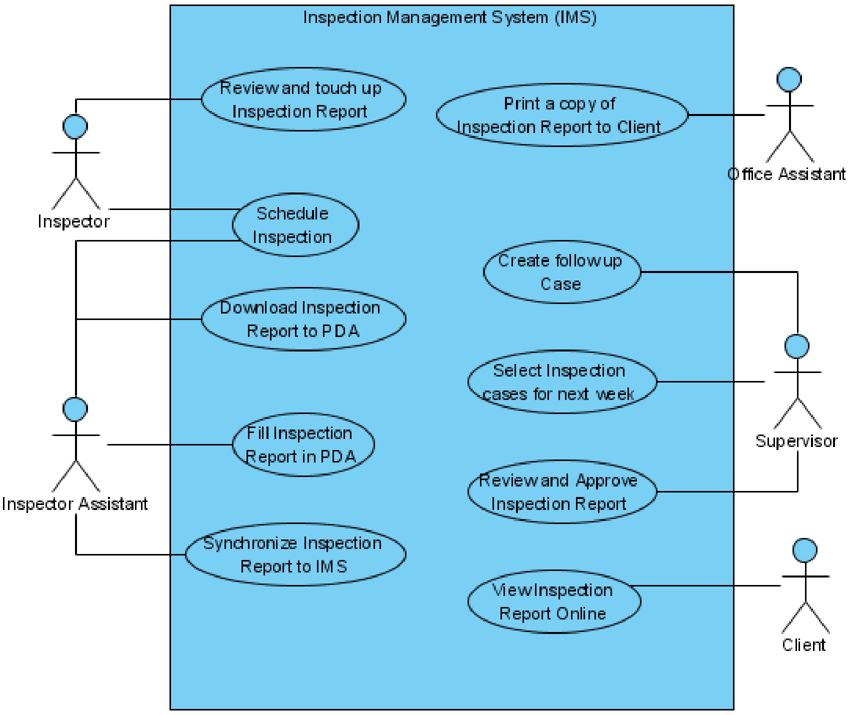

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
View Inspection Report Online
Client

Figure 3-68 Modified branch project

2. Open HEAD project.

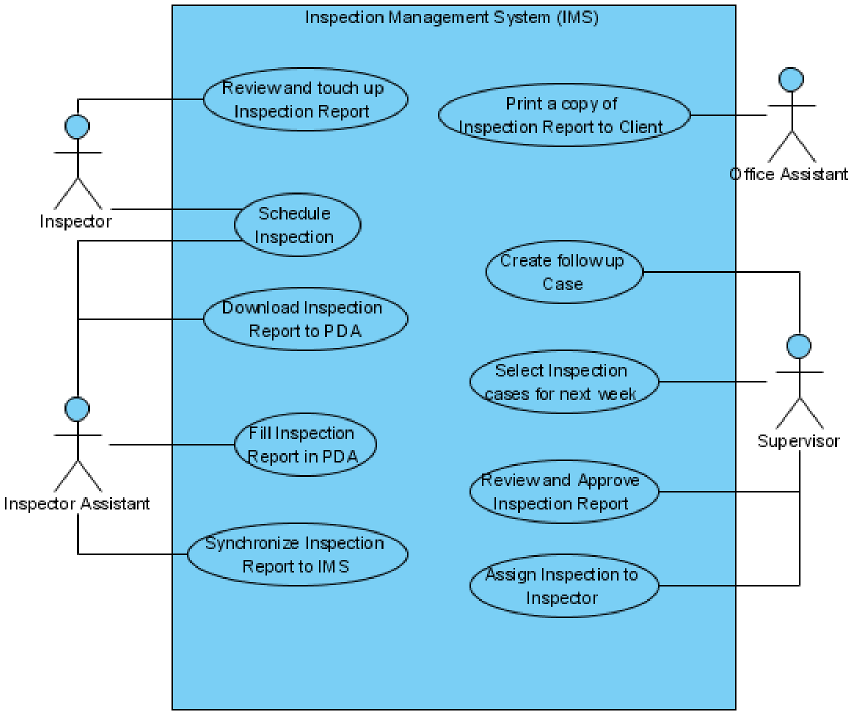
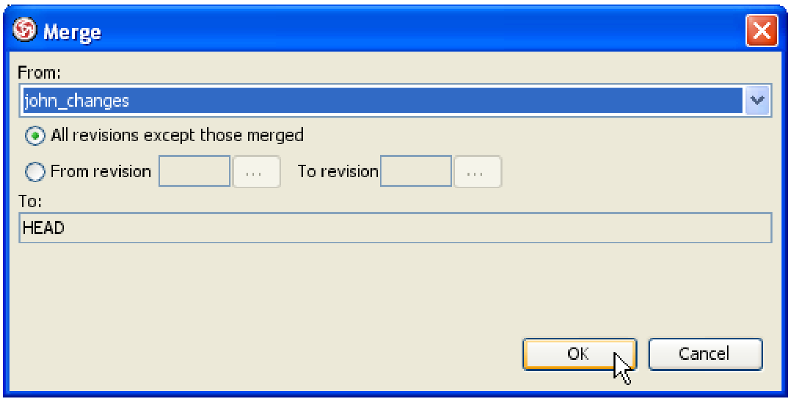

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 3-69 HEAD project

3. Select **Project > Merge...** from menu, or click **Merge...** button from toolbar.
4. Select a branch name in **From** combo box.



Merge

From:

john_changes

● All revisions except those merged

○ From revision ... To revision ...

To:

HEAD

OK Cancel

Figure 3-70 Merge dialog

5. The **Merge Project** dialog show a list of changes similar to **Commit** and **Update** dialog.
6. Finally, review the changes in **Commit** dialog. Input comment and click **OK** to commit.

7. The HEAD project now contains the changes from branch.

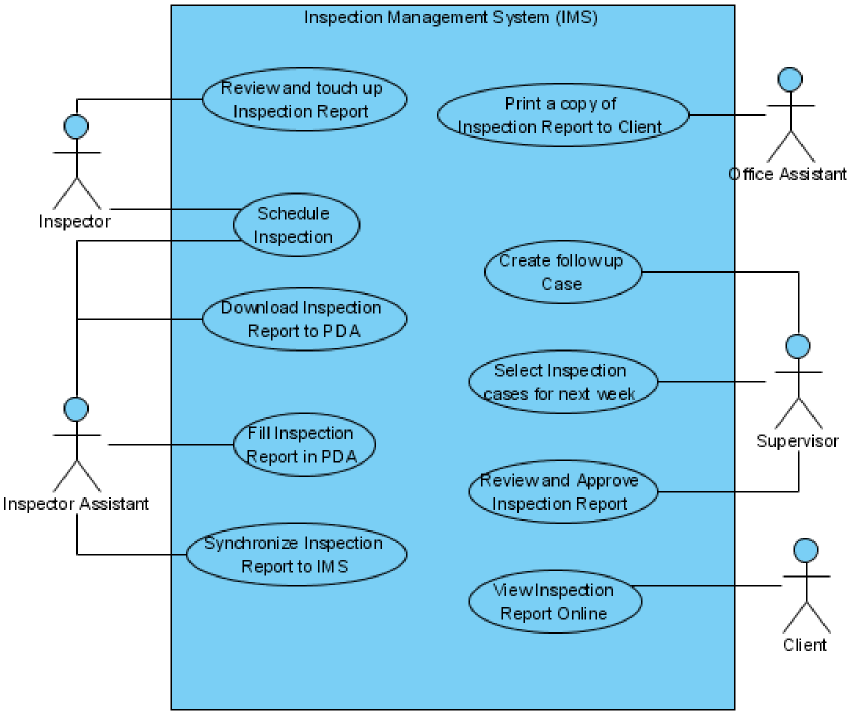

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
View Inspection Report Online
Client

Figure 3-71 *HEAD project merged from branch*

Delete Branch

1. Select **Project > Delete Branch...** from menu

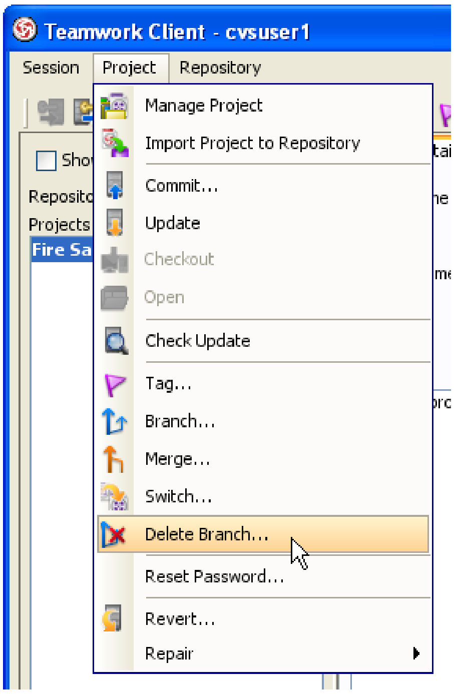


Figure 3-72 *Delete Branch from menu*

Or click **Delete Branch ...** button from toolbar.

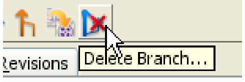
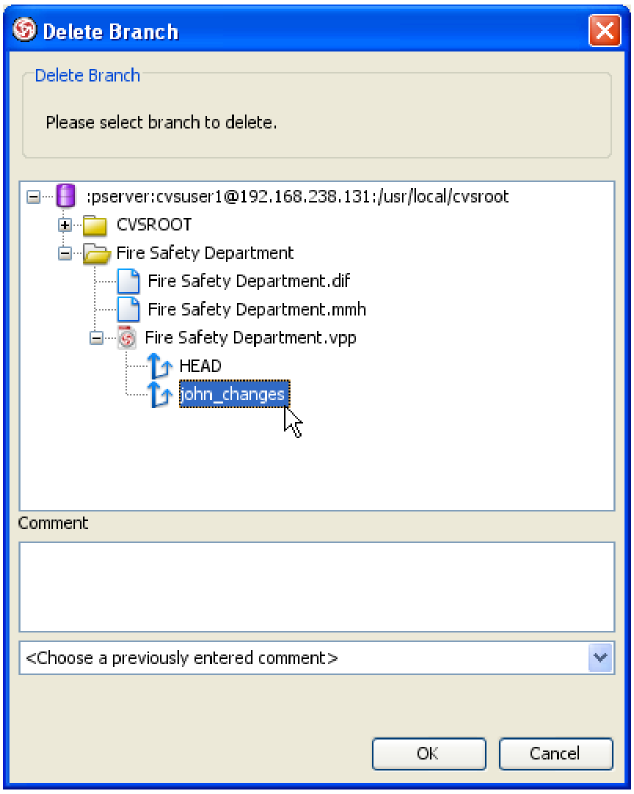


Figure 3-73 *Delete Branch from toolbar*

2. Expand the repository and folder node, select the branch to delete. Click **OK** button to continue.



Delete Branch

Delete Branch

Please select branch to delete.

```
:pserver:cvsuser1@192.168.238.131:/usr/local/cvsroot
    CVSROOT
    Fire Safety Department
        Fire Safety Department.dif
        Fire Safety Department.mmh
        Fire Safety Department.vpp
            HEAD
            john_changes
```

Comment

<Choose a previously entered comment>

Figure 3-74 *Delete branch dialog*

3. Click **Yes** button on confirmation dialog.

Marking release or milestone with Tags

Tags and branches are almost the same, with the only difference - tags are read only. Creating read only tags ensure you are able open the release version project later.

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Tag...** from menu

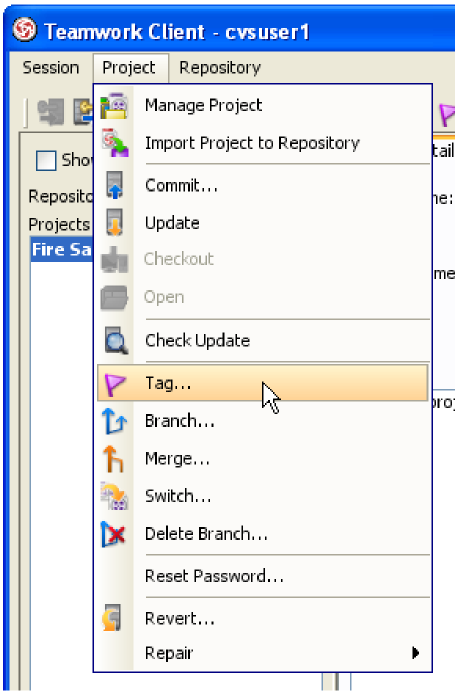


Figure 3-75 *Tag from menu*

Or click **Branch...** button from toolbar.

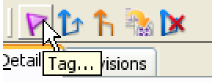
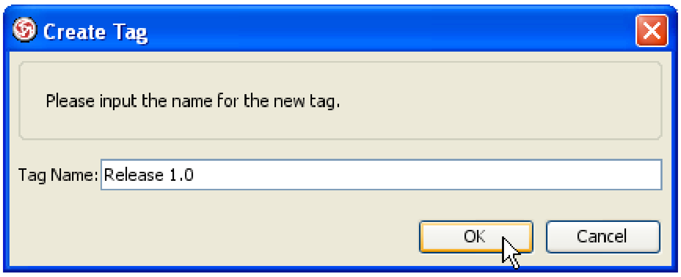


Figure 3-76 *Tag from toolbar*

4. Fill in the **Tag Name** in **Create Tag** dialog.



Please input the name for the new tag.

Tag Name: Release 1.0

Figure 3-77 *Create Tag dialog*

5. Click **OK** button, the tag will be created.

Importing Projects to Perforce

1. Start **VP-UML**.
2. Select **Tools > Teamwork > Open Teamwork Client...** from the main menu, or **Open Teamwork Client** icon from toolbar to open teamwork client.

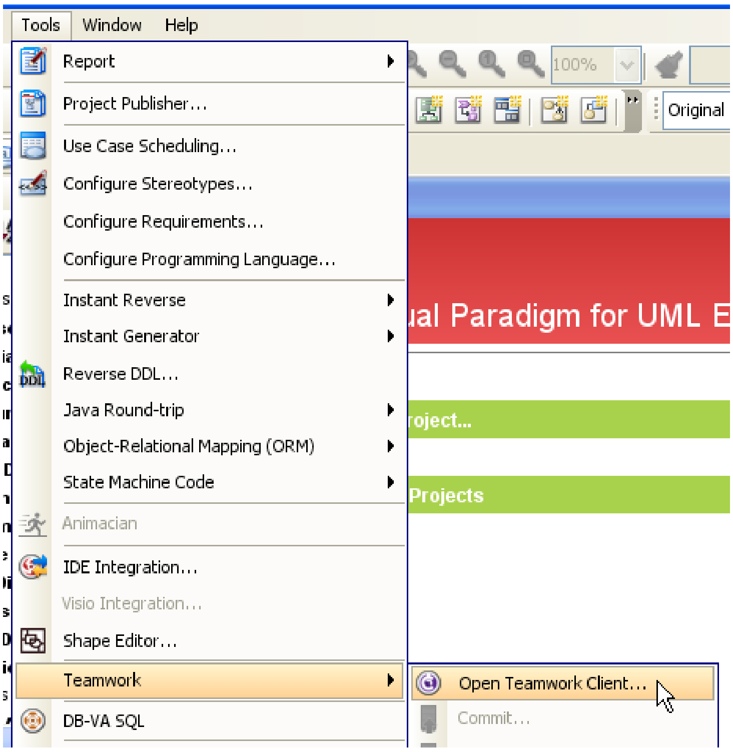
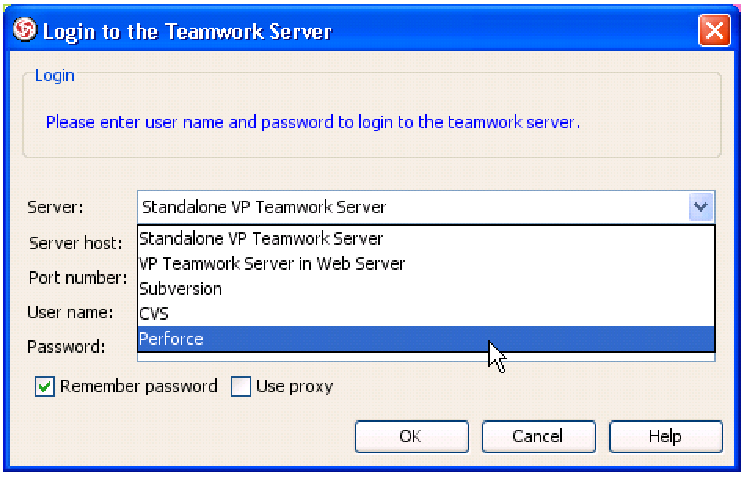


Figure 4-1 Open Teamwork Client from main menu

3. Select **Subversion** in **Server** field.


Login to the Teamwork Server

Login

Please enter user name and password to login to the teamwork server.

Server: Standalone VP Teamwork Server

Server host:

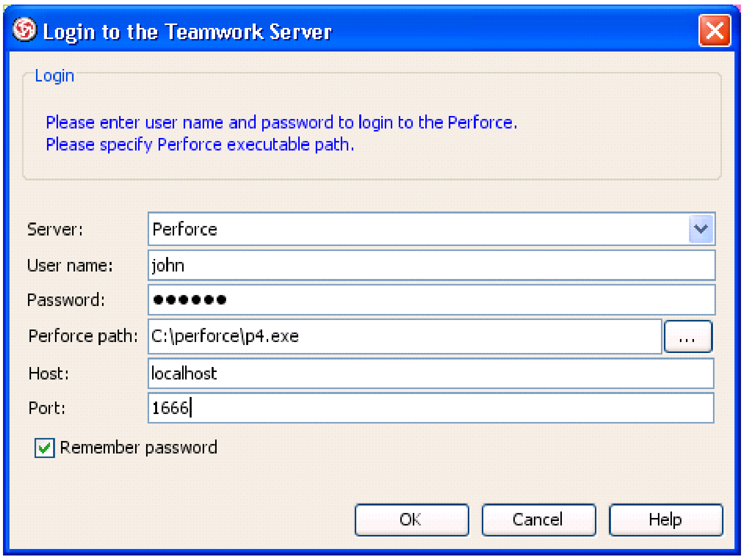
Port number:

User name:

Password:

Figure 4-2 perforce teamwork server

4. Fill in the server and user information to login teamwork server.



Login to the Teamwork Server

Login

Please enter user name and password to login to the Perforce.
Please specify Perforce executable path.

Server: Perforce

User name: john

Password: ●●●●●●

Perforce path: C:\perforce\p4.exe

Host: localhost

Port: 1666

☑ Remember password

Figure 4-2 Login teamwork server

5. Select **Project > Import Project to Repository** from the menu.

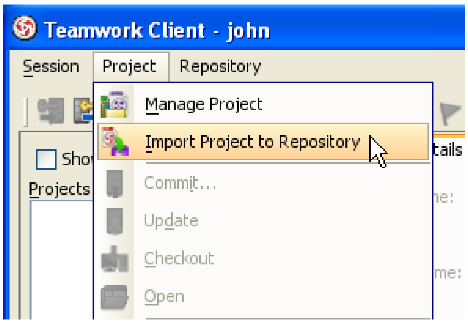


Figure 4-3 Import project from menu

Or click the **Import Project to Repository** button on toolbar.

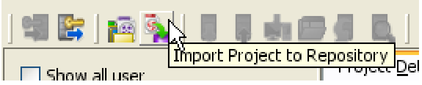
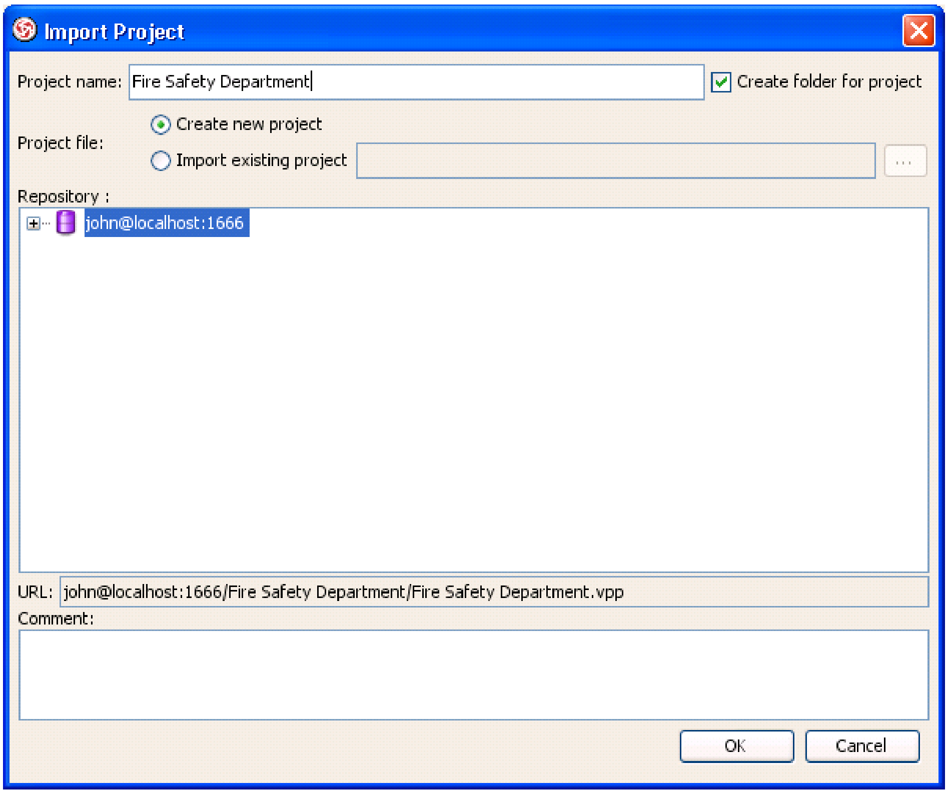


Figure 4-4 Import project from toolbar

6. In the **Import Project** dialog, fill in the **Project name**.



Import Project

Project name: Fire Safety Department ☑ Create folder for project

Project file: ● Create new project ○ Import existing project

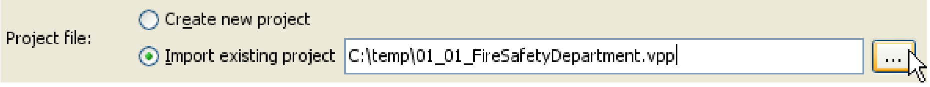
Repository : john@localhost:1666

URL: john@localhost:1666/Fire Safety Department/Fire Safety Department.vpp

Comment:

Figure 4-5 *Fill project name in Import Project dialog*

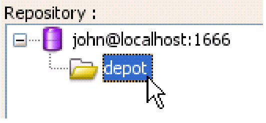
7. Choose **Create new project** or **Import existing project** from **Project file**. **Create new project** will import a blank project into teamwork server, **Import existing project** will import an existing VP-UML Project file (*.vpp) into teamwork server as first revision.



Project file: ○ Create new project ● Import existing project C:\temp\01_01_FireSafetyDepartment.vpp

Figure 4-6 *Specify project file*

8. Select the **depot** node for import the project.



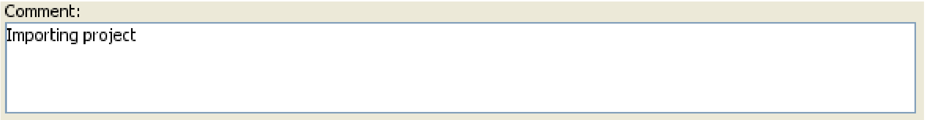
Repository :

john@localhost:1666

depot

Figure 4-7 *Select repository*

9. Fill in the comment.



Comment:

Importing project

Figure 4-8 *Input comment*

10. Click **OK** button to import project.

NOTE: Please make sure the user login to **Teamwork Client** has **Create Project** permission.

Checkout Project from Perforce

Checkout and open Perforce Projects

1. Open **Teamwork Client** dialog.
2. Select **Project > Manage Project** from the menu.

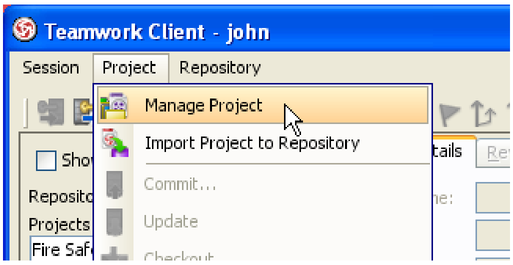


Figure 4-9 *Manage project from main menu*

Or click **Manage Project** icon on the toolbar.

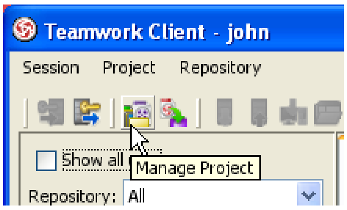


Figure 4-10 *Manage project from toolbar*

3. In **Manage Project** dialog, expand the repository node and select the project to checkout, click **>** button to add project. Click **OK** button to close the dialog.

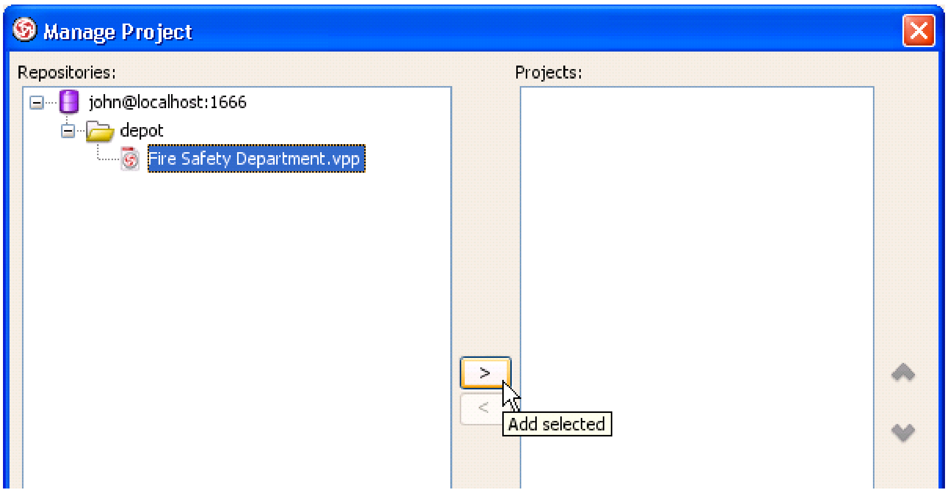


Figure 4-11 *Manage projects*

4. Select the project in project list.

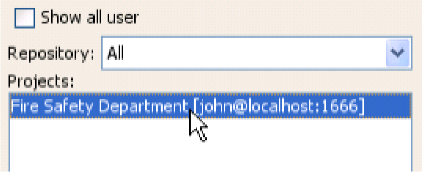


Figure 4-12 *Select project from the list*

5. Click **Open Project** button to checkout and open the project.

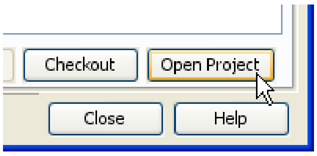
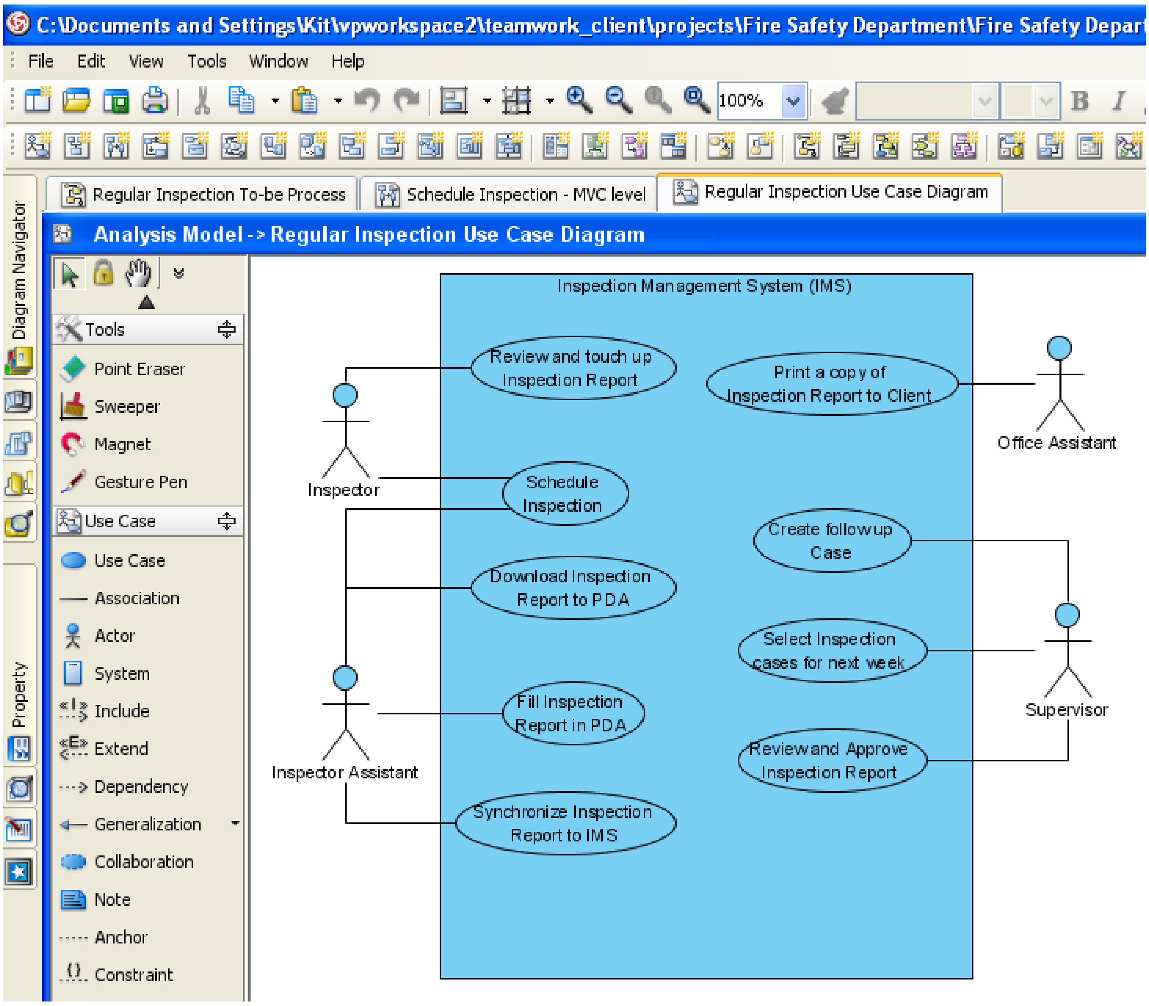


Figure 4-13 *Open project*

6. The project opened in **VP-UML**.



C:\Documents and Settings\Kit\vpworkspace2\teamwork_client\projects\Fire Safety Department\Fire Safety Depart
File
Edit
View
Tools
Window
Help
100%
Regular Inspection To-be Process
Schedule Inspection - MVC level
Regular Inspection Use Case Diagram
Analysis Model -> Regular Inspection Use Case Diagram
Diagram Navigator
Property
Tools
Point Eraser
Sweeper
Magnet
Gesture Pen
Use Case
Use Case
Association
Actor
System
Include
Extend
Dependency
Generalization
Collaboration
Note
Anchor
Constraint
Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS

Figure 4-14 *Project opened*

Committing local modification to Perforce

1. Modify the project.

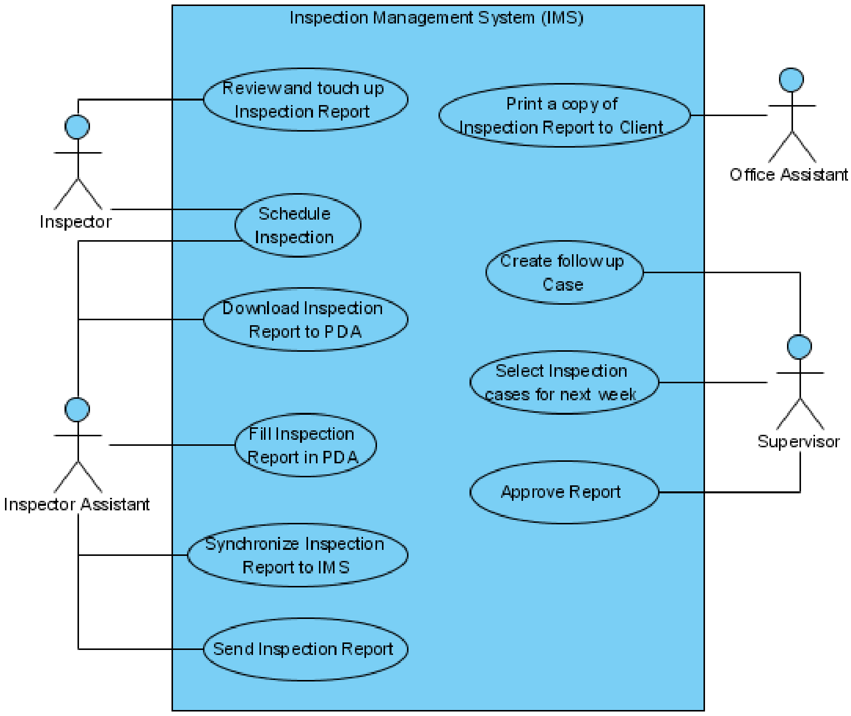

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Approve Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 4-15 *Modified project*

2. Show **Teamwork Toolbar** by right click on the Toolbar, select **Teamwork** if not selected already.

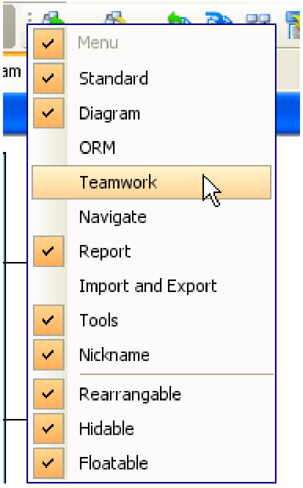


Figure 4-16 *Show teamwork toolbar*

3. Click the **Commit** button on **Teamwork Toolbar**.

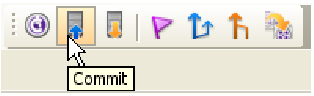
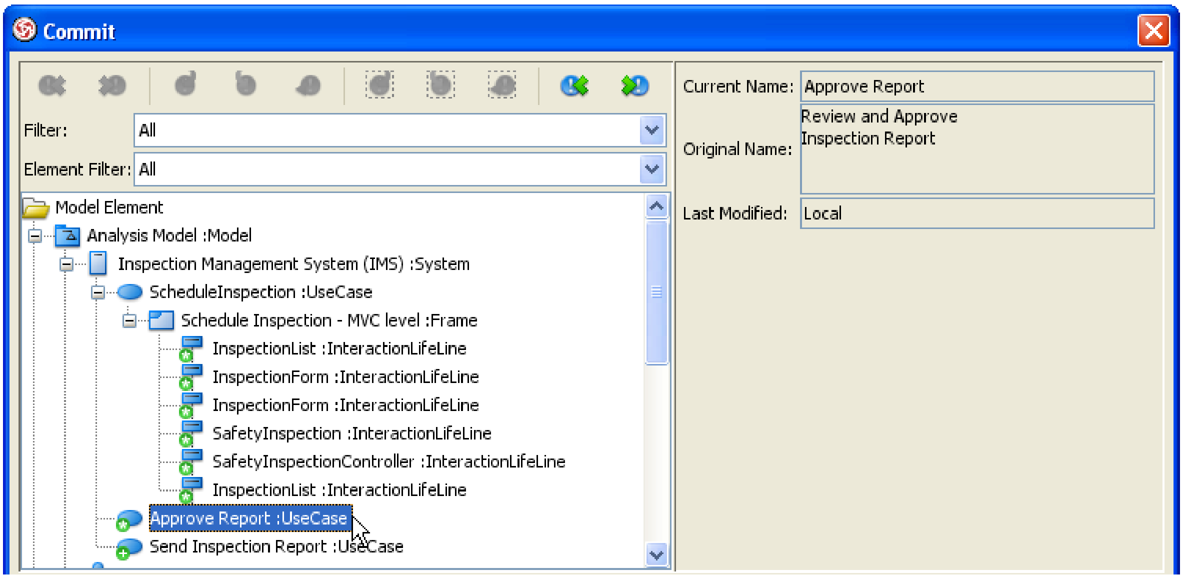


Figure 4-17 *Commit project*

4. In **Commit** dialog, you can review the changes for commit. On the left of dialog, you can see a list of changes shapes and model elements, click on it to view the detail changes on the right.



Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionList :InteractionLifeLine
          - InspectionForm :InteractionLifeLine
          - InspectionForm :InteractionLifeLine
          - SafetyInspection :InteractionLifeLine
          - SafetyInspectionController :InteractionLifeLine
          - InspectionList :InteractionLifeLine
      - Approve Report :UseCase
      - Send Inspection Report :UseCase

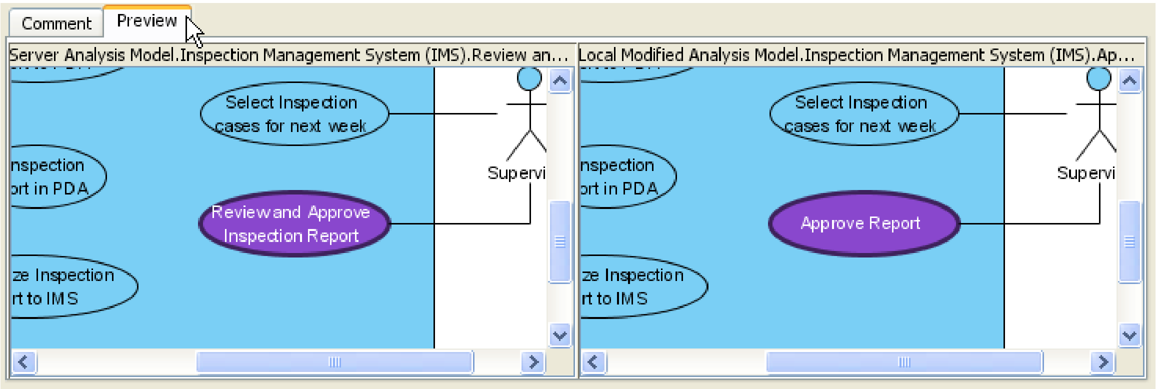
Current Name: Approve Report

Original Name: Review and Approve Inspection Report

Last Modified: Local

Figure 4-18 Review commit changes

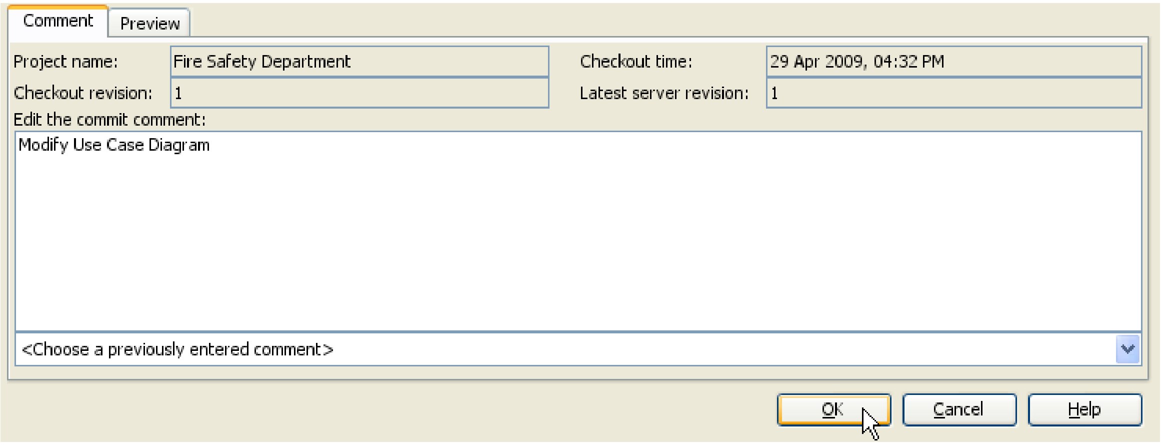
5. On the bottom of dialog, click the **Preview** tab to visually preview the changes in diagram. The selected shape is highlighed in purple.



Comment
Preview
Server Analysis Model.Inspection Management System (IMS).Review an...
Select Inspection cases for next week
nspection ort in PDA
Supervi
Review and Approve Inspection Report
ze Inspection rt to IMS
Local Modified Analysis Model.Inspection Management System (IMS).Ap...
Select Inspection cases for next week
nspection ort in PDA
Supervi
Approve Report
ze Inspection rt to IMS

Figure 4-19 Preview commit changes

6. After review the changes, click the **Comment** tab and input the comment for commit. Click **OK** button to start commit.



Project name: Fire Safety Department

Checkout time: 29 Apr 2009, 04:32 PM

Checkout revision: 1

Latest server revision: 1

Edit the commit comment:

Modify Use Case Diagram

<Choose a previously entered comment>

Figure 4-20 Input commit comment

Resolving conflicts

1. Modify the project.

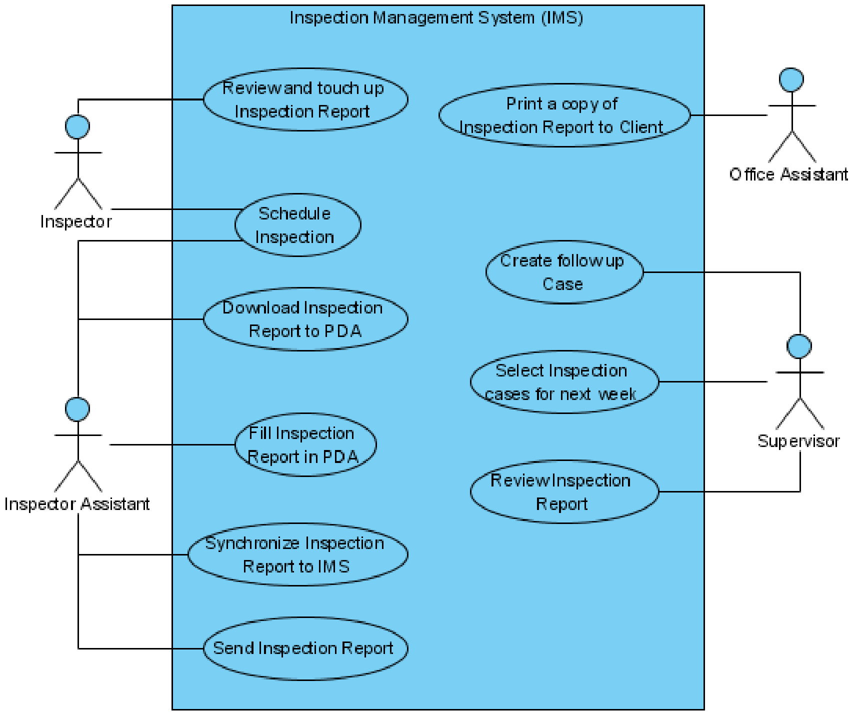
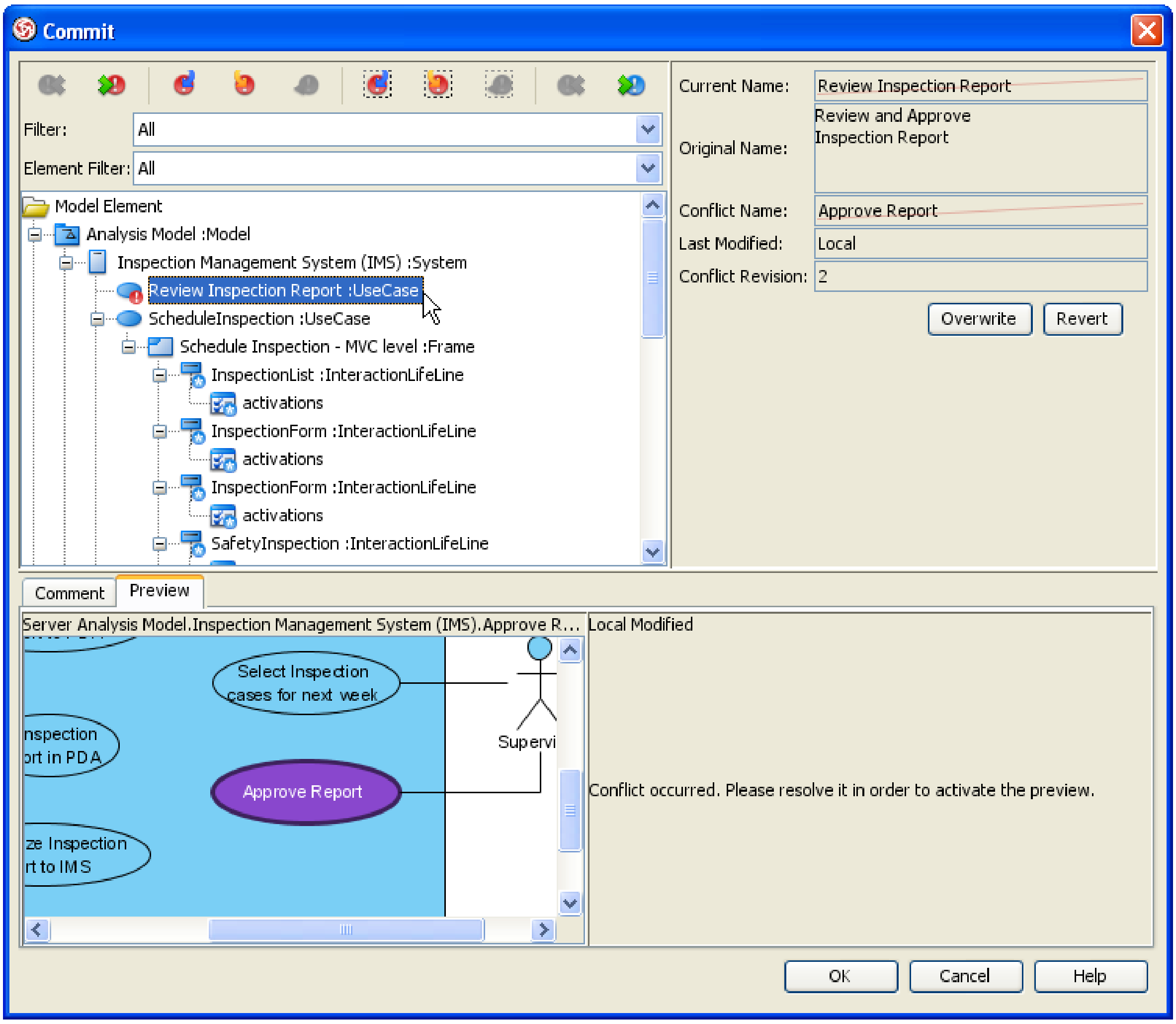

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Review Inspection Report
Inspector Assistant
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 4-21 *Modified project*

2. Click **Commit** button on the **Teamwork Toolbar**.

3. In **Commit** dialog, you may found some shapes or model elements show with red icon. This indicate there is conflict when commit, it is caused by someone modified the same content and commit after you checkout. You can review the current value, original value (the value when you checkout), and conflict value (the value changed by other users). And dialog preview is disabled until you resolve the conflict.



Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Review Inspection Report :UseCase
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionList :InteractionLifeLine
            - activations
          - InspectionForm :InteractionLifeLine
            - activations
          - InspectionForm :InteractionLifeLine
            - activations
          - SafetyInspection :InteractionLifeLine

Current Name: ~~Review Inspection Report~~

Original Name: Review and Approve Inspection Report

Conflict Name: ~~Approve Report~~

Last Modified: Local

Conflict Revision: 2

Overwrite
Revert
Comment
Preview
Server Analysis Model.Inspection Management System (IMS).Approve R...
Select Inspection cases for next week
nspection
ort in PDA
Approve Report
ze Inspection
rt to IMS
Supervi
Local Modified

Conflict occurred. Please resolve it in order to activate the preview.

Figure 4-22 *Commit with conflicts*

4. Select the conflict element, click **Overwrite selected conflicts** button on the toolbar to overwrite other user's change.

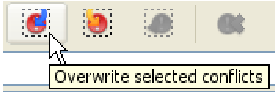


Figure 4-23 *Overwrite selected conflicts*

Or click **Revert selected conflicts** to revert your own changes.

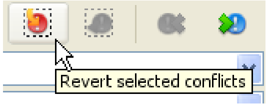


Figure 4-24 *Revert selected conflicts*

You can also click **Overwrite all conflicts** or **Revert all conflicts** to overwrite or revert all conflicts at once.

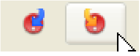


Figure 4-25 *Overwrite/Revert all conflicts*

5. After resolve conflict, the preview will be enabled to visualize the final result.

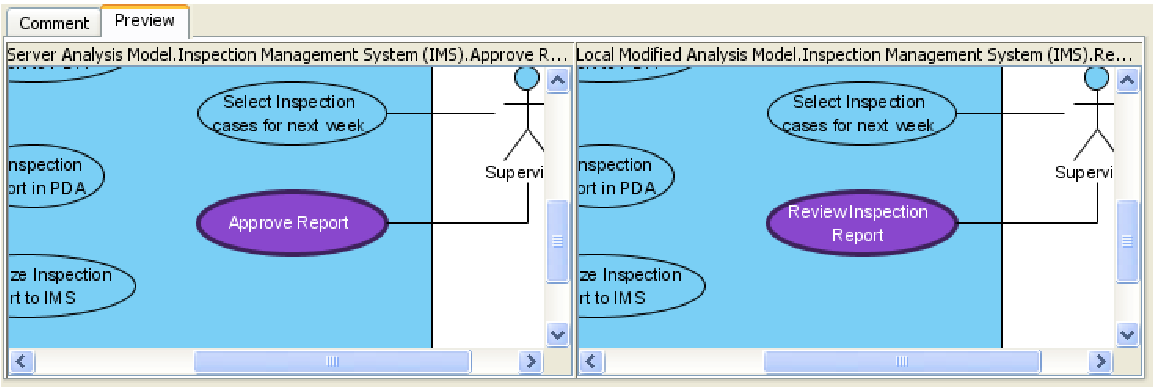

Comment
Preview
Server Analysis Model.Inspection Management System (IMS).Approve R...
Select Inspection cases for next week
nspection ort in PDA
Approve Report
ze Inspection rt to IMS
Supervi
Local Modified Analysis Model.Inspection Management System (IMS).Re...
Select Inspection cases for next week
nspection ort in PDA
ReviewInspection Report
ze Inspection rt to IMS
Supervi

Figure 4-26 *Preview resolved conflict*

6. You can change your mind by click **Reset selected conflicts** or **Reset all conflicts** button. Then overwrite or revert the conflict again.

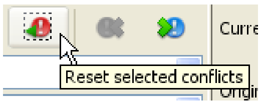


Figure 4-27 *Reset conflicts*

7. After all conflicts was resolved, you can now input comment and click **OK** button to commit.

Updating latest revision from Perforce

Updating modification from Perforce

1. Click **Update** button on **Teamwork Toolbar**.

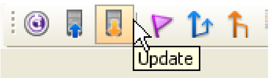
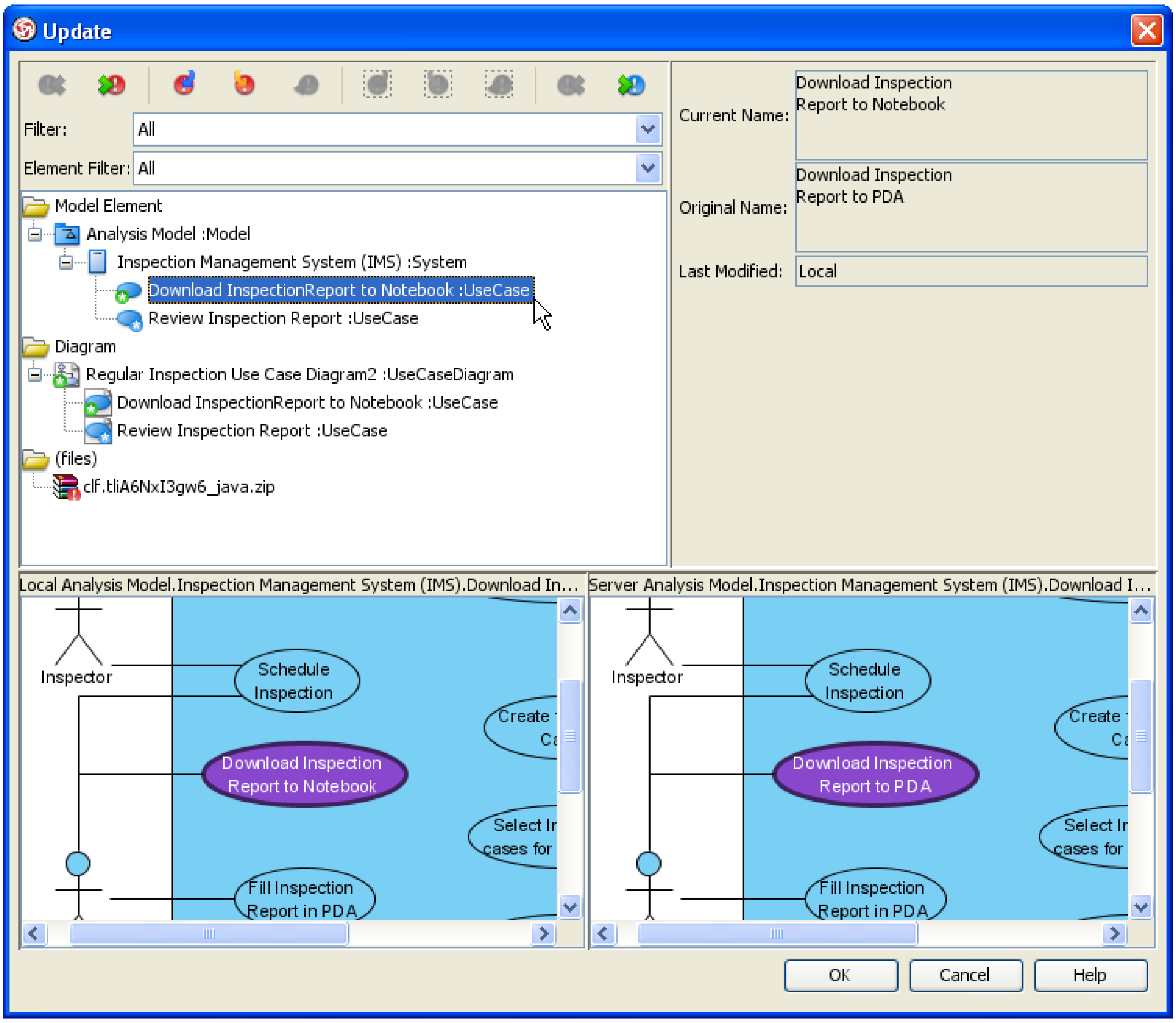


Figure 4-28 *Update from toolbar*

2. Similar to commit, you can review the change and preview the diagram in **Update** dialog.



Update

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Download InspectionReport to Notebook :UseCase
      - Review Inspection Report :UseCase
- Diagram
  - Regular Inspection Use Case Diagram2 :UseCaseDiagram
    - Download InspectionReport to Notebook :UseCase
    - Review Inspection Report :UseCase
- (files)
  - clf.tliA6NxI3gw6_java.zip

Current Name: Download Inspection Report to Notebook

Original Name: Download Inspection Report to PDA

Last Modified: Local

Local Analysis Model.Inspection Management System (IMS).Download In...

Server Analysis Model.Inspection Management System (IMS).Download I...

Inspector
Schedule Inspection
Create Ca
Download Inspection Report to Notebook
Select In cases for
Fill Inspection Report in PDA
Inspector
Schedule Inspection
Create Ca
Download Inspection Report to PDA
Select In cases for
Fill Inspection Report in PDA
OK
Cancel
Help

Figure 4-29 *Update dialog*

3. If there are conflicts, resolve it similar to commit, but the changes will apply to local project instead of commit to server immediately.

4. Click **OK** button to update. **VP-UML** will open the updated project.

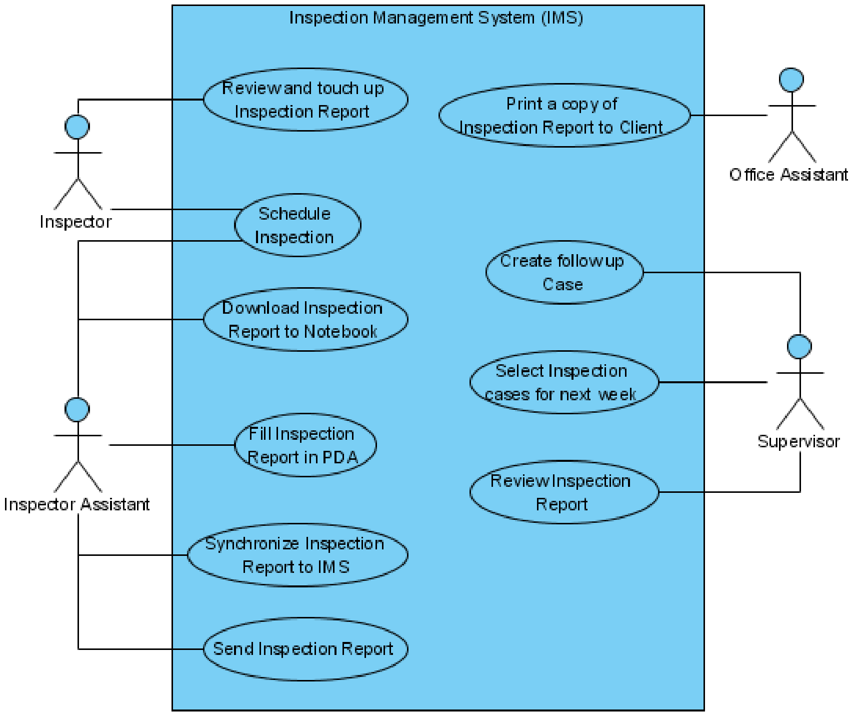

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to Notebook
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 4-30 *Updated project*

Checking status

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Check Update** from menu.

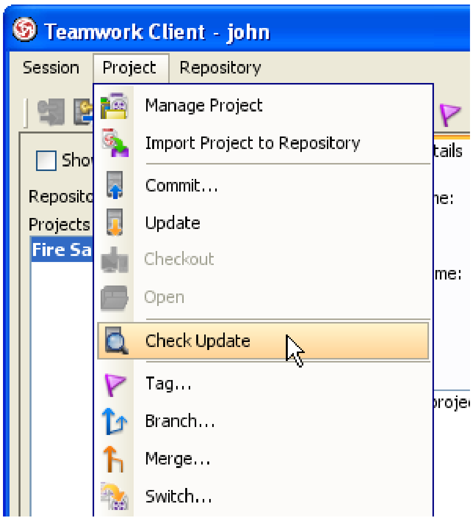


Figure 4-31 *Update project from menu*

Or click **Check Update** button from toolbar.

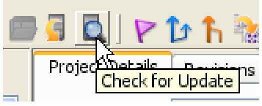
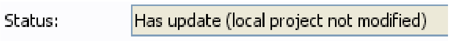


Figure 4-32 *Update project from toolbar*

4. You can see the status showing **Has update** or **Up-to-date**, it also indicate local project status **(local project not modified)** or **(local project modified)**.



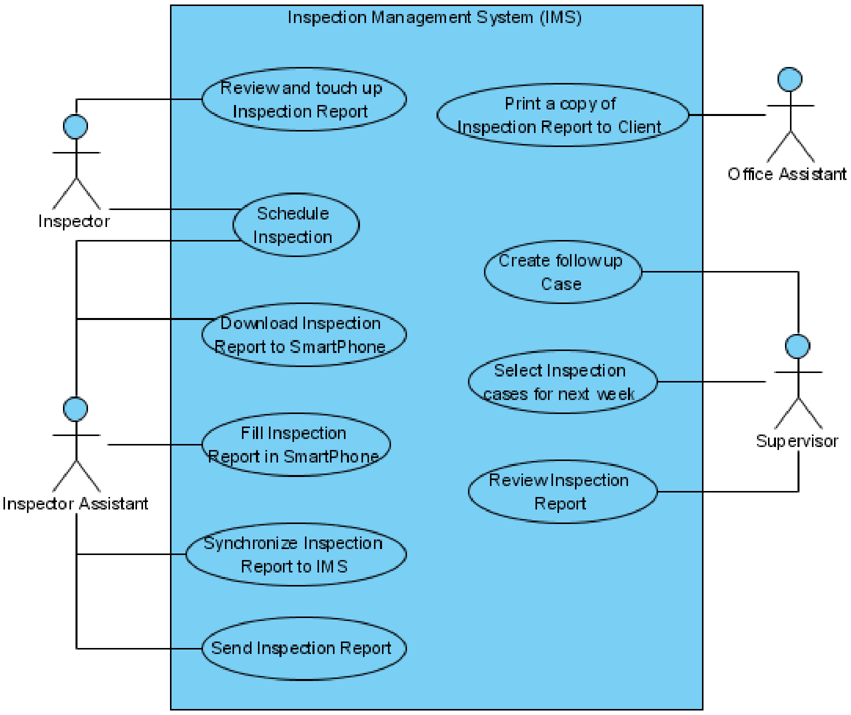
Status: Has update (local project not modified)

Figure 4-33 *Project status*

Reverting Changes

Revert local modifications

1. Modify the project.



Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to SmartPhone
Select Inspection cases for next week
Supervisor
Fill Inspection Report in SmartPhone
Review Inspection Report
Inspector Assistant
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 4-34 *Modify project*

2. Open **Teamwork Client** dialog.
3. Select the teamwork project in the list.

4. Select **Project > Revert...** from menu

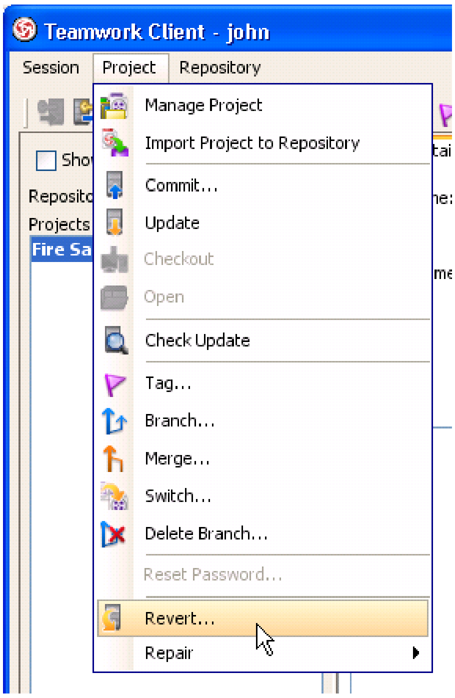


Figure 4-35 *Revert from menu*

Or click **Revert...** button from toolbar.

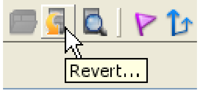


Figure 4-36 *Revert from toolbar*

5. Click **Yes** button from the confirmation dialog.
6. The reverted project opened in **VP-UML**.

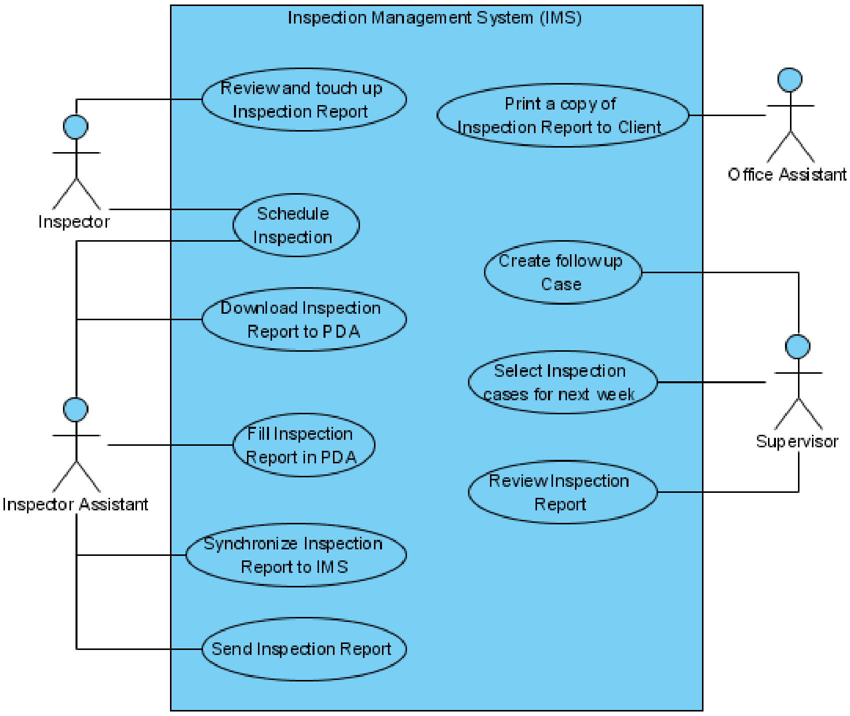

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 4-37 *Reverted project*

Revert server modifications

1. Open project.

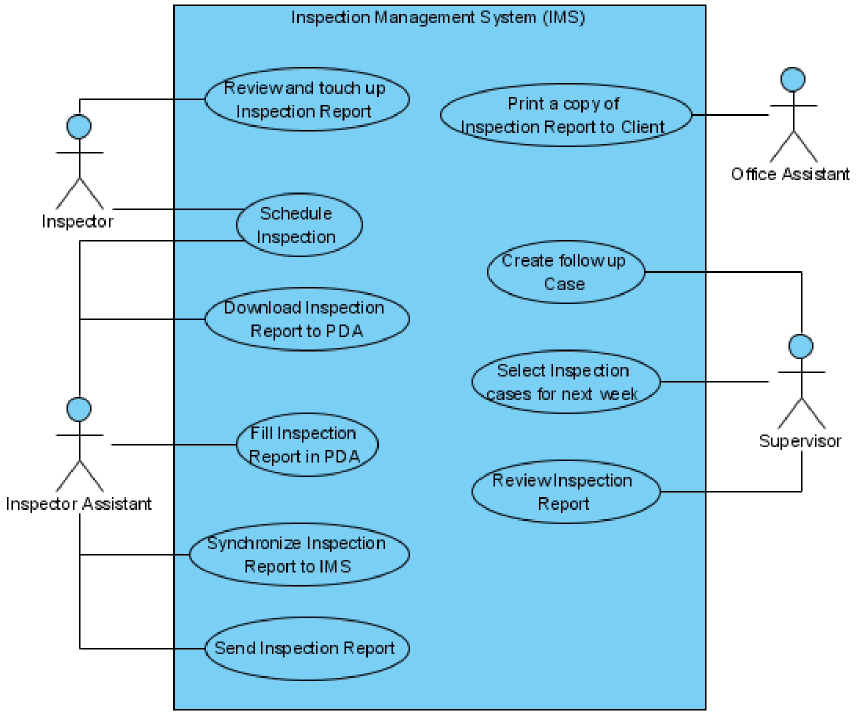
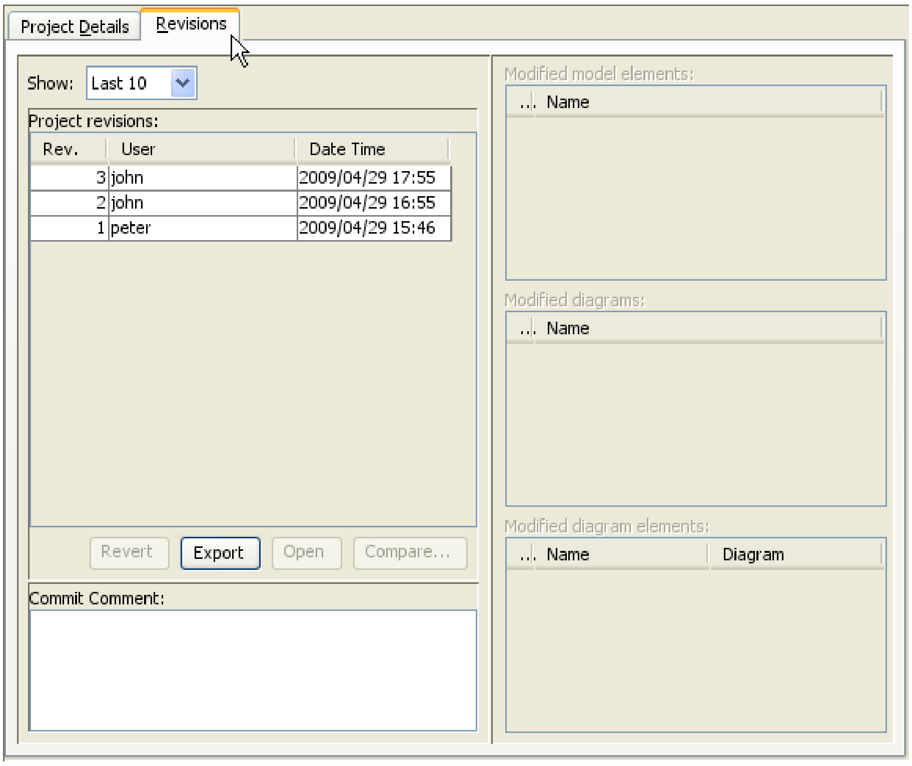

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 4-38 Open project

2. Open **Teamwork Client** dialog.
3. Select the teamwork project in the list.
4. Click the **Revisions** tab on the right of the project list.



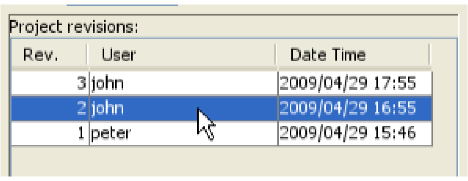
Show: Last 10

Project revisions:

Rev.	User	Date Time
3	john	2009/04/29 17:55
2	john	2009/04/29 16:55
1	peter	2009/04/29 15:46

Figure 3-39 Revision tab

5. Select the revision to revert.



Project revisions:

Rev.	User	Date Time
3	john	2009/04/29 17:55
2	john	2009/04/29 16:55
1	peter	2009/04/29 15:46

Figure 3-40 Select revision

6. Click **Revert** button.

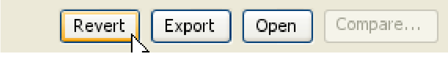
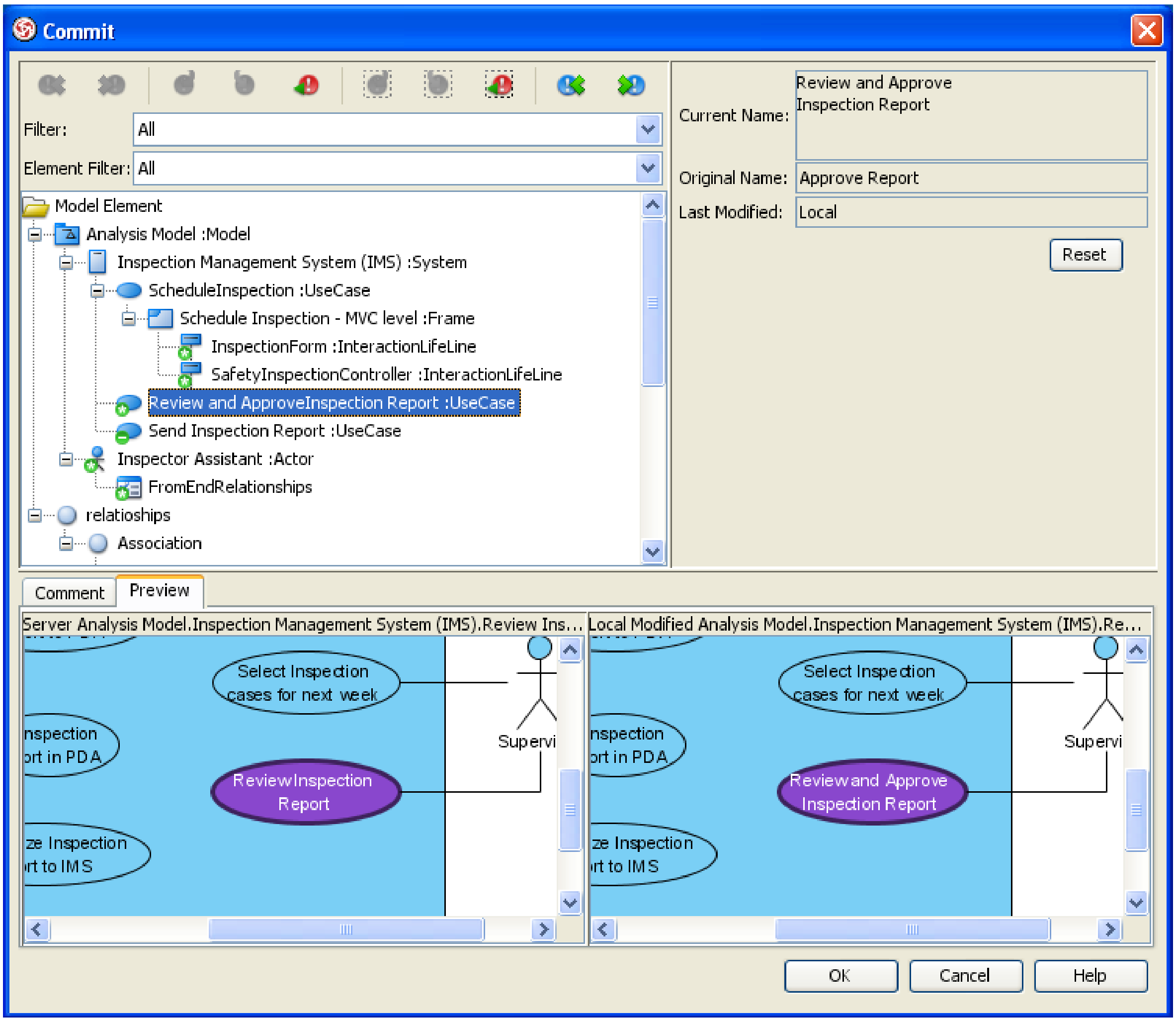


Figure 3-41 Revert button

7. The **Commit** dialog show a list of shapes and model elements reverted for review the changes and preview diagram, input the comment and click **OK** button to commit the revert.


Commit
Model Element
Analysis Model :Model
Inspection Management System (IMS) :System
ScheduleInspection :UseCase
Schedule Inspection - MVC level :Frame
InspectionForm :InteractionLifeLine
SafetyInspectionController :InteractionLifeLine
Review and ApproveInspection Report :UseCase
Send Inspection Report :UseCase
Inspector Assistant :Actor
FromEndRelationships
relatioships
Association
Reset
Comment
Preview
Server Analysis Model.Inspection Management System (IMS).Review Ins...
Local Modified Analysis Model.Inspection Management System (IMS).Re...
Select Inspection cases for next week
nspection ort in PDA
Review Inspection Report
ze Inspection rt to IMS
Supervi
Review and Approve Inspection Report
OK
Cancel
Help

Filter: All

Element Filter: All

Current Name: Review and Approve Inspection Report

Original Name: Approve Report

Last Modified: Local

Figure 4-42 Commit dialog with reverted changes

8. A new revision with the reverted changes was created in server, and opened in **VP-UML**.

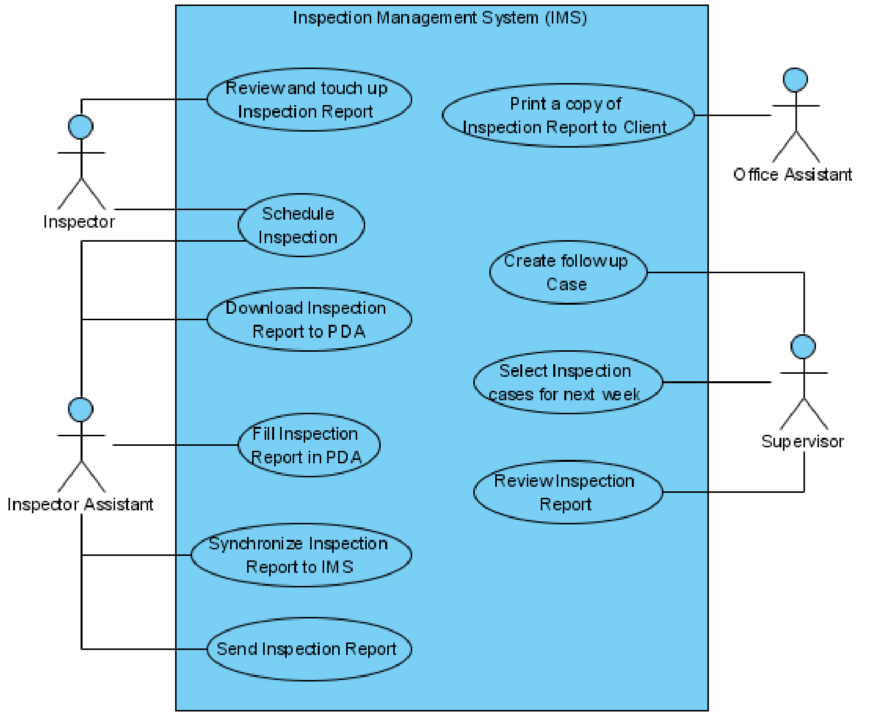

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review Inspection Report
Synchronize Inspection Report to IMS
Send Inspection Report

Figure 4-43 *Reverted project*

Browsing change histories (old revisions)

Checkout old revisions

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select the revision to open.
5. Click Open button.

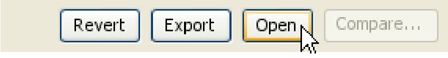
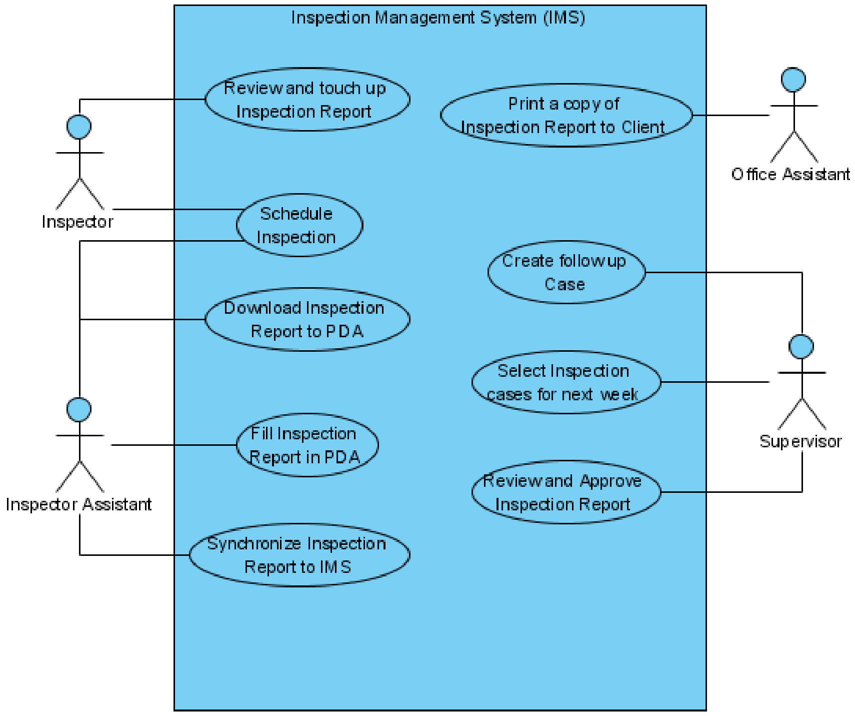


Figure 4-44 *Open button*

6. Selected revision opened in **VP-UML**.

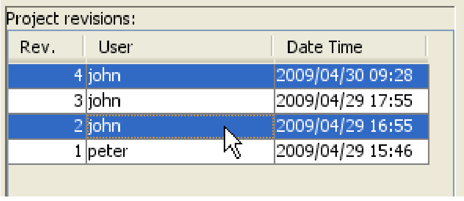


Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS

Figure 4-45 *Open selected revision*

Showing differences between revisions visually

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select a revision for compare.
5. Select another revision by **Ctrl+Click**.



Project revisions:

Rev.	User	Date Time
4	john	2009/04/30 09:28
3	john	2009/04/29 17:55
2	john	2009/04/29 16:55
1	peter	2009/04/29 15:46

Figure 4-46 *Select two revisions*

6. Click the **Compare...** button.

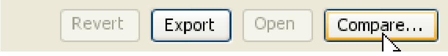
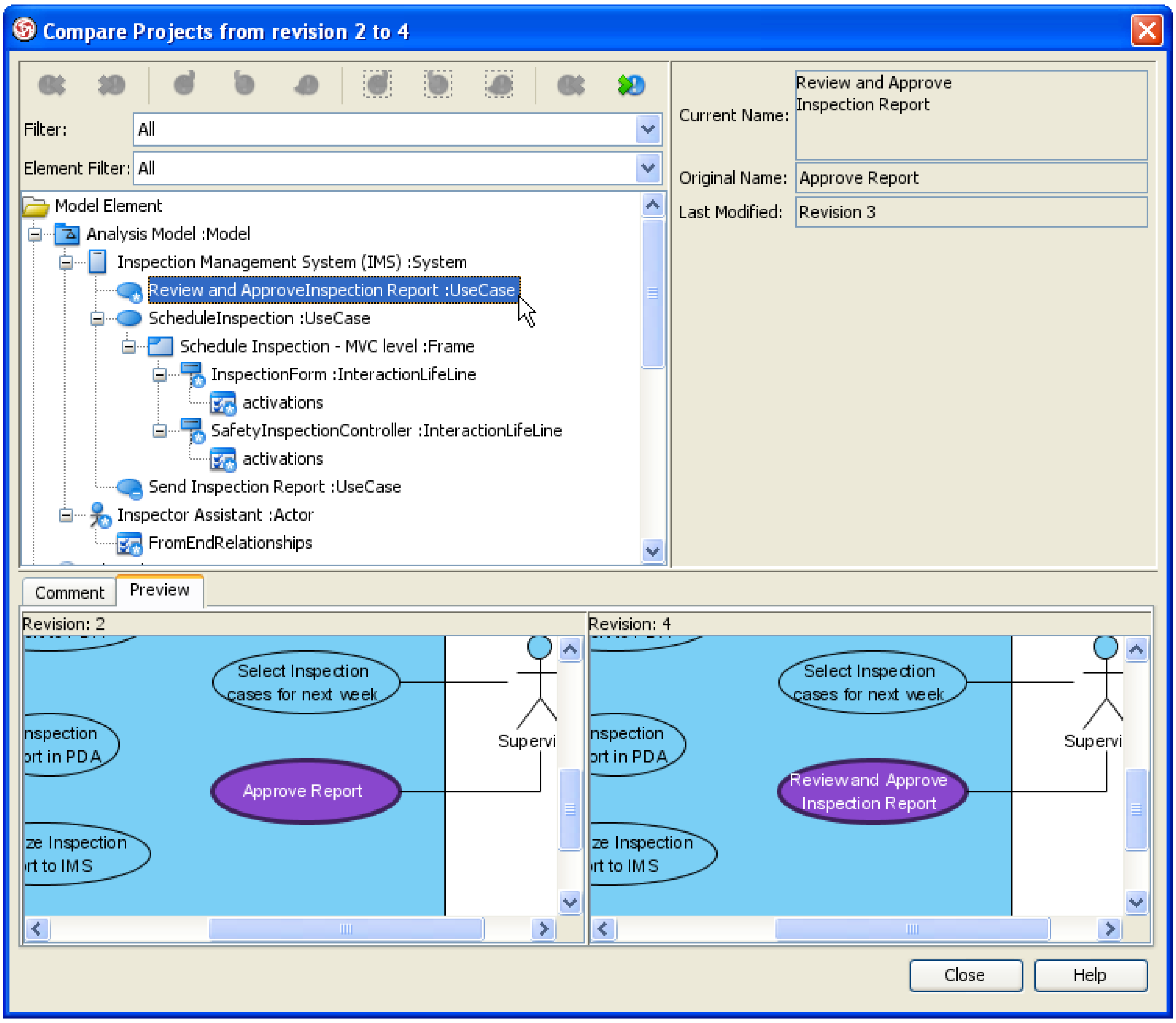


Figure 4-47 *Compare button*

7. Similar to **Commit** and **Update** dialog, the **Compare Projects** dialog show a list of differences between the selected revisions. You can also view the differences visually in diagram on the preview tab.



Compare Projects from revision 2 to 4

Filter: All

Element Filter: All

- Model Element
  - Analysis Model :Model
    - Inspection Management System (IMS) :System
      - Review and ApproveInspection Report :UseCase
      - ScheduleInspection :UseCase
        - Schedule Inspection - MVC level :Frame
          - InspectionForm :InteractionLifeLine
            - activations
          - SafetyInspectionController :InteractionLifeLine
            - activations
      - Send Inspection Report :UseCase
    - Inspector Assistant :Actor
      - FromEndRelationships

Current Name: Review and Approve Inspection Report

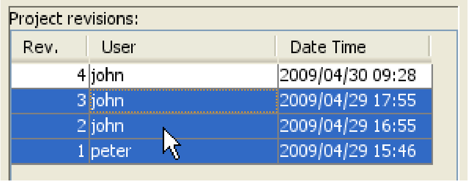
Original Name: Approve Report

Last Modified: Revision 3

Figure 4-48 Compare differences

Export multiple revisions to local

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Click the **Revisions** tab.
4. Select multiple revisions for export, by **Ctrl+Click** or **Shift+Click**.



Project revisions:

Rev.	User	Date Time
4	john	2009/04/30 09:28
3	john	2009/04/29 17:55
2	john	2009/04/29 16:55
1	peter	2009/04/29 15:46

Figure 4-49 Select multiple revisions

5. Click the **Export** button.

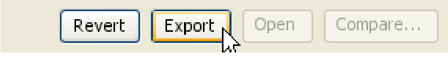
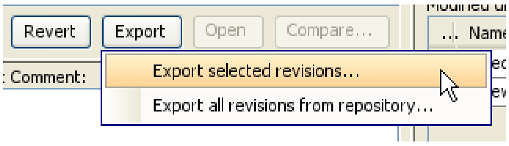


Figure 4-50 Export buttons

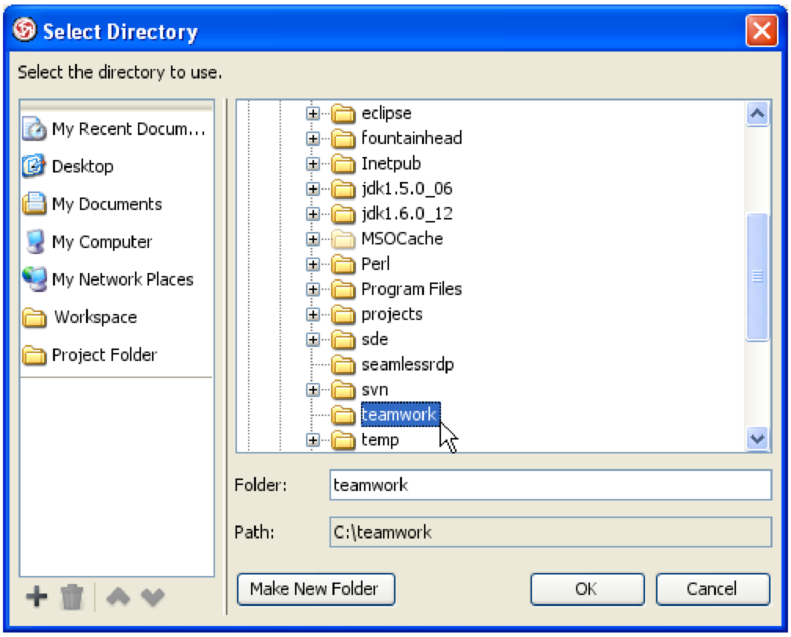
6. Select **Export selected revisions...** from the popup.



Revert
Export
Open
Compare...
Comment:
... Name
Export selected revisions...
Export all revisions from repository...

Figure 4-51 Export selected revisions

7. Select the directory to save the exported project revisions.



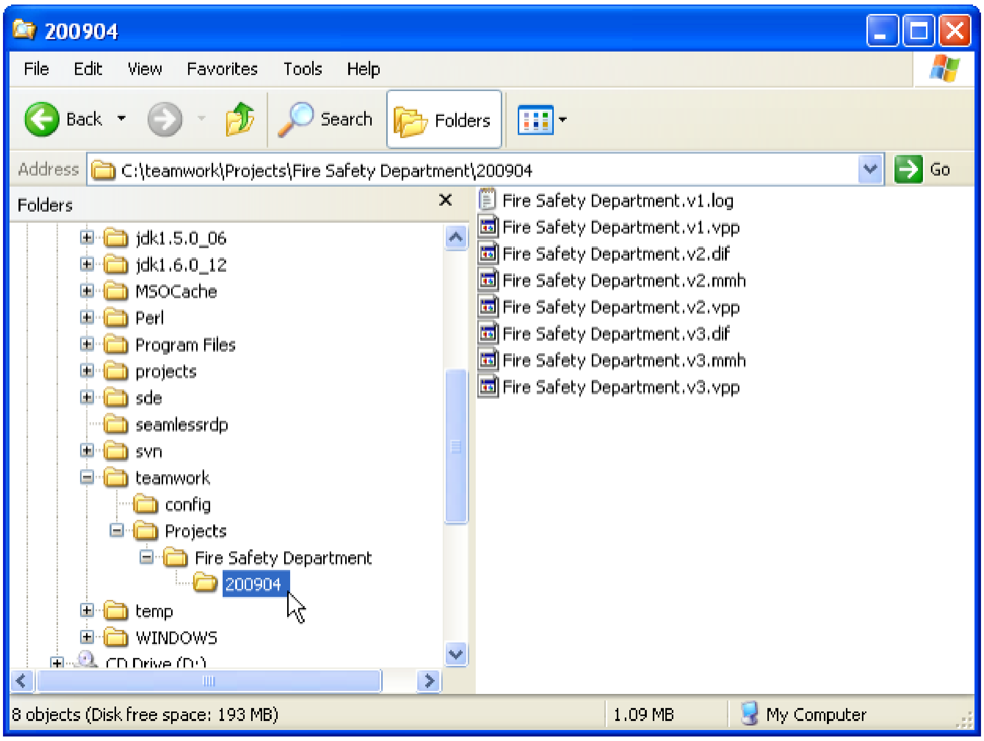
Select the directory to use.

Folder: teamwork

Path: C:\teamwork

__Figure 4-52__ Select directory

8. You'll found the selected revisions was exported to the select directory.



Address C:\teamwork\Projects\Fire Safety Department\200904

Fire Safety Department.v1.log
Fire Safety Department.v1.vpp
Fire Safety Department.v2.dif
Fire Safety Department.v2.mmh
Fire Safety Department.v2.vpp
Fire Safety Department.v3.dif
Fire Safety Department.v3.mmh
Fire Safety Department.v3.vpp

8 objects (Disk free space: 193 MB) 1.09 MB My Computer

__Figure 4-53__ Exported revisions

Isolating last long modifications with Branches

Creating Branch

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Branch...** from menu

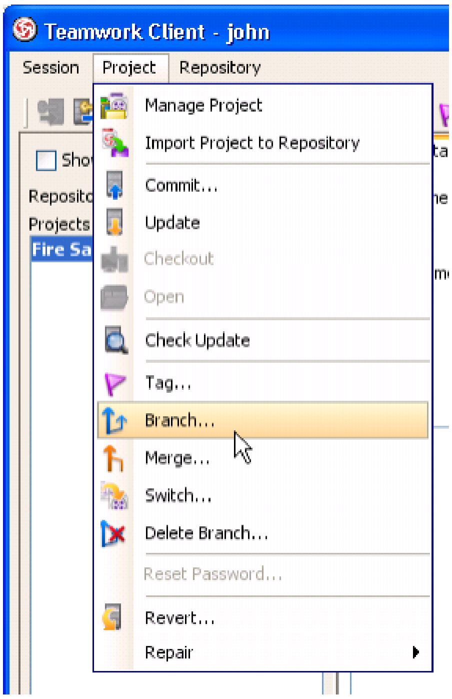


Figure 4-54 Branch from menu

Or click **Branch...** button from toolbar.

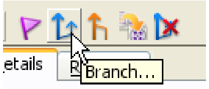
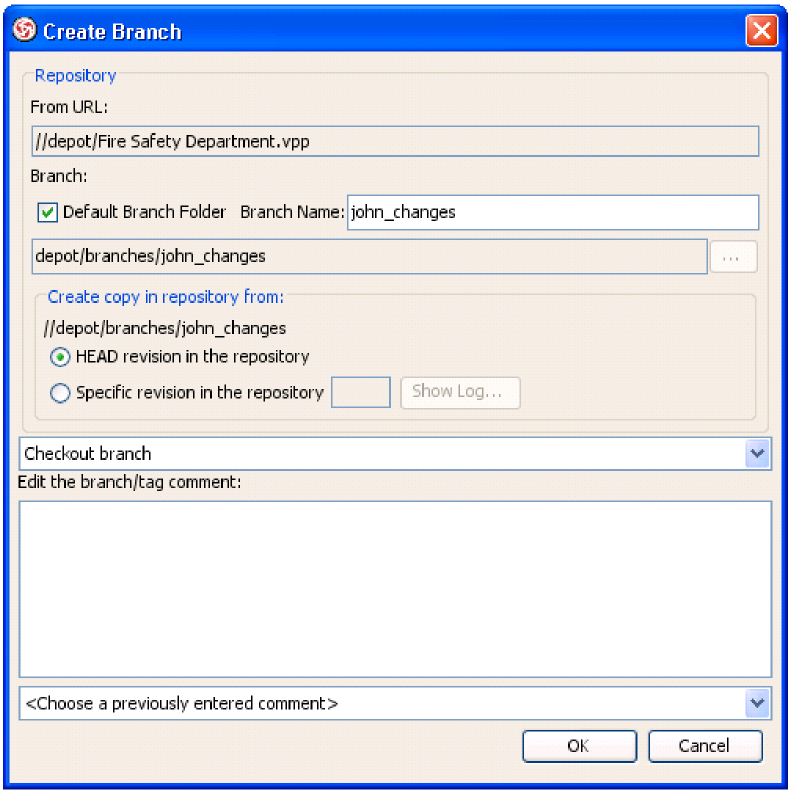


Figure 4-55 Branch from toolbar

4. Fill in the **Branch Name** in **Create Branch** dialog.



Create Branch

Repository

From URL:

//depot/Fire Safety Department.vpp

Branch:

☑ Default Branch Folder Branch Name: john_changes

depot/branches/john_changes

Create copy in repository from:

//depot/branches/john_changes

● HEAD revision in the repository

○ Specific revision in the repository

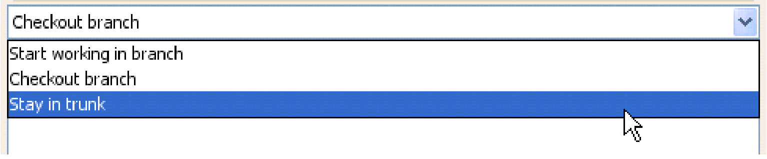
Checkout branch

Edit the branch/tag comment:

<Choose a previously entered comment>

Figure 4-56 *Create Branch dialog*

5. Select **Stay in trunk** in the list.



Checkout branch

Start working in branch

Checkout branch

Stay in trunk

Figure 4-57 *Stay in trunk*

Switch local copy between Branches

1. Select **Project > Switch...** from menu

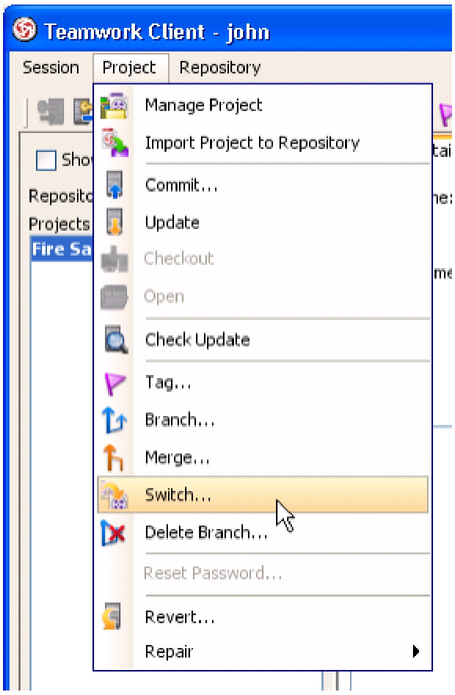


Figure 4-58 *Switch from menu*

Or click **Switch...** button from toolbar.

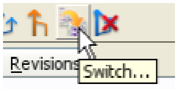
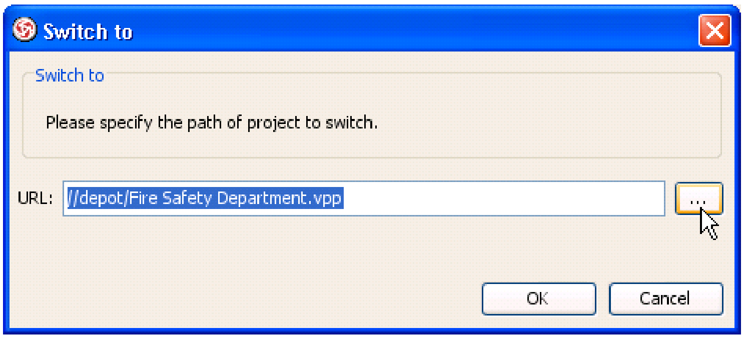


Figure 4-59 *Switch from toolbar*

2. Click **...** to browser the path.


Switch to

Switch to

Please specify the path of project to switch.

URL: //depot/Fire Safety Department.vpp

Figure 4-60 *Select branch*

3. Select the branch to switch, and click **OK**.

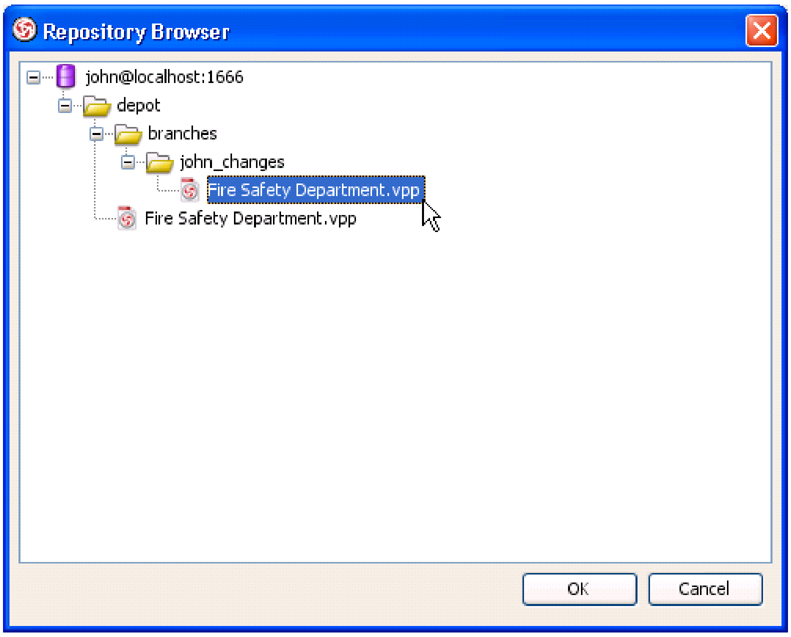


Figure 4-61 *Select branch path*

4. The branch is opened in **VP-UML**.

Merging from Trunk To Branch

1. Open, modify and commit trunk project.

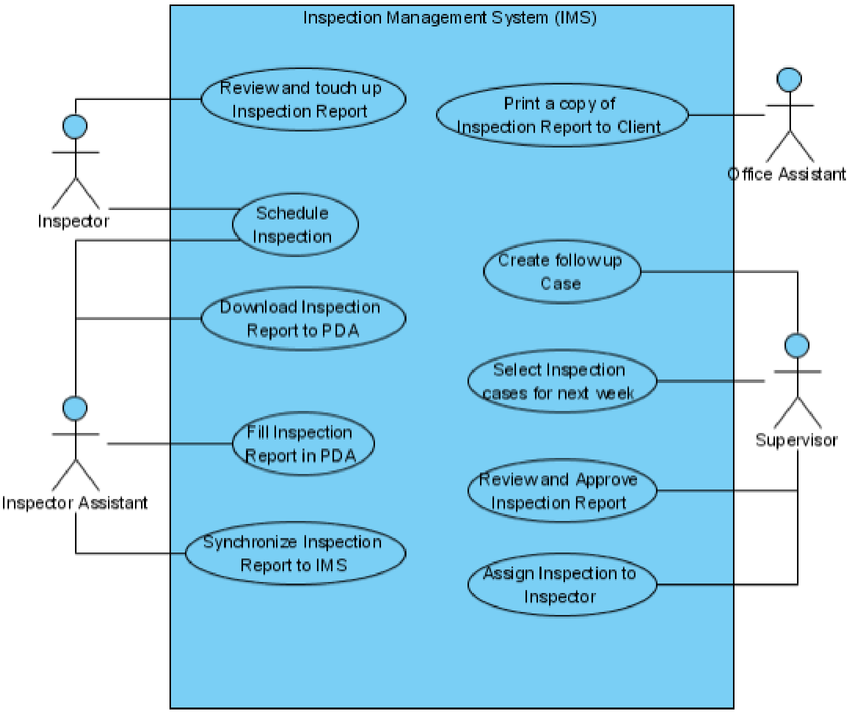

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Inspector Assistant
Fill Inspection Report in PDA
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 4-62 *Modified trunk project*

2. Open branch project.

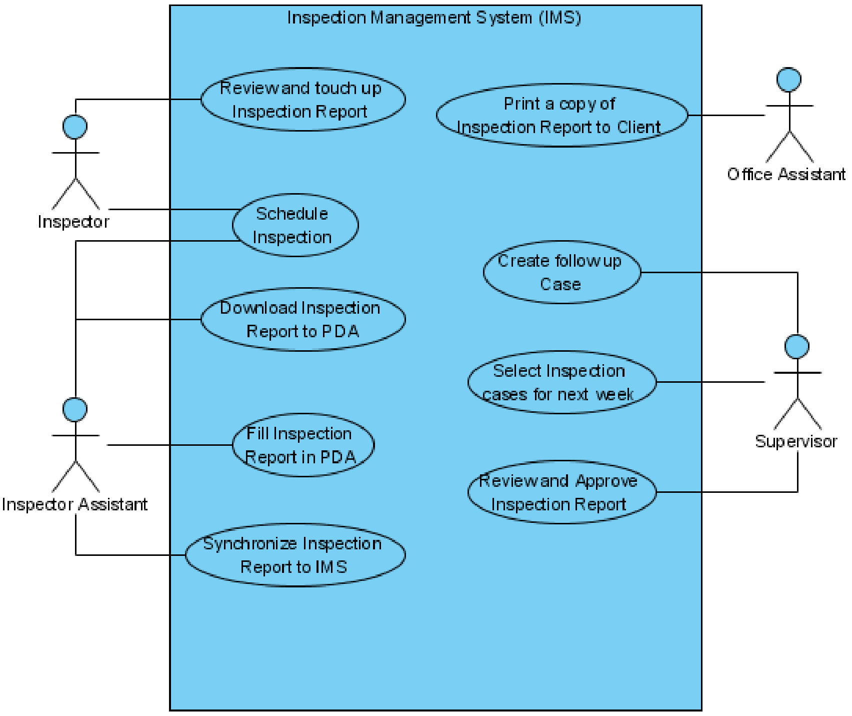

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS

Figure 4-63 Branch project

3. Select **Project > Merge...** from menu.

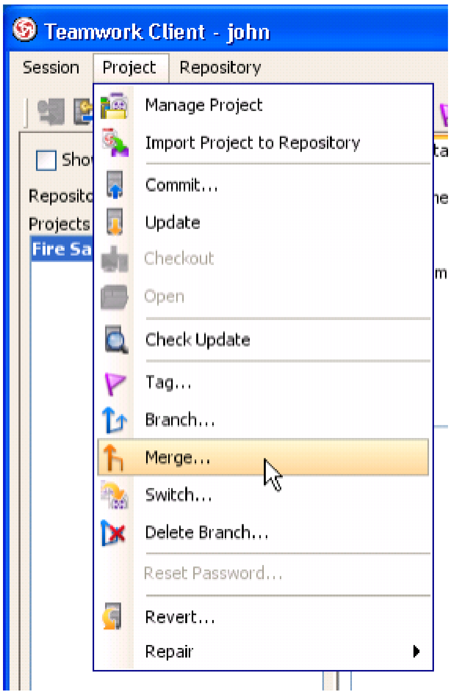


Figure 4-64 Merge from menu

Or click **Merge...** button from toolbar.

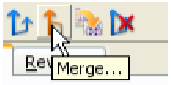
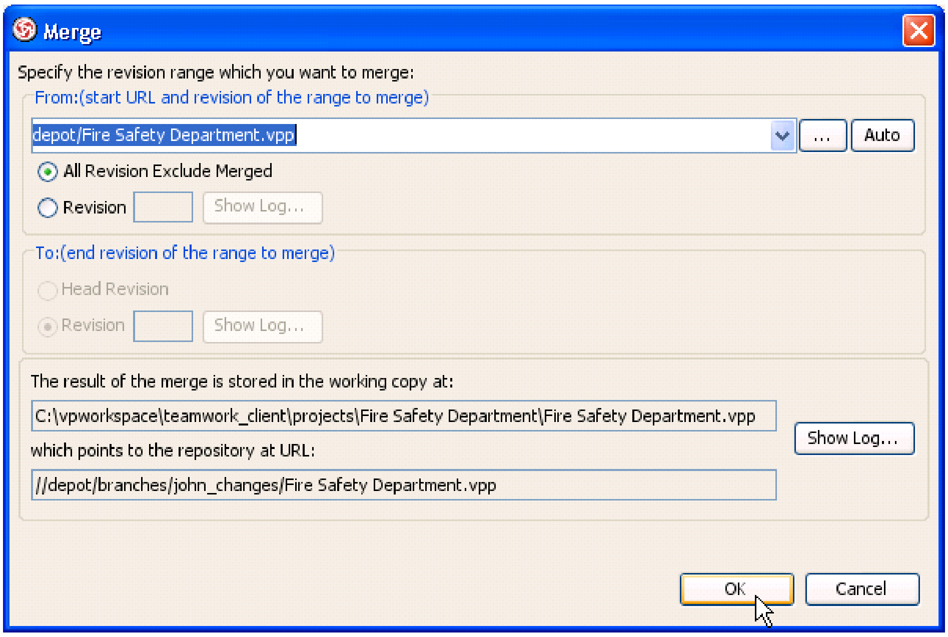


Figure 4-65 Merge from toolbar

4. Select the trunk path(*depot/FireSafetyDepartment.vpp*) as **From** in the **Merge** dialog, and press **OK**.



Merge

Specify the revision range which you want to merge:

From:(start URL and revision of the range to merge)

depot/Fire Safety Department.vpp

● All Revision Exclude Merged

○ Revision

To:(end revision of the range to merge)

○ Head Revision

● Revision

The result of the merge is stored in the working copy at:

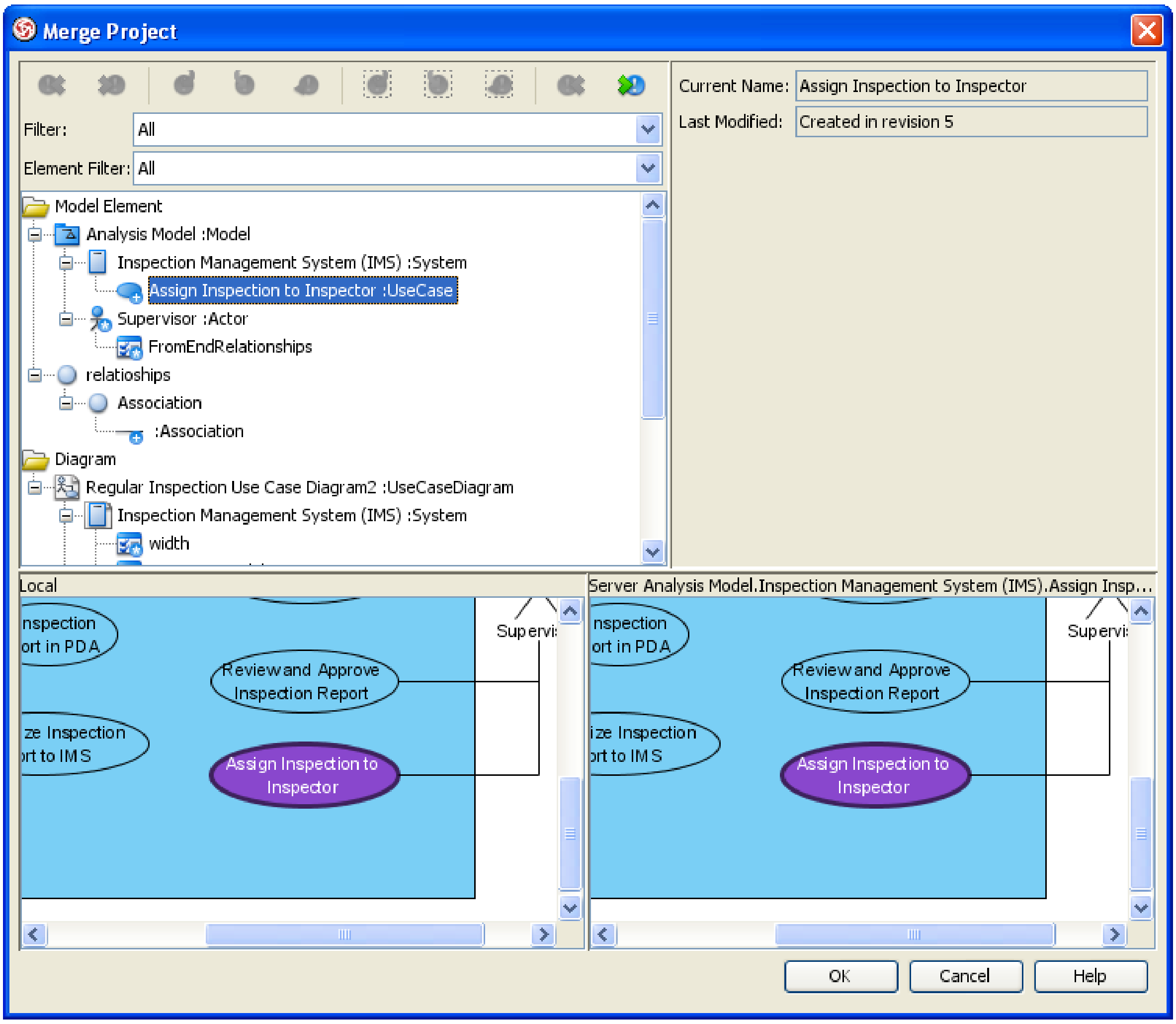
C:\vpworkspace\teamwork_client\projects\Fire Safety Department\Fire Safety Department.vpp

which points to the repository at URL:

//depot/branches/john_changes/Fire Safety Department.vpp

Figure 4-66 *Merge dialog*

5. The **Merge Project** dialog show a list of changes similar to **Commit** and **Update** dialog.



Merge Project

Filter: All

Element Filter: All

Current Name: Assign Inspection to Inspector

Last Modified: Created in revision 5

```
Model Element
  Analysis Model :Model
    Inspection Management System (IMS) :System
      Assign Inspection to Inspector :UseCase
    Supervisor :Actor
      FromEndRelationships
  relatioships
    Association
      :Association
Diagram
  Regular Inspection Use Case Diagram2 :UseCaseDiagram
    Inspection Management System (IMS) :System
      width
```

Local

Server Analysis Model.Inspection Management System (IMS).Assign Insp...

nspection
ort in PDA
Review and Approve Inspection Report
ze Inspection
ort to IMS
Assign Inspection to Inspector
Supervi
nspection
ort in PDA
Review and Approve Inspection Report
ize Inspection
ort to IMS
Assign Inspection to Inspector
Supervi
OK
Cancel
Help

Figure 4-67 *Merge project dialog*

6. Finally, review the changes in **Commit** dialog. Input comment and click **OK** to commit.

7. The branch project now contains the changes from trunk.

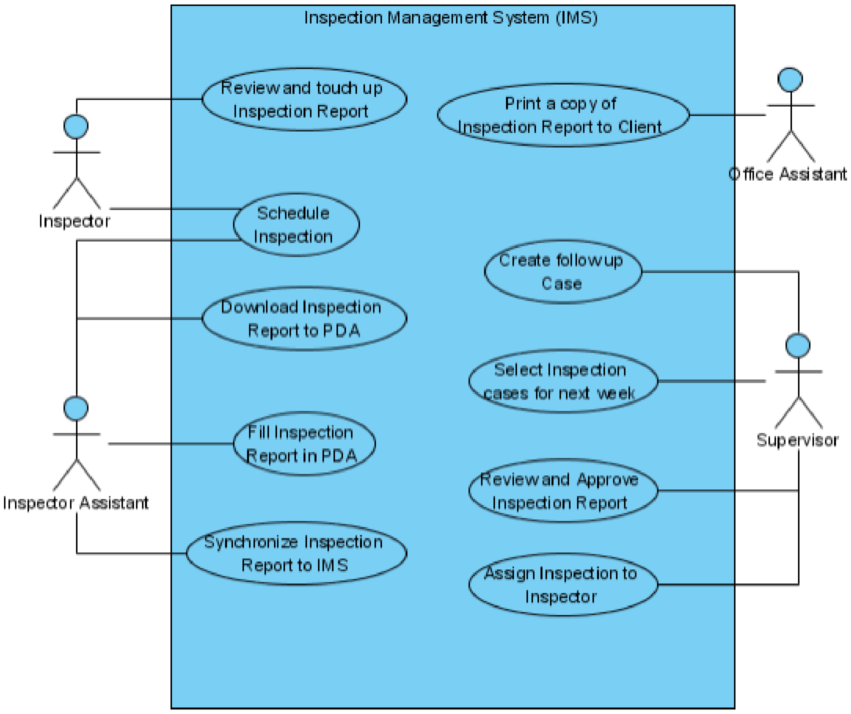

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 4-68 *Branch project merged from trunk*

Merging from Branch to Trunk

1. Open, modify and commit branch project.

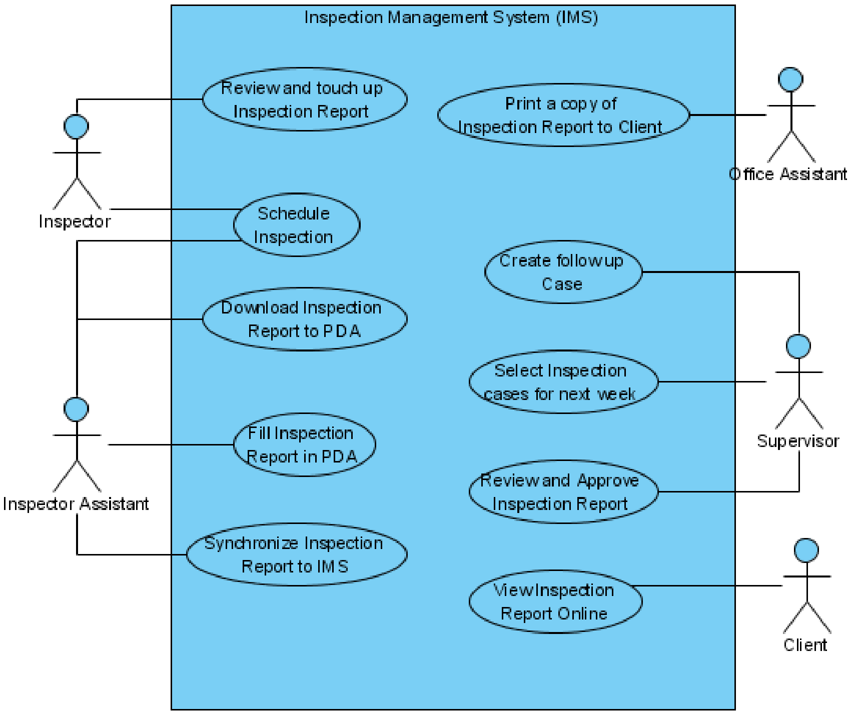

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
View Inspection Report Online
Client

Figure 4-69 *Modified branch project*

2. Open trunk project.

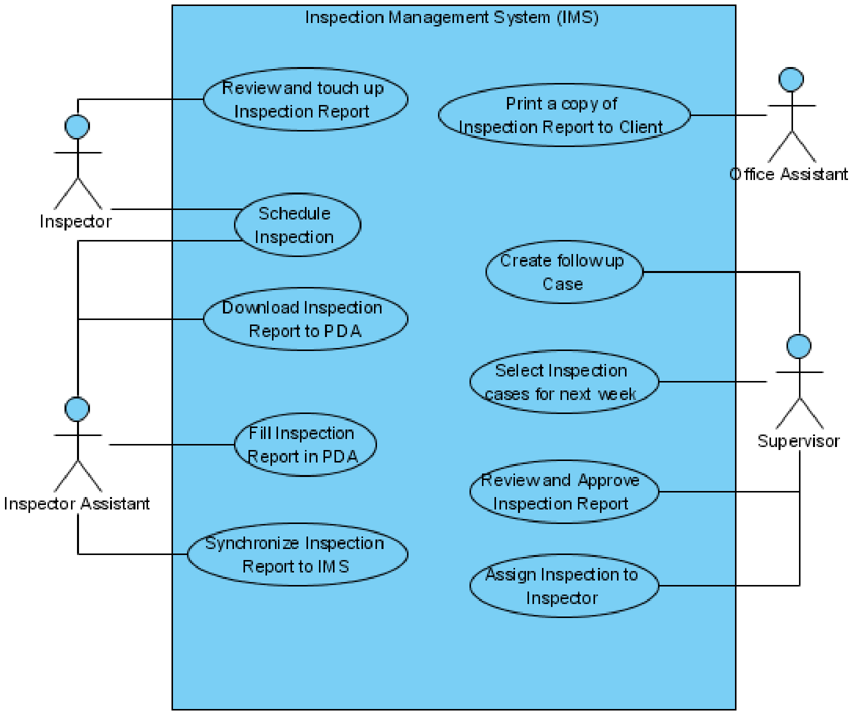
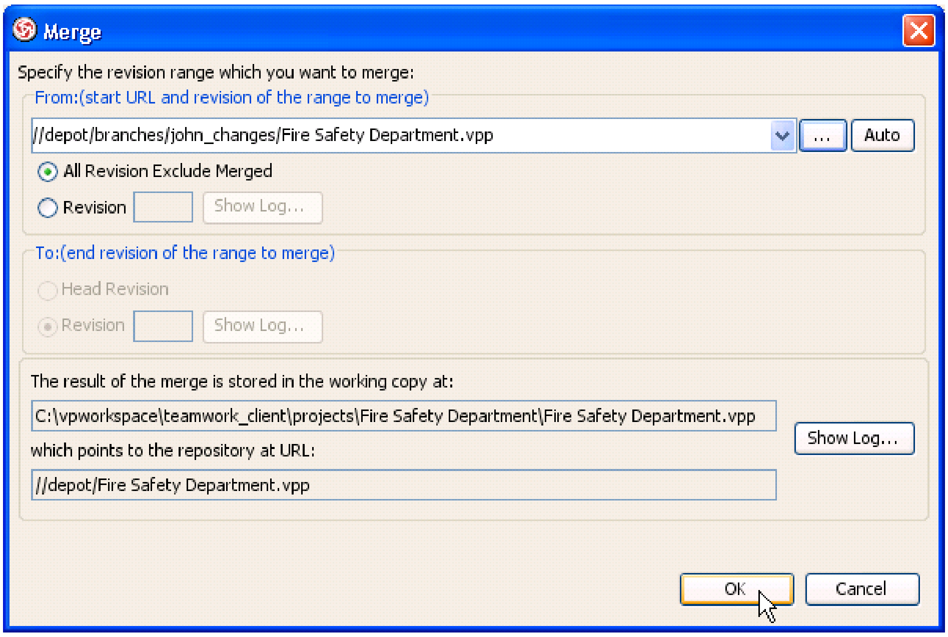

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Assign Inspection to Inspector

Figure 4-70 *Trunk project*

3. Select **Project > Merge...** from menu, or click **Merge...** button from toolbar.
4. Select the branch path(*//depot/branches/john_changes/FireSafetyDepartment.vpp*) as **From** in the **Merge** dialog, and press **OK**.


Merge

Specify the revision range which you want to merge:

From:(start URL and revision of the range to merge)

//depot/branches/john_changes/Fire Safety Department.vpp

● All Revision Exclude Merged

○ Revision

To:(end revision of the range to merge)

○ Head Revision

● Revision

The result of the merge is stored in the working copy at:

C:\vpworkspace\teamwork_client\projects\Fire Safety Department\Fire Safety Department.vpp

which points to the repository at URL:

//depot/Fire Safety Department.vpp

Figure 4-71 *Merge dialog*

5. The **Merge Project** dialog show a list of changes similar to **Commit** and **Update** dialog.
6. Finally, review the changes in **Commit** dialog. Input comment and click **OK** to commit.

7. The trunk project now contains the changes from branch.

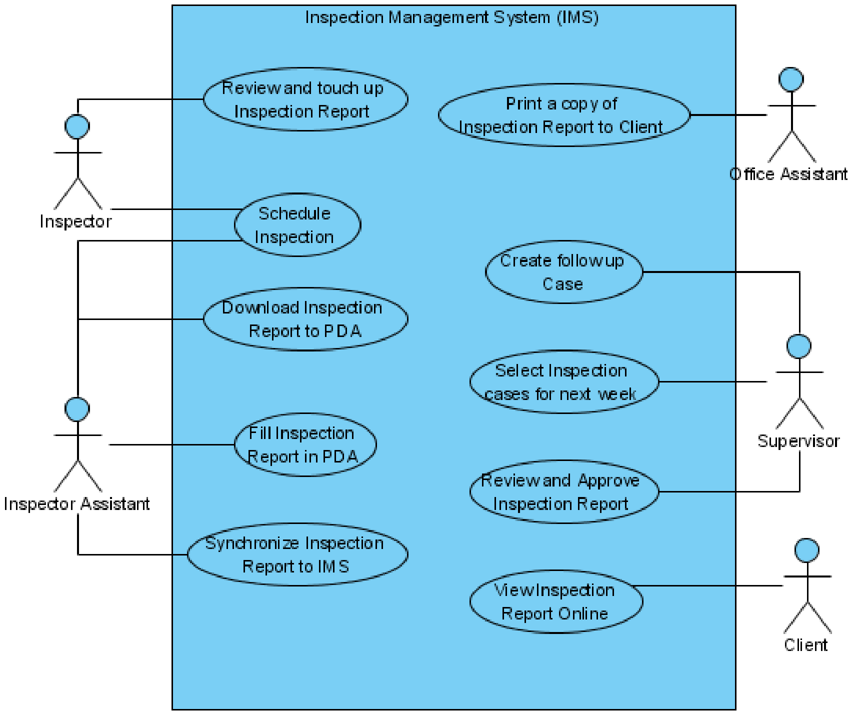

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Office Assistant
Inspector
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Supervisor
Fill Inspection Report in PDA
Inspector Assistant
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
View Inspection Report Online
Client

Figure 4-72 Trunk project merged from branch

Delete Branch

1. Select **Project > Delete Branch...** from menu

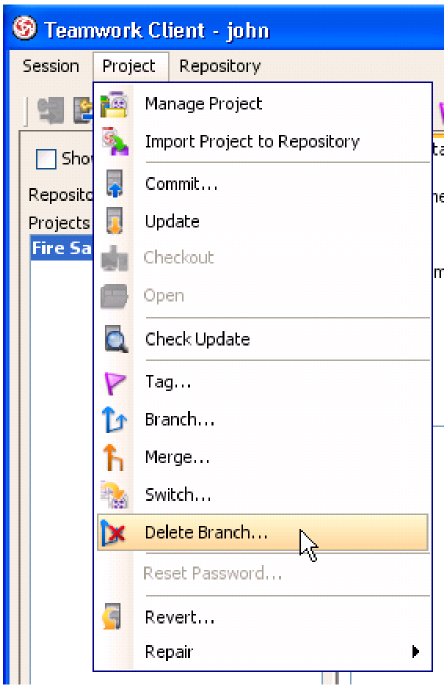


Figure 4-73 Delete Branch from menu

Or click **Delete Branch ...** button from toolbar.

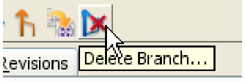
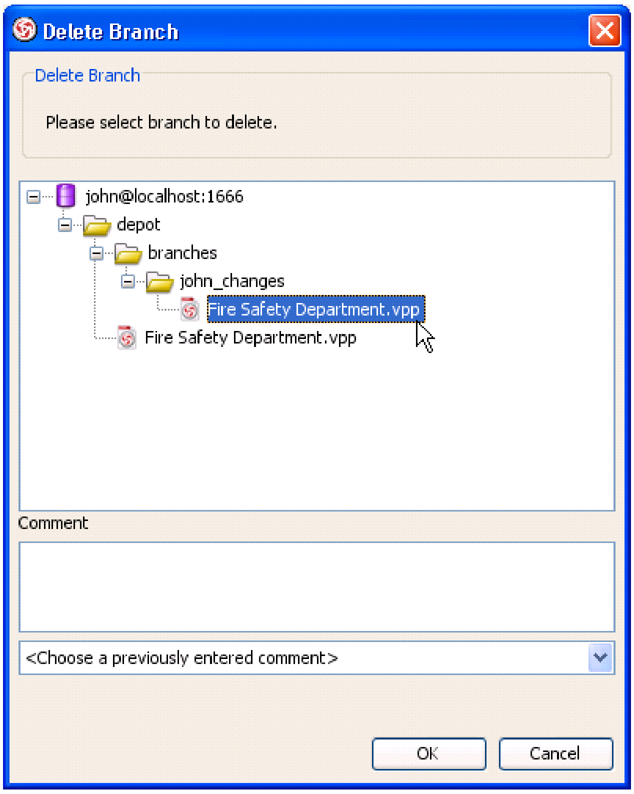


Figure 4-74 Delete Branch from toolbar

2. Expand the repository and folder node, select the branch to delete. Click **OK** button to continue.



Please select branch to delete.

<Choose a previously entered comment>

Figure 4-75 *Delete branch dialog*

3. Click **Yes** button on confirmation dialog.

Marking release or milestone with Tags

Tags and branches are almost the same, with the only difference - tags are read only. Creating read only tags ensure you are able open the release version project later.

1. Open **Teamwork Client** dialog.
2. Select the teamwork project in the list.
3. Select **Project > Tag...** from menu

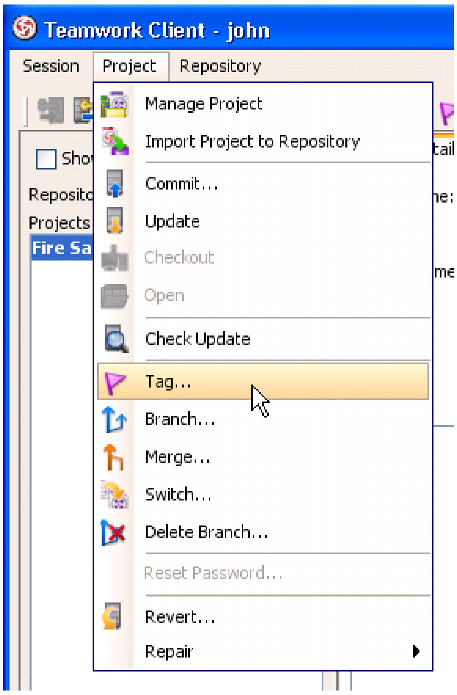


Figure 4-76 *Tag from menu*

Or click **Branch...** button from toolbar.

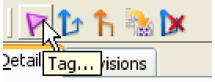
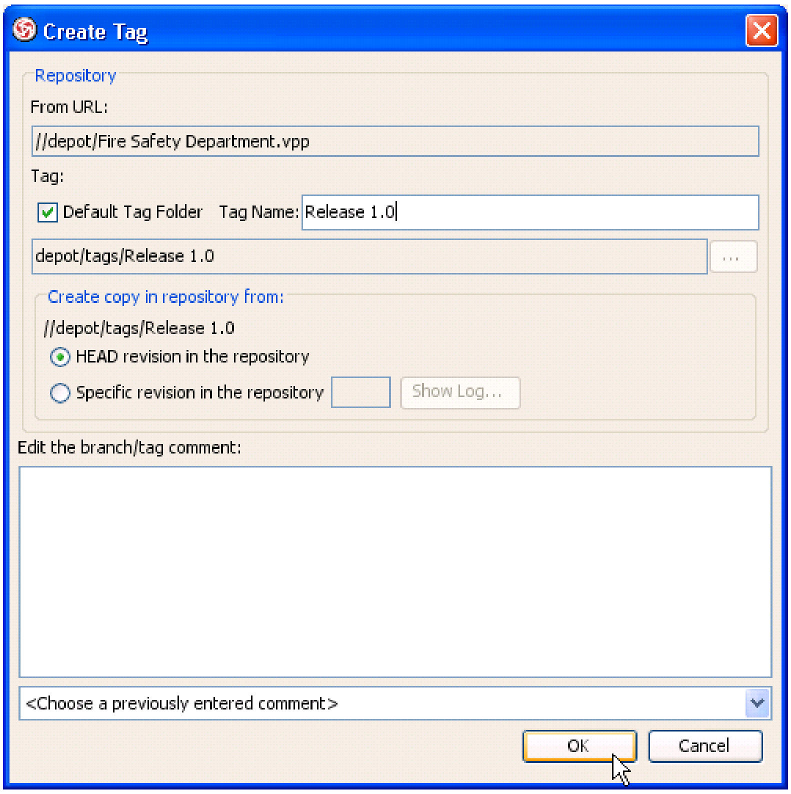


Figure 4-77 *Tag from toolbar*

4. Fill in the **Tag Name** in **Create Tag** dialog.



Create Tag

Repository

From URL: //depot/Fire Safety Department.vpp

Tag:

☑ Default Tag Folder Tag Name: Release 1.0

depot/tags/Release 1.0

Create copy in repository from:

//depot/tags/Release 1.0

● HEAD revision in the repository

○ Specific revision in the repository

Edit the branch/tag comment:

<Choose a previously entered comment>

Figure 4-78 Create Tag dialog

5. Click **OK** button, the tag will be created.

Generating PDF Report

The Generate PDF dialog box provides a set of options for changing the report style. To display the dialog box, perform one of the following actions:

Select Tools > Report > Generate PDF Report... from main menu.

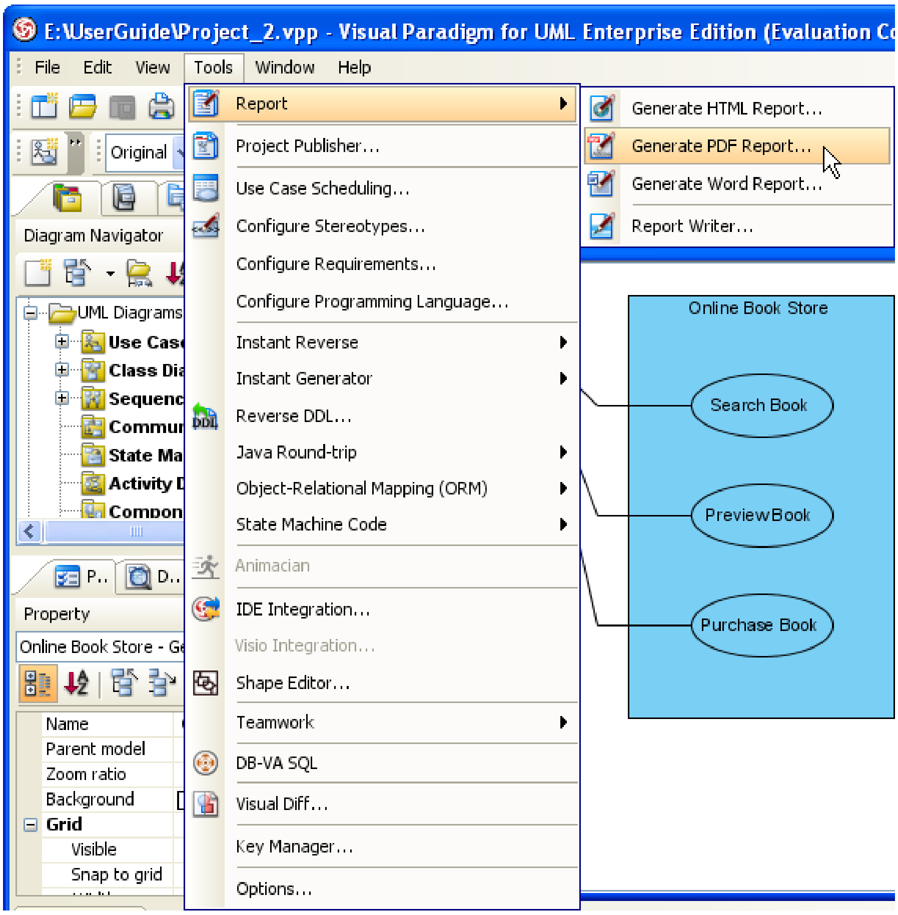


Figure 1-1 *Generate PDF from main menu*

Or click on the Generate PDF Report button on the toolbar.

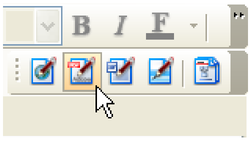
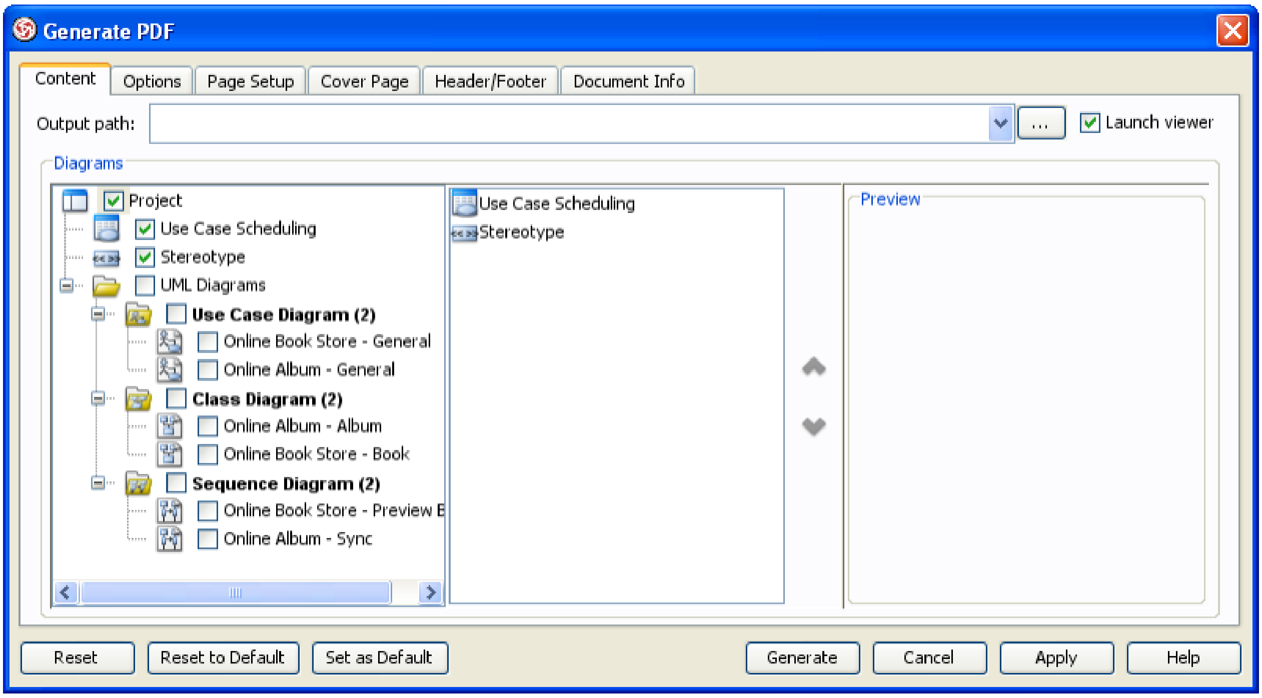


Figure 1-2 *Generate PDF from tool bar*

Generate PDF dialog



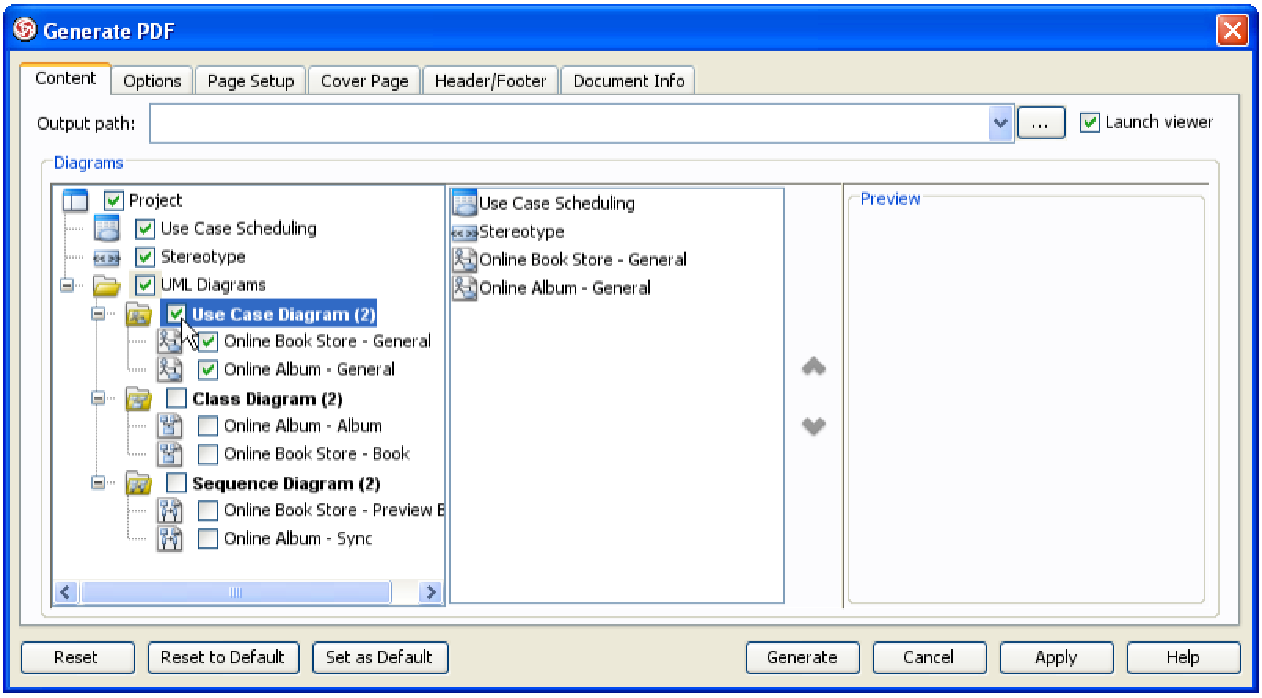
☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - □ UML Diagrams
    - □ **Use Case Diagram (2)**
      - □ Online Book Store - General
      - □ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Stereotype

__Figure 1-3__ Generate PDF dialog

To generate a PDF Report, open the Generate PDF dialog box and select diagrams for the generation.



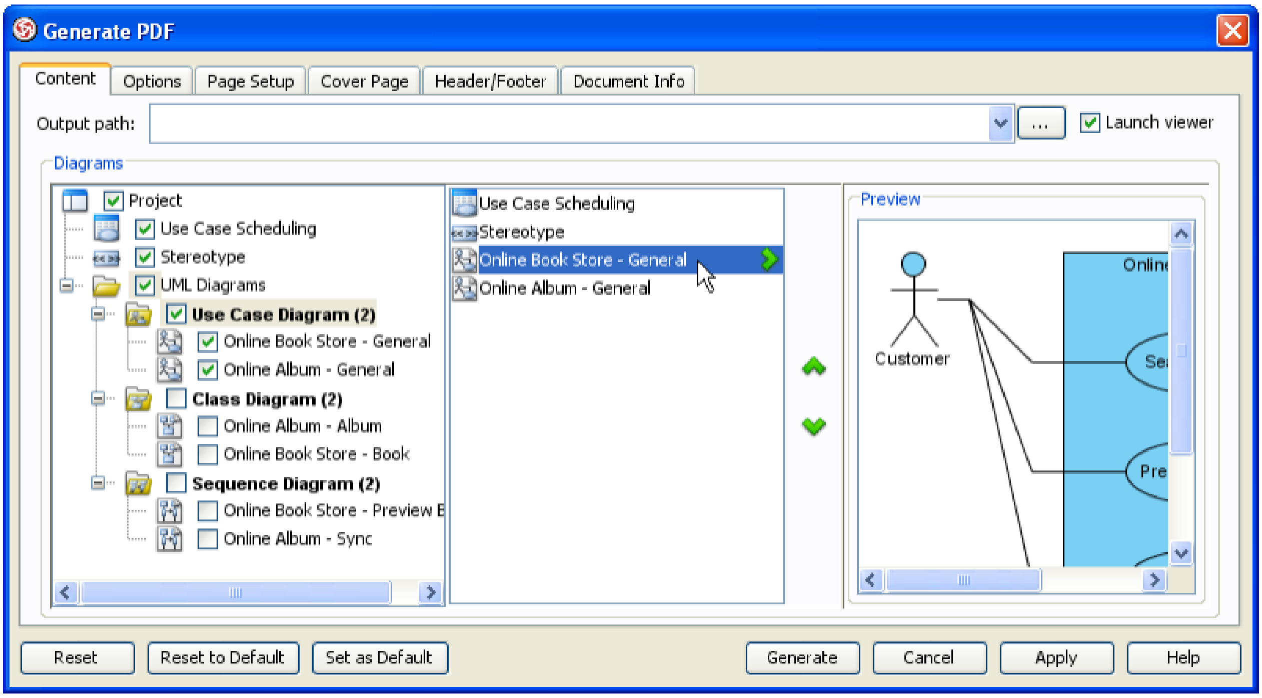
☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - ☑ UML Diagrams
    - ☑ **Use Case Diagram (2)**
      - ☑ Online Book Store - General
      - ☑ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

__Figure 1-4__ Selecting diagram to generate PDF

It is possible to preview diagram by selecting it as below.



Output path:

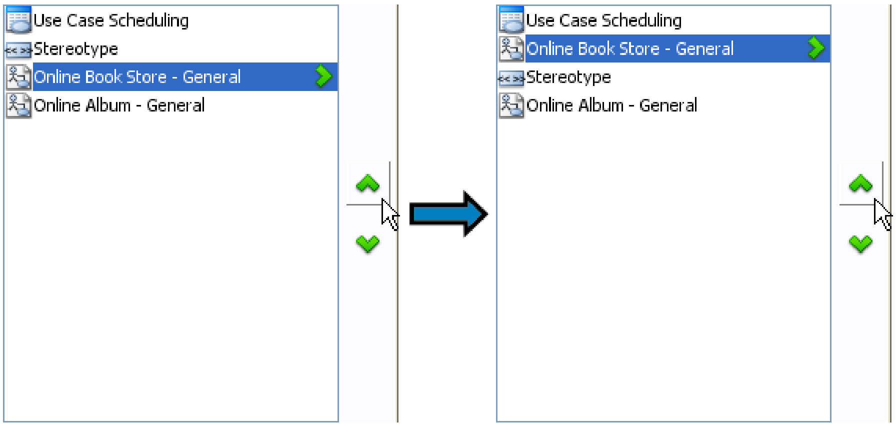
☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - ☑ UML Diagrams
    - ☑ **Use Case Diagram (2)**
      - ☑ Online Book Store - General
      - ☑ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

__Figure 1-5__ Showing preview of diagram

Also ▲ or ▼ may enable for reordering.



Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

Use Case Scheduling
Online Book Store - General
Stereotype
Online Album - General

__Figure 1-6__ Reording diagram

To configure diagram's element, press ▶ to change view first.

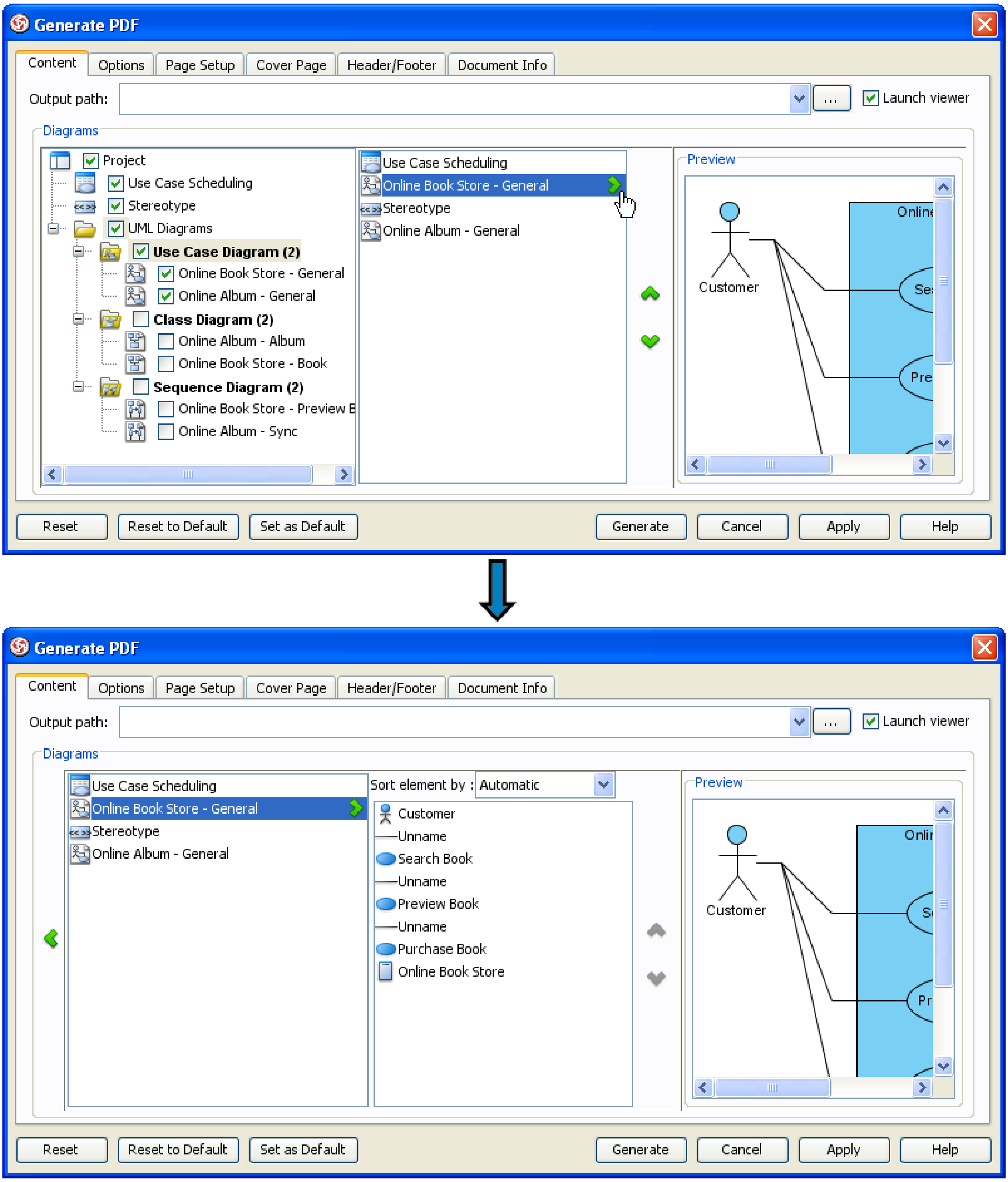
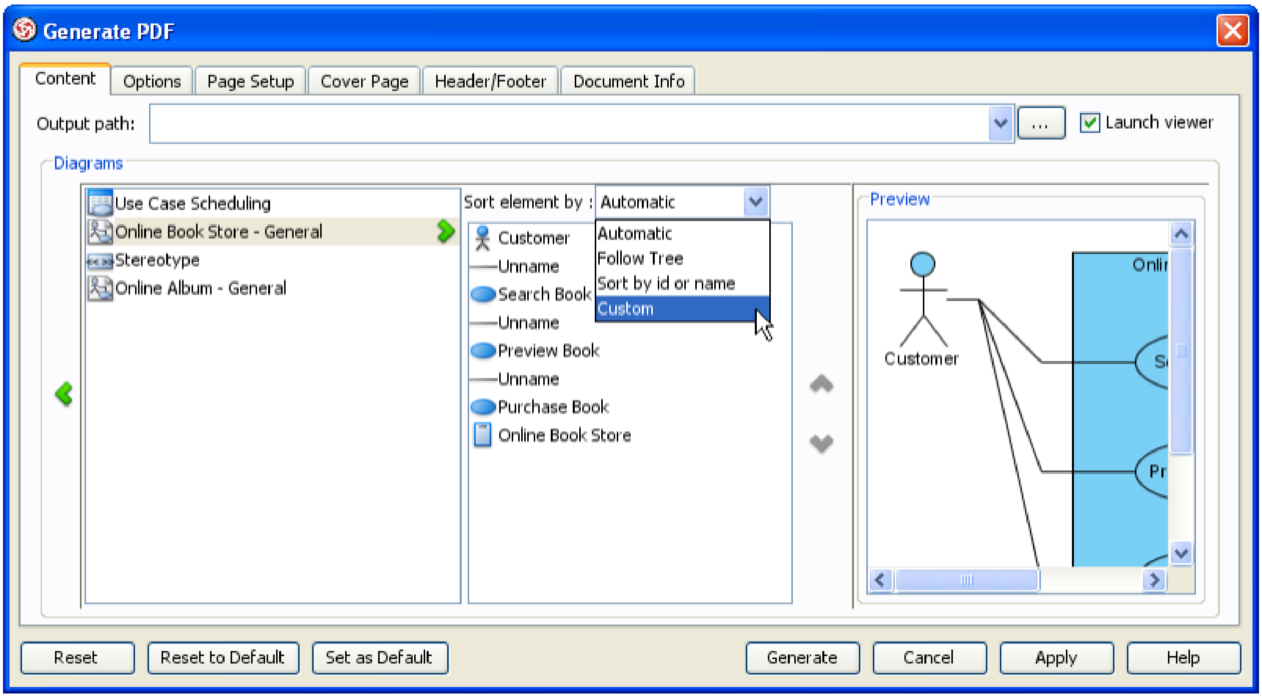


***Figure 1-7** View diagram elements of diagram*

There is four kinds of sorting for element.



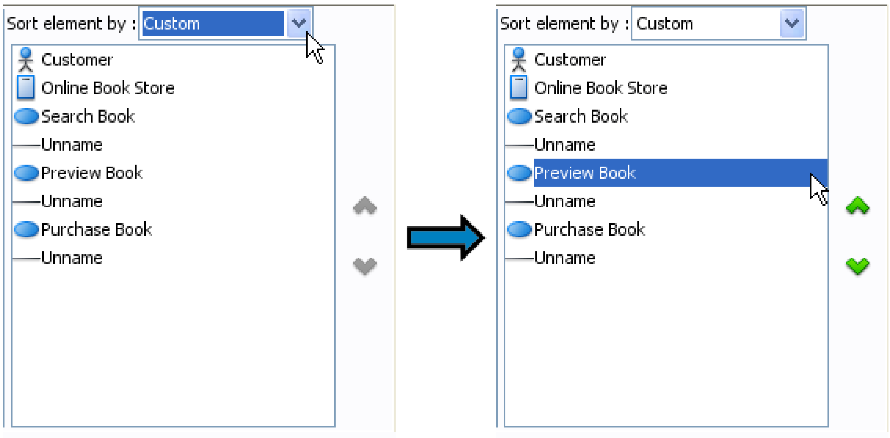
Output path:

☑ Launch viewer

Sort element by : Automatic

__Figure 1-8__ Kinds of sort element

Sort element by Custom allow to reorder element manually.

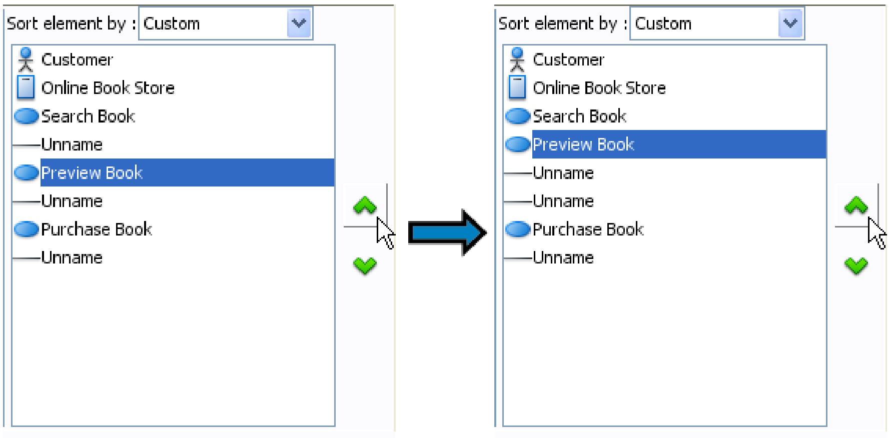


Sort element by : Custom

Sort element by : Custom

__Figure 1-9__ Selecting element

Press ▲ or ▼ to reorder element as preferred manually.

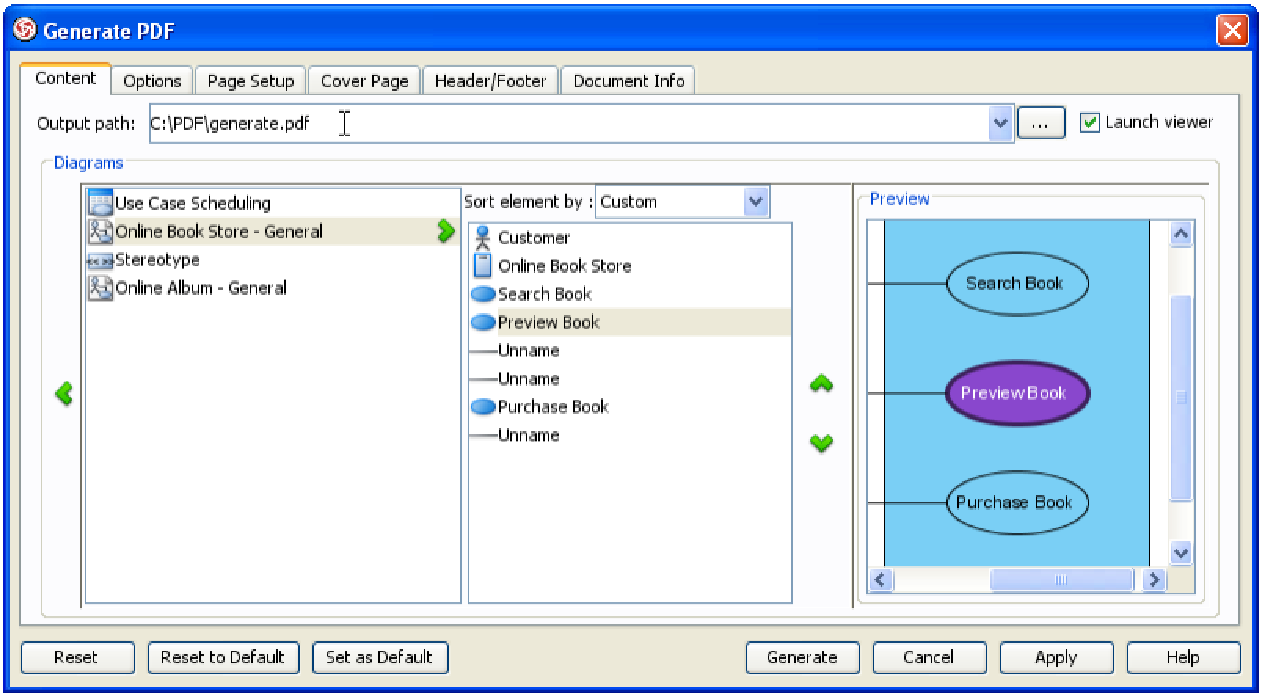


Sort element by : Custom

Sort element by : Custom

__Figure 1-10__ Reordering element

Don't forget to input Output path of the generate PDF.



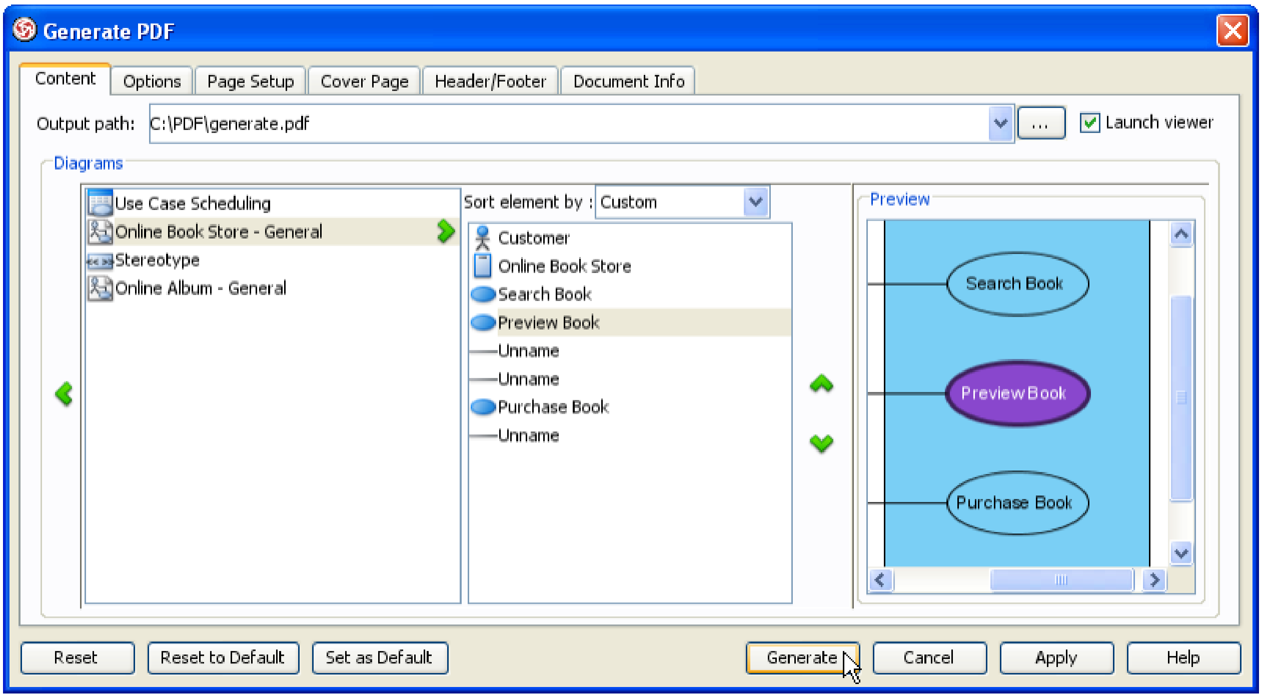
Output path: C:\PDF\generate.pdf

☑ Launch viewer

Sort element by : Custom

Figure 1-11 *Inputting output path*

Press Generate button to perform the PDF generation.



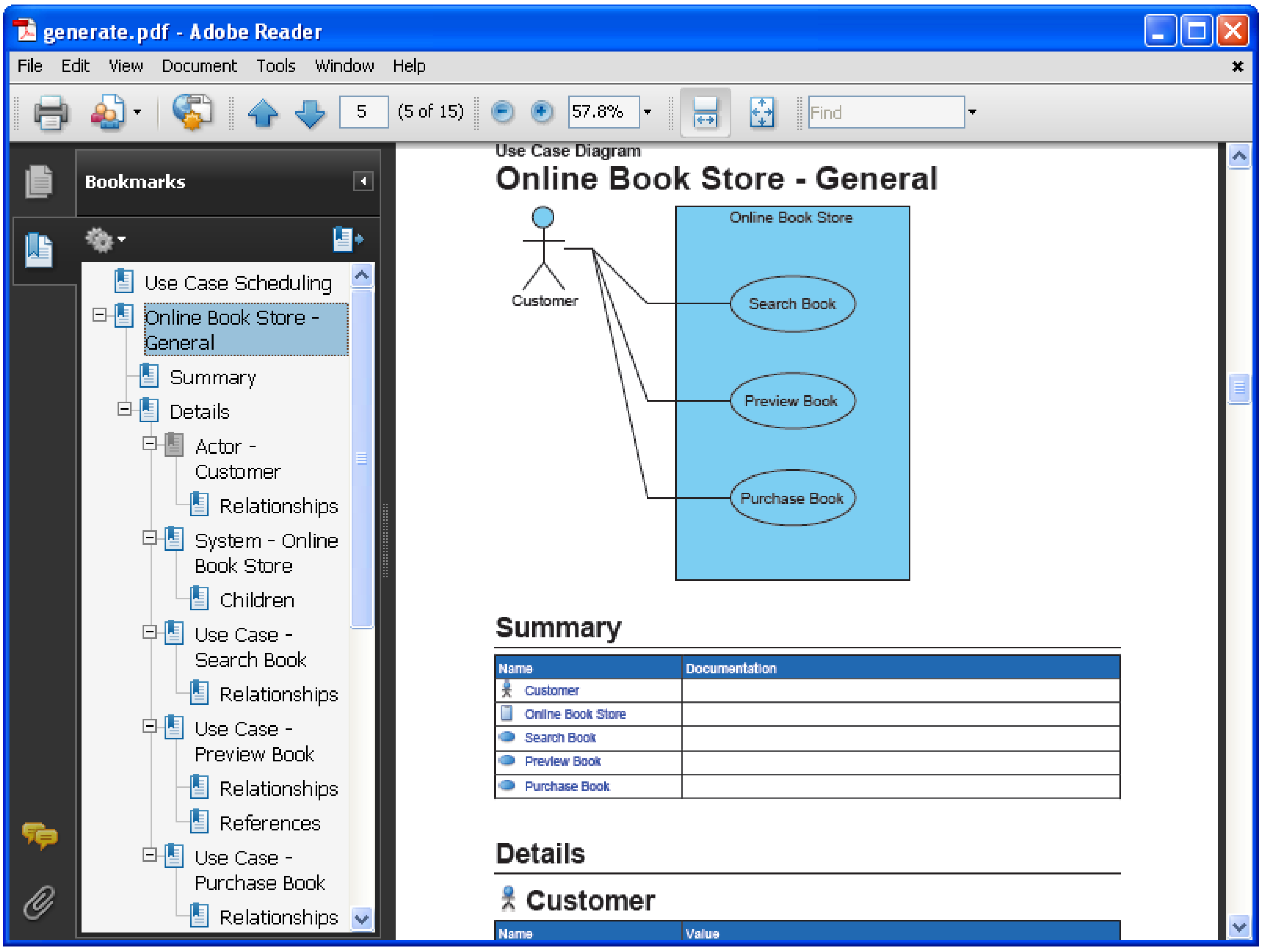
Output path: C:\PDF\generate.pdf

☑ Launch viewer

Sort element by : Custom

Figure 1-12 *Generate PDF*

Generated PDF will looks like below.



Use Case Diagram

Online Book Store - General

Summary

Name	Documentation
Customer	
Online Book Store	
Search Book	
Preview Book	
Purchase Book	

Details

Customer

Name	Value

Figure 1-13 Generated PDF

Customizing Report generation options

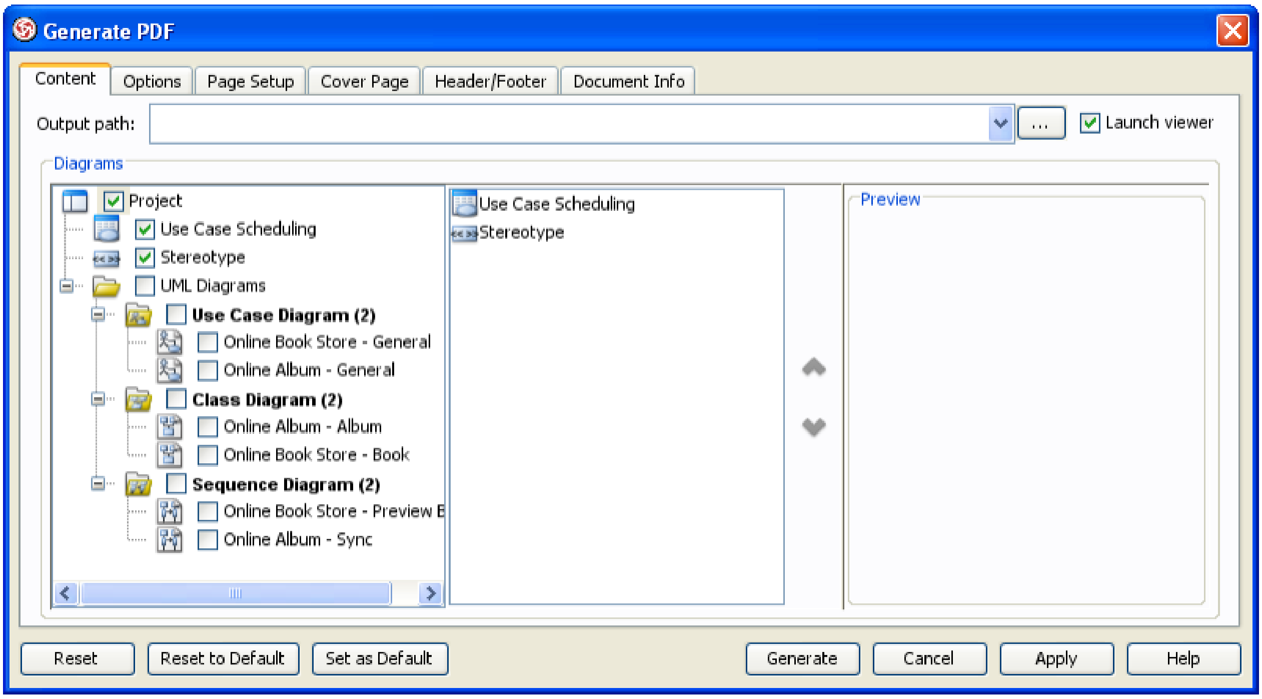


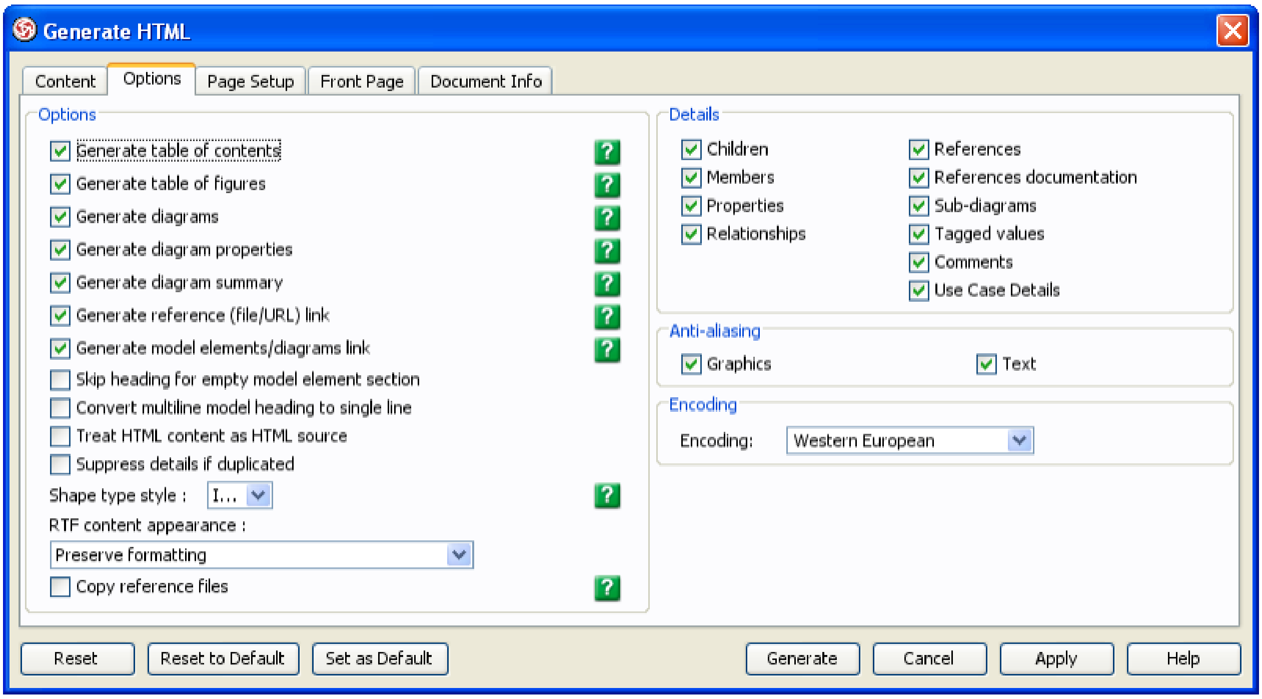
Figure 1-14 *Content tab of generate PDF dialog*

Details of each field in Content tab are shown as below:

Field	Description
Output path	To select the destination file for the generated report. You can type the path in the text field or you can browse the location by clicking on the ...button.
Launch viewer	If this option is selected, the default browser of the system will be opened automatically to show the generated document.
Sort element by	Automatic - sorting elements by listing them in the most logical order, which is to follow most users&rsquo; understanding of that kind of diagram Follow tree - sorting elements by following the sort order of the diagram tree in the tool id or name - sorting elements by their ID or names You can refer to the section 'Sorting Elements in Report'.

Table 1-1 *Details of generate PDF dialog's content tab*

Customizing Report generation options



Options

- ☑ Generate table of contents
- ☑ Generate table of figures
- ☑ Generate diagrams
- ☑ Generate diagram properties
- ☑ Generate diagram summary
- ☑ Generate reference (file/URL) link
- ☑ Generate model elements/diagrams link
- □ Skip heading for empty model element section
- □ Convert multiline model heading to single line
- □ Treat HTML content as HTML source
- □ Suppress details if duplicated

Shape type style : I...

RTF content appearance : Preserve formatting

- □ Copy reference files

Details

- ☑ Children
- ☑ Members
- ☑ Properties
- ☑ Relationships
- ☑ References
- ☑ References documentation
- ☑ Sub-diagrams
- ☑ Tagged values
- ☑ Comments
- ☑ Use Case Details

Anti-aliasing

- ☑ Graphics
- ☑ Text

Encoding

Encoding: Western European

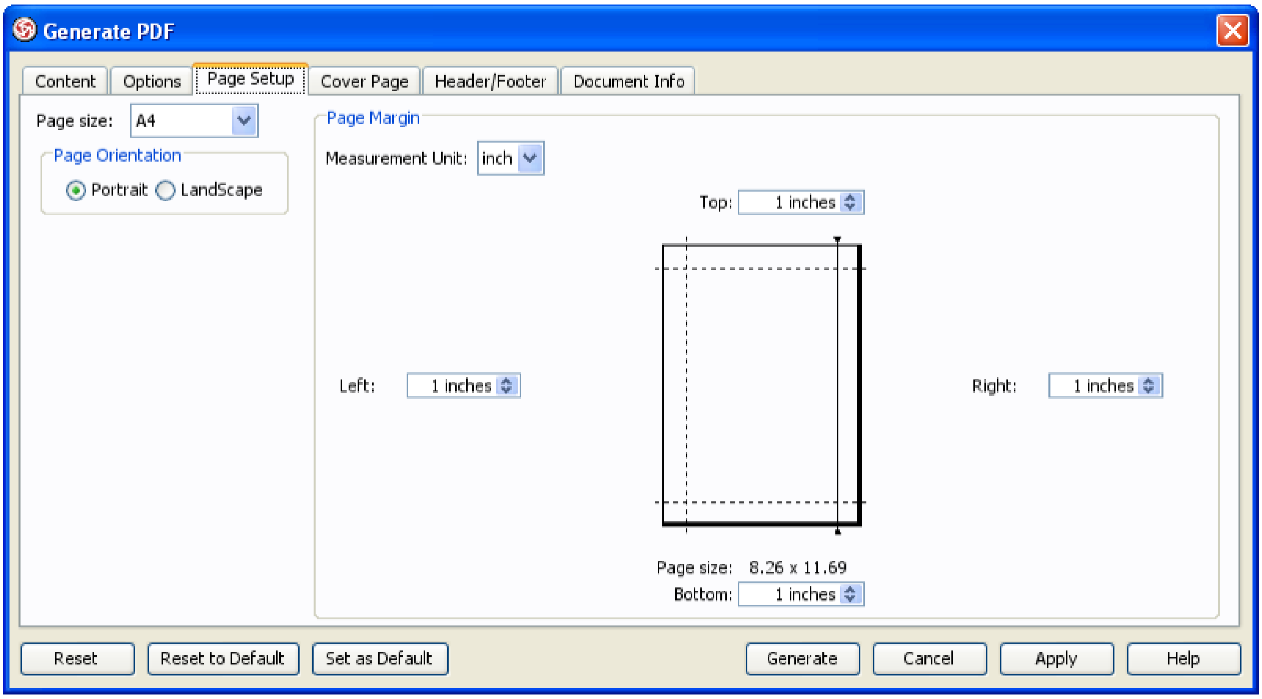
Figure 1-15 *Options tab of generate PDF dialog*

Details of each field in Options tab are shown as below:

Field	Description
Options	
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.
Image type	SVG - using SVG image format for diagram image PNG - using PNG image format for diagram image
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.
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.
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.
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
Details	
Children	Select to generate children of model
Members	Select to generate members of model
Properties	Select to generate properties of model
Relationships	Select to generate relationships of model
References	Select to generate references of model
References documentation	Select to generate references documentation
Sub-diagrams	Select to generate sub-diagrams of model
Tagged values	Select to generate tagged values of model
Comments	Select to generate comments of model
Use Case Details	Select to generate use case details
Anti-aliasing	
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	
Font	Specify font to be used.

***Table 1-2** Details of generate PDF dialog's options tab*

Page Setup



Generate PDF

Page size: A4

Page Orientation

● Portrait ○ LandScape

Page Margin

Measurement Unit: inch

Top: 1 inches

Left: 1 inches

Right: 1 inches

Page size: 8.26 x 11.69

Bottom: 1 inches

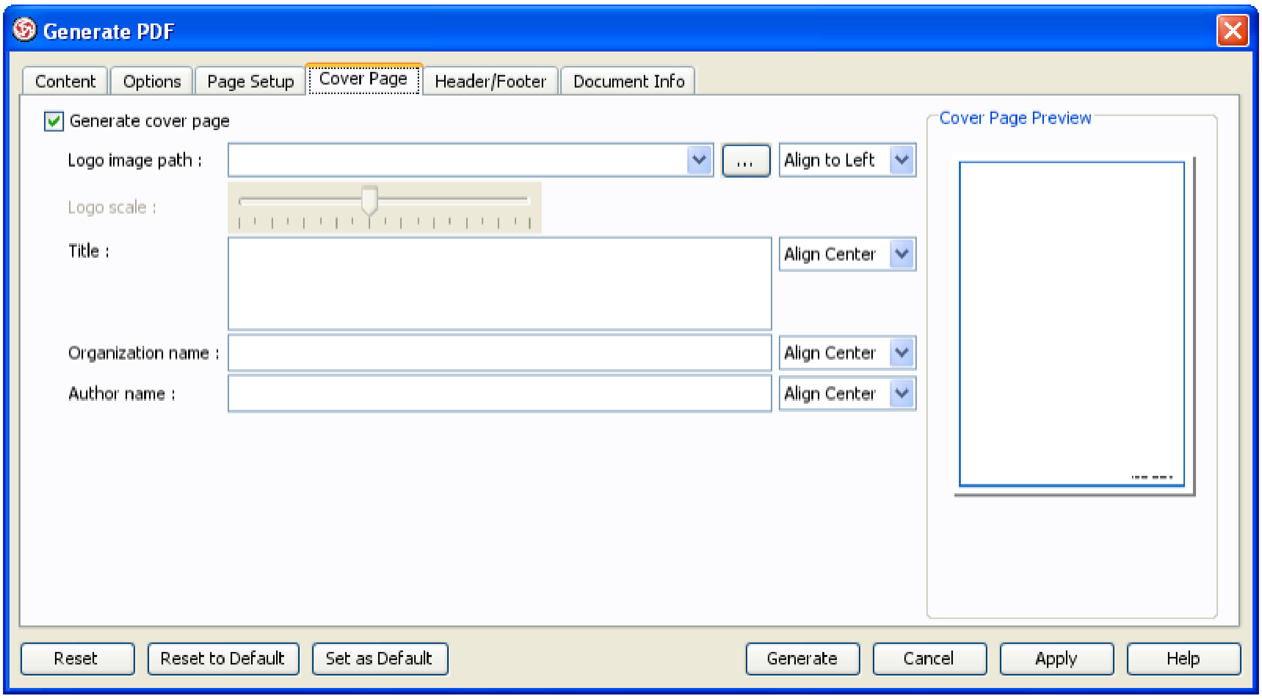
***Figure 1-16** Page setup tab of generate PDF dialog*

Details of each field in Page Setup tab are shown as below:

Field	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).
Page Margin	To specify the page margins of the report.

***Table 1-3** Details of generate PDF dialog's page setup tab*

Customizing Cover Page



☑ Generate cover page

Logo image path : Align to Left

Logo scale :

Title : Align Center

Organization name : Align Center

Author name : Align Center

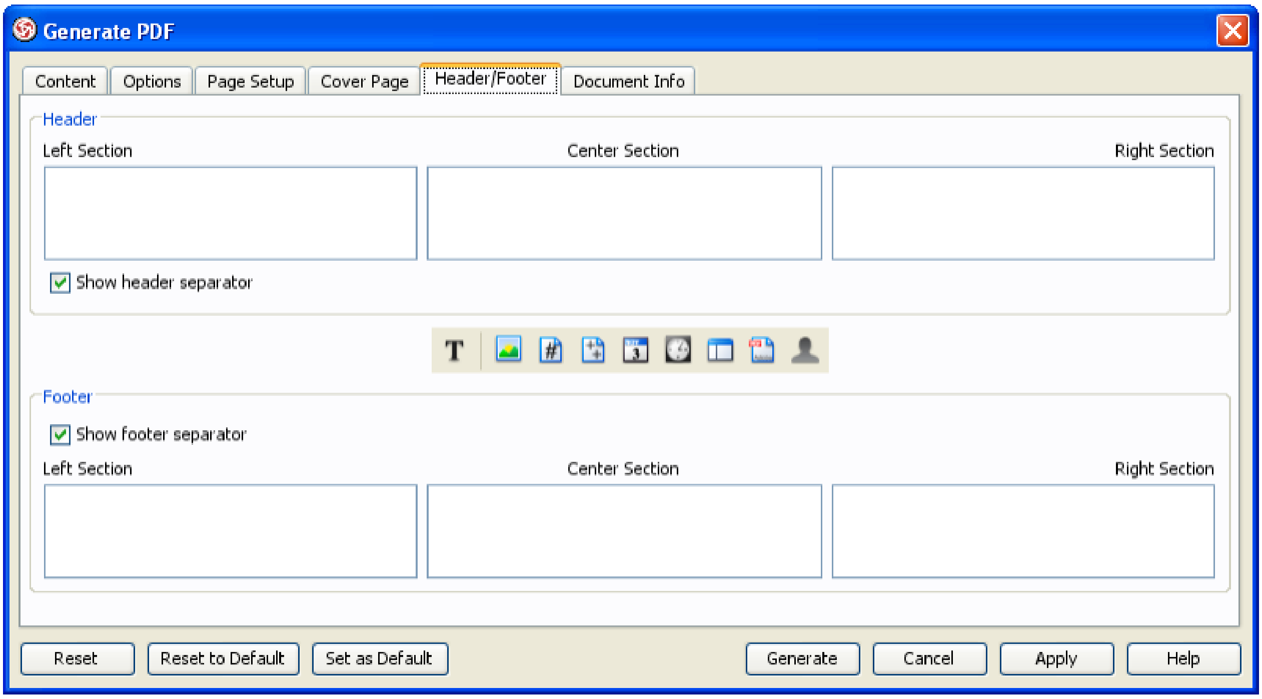
Figure 1-17 *Cover page tab of generate PDF dialog*

Details of each field in Cover Page tab are shown as below:

Field	Description
Generate cover page	If this option is selected, cover page for this document will be generated to the report
Logo image path	PDF report's logo image path
Logo scale	PDF report's logo scale
Title	PDF report's title
Organization name	PDF report's organization name
Author name	PDF report's author name

Table 1-4 *Details of generate PDF dialog's cover page tab*

Customizing Header/Footer


Generate PDF

Content | Options | Page Setup | Cover Page | Header/Footer | Document Info

Header

Left Section | Center Section | Right Section

☑ Show header separator

Footer

☑ Show footer separator

Left Section | Center Section | Right Section

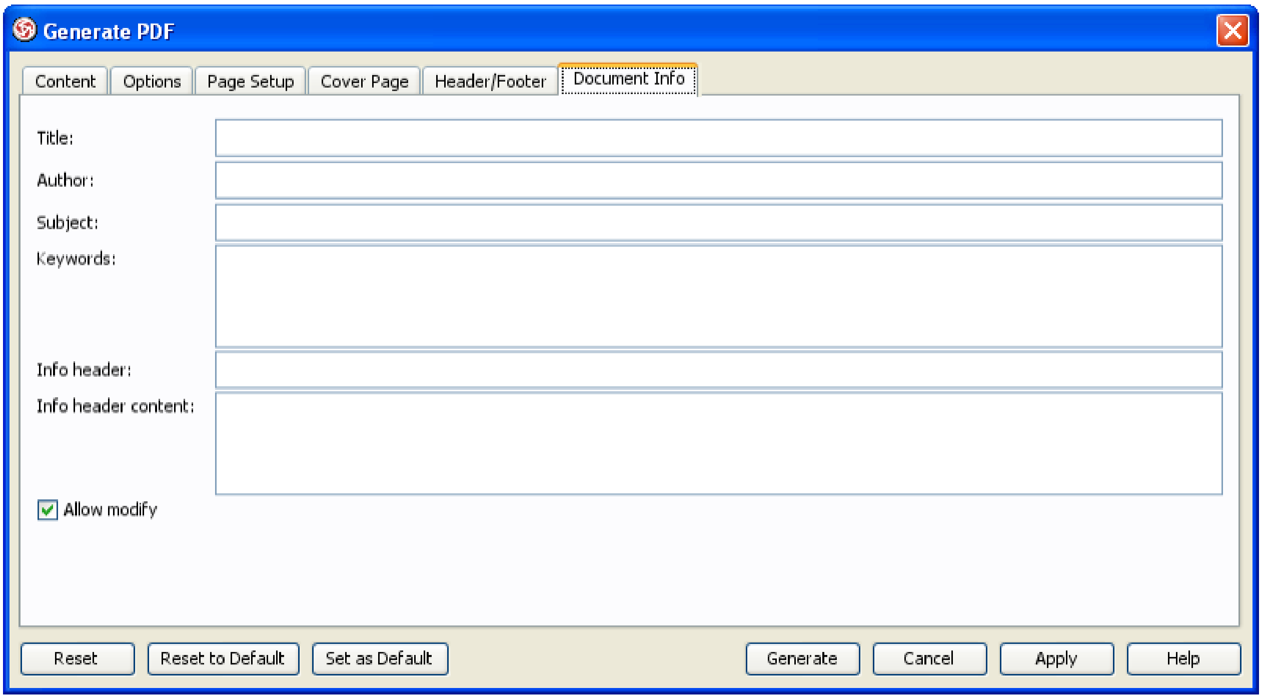
Figure 1-18 Header/Footer tab of generate PDF dialog

Details of each field in Header/Footer tab are shown as below:

Field	Description
T	To format text style, such as font style, size and color.
🖼	To insert an image to the header or footer
#	Insert page number
+	Insert page count
3	Insert date
🕒	Insert time
▭	Insert project name
📄	Insert report file name
👤	Insert user name

Table 1-5 Details of generate PDF dialog's header/footer tab

Filling Document Info



Title:

Author:

Subject:

Keywords:

Info header:

Info header content:

☑ Allow modify

Figure 1-19 Document info tab of generate PDF dialog

Details of each field in Document Info tab are shown as below:

Field	Description
Title	The title of the report.
Author	The author of the report.
Subject	The subject of the report.
Keywords	The keywords of the report.
Info header	The info header of the report.
Info header content	The info header content of the report.
Allow modify	Select to allow modification on the report.

Table 1-6 Details of generate PDF dialog's document info tab

Generating Word Report

The Generate Word dialog box provides a set of options for changing the report style. To display the dialog box, perform one of the following actions:

Select Tools > Report > Generate Word Report... from main menu.

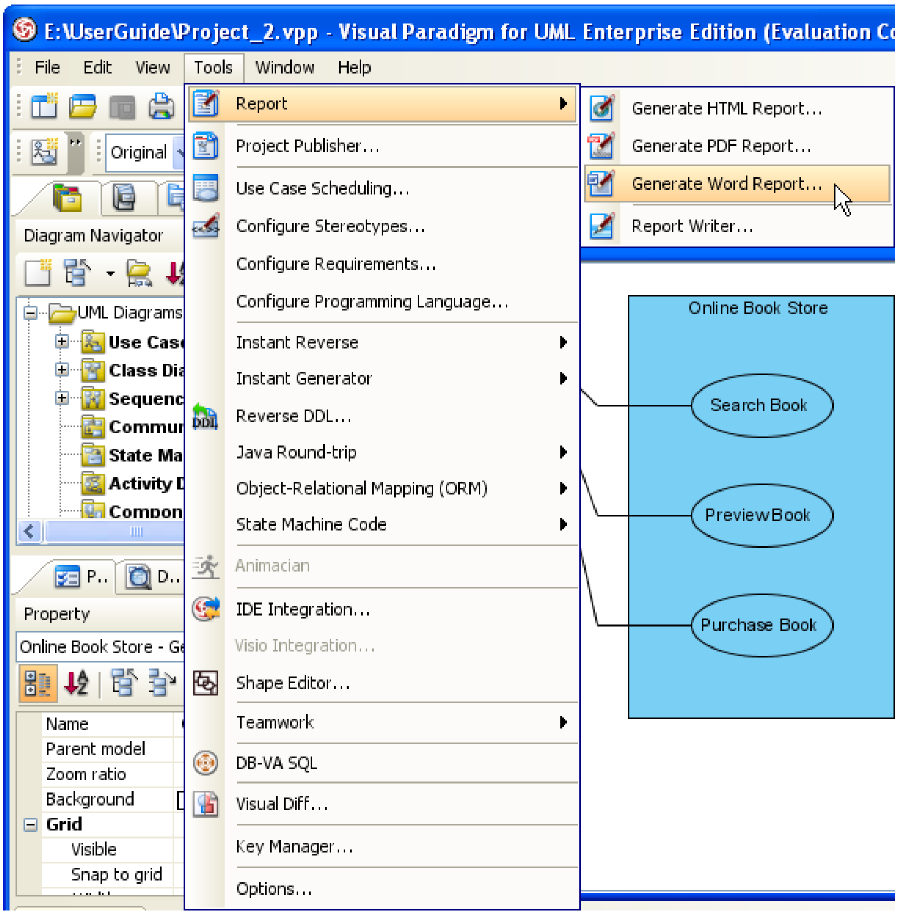


Figure 2-1 Generate Word from main menu

Or click on the Generate Word Report button on the toolbar.

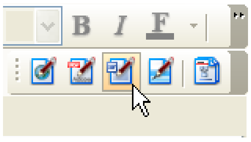
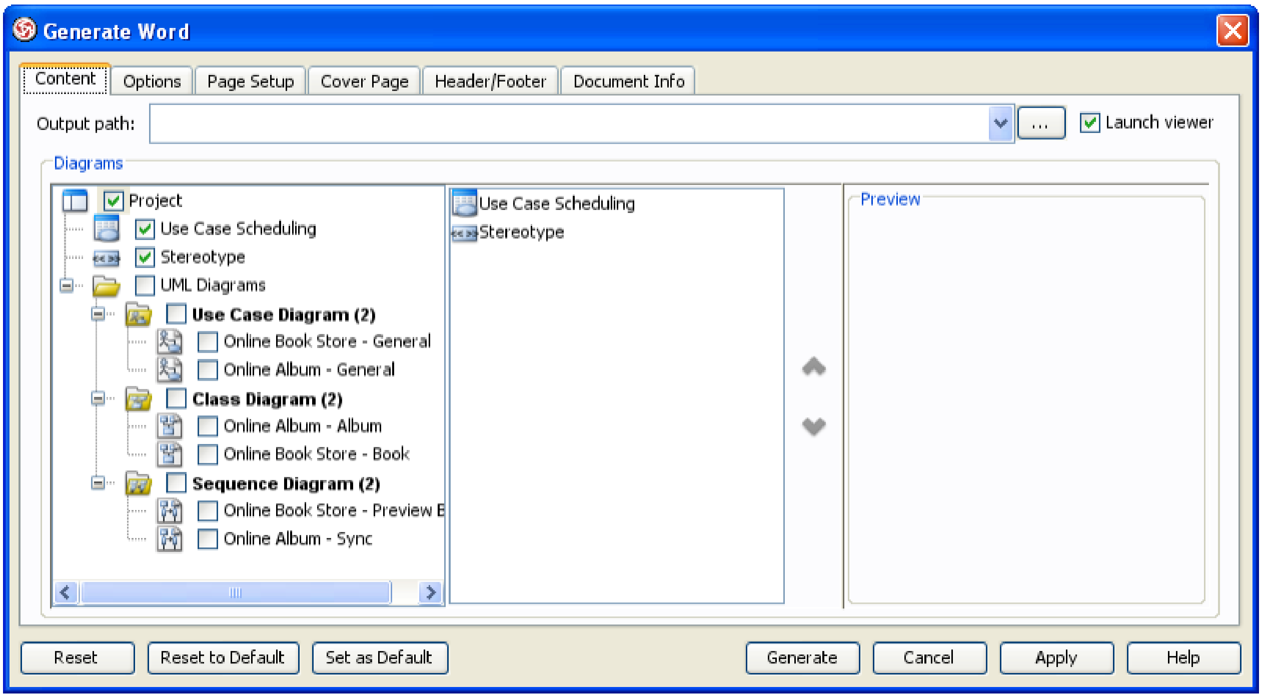


Figure 2-2 Generate Word from tool bar

Generate Word dialog



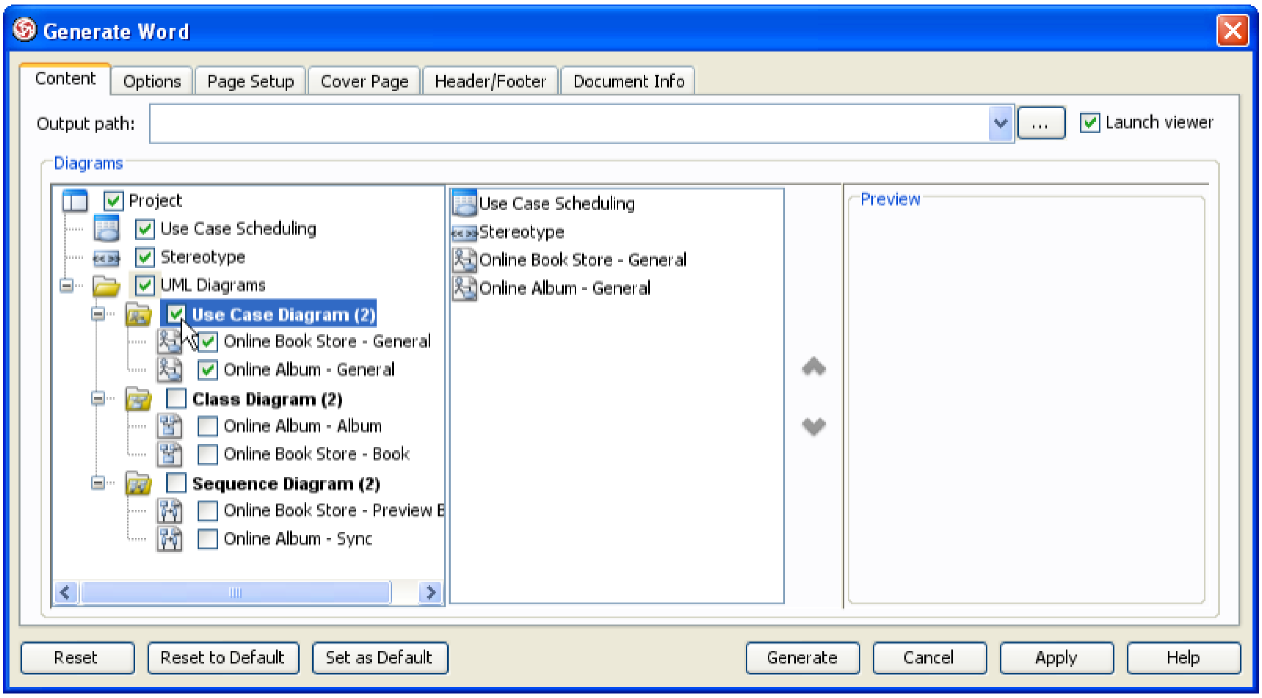
☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - ☐ UML Diagrams
    - ☐ **Use Case Diagram (2)**
      - ☐ Online Book Store - General
      - ☐ Online Album - General
    - ☐ **Class Diagram (2)**
      - ☐ Online Album - Album
      - ☐ Online Book Store - Book
    - ☐ **Sequence Diagram (2)**
      - ☐ Online Book Store - Preview E
      - ☐ Online Album - Sync

Use Case Scheduling
Stereotype

Figure 2-3 *Generate Word dialog*

To generate a Word Report, open the Generate Word dialog box and select diagrams for the generation.



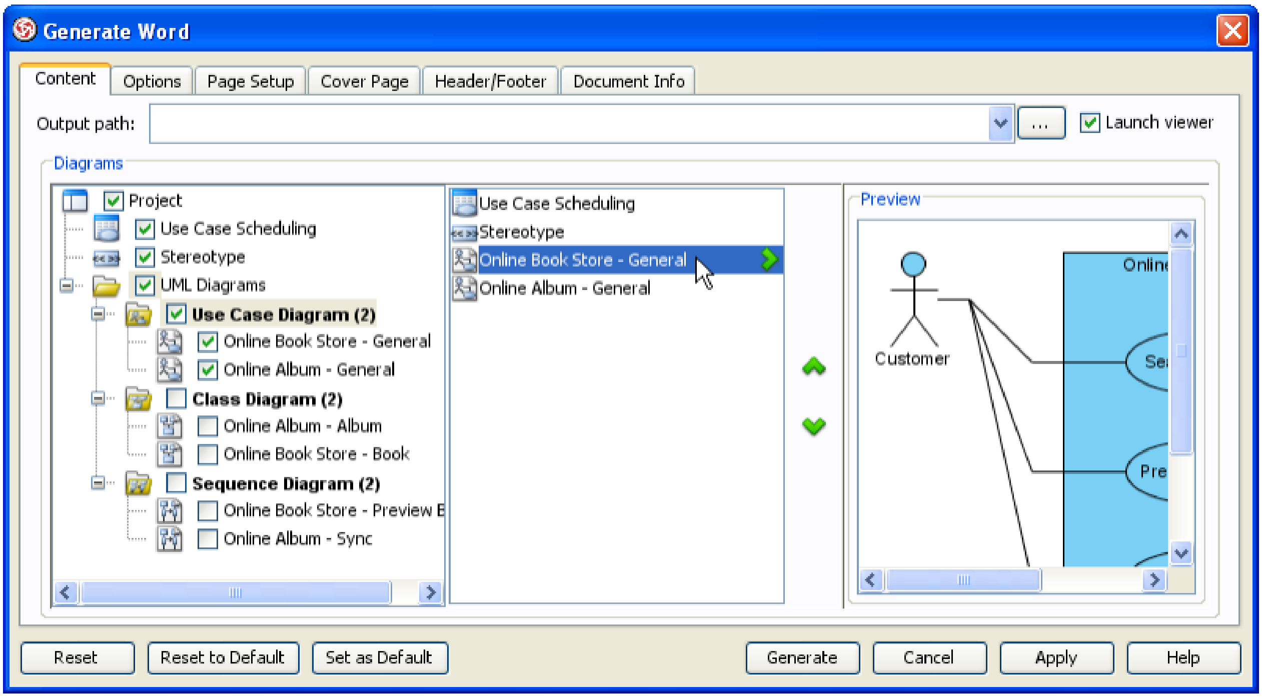
☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - ☑ UML Diagrams
    - ☑ **Use Case Diagram (2)**
      - ☑ Online Book Store - General
      - ☑ Online Album - General
    - ☐ **Class Diagram (2)**
      - ☐ Online Album - Album
      - ☐ Online Book Store - Book
    - ☐ **Sequence Diagram (2)**
      - ☐ Online Book Store - Preview E
      - ☐ Online Album - Sync

Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

Figure 2-4 *Selecting diagram to generate Word*

It is possible to preview diagram by selecting it as below.


Generate Word

Content | Options | Page Setup | Cover Page | Header/Footer | Document Info

Output path: ☑ Launch viewer

Diagrams

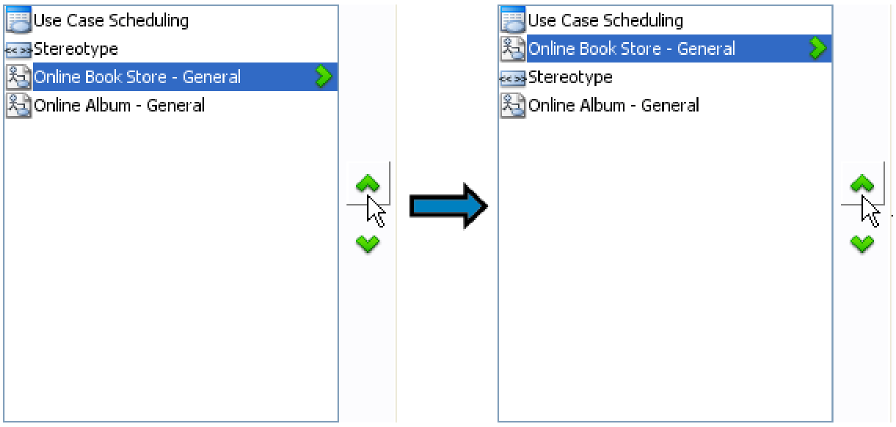
- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - ☑ UML Diagrams
    - ☑ **Use Case Diagram (2)**
      - ☑ Online Book Store - General
      - ☑ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

Preview

Figure 2-5 *Showing preview of diagram*

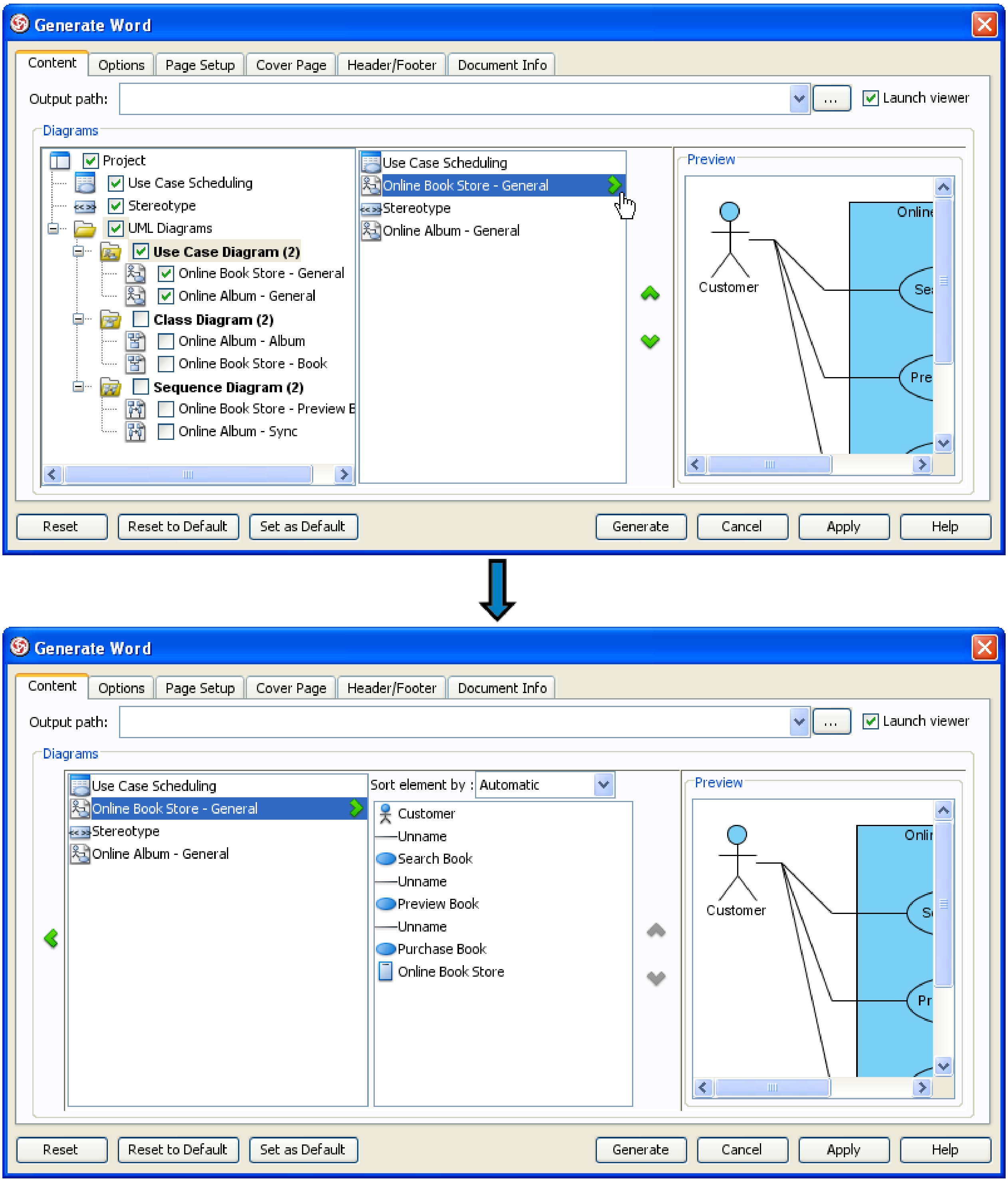
Also ▲ or ▼ may enable for reordering.


Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

Use Case Scheduling
Online Book Store - General
Stereotype
Online Album - General

Figure 2-6 *Reording diagram*

To configure diagram's element, press ➤ to change view first.



☑ Launch viewer

☑ Project
☑ Use Case Scheduling
☑ Stereotype
☑ UML Diagrams
☑ **Use Case Diagram (2)**
☑ Online Book Store - General
☑ Online Album - General
☐ **Class Diagram (2)**
☐ Online Album - Album
☐ Online Book Store - Book
☐ **Sequence Diagram (2)**
☐ Online Book Store - Preview B
☐ Online Album - Sync

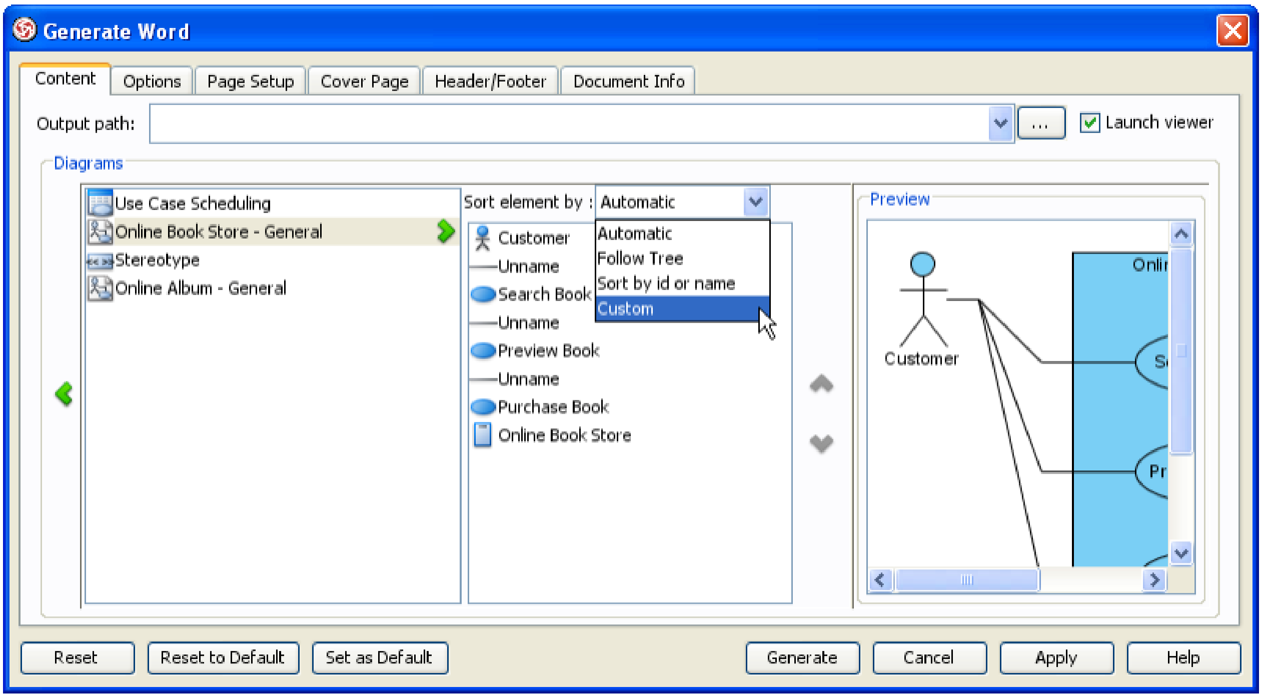
Use Case Scheduling
Online Book Store - General
Stereotype
Online Album - General

Sort element by : Automatic

Customer
Unname
Search Book
Unname
Preview Book
Unname
Purchase Book
Online Book Store

Figure 2-7 View diagram elements of diagram

There is four kinds of sorting for element.

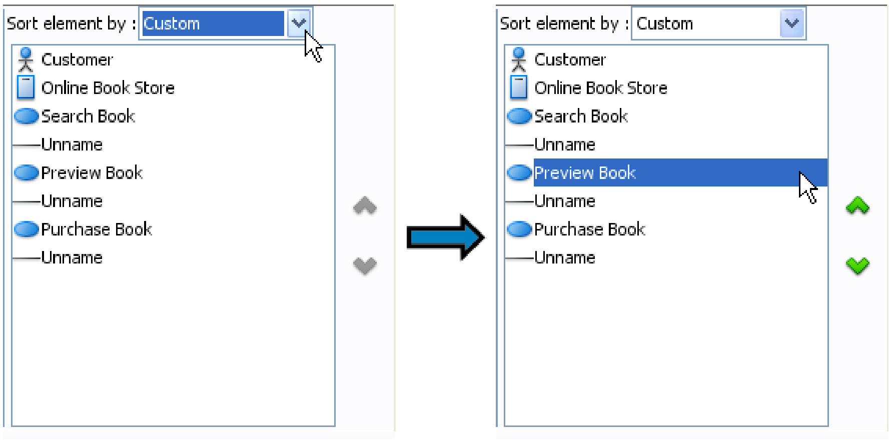


Sort element by : Automatic

☑ Launch viewer

Figure 2-8 Kinds of sort element

Sort element by Custom allow to reorder element manually.

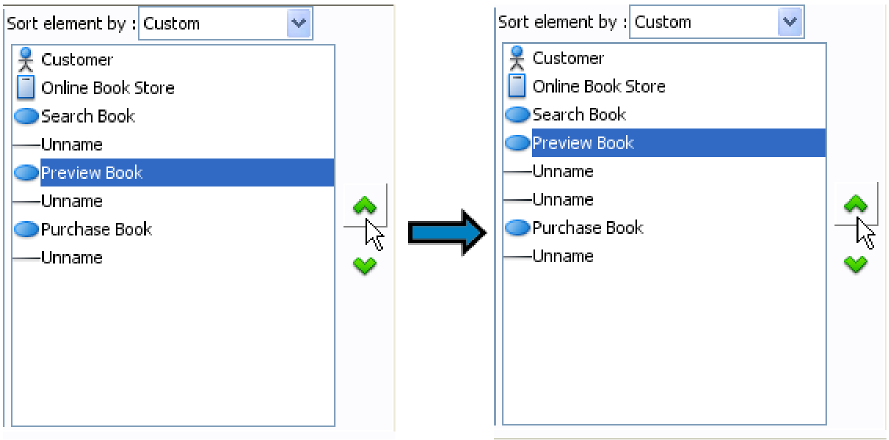


Sort element by : Custom

Sort element by : Custom

Figure 2-9 Selecting element

Press ▲ or ▼ to reorder element as preferred manually.

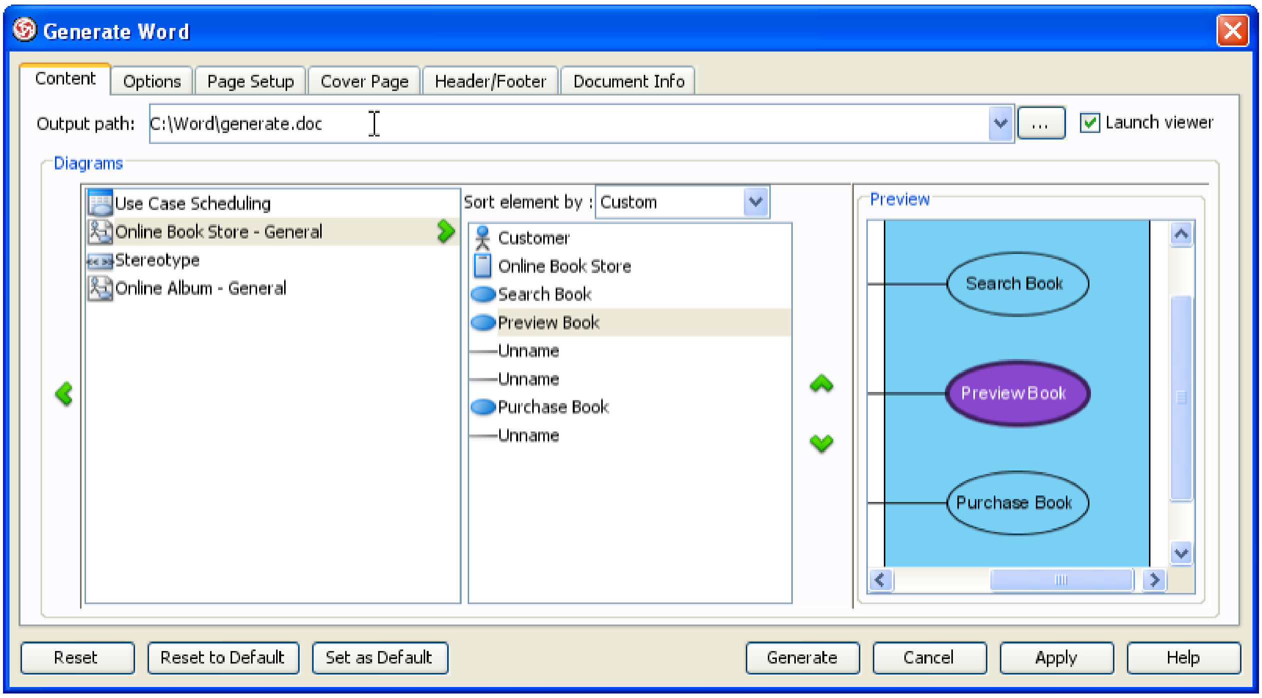


Sort element by : Custom

Sort element by : Custom

Figure 2-10 Reordering element

Don't forget to input Output path of the generate Word.

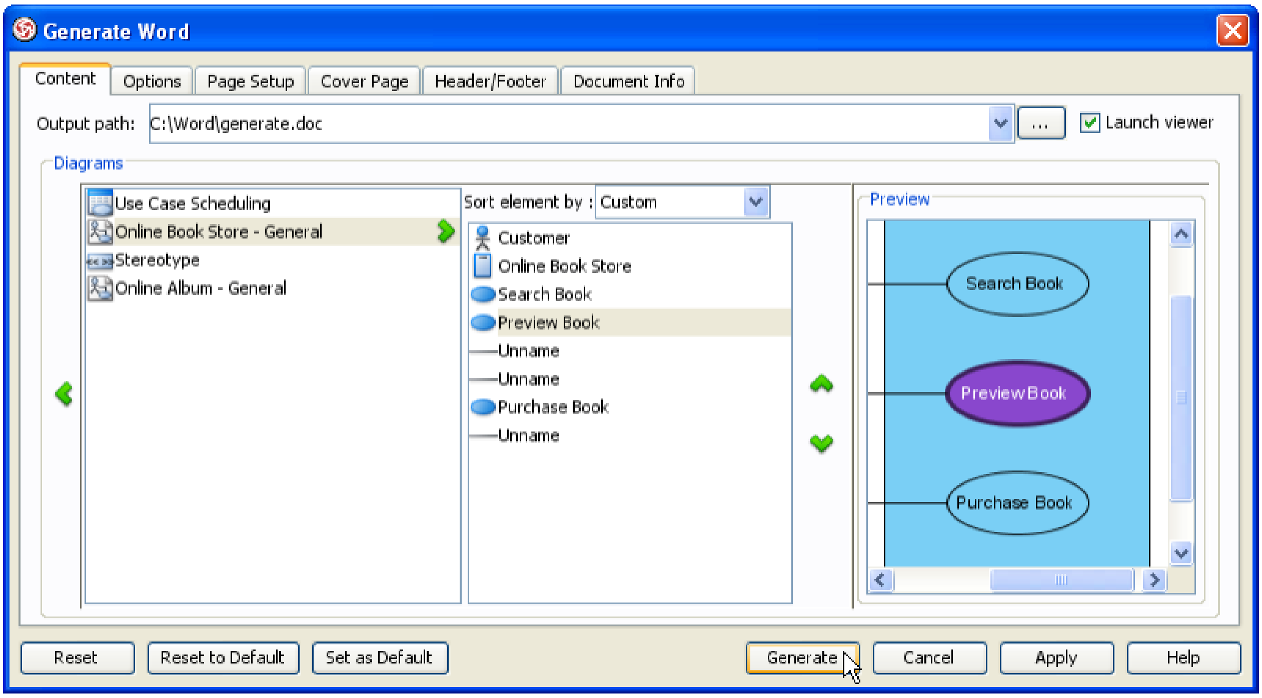

Output path: C:\Word\generate.doc

☑ Launch viewer

Sort element by : Custom

Figure 2-11 Inputting output path

Press Generate button to perform the Word generation.

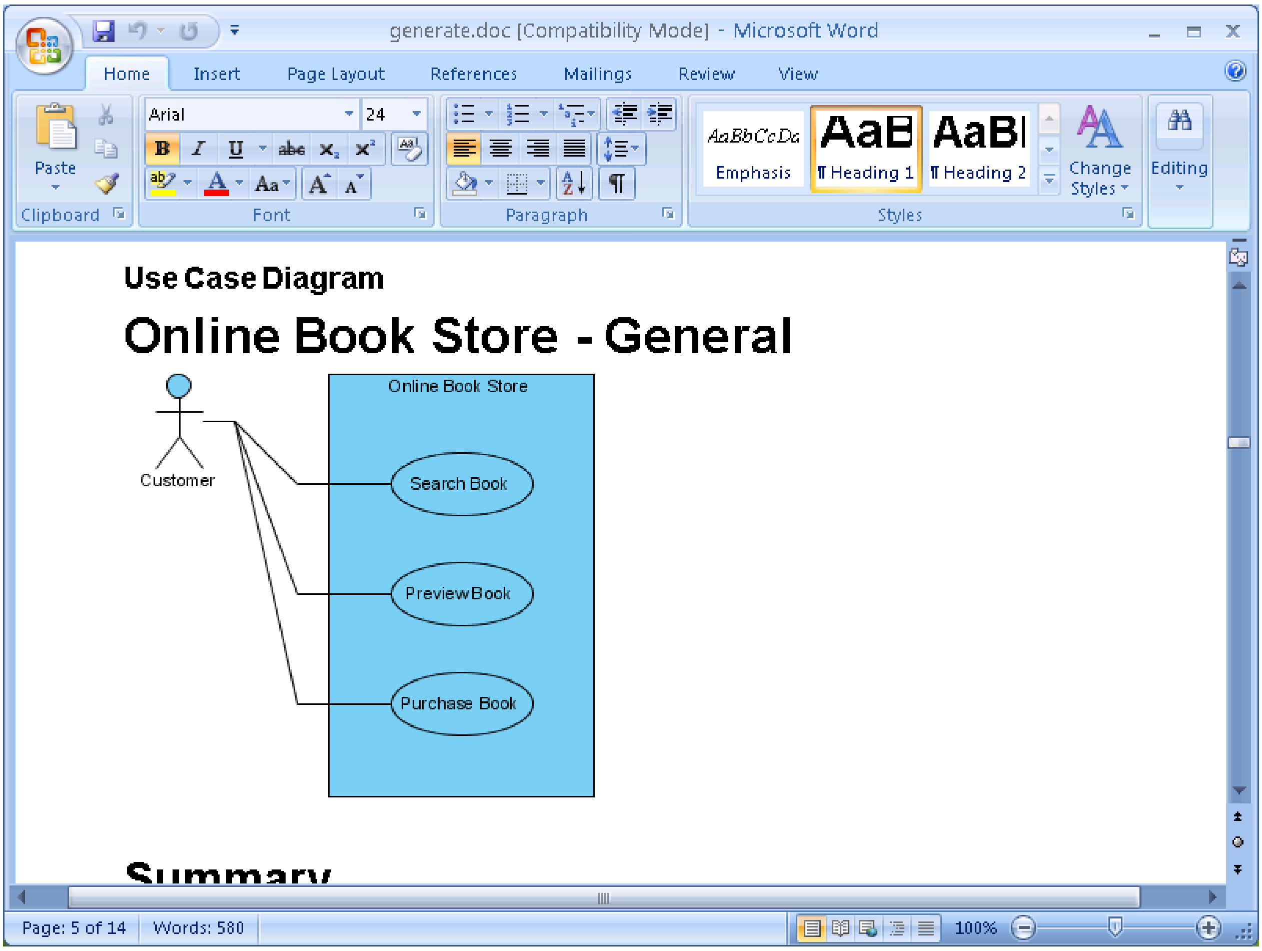

Output path: C:\Word\generate.doc

☑ Launch viewer

Sort element by : Custom

Figure 2-12 Generate Word

Generated Word will looks like below.



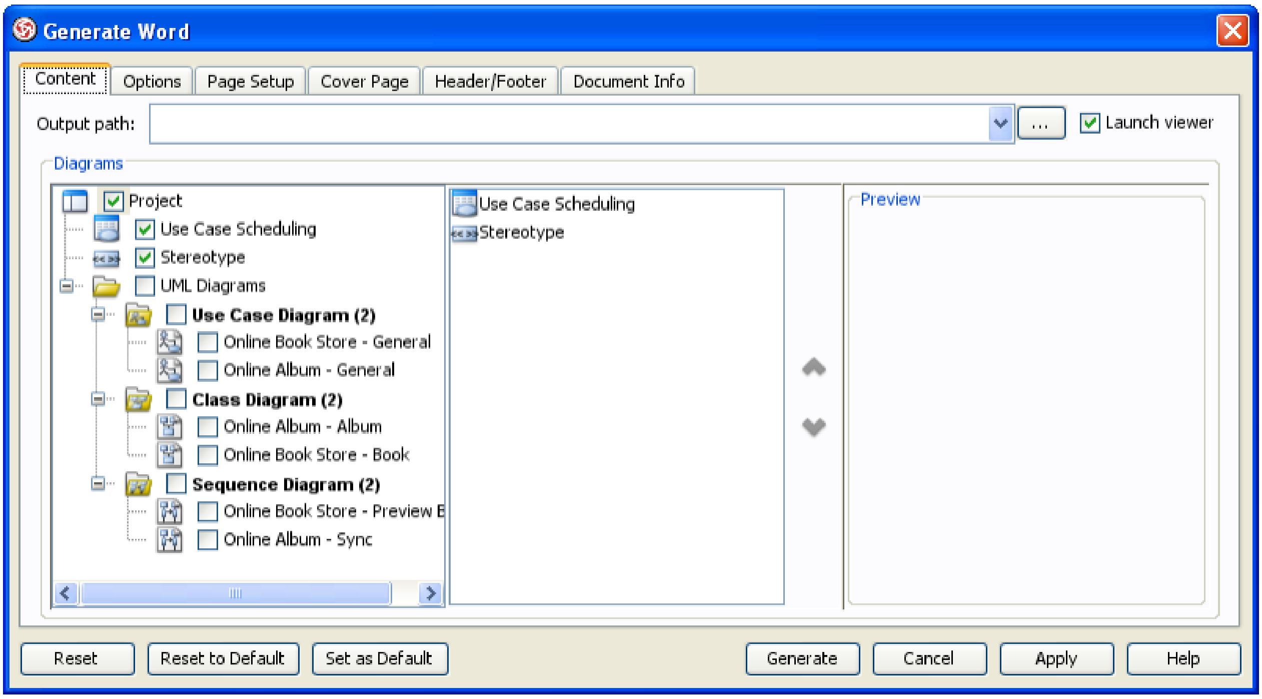
Use Case Diagram

Online Book Store - General

Summary

Figure 2-13 *Generated Word*

Customizing Report generation options



Output path:

☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - □ UML Diagrams
    - □ **Use Case Diagram (2)**
      - □ Online Book Store - General
      - □ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Stereotype

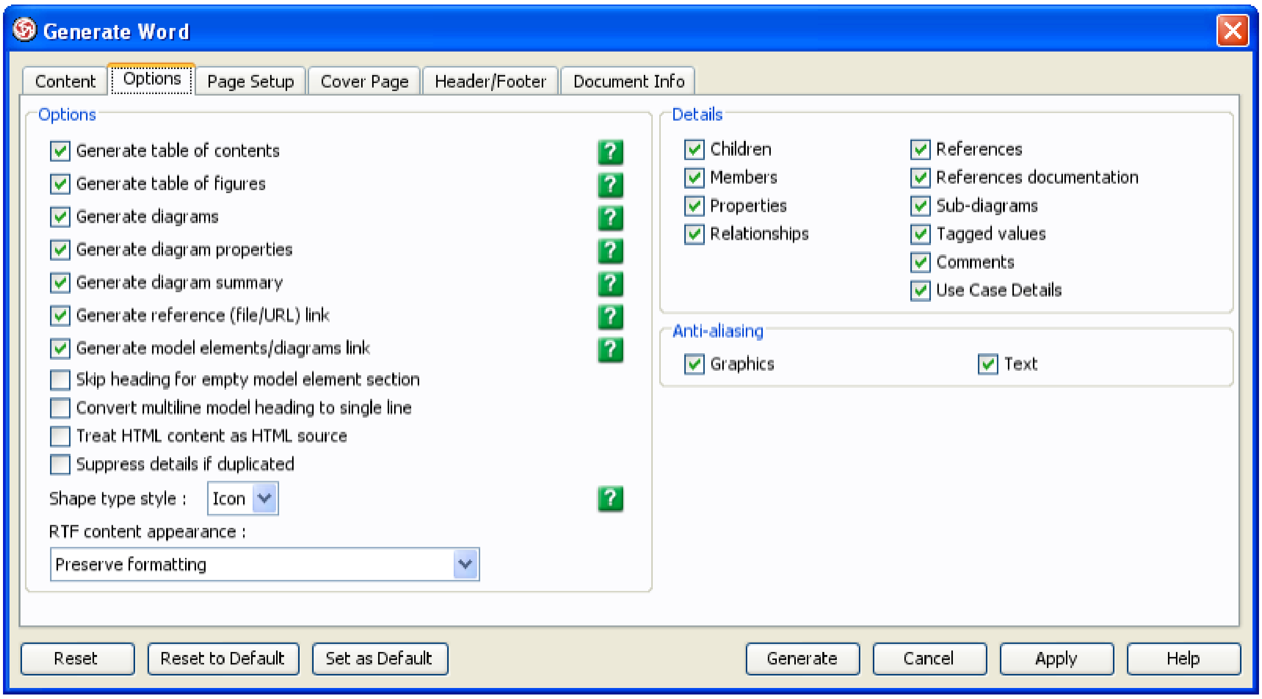
Figure 2-14 *Content tab of generate Word dialog*

Details of each field in Content tab are shown as below:

Field	Description
Output path	To select the destination file for the generated report. You can type the path in the text field or you can browse the location by clicking on the ...button.
Launch viewer	If this option is selected, the default browser of the system will be opened automatically to show the generated document.
Sort element by	Automatic - sorting elements by listing them in the most logical order, which is to follow most users&rsquo; understanding of that kind of diagram Follow tree - sorting elements by following the sort order of the diagram tree in the tool id or name - sorting elements by their ID or names You can refer to the section 'Sorting Elements in Report'.

Table 2-1 *Details of generate Word dialog's content tab*

Customizing Report generation options



Options

- ☑ Generate table of contents
- ☑ Generate table of figures
- ☑ Generate diagrams
- ☑ Generate diagram properties
- ☑ Generate diagram summary
- ☑ Generate reference (file/URL) link
- ☑ Generate model elements/diagrams link
- ☐ Skip heading for empty model element section
- ☐ Convert multiline model heading to single line
- ☐ Treat HTML content as HTML source
- ☐ Suppress details if duplicated

Shape type style : Icon

RTF content appearance : Preserve formatting

Details

- ☑ Children
- ☑ Members
- ☑ Properties
- ☑ Relationships
- ☑ References
- ☑ References documentation
- ☑ Sub-diagrams
- ☑ Tagged values
- ☑ Comments
- ☑ Use Case Details

Anti-aliasing

- ☑ Graphics
- ☑ Text

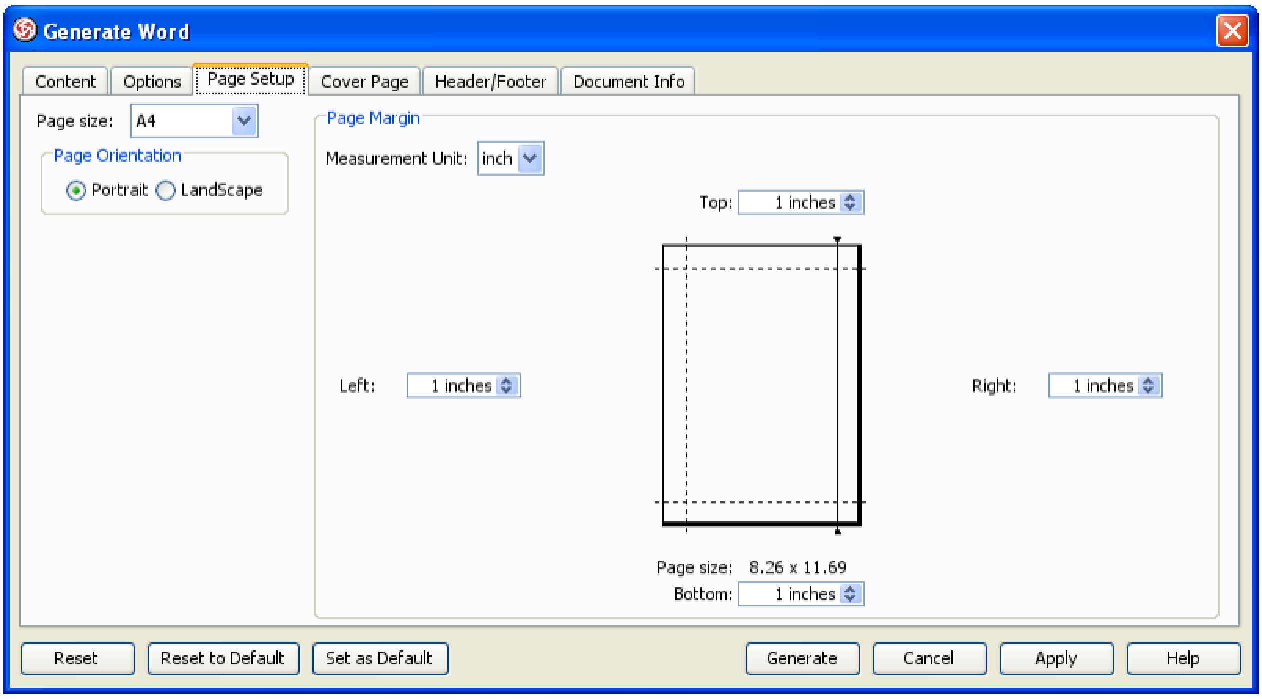
Figure 2-15 *Options tab of generate Word dialog*

Details of each field in Options tab are shown as below:

Field	Description
Options	
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.
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.
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.
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.
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
Details	
Children	Select to generate children of model
Members	Select to generate members of model
Properties	Select to generate properties of model
Relationships	Select to generate relationships of model
References	Select to generate references of model
References documentation	Select to generate references documentation
Sub-diagrams	Select to generate sub-diagrams of model
Tagged values	Select to generate tagged values of model
Comments	Select to generate comments of model
Use Case Details	Select to generate use case details
Anti-aliasing	
Graphics	To enable/disable the graphic anti-aliasing of the diagram images.
Text	To enable/disable the text anti-aliasing of the diagram images.

Table 2-2 *Details of generate Word dialog's options tab*

Page Setup



Page size: A4

Page Orientation: ● Portrait ○ LandScape

Measurement Unit: inch

Top: 1 inches

Left: 1 inches

Right: 1 inches

Page size: 8.26 x 11.69

Bottom: 1 inches

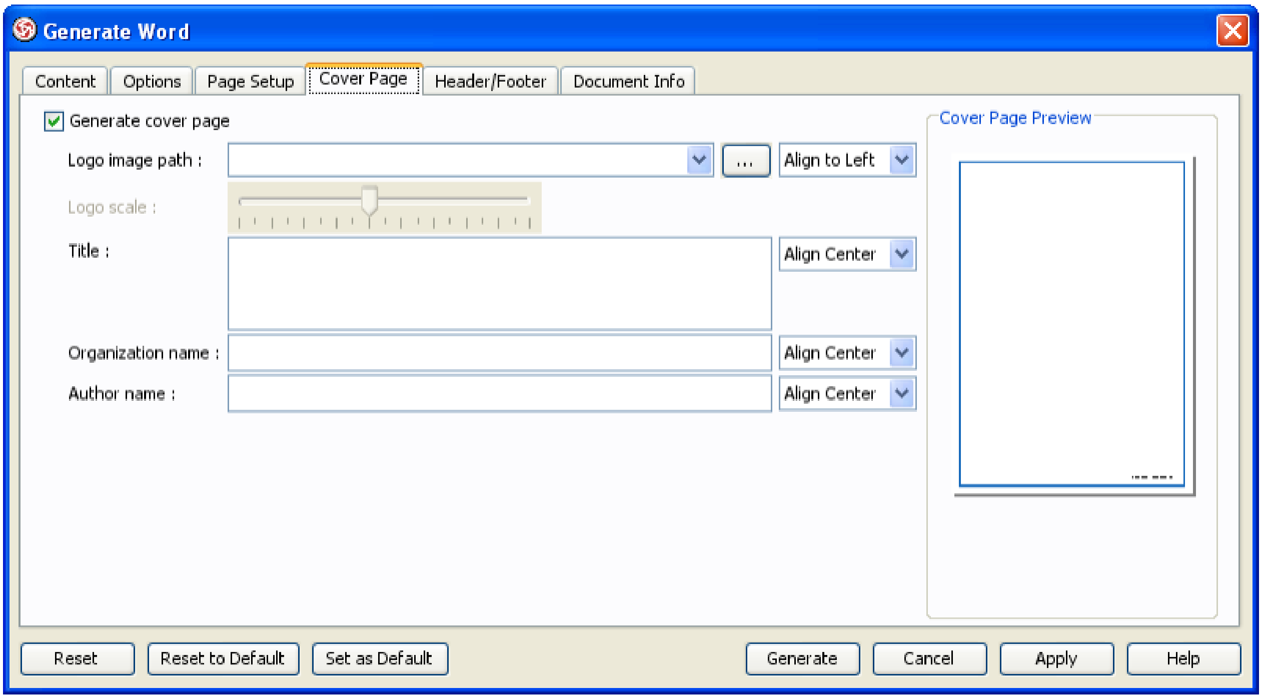
Figure 2-16 *Page setup tab of generate Word dialog*

Details of each field in Page Setup tab are shown as below:

Field	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).
Page Margin	To specify the page margins of the report.

Table 2-3 *Details of generate Word dialog's page setup tab*

Customizing Cover Page



☑ Generate cover page

Logo image path : Align to Left

Logo scale :

Title : Align Center

Organization name : Align Center

Author name : Align Center

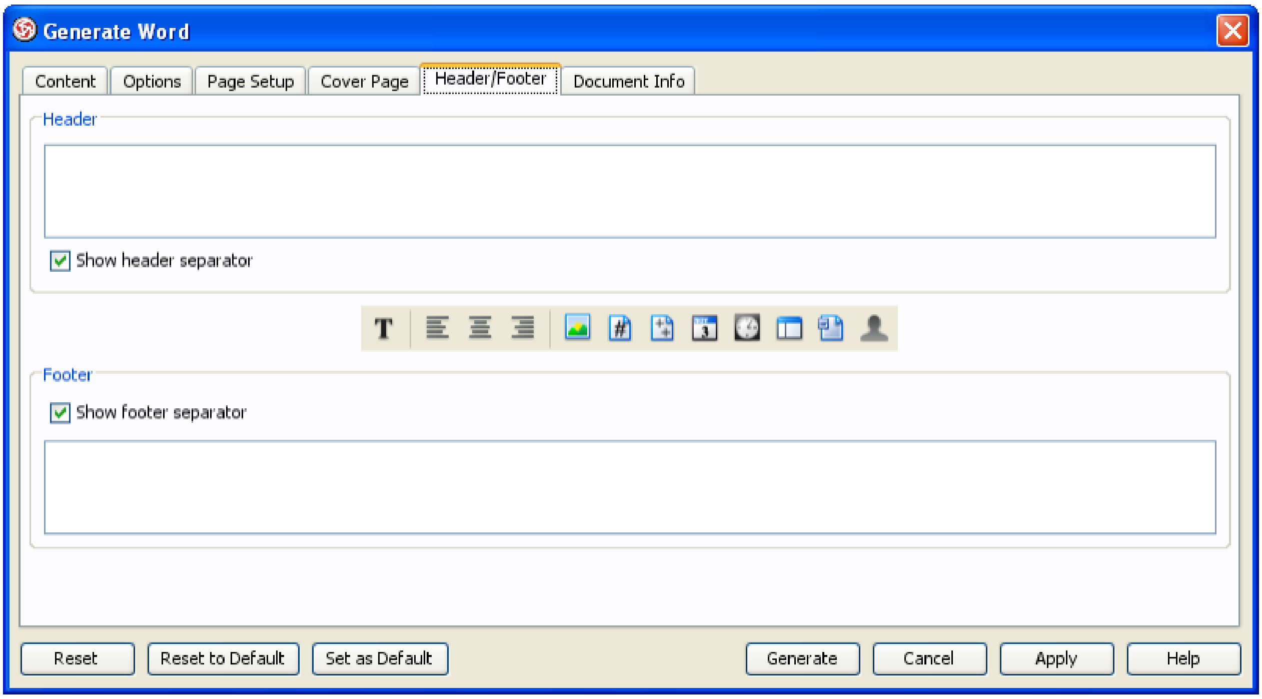
***Figure 2-17** Cover page tab of generate Word dialog*

Details of each field in Cover Page tab are shown as below:

Field	Description
Generate cover page	If this option is selected, cover page for this document will be generated to the report
Logo image path	Word report's logo image path
Logo scale	Word report's logo scale
Title	Word report's title
Organization name	Word report's organization name
Author name	Word report's author name

***Table 2-4** Details of generate Word dialog's cover page tab*

Customizing Header/Footer



☑ Show header separator

☑ Show footer separator

__Figure 2-18__ Header/Footer tab of generate Word dialog

Details of each field in Header/Footer tab are shown as below:

Field	Description
T	To format text style, such as font style, size and color.
≡	Align text to left
≡	Align text to center
≡	Align text to right
🖼	To insert an image to the header or footer
#	Insert page number
+	Insert page count
3	Insert date
🕒	Insert time
▭	Insert project name
📄	Insert report file name
👤	Insert user name

__Table 2-5__ Details of generate Word dialog's header/footer tab

Filling Document Info

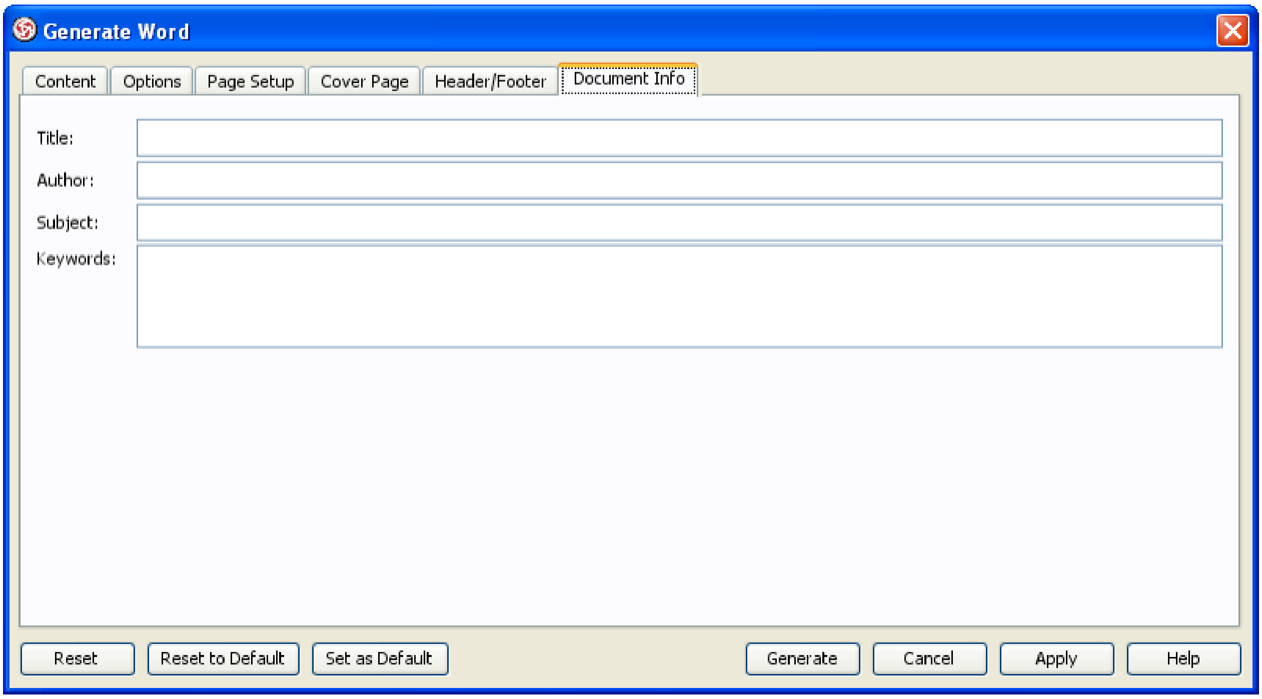


Figure 2-19 *Document info tab of generate Word dialog*

Details of each field in Document Info tab are shown as below:

Field	Description
Title	The title of the report.
Author	The author of the report.
Subject	The subject of the report.
Keywords	The keywords of the report.

Table 2-6 *Details of generate Word dialog's document info tab*

Generating HTML Report

The Generate HTML dialog box provides a set of options for changing the report style. To display the dialog box, perform one of the following actions:

Select Tools > Report > Generate HTML Report... from main menu.

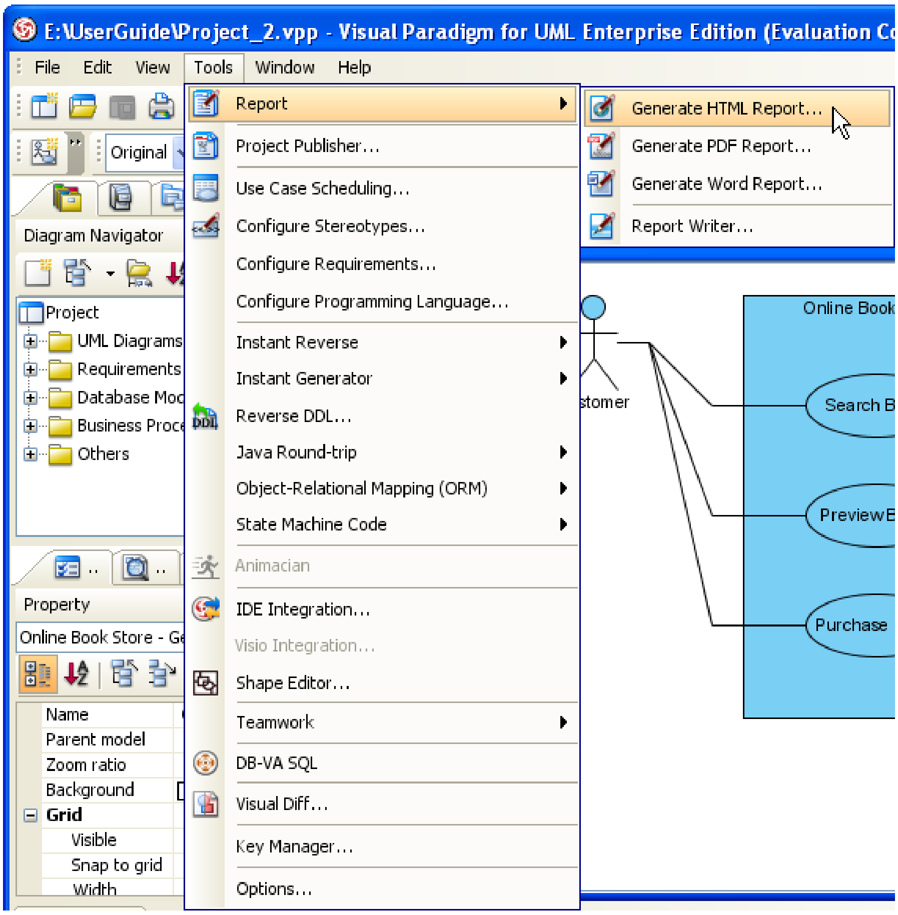


Figure 3-1 Generate HTML from main menu

Or click on the Generate HTML Report button on the toolbar.

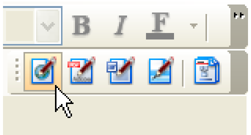
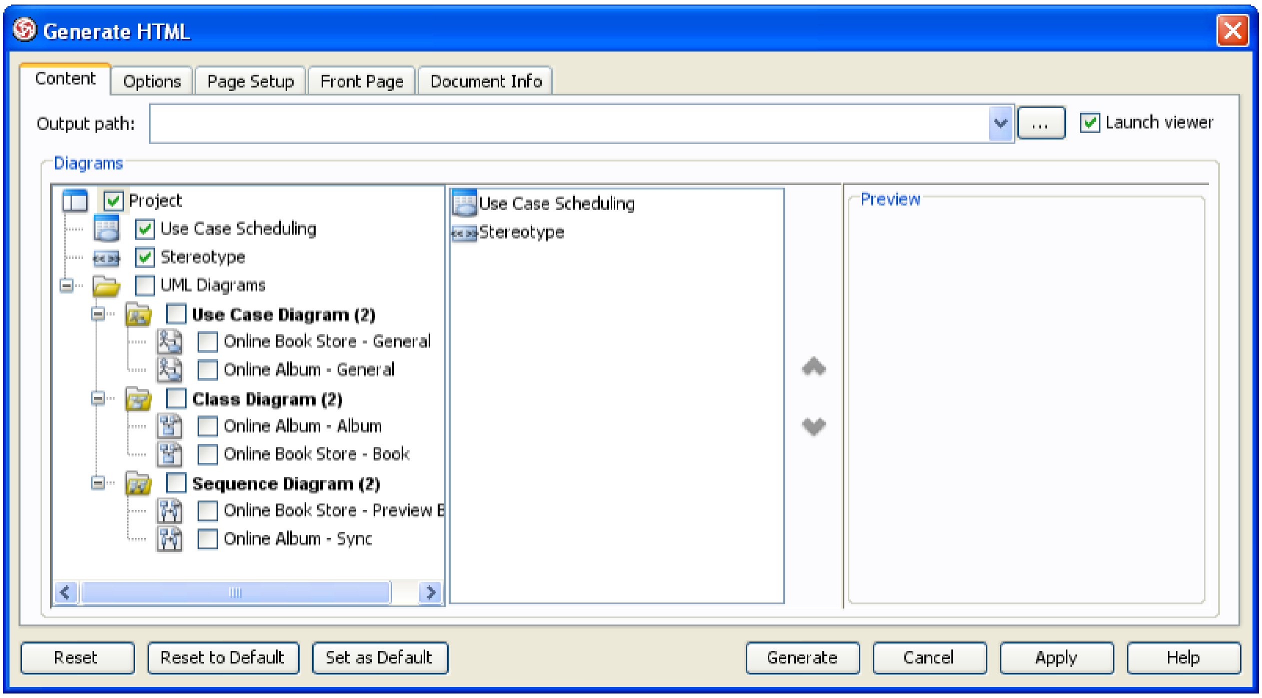


Figure 3-2 Generate HTML from tool bar

Generate HTML dialog



Output path:

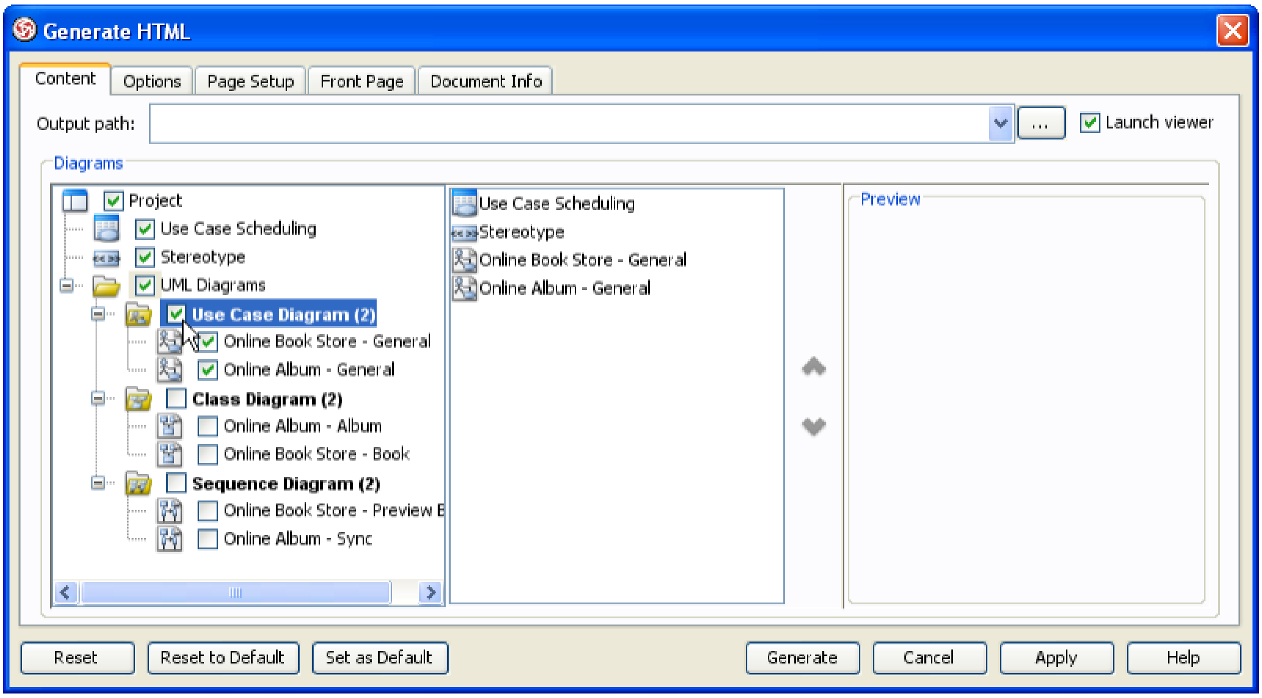
☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - □ UML Diagrams
    - □ **Use Case Diagram (2)**
      - □ Online Book Store - General
      - □ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Stereotype

Figure 3-3 Generate HTML dialog

To generate a HTML Report, open the Generate HTML dialog box and select diagrams for the generation.



Output path:

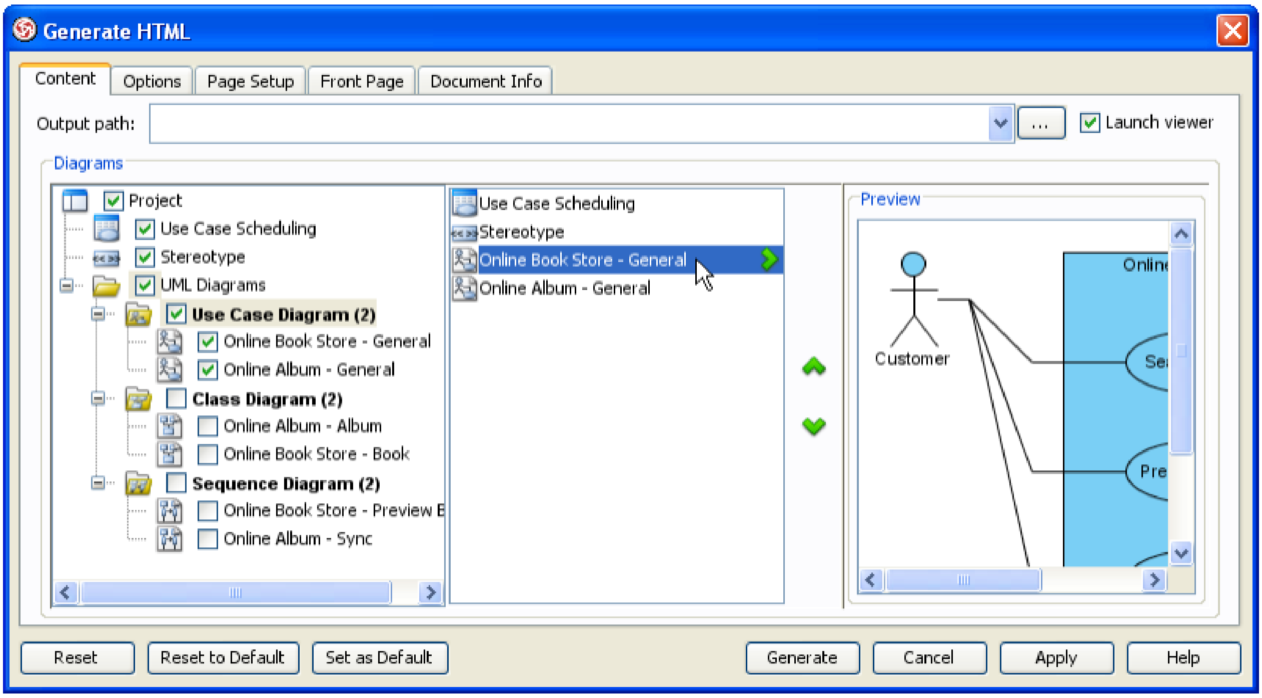
☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - ☑ UML Diagrams
    - ☑ **Use Case Diagram (2)**
      - ☑ Online Book Store - General
      - ☑ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

Figure 3-4 Selecting diagram to generate HTML

It is possible to preview diagram by selecting it as below.



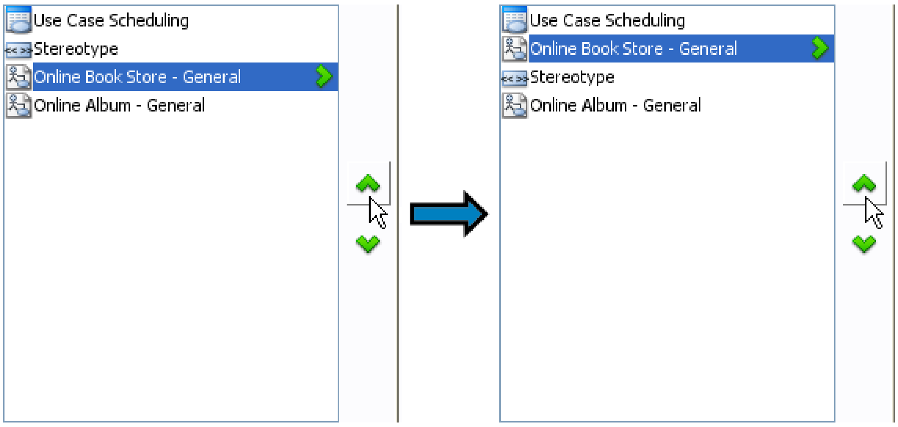
Output path:
☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - ☑ UML Diagrams
    - ☑ **Use Case Diagram (2)**
      - ☑ Online Book Store - General
      - ☑ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview B
      - □ Online Album - Sync

Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

Figure 3-5 Showing preview of diagram

Also ▲ or ▼ may enable for reordering.

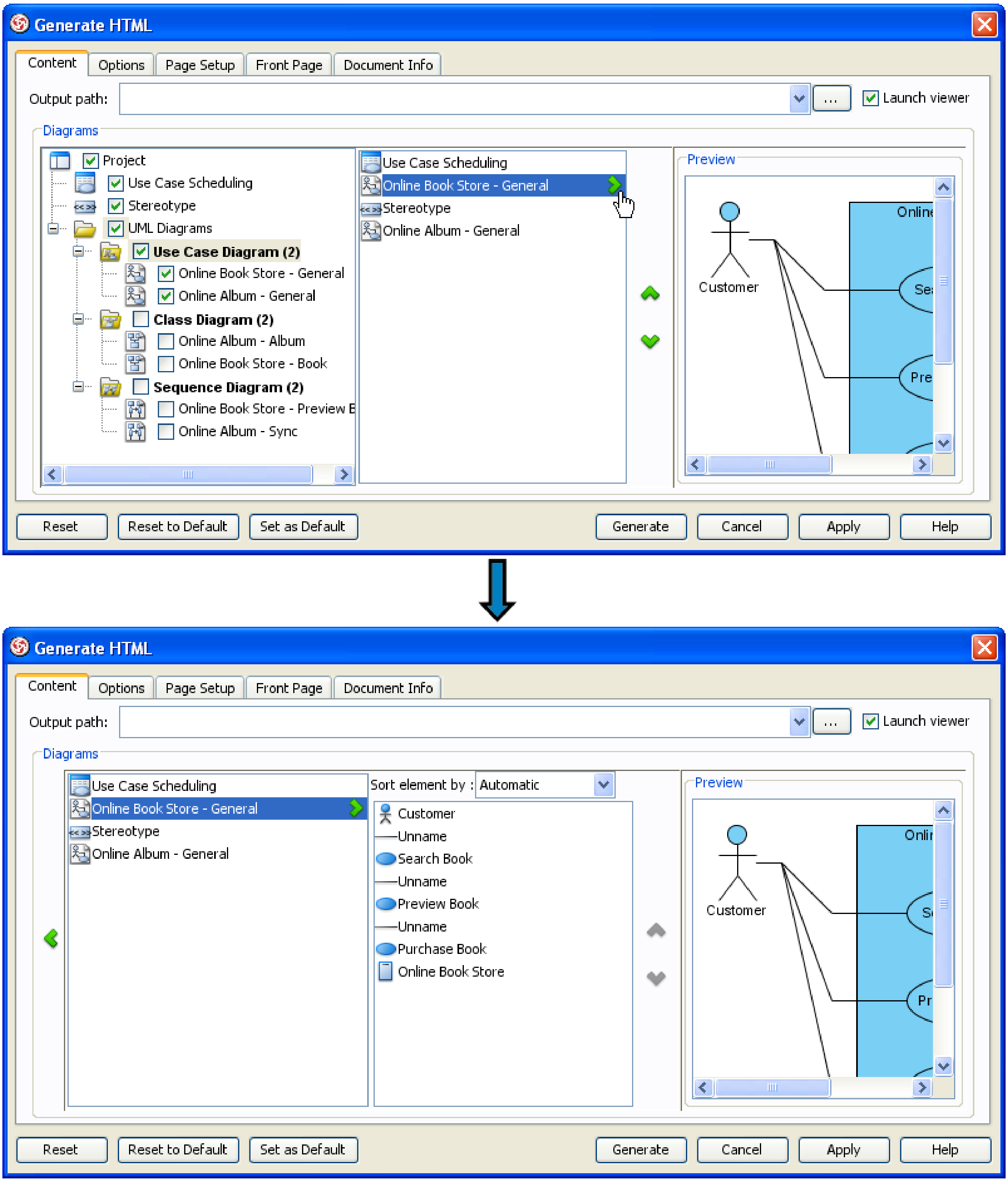


Use Case Scheduling
Stereotype
Online Book Store - General
Online Album - General

Use Case Scheduling
Online Book Store - General
Stereotype
Online Album - General

***Figure 3-6** Reording diagram*

To configure diagram's element, press ➔ to change view first.



Output path:

☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - ☑ UML Diagrams
    - ☑ Use Case Diagram (2)
      - ☑ Online Book Store - General
      - ☑ Online Album - General
    - □ Class Diagram (2)
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ Sequence Diagram (2)
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Online Book Store - General
Stereotype
Online Album - General

Sort element by : Automatic

Customer
Unname
Search Book
Unname
Preview Book
Unname
Purchase Book
Online Book Store

Figure 3-7 View diagram elements of diagram

There is four kinds of sorting for element.

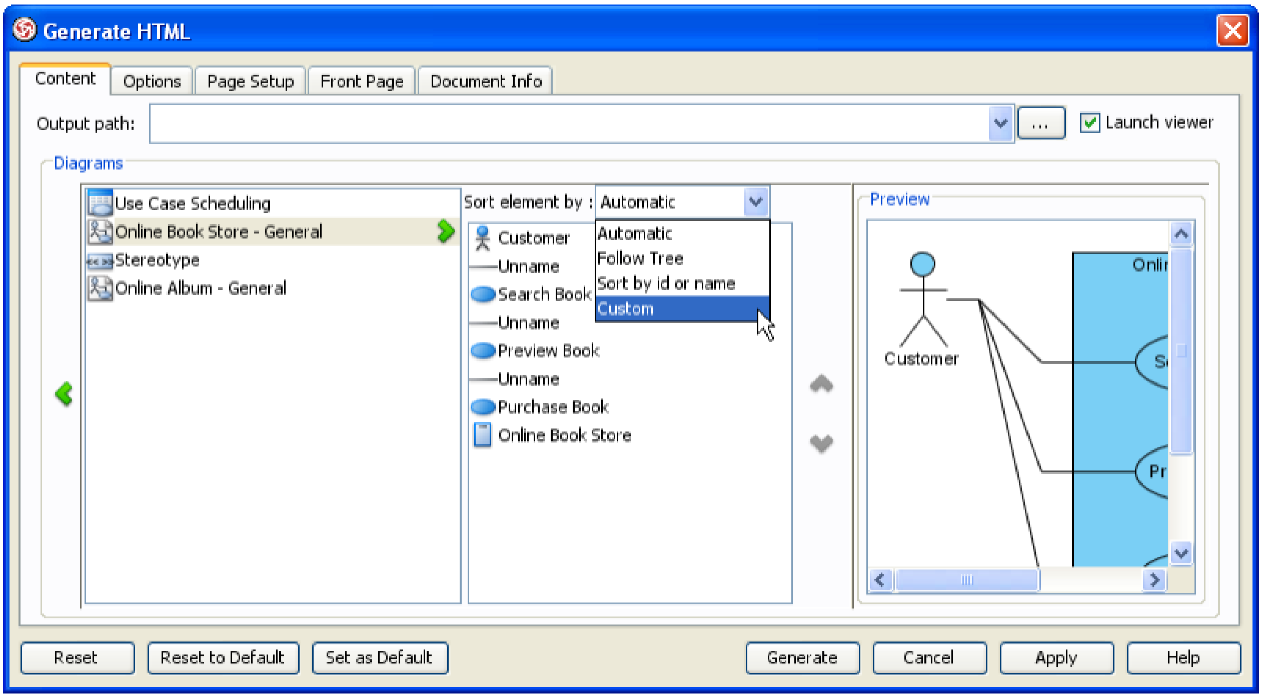


Figure 3-8 *Kinds of sort element*

Sort element by Custom allow to reorder element manually.

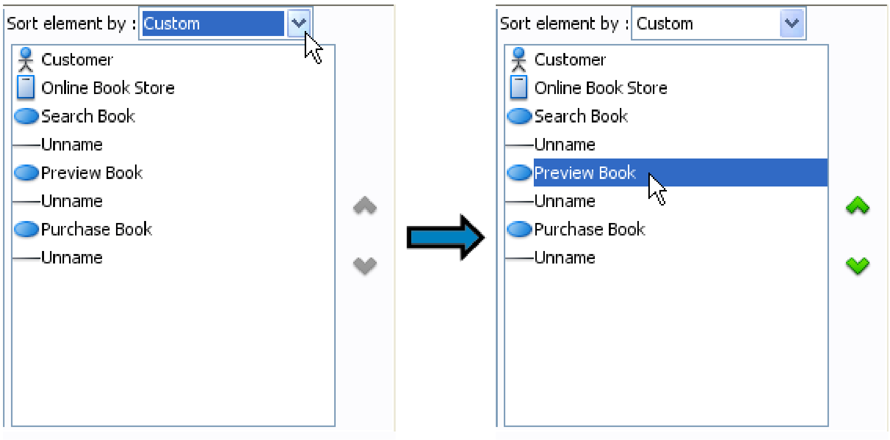


Figure 3-9 *Selecting element*

Press ▲ or ▼ to reorder element as preferred manually.

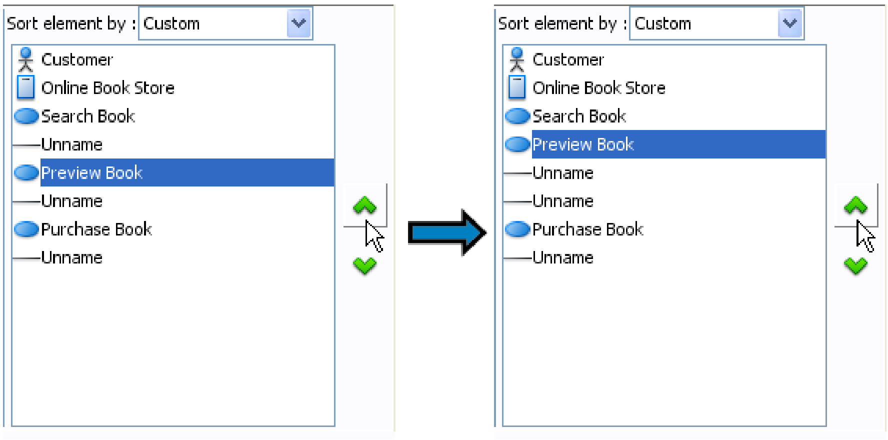
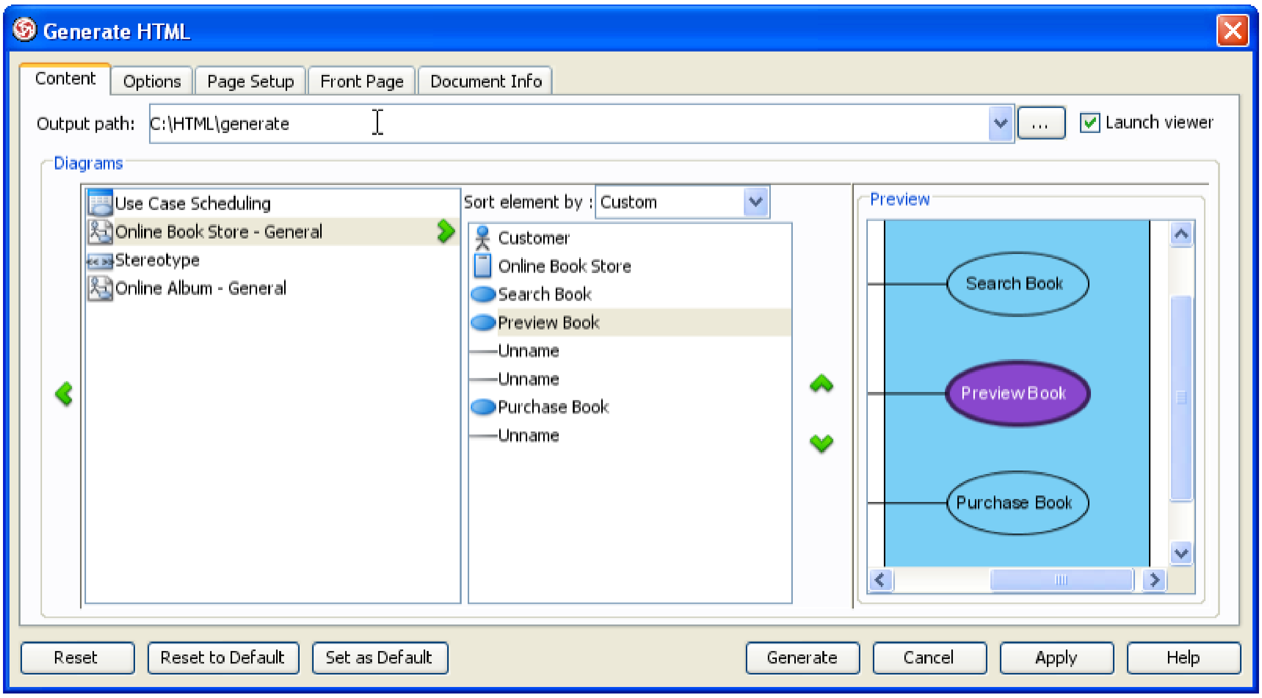


Figure 3-10 *Reordering element*

Don't forget to input Output path of the generate HTML.

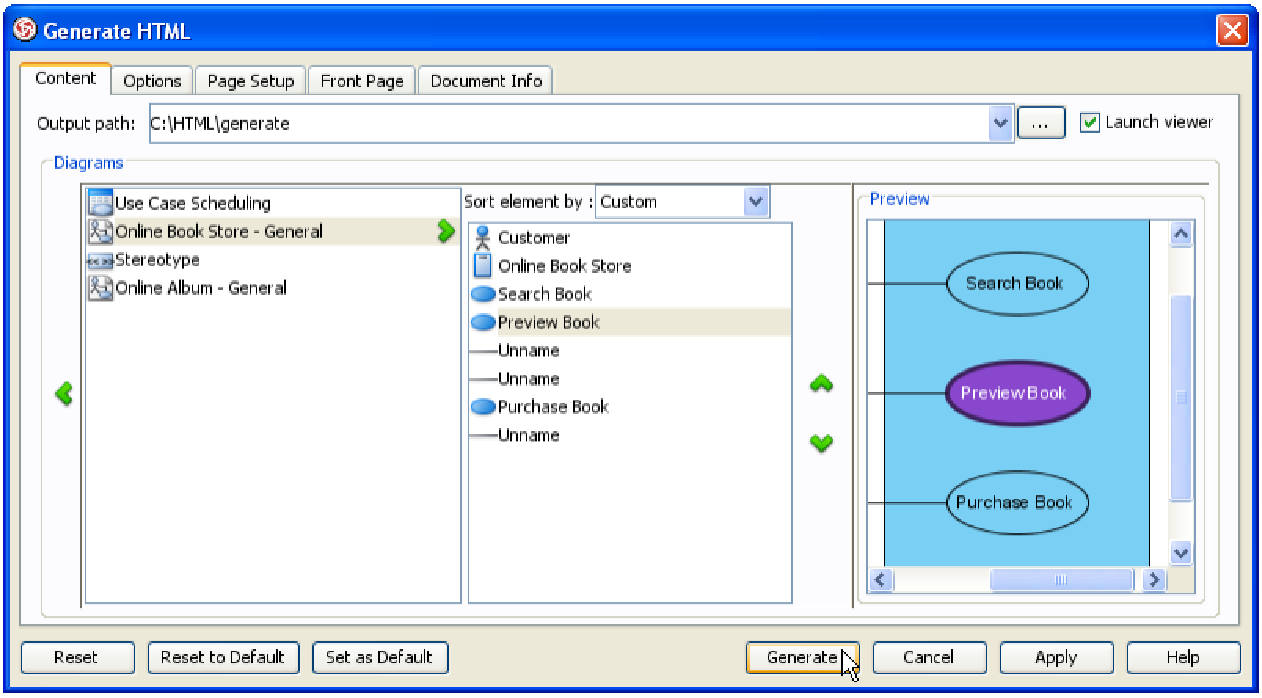

Output path: C:\HTML\generate

☑ Launch viewer

Sort element by : Custom

Figure 3-11 *Inputting output path*

Press Generate button to perform the HTML generation.

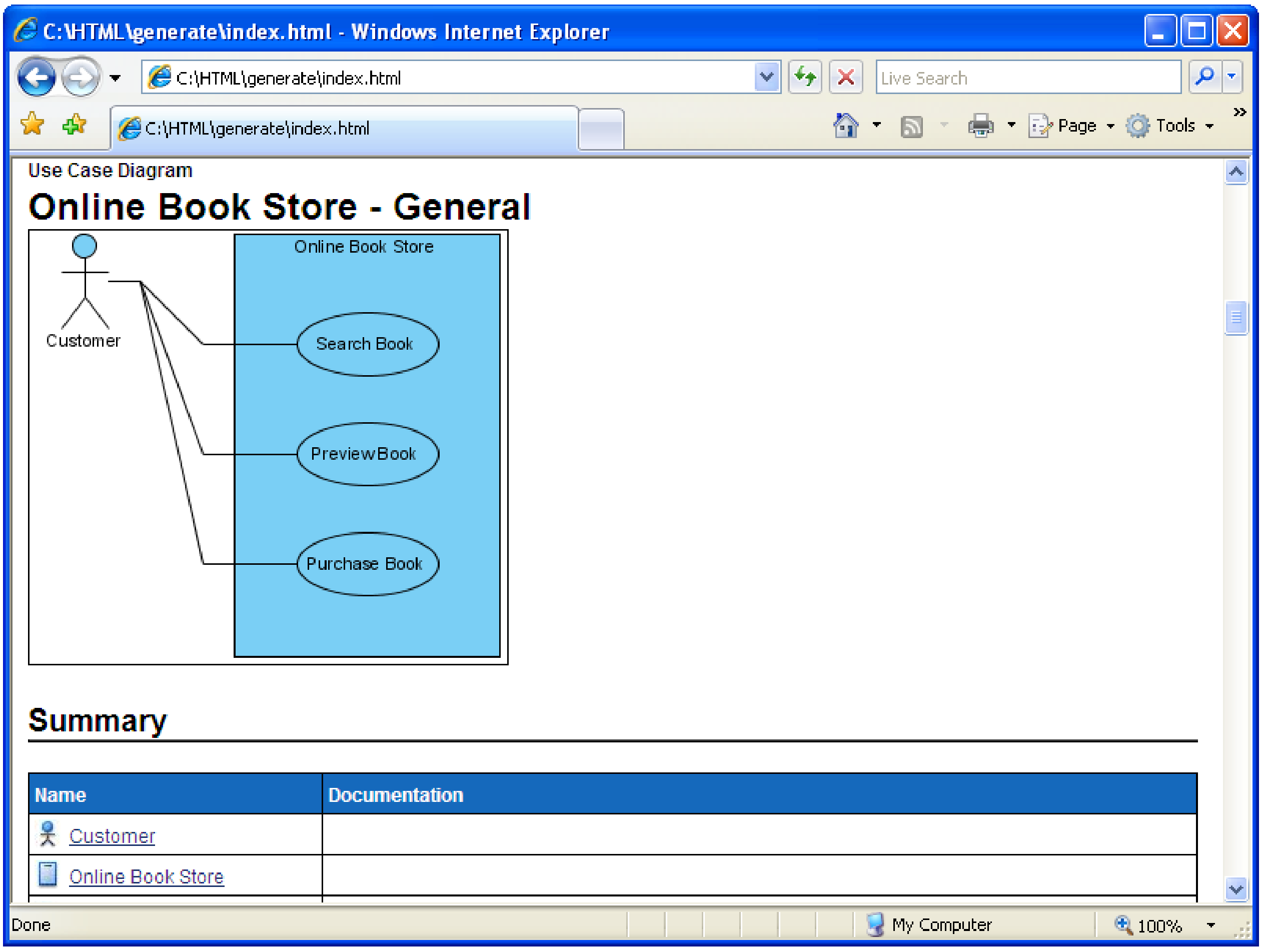

Output path: C:\HTML\generate

☑ Launch viewer

Sort element by : Custom

Figure 3-12 *Generate HTML*

Generated HTML will looks like below.


C:\HTML\generate\index.html - Windows Internet Explorer
C:\HTML\generate\index.html
Live Search
C:\HTML\generate\index.html
Page
Tools
Done
My Computer
100%

Use Case Diagram

Online Book Store - General

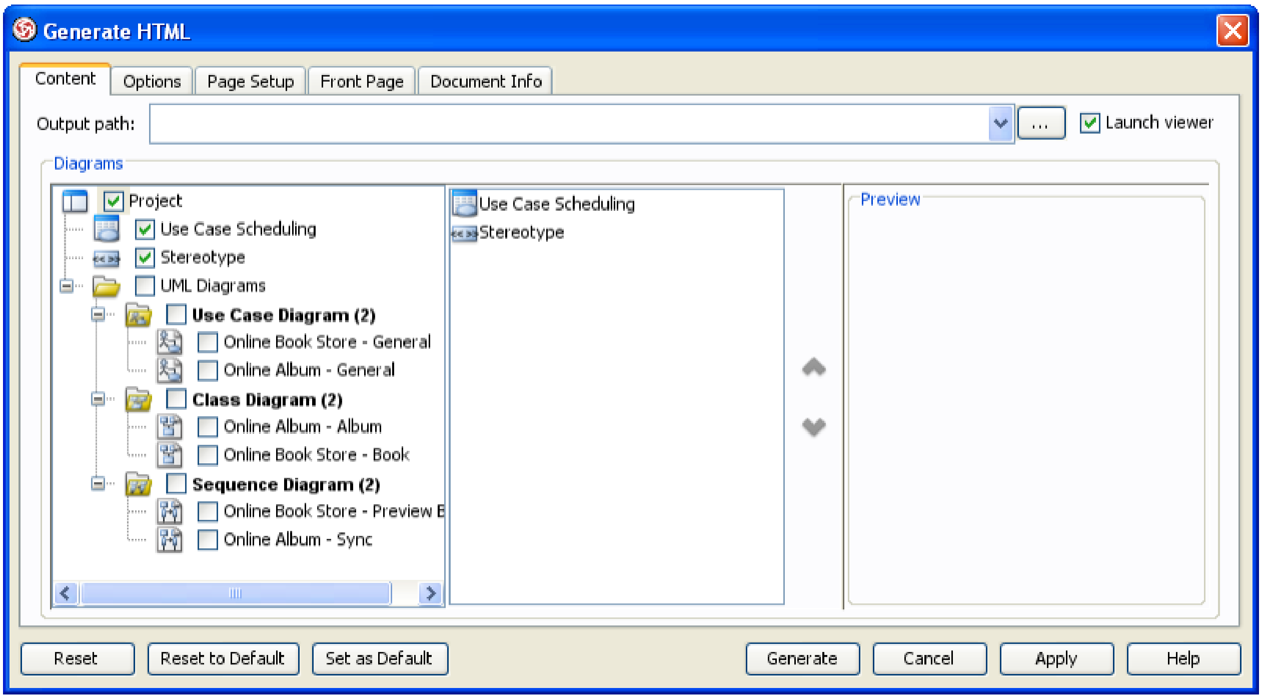
Online Book Store
Customer
Search Book
PreviewBook
Purchase Book

Summary

Name	Documentation
Customer	
Online Book Store	

Figure 3-13 Generated HTML

Customizing Report generation options



☑ Launch viewer

- ☑ Project
  - ☑ Use Case Scheduling
  - ☑ Stereotype
  - □ UML Diagrams
    - □ **Use Case Diagram (2)**
      - □ Online Book Store - General
      - □ Online Album - General
    - □ **Class Diagram (2)**
      - □ Online Album - Album
      - □ Online Book Store - Book
    - □ **Sequence Diagram (2)**
      - □ Online Book Store - Preview E
      - □ Online Album - Sync

Use Case Scheduling
Stereotype

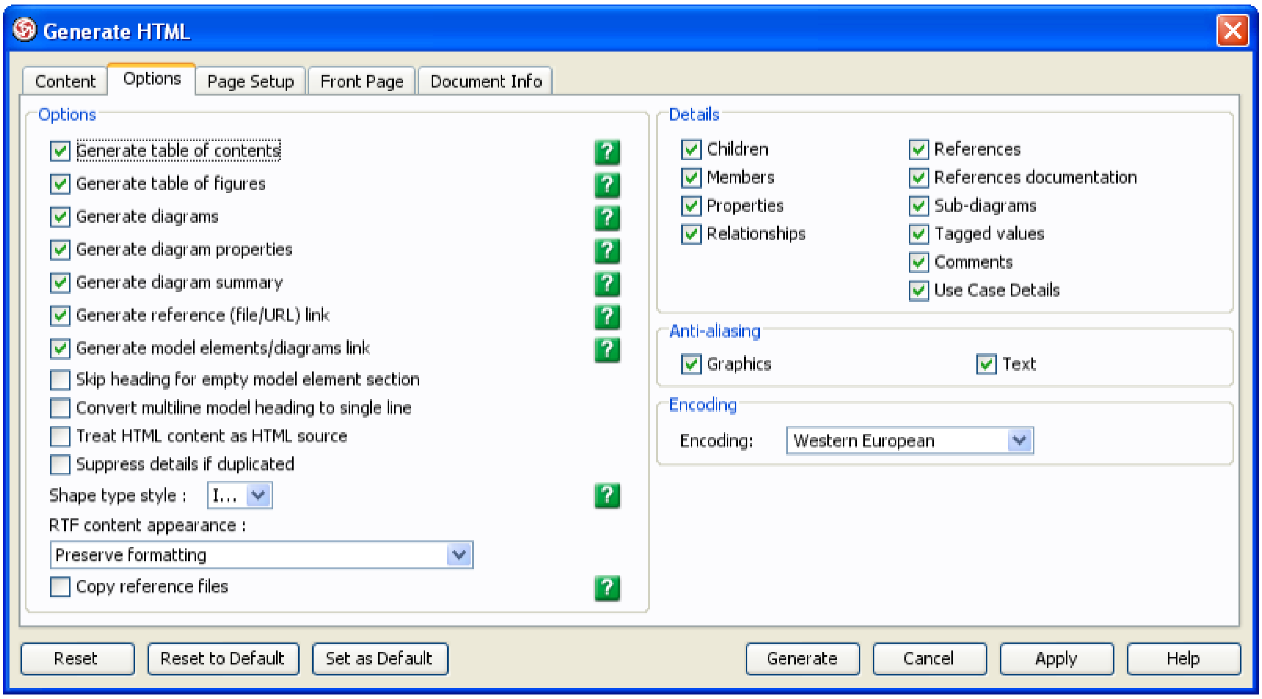
Figure 3-14 *Content tab of generate HTML dialog*

Details of each field in Content tab are shown as below:

Field	Description
Output path	To select the destination file for the generated report. You can type the path in the text field or you can browse the location by clicking on the ...button.
Launch viewer	If this option is selected, the default browser of the system will be opened automatically to show the generated document.
Sort element by	Automatic - sorting elements by listing them in the most logical order, which is to follow most users&rsquo; understanding of that kind of diagram Follow tree - sorting elements by following the sort order of the diagram tree in the tool id or name - sorting elements by their ID or names You can refer to the section 'Sorting Elements in Report'.

Table 3-1 *Details of generate HTML dialog's content tab*

Customizing Report generation options



Options

☑ Generate table of contents

☑ Generate table of figures

☑ Generate diagrams

☑ Generate diagram properties

☑ Generate diagram summary

☑ Generate reference (file/URL) link

☑ Generate model elements/diagrams link

□ Skip heading for empty model element section

□ Convert multiline model heading to single line

□ Treat HTML content as HTML source

□ Suppress details if duplicated

Shape type style : I...

RTF content appearance :

Preserve formatting

□ Copy reference files

Details

☑ Children

☑ Members

☑ Properties

☑ Relationships

☑ References

☑ References documentation

☑ Sub-diagrams

☑ Tagged values

☑ Comments

☑ Use Case Details

Anti-aliasing

☑ Graphics

☑ Text

Encoding

Encoding: Western European

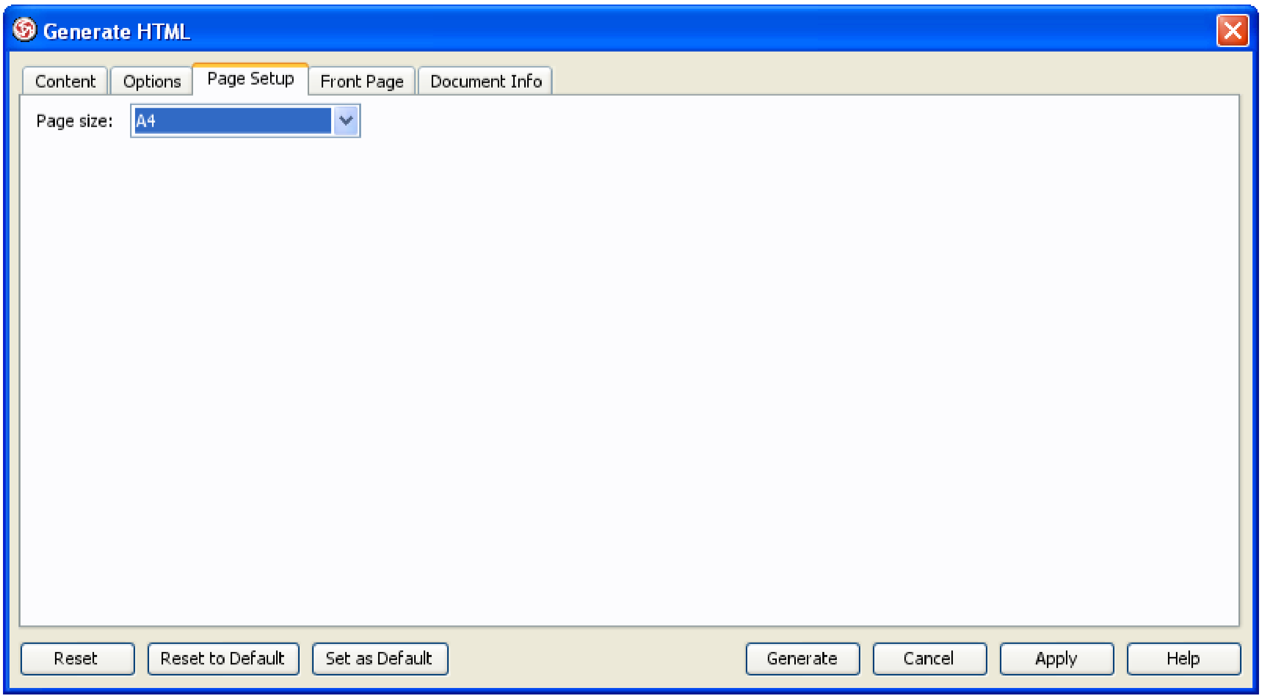
Figure 3-15 *Options tab of generate HTML dialog*

Details of each field in Options tab are shown as below:

Field	Description
Options	
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.
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.
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.
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.
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, reference files will be copy to generated report.
Details	
Children	Select to generate children of model
Members	Select to generate members of model
Properties	Select to generate properties of model
Relationships	Select to generate relationships of model
References	Select to generate references of model
References documentation	Select to generate references documentation
Sub-diagrams	Select to generate sub-diagrams of model
Tagged values	Select to generate tagged values of model
Comments	Select to generate comments of model
Use Case Details	Select to generate use case details
Anti-aliasing	
Graphics	To enable/disable the graphic anti-aliasing of the diagram images.
Text	To enable/disable the text anti-aliasing of the diagram images.
Encoding	
Encoding	Specify encoding to be used.

Table 3-2 *Details of generate HTML dialog's options tab*

Page Setup



Page size: A4

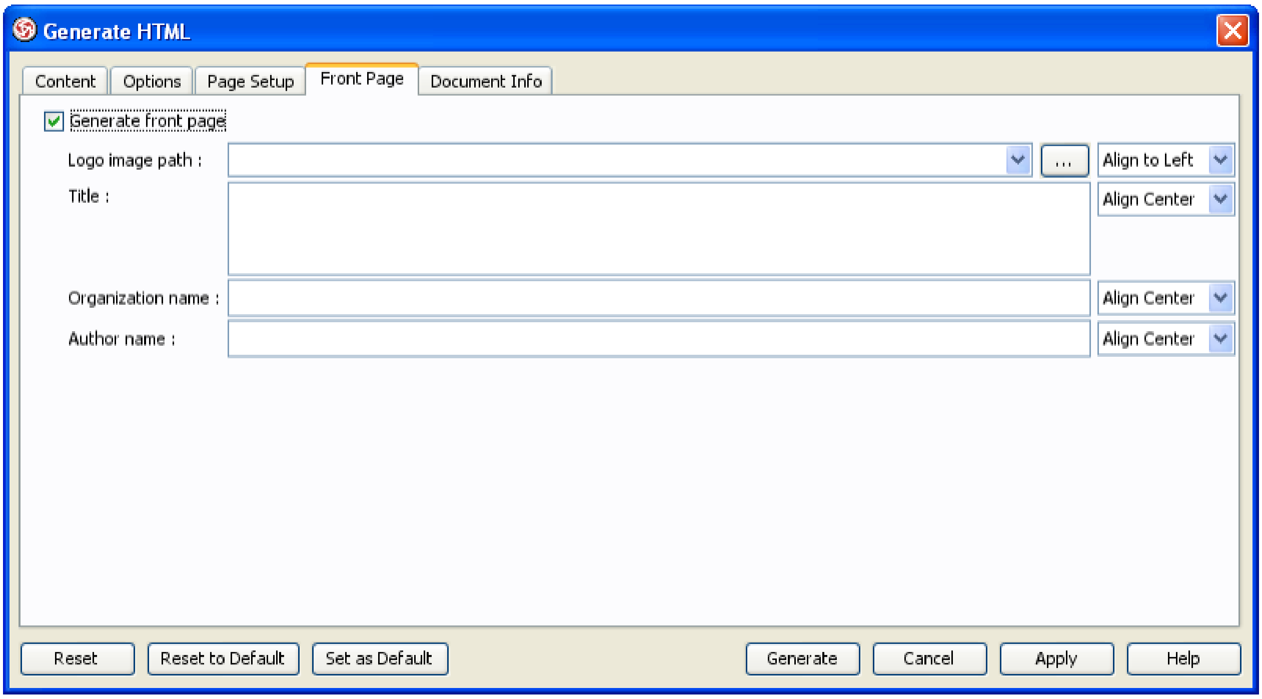
Figure 3-16 *Page setup tab of generate HTML dialog*

Details of each field in Page Setup tab are shown as below:

Field	Description
Page size	To select the paper size of the generated report.

Table 3-3 *Details of generate HTML dialog's page setup tab*

Customizing Cover Page



Generate HTML

Content | Options | Page Setup | Front Page | Document Info

☑ Generate front page

Logo image path : Align to Left

Title : Align Center

Organization name : Align Center

Author name : Align Center

Figure 3-17 *Cover page tab of generate HTML dialog*

Details of each field in Cover Page tab are shown as below:

Field	Description
Generate front page	If this option is selected, front page for this document will be generated to the report
Logo image path	HTML report's logo image path
Title	HTML report's title
Organization name	HTML report's organization name
Author name	HTML report's author name

Table 3-4 *Details of generate HTML dialog's cover page tab*

Filling Document Info

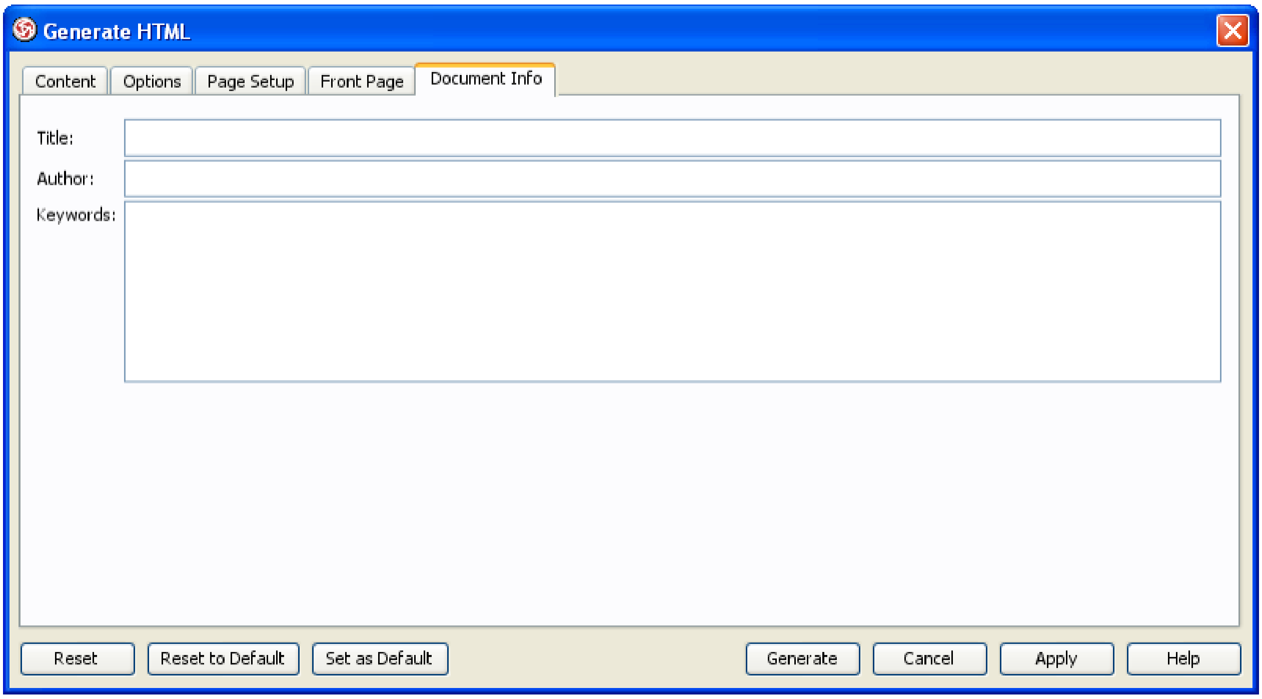


Figure 3-18 *Document info tab of generate HTML dialog*

Details of each field in Document Info tab are shown as below:

Field	Description
Title	The title of the report.
Author	The author of the report.
Keywords	The keywords of the report.

Table 3-5 *Details of generate HTML dialog's document info tab*

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, perform one of the following actions:

Select Tools > Project Publisher...from main menu.

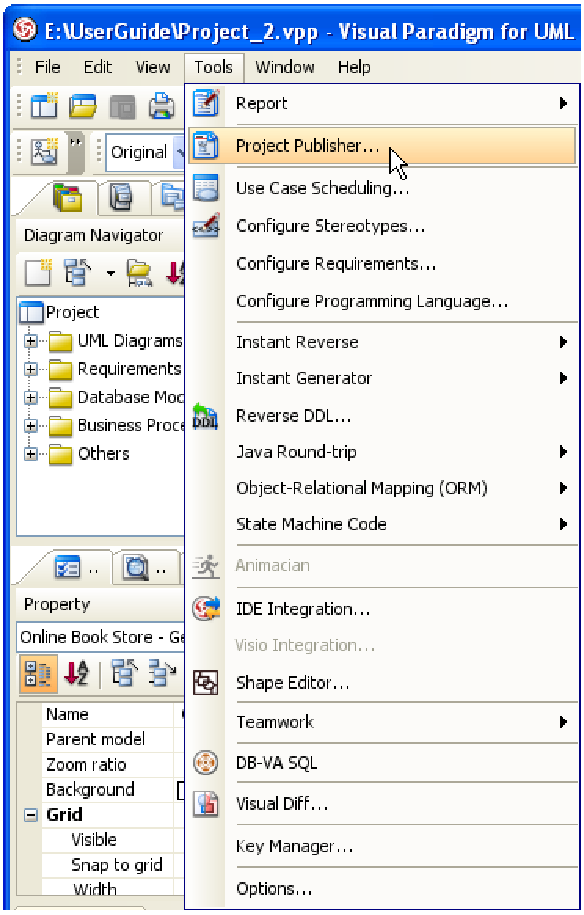


Figure 4-1 *Start project publisher from main menu*

Or click on the Project Publisher button on the toolbar.

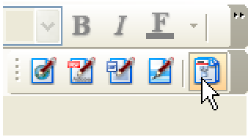
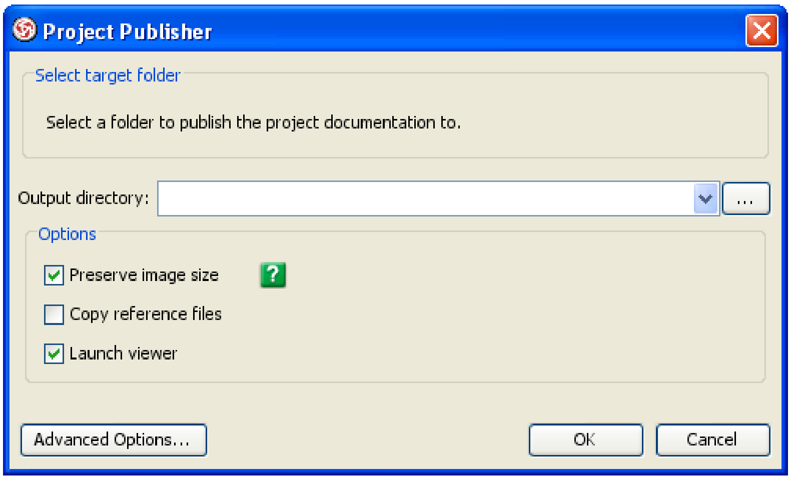


Figure 4-2 *Start project publisher from tool bar*

The Project Publisher dialog box appears.



Select a folder to publish the project documentation to.

Output directory:

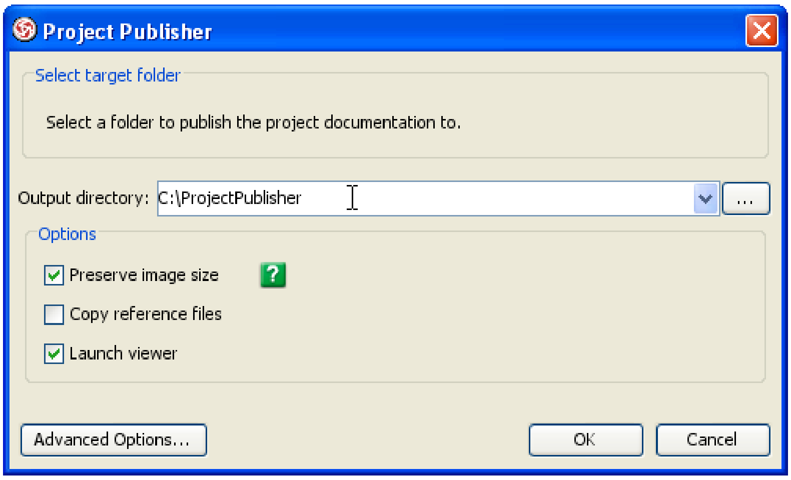
☑ Preserve image size

☐ Copy reference files

☑ Launch viewer

Figure 4-3 *Project publisher dialog*

To publish the project, you need to enter the Output directory where the published files will be saved to. You may select the Launch viewer option so the default web browser on your computer will open the index page of the published project when the process is completed.



Select a folder to publish the project documentation to.

Output directory: C:\ProjectPublisher

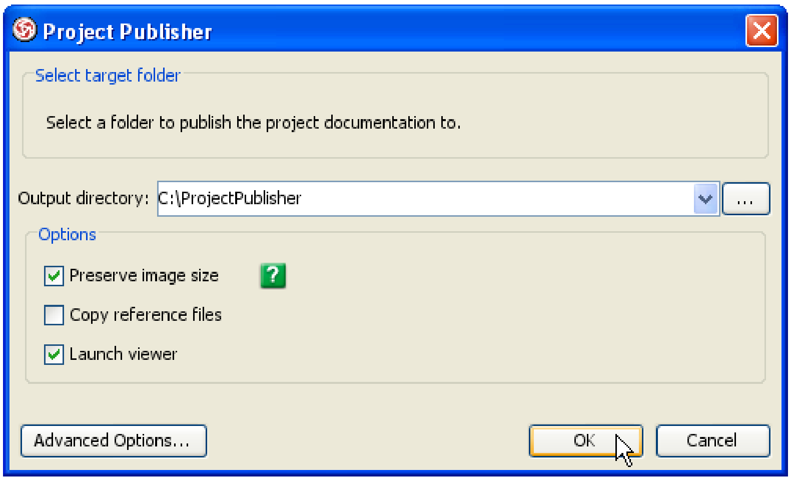
☑ Preserve image size

☐ Copy reference files

☑ Launch viewer

Figure 4-4 *Inputting output directory*

Click OK button to start publishing. The progress dialog box will appear while generating the content and a ' Project publishing complete' message will show once it is done.



Select a folder to publish the project documentation to.

Output directory: C:\ProjectPublisher

☑ Preserve image size

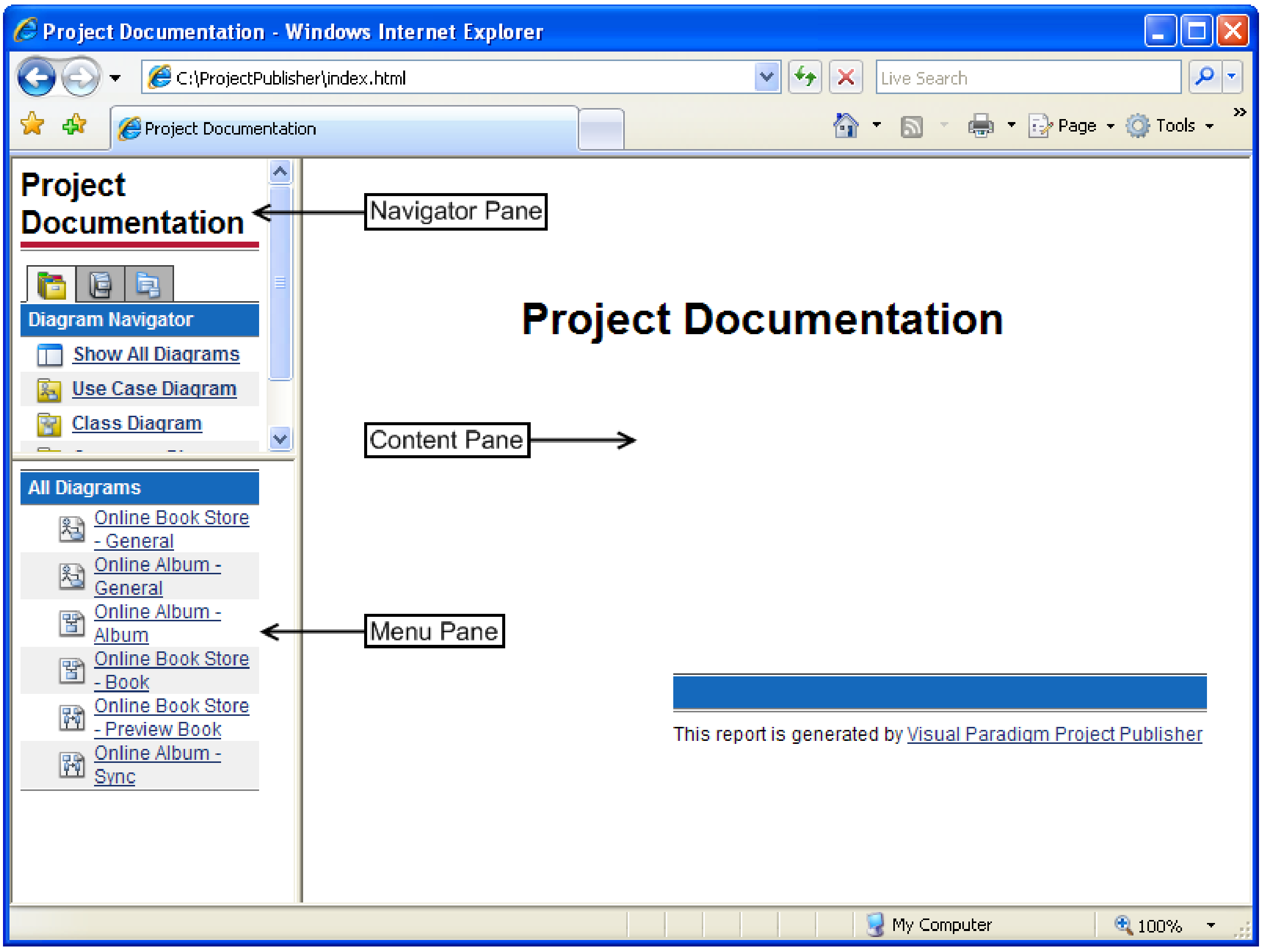
☐ Copy reference files

☑ Launch viewer

Figure 4-5 *Publishing project*

Using the published content

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.



This report is generated by Visual Paradigm Project Publisher

Figure 4-6 Published project

Navigator Pane

It comprises of the Diagram Navigator, Model Navigator and Class 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.

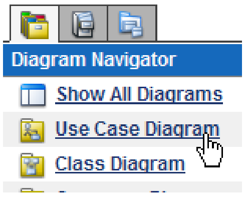


Figure 4-7 Diagram navigator

Model Navigator 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.

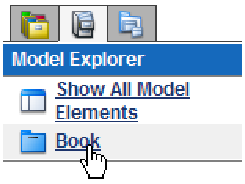
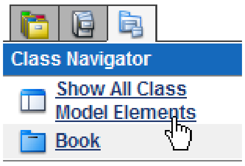


Figure 4-8 Model Navigator

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

Show All Class Model Elements

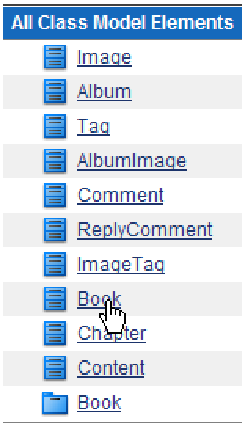
Book

Figure 4-9 *Class navigator*

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.

To view the details of an item (diagram, model or package/class), click on its link in the Menu Pane and its details will be shown in the Content Pane.



All Class Model Elements

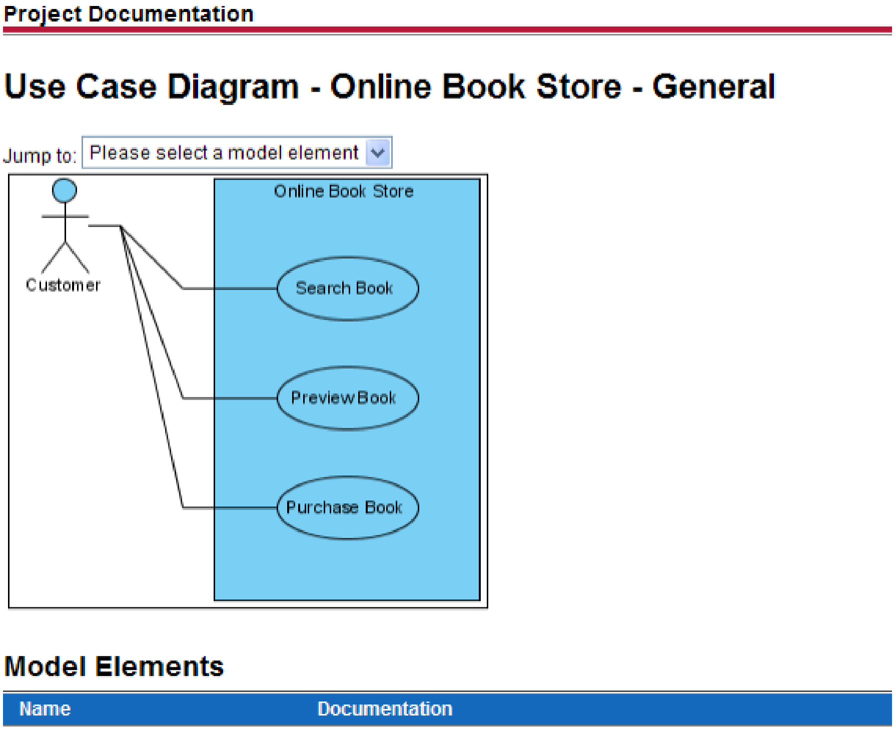
Image
Album
Tag
AlbumImage
Comment
ReplyComment
ImageTag
Book
Chapter
Content
Book

Figure 4-10 *Menu navigator*

Content Pane

It shows the details of the item (diagram, model or package/class) you clicked in the Menu Pane or Content Pane.

Diagram Content



Project Documentation

Use Case Diagram - Online Book Store - General

Jump to: Please select a model element

Model Elements

Name	Documentation

Figure 4-11 *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.

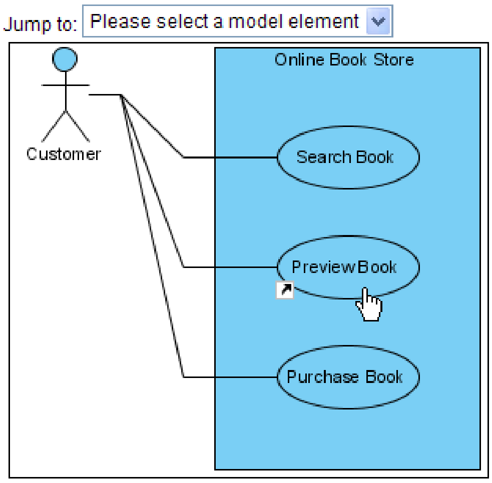


Figure 4-12 *Shape link to descriptions*

Using Jump to

The Jump to combo box 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.

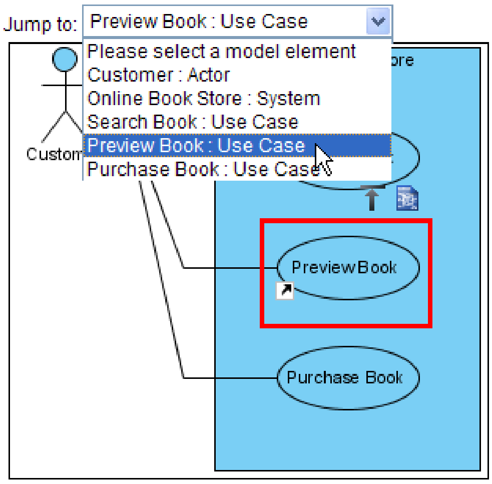


Figure 4-13 *Jump to*

Besides, there will be two shortcut buttons above the selected shape.

The Back to top button brings you to the top of the page.

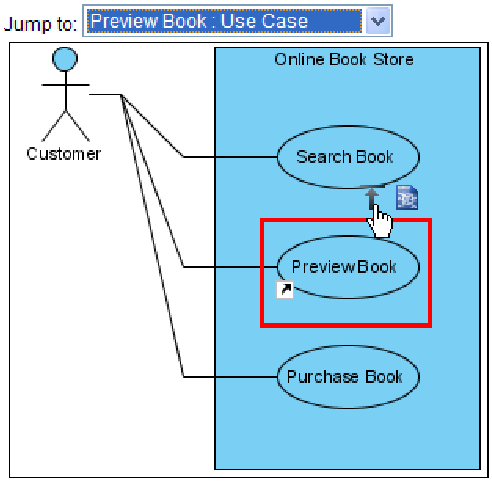


Figure 4-14 *Back to top*

The Open specification button brings you to the details page of the shape.

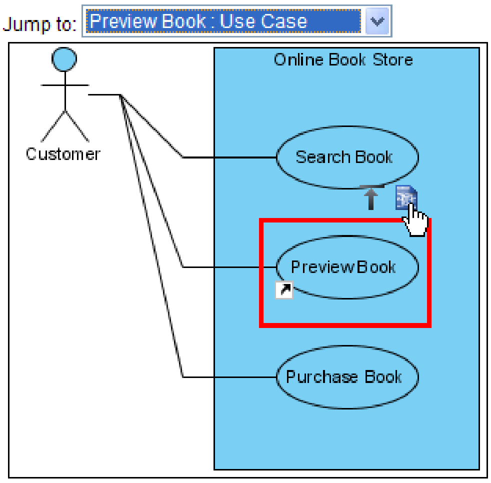
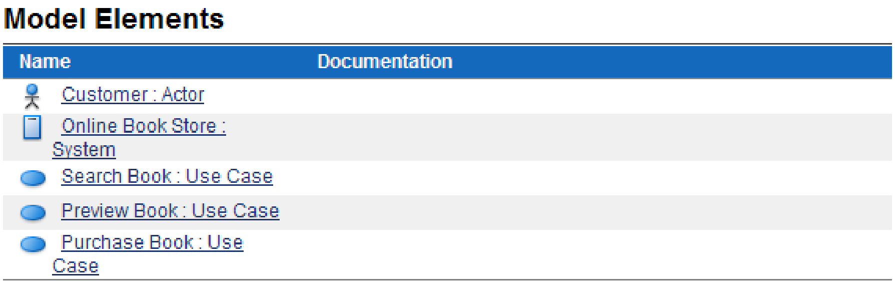

Jump to: Preview Book : Use Case

Figure 4-15 *Open specification*

Models

The Models section of the diagram content page shows the name, type and documentation of the models of all shapes in the diagram. You can click on the link of a model to view its details.

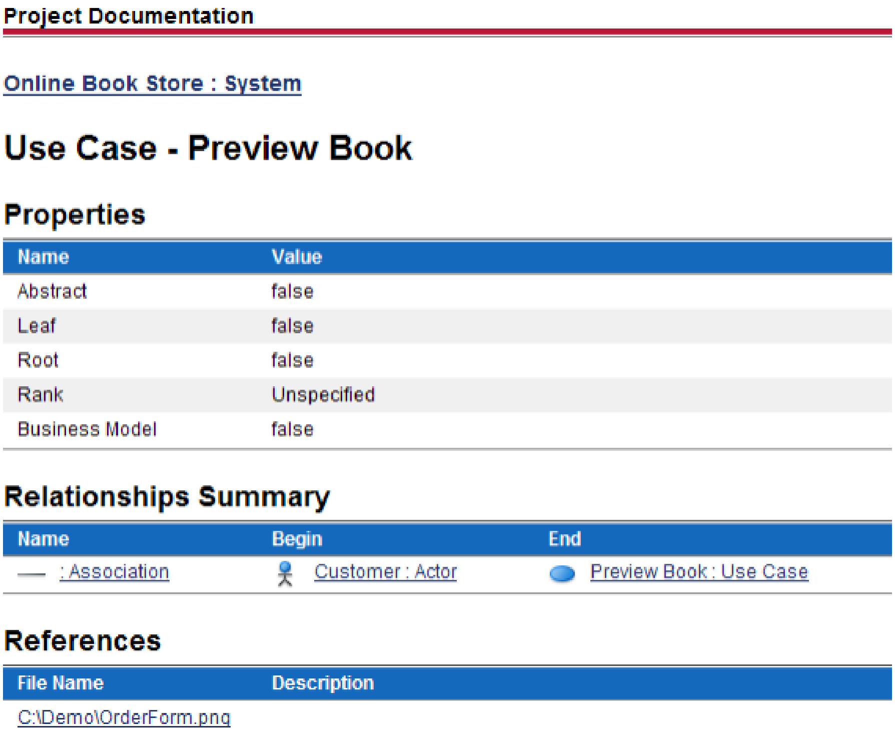


Model Elements

Name	Documentation
Customer : Actor	
Online Book Store : System	
Search Book : Use Case	
Preview Book : Use Case	
Purchase Book : Use Case	

Figure 4-16 *Model elements*

Model Content



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

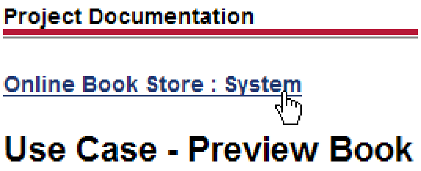
File Name	Description
C:\Demo\OrderForm.png	

Figure 4-17 *Model content*

The type, name and general model properties of a model are shown in the content page.

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

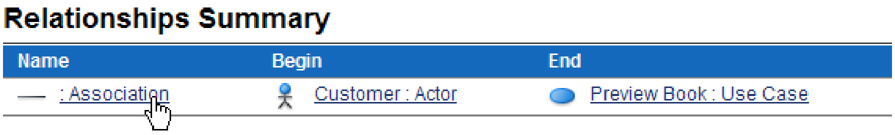
Online Book Store : System

Use Case - Preview Book

__Figure 4-18__ Parent hierarchy

Relationships

The summary of the relationships of the model is shown in the Relationships Summary section. Click on a relationship and it will take you to the Relationships Detail section.

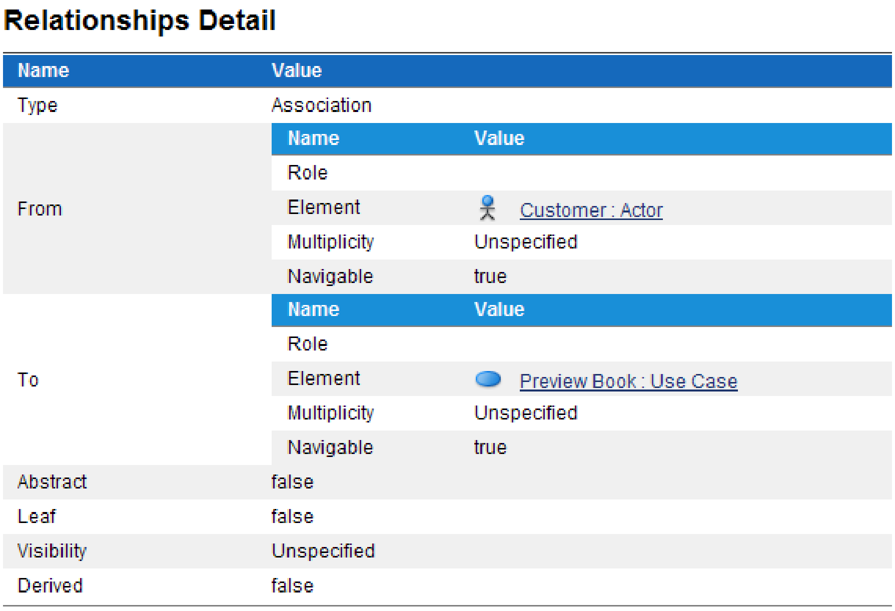


Relationships Summary

Name	Begin	End
— : Association	Customer : Actor	Preview Book : Use Case

__Figure 4-19__ Relationships summary

Relationships Detail



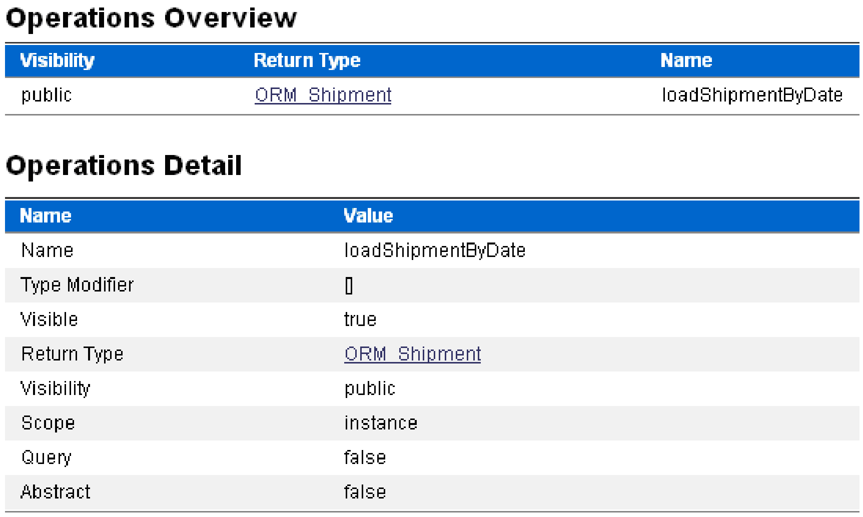
Relationships Detail

Name	Value	
Type	Association	
From	Name	Value
	Role	
	Element	Customer : Actor
	Multiplicity	Unspecified
	Navigable	true
To	Name	Value
	Role	
	Element	Preview Book : Use Case
	Multiplicity	Unspecified
	Navigable	true
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

__Figure 4-20__ Relationships detail

Other Model Details

Certain types of model have their own properties, for example, attributes and operations of class, or columns of ERD table. They are also included in the content page as custom sections. For instance, the Operations Overview and the Operations Detail sections show the overview and details of the operations of a class respectively.



Operations Overview

Visibility	Return Type	Name
public	ORM_Shipment	loadShipmentByDate

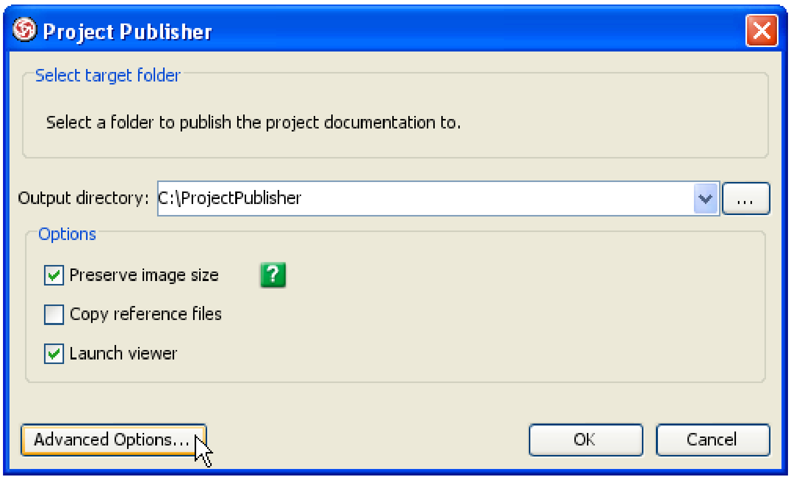
Operations Detail

Name	Value
Name	loadShipmentByDate
Type Modifier	[]
Visible	true
Return Type	ORM_Shipment
Visibility	public
Scope	instance
Query	false
Abstract	false

Figure 4-21 *Other model detail*

Setting Advanced Options

There are Advanced Options available in Project Publisher. To configure Advanced Options, open Project Publisher dialog and press Advanced Options... button.



Project Publisher

Select target folder

Select a folder to publish the project documentation to.

Output directory: C:\ProjectPublisher

Options

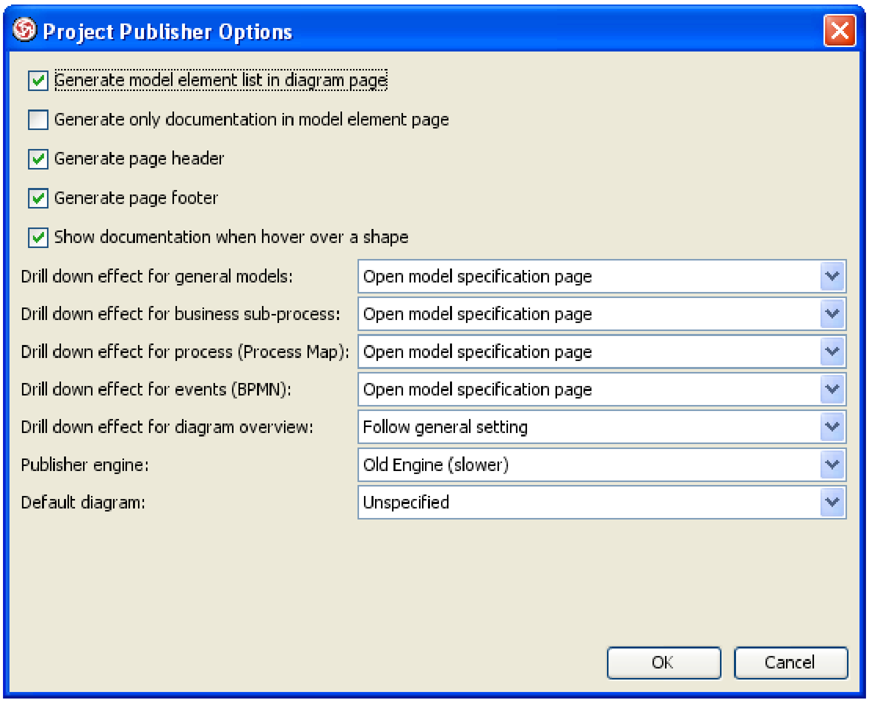
☑ Preserve image size

□ Copy reference files

☑ Launch viewer

Figure 4-22 *Project publisher dialog*

Project Publisher Options dialog will be shown.



Project Publisher Options

☑ Generate model element list in diagram page

□ Generate only documentation in model element page

☑ Generate page header

☑ Generate page footer

☑ Show documentation when hover over a shape

Drill down effect for general models: Open model specification page

Drill down effect for business sub-process: Open model specification page

Drill down effect for process (Process Map): Open model specification page

Drill down effect for events (BPMN): Open model specification page

Drill down effect for diagram overview: Follow general setting

Publisher engine: Old Engine (slower)

Default diagram: Unspecified

Figure 4-23 *Project publisher options dialog*

Details of each field in Project Publisher Options dialog are shown as below:

Field	Description
Generate model element list in diagram page	If this option is selected, model element list will be generated in diagram page.
Generate only documentation in model element page	If this option is selected, model element page will only include documentation.
Generate page header	If this options is selected, page header will be generated.
Generate page footer	If this options is selected, page footer will be generated.
Show documentation when hover over a shape	If this options is selected, hover over a shape in published project will show documentation.
Drill down effect for general models	Open model specification page - Open model specification page when drilling down general models. Open subdiagrams - Open subdiagrams when drilling down general models. Open references - Open references when drilling down general models.
Drill down effect for business sub-process	Follow general setting - Follow setting of drill down general models. Open model specification page - Open model specification page when drilling down business sub-process. Open process diagram page - Open process diagram page when drilling down business sub-process. Open subdiagrams - Open subdiagrams when drilling down business sub-process. Open references - Open references when drilling down business sub-process.
Drill down effect for process (Process Map)	Follow general setting - Follow setting of drill down general models. Open model specification page - Open model specification page when drilling down process (Process Map). Open subdiagrams - Open subdiagrams when drilling down process (Process Map). Open references - Open references when drilling down process (Process Map).
Drill down effect for events (BPMN)	Follow general setting - Follow setting of drill down general models. Open model specification page - Open model specification page when drilling down events (BPMN). Open references - Open references when drilling down events (BPMN).
Drill down effect for diagram overview	Follow general setting - Follow setting of drill down general models. Open model specification page - Open model specification page when drilling down diagram overview. Open overview diagram page - Open overview diagram page when drilling down business sub-process. Open subdiagrams - Open subdiagrams when drilling down diagram overview. Open references - Open references when drilling down diagram overview.
Publisher engine	Old Engine (slower) - Using old engine to publish project which will be slower. New Engine (faster) - Using new engine to publish project which will be faster.
Default diagram	Default diagram of published project.

Table 4-1 *Details of project publisher options dialog*

Installing Report Writer

The Report Writer is a sophisticated tool for report creation. Users can output the existing project as reports by documenting their project within VP-UML. VP-UML offers seamless integration of UML modeling tool with word processors to provide a unified documenting environment. By dragging the models from VP-UML to Report Writer, data is extracted from models and content is created in Report Writer.

Retaining the conformance between documentation and design is a tedious task. Report Writer maintains the consistency between them. If you create a new model, the content will be appended to the existing one. If you remove a model, the generated element will be removed. If you re-edit the models, the content will be refreshed.

Users can also apply their own style for the generated element, to the Report Writer more flexible.

Launching Report Writer

To launch Report Writer, perform one of the following actions:

Select Tools > Report > Report Writer...from main menu.

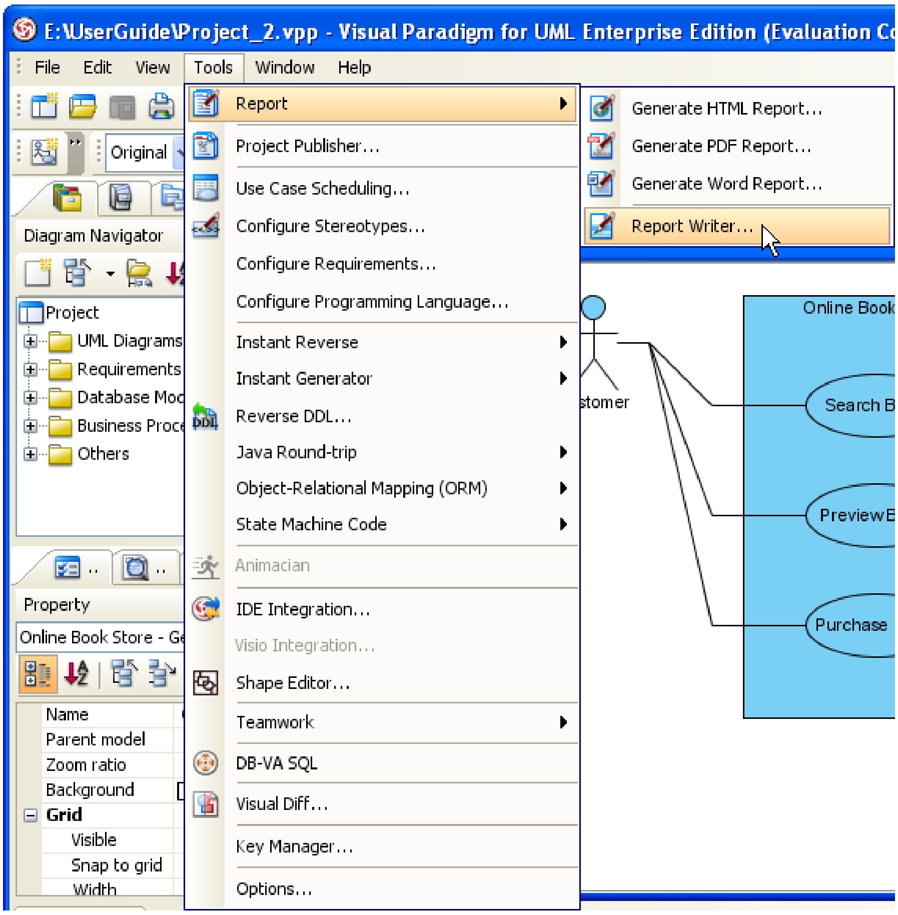


Figure 5-1 *Launch report writer from main menu*

Click on the Report Writer button on the toolbar.

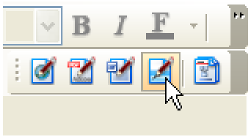
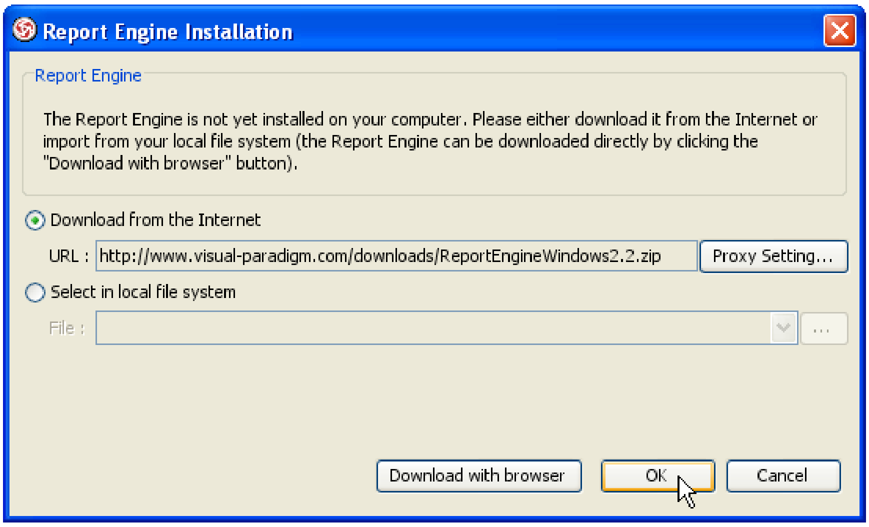


Figure 5-2 *Launch report writer from toolbar*

Installing Report Engine

If it is the first time you have started the Report Writer, the Report Engine Installation dialog box will be displayed asking for the installation of Report Engine.



Report Engine Installation

Report Engine

The Report Engine is not yet installed on your computer. Please either download it from the Internet or import from your local file system (the Report Engine can be downloaded directly by clicking the "Download with browser" button).

● Download from the Internet

URL : http://www.visual-paradigm.com/downloads/ReportEngineWindows2.2.zip

○ Select in local file system

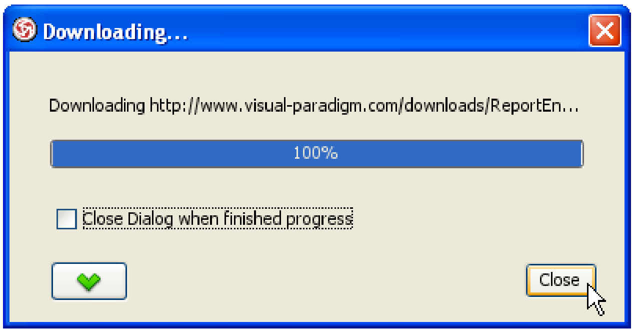
File :

Figure 5-3 *Report engine installation dialog*

To install Report Engine, perform one of the following actions:

Choose from the Report Engine Installation dialog box the option Download from Internet and click OK. This downloads the Report Engine from the Internet and automatically proceeds with Report Engine installation once the download has been completed.

Choose from the Report Engine Installation dialog box the option 'Select in local file system', locate the report engine and then click OK to start the Report Engine installation. The Report Engine can be obtained by clicking 'Download with browser'. You can enter the path directly into the text field or click ... to locate the file from the file chooser.



Downloading...

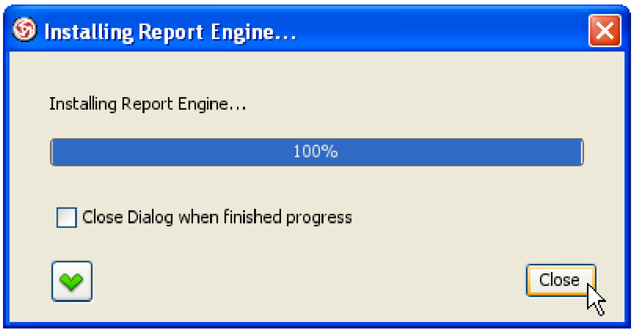
Downloading http://www.visual-paradigm.com/downloads/ReportEn...

100%

□ Close Dialog when finished progress

Figure 5-4 *Downloading report engine*

After downloading completed, installation will be performed.



Installing Report Engine...

Installing Report Engine...

100%

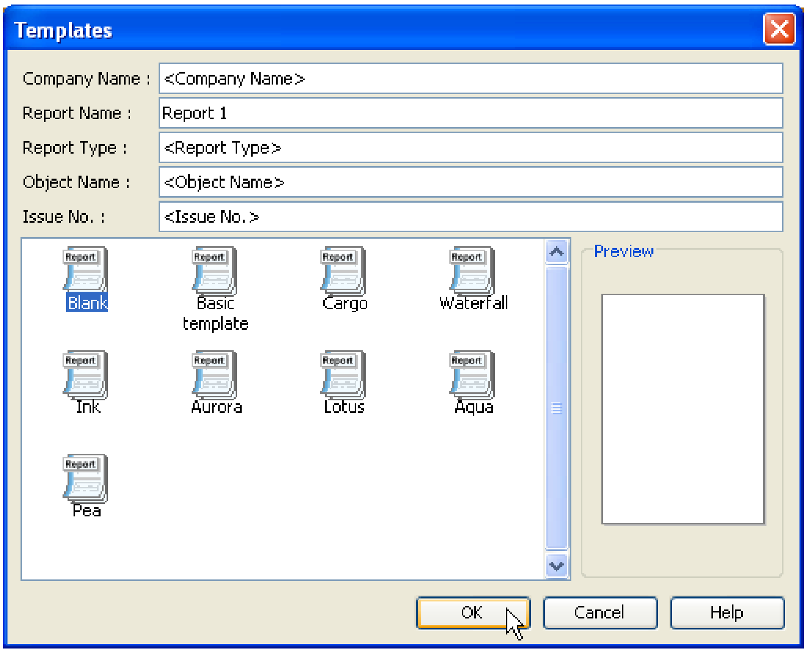
□ Close Dialog when finished progress

Figure 5-5 *Installing report engine*

Report writer will be available when installation is done.

Creating Report

Upon launching Report Writer for the first time, the Template dialog box will be displayed and ask for the information of the new report. Enter the report information and select a desired report theme for the report, preview of the selected theme is shown on the preview pane. Click OK to start Report Writer when everything is ready.



Templates

Company Name : <Company Name>

Report Name : Report 1

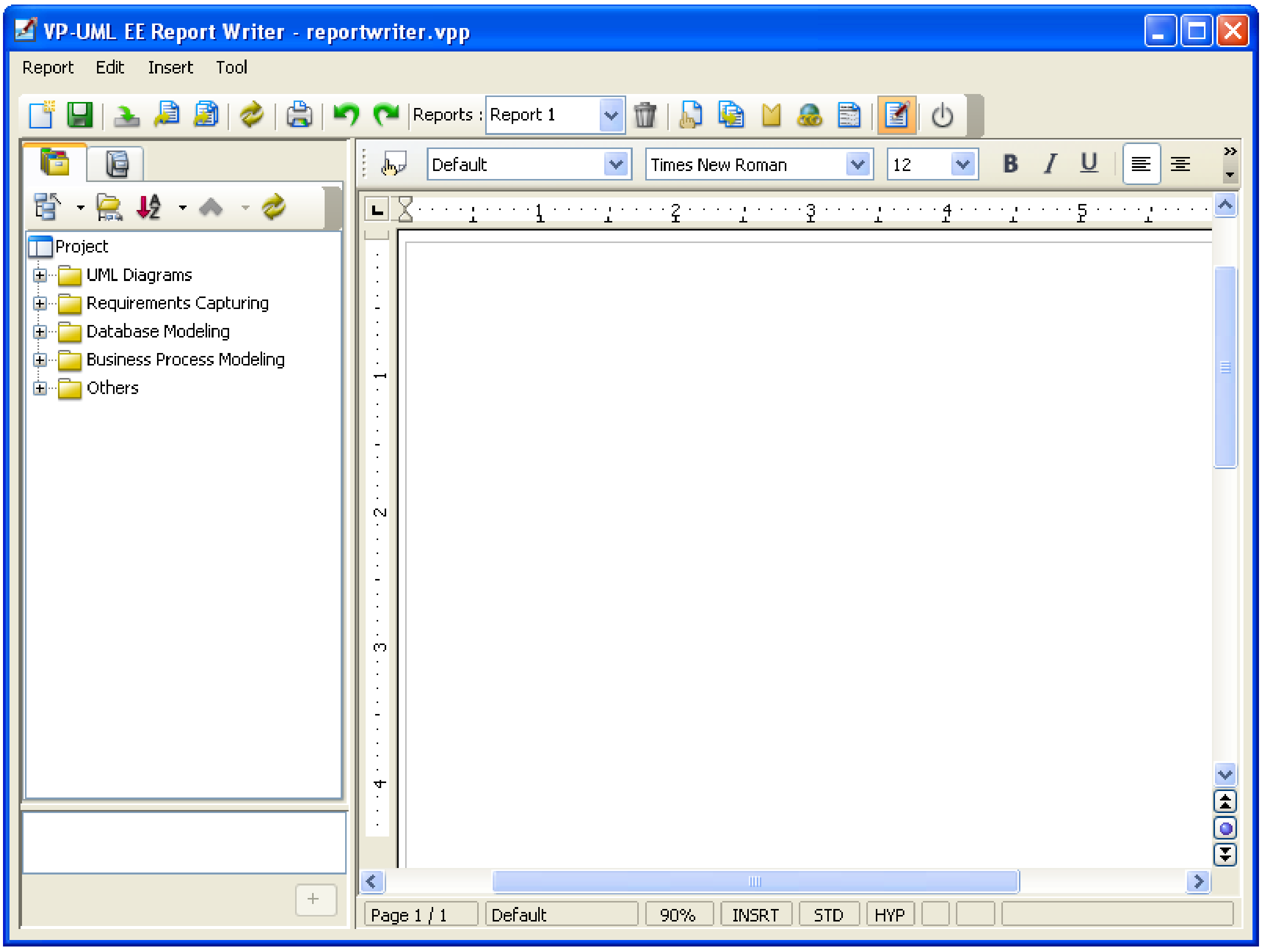
Report Type : <Report Type>

Object Name : <Object Name>

Issue No. : <Issue No.>

Figure 5-6 *Templates dialog*

Report 1 is created as shown.



VP-UML EE Report Writer - reportwriter.vpp

Reports : Report 1

Figure 5-7 *Empty report in report writer*

Introduction to Report Writer User Interface

When Report Writer is launched you are taken to the Report Writer environment where you can create and edit your reports. Three distinct panes are presented on the screen: the Project Explorer, Template Pane and the Writer Pane.

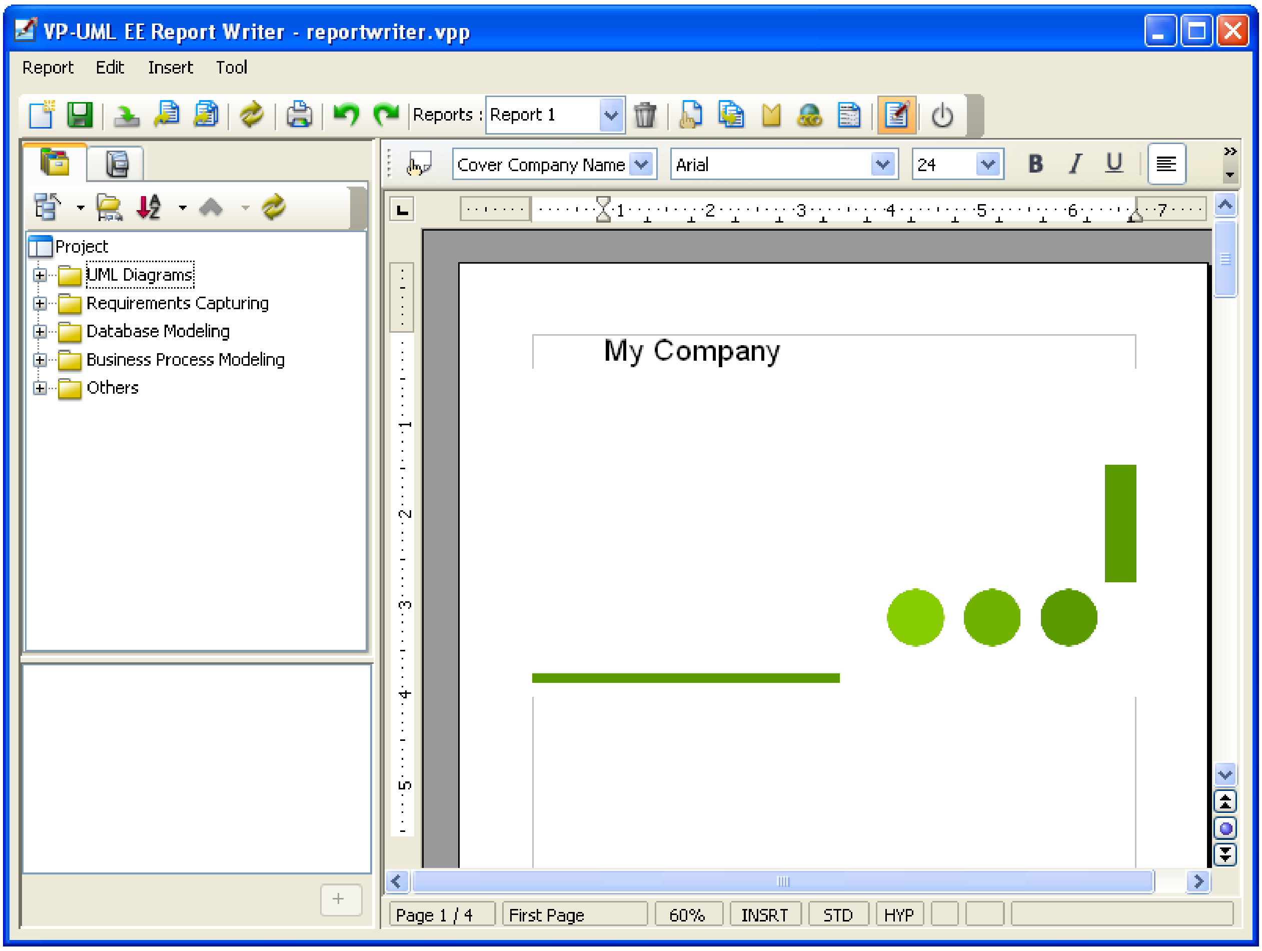


Figure 5-8 *Report writer*

Diagram Navigator

The Diagram Navigator displays all diagrams within the project in a form of a project tree and organizes them by their diagram type. Through the use of a folding tree structure you can browse the names of these diagrams by either expanding or collapsing the folders and perform sorting by diagram type and name.

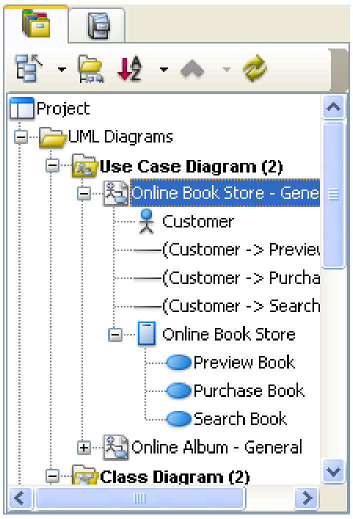


Figure 5-9 *Diagram navigator*

Details of each button in Diagram Navigator are show as below:

Button	Description
	To collapse all the nodes within the project tree.
	To expand all the nodes within the project tree.
	To show only diagrams but not models in the tree.
	To sort diagrams within the project tree by alphabetical order of their names.
	To sort diagrams within the project tree by their diagram type.
	To sort diagrams within the project tree by their diagram type.

Table 5-1 *Details of diagram navigator*

Model Tree

The Model Pane displays models within the project in a form of a project tree. Notice that not all the model elements are displayed, and only the elements that are available for generating report content are shown.

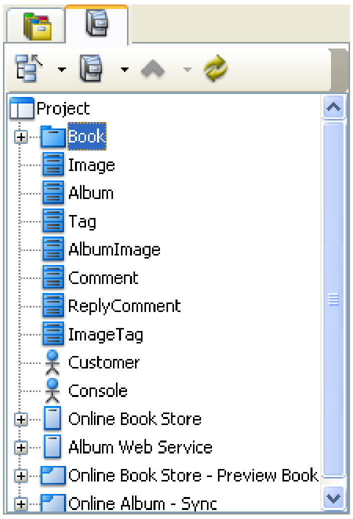


Figure 5-10 *Model tree*

Details of each button in Model Tree are shown as below:

Button	Description
	To collapse all the nodes within the project tree.
	To expand all the nodes within the project tree.
	To display the models within the project without sorting. Ordering of models will be based on their order of creation.
	To sort models within the project tree by alphabetical order of their names.
	To sort models within the project tree by their model type.
	To refresh the project tree within the Model Pane.

Table 5-2 *Details of model tree*

Template Pane

The Template Pane displays all the templates available for the model or diagram selected in Property Pane.

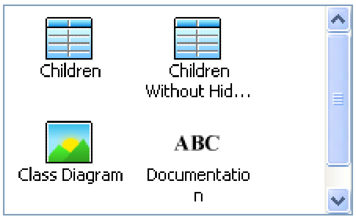


Figure 5-11 *Template pane*

Each template represents the corresponding report content of a particular model or diagram. By dragging a template into the Writer Pane, the report content will be printed on the report. There are three types of template: Text, Image and Table. Each of them has its own appearance in the report content.

Button	Description
ABC	The generated element block is mainly composed of text. It is mainly used in the documentation template of elements.
🖼	The generated element block is mainly composed of images. It is used in the diagram template for UML Diagrams.
▦	The generated element block is mainly composed of tables. Most of the content-related templates use this type of template.

Table 5-3 *Details of template pane*

Writer Pane

Writer Pane embeds a word processor to provide a report editing environment.

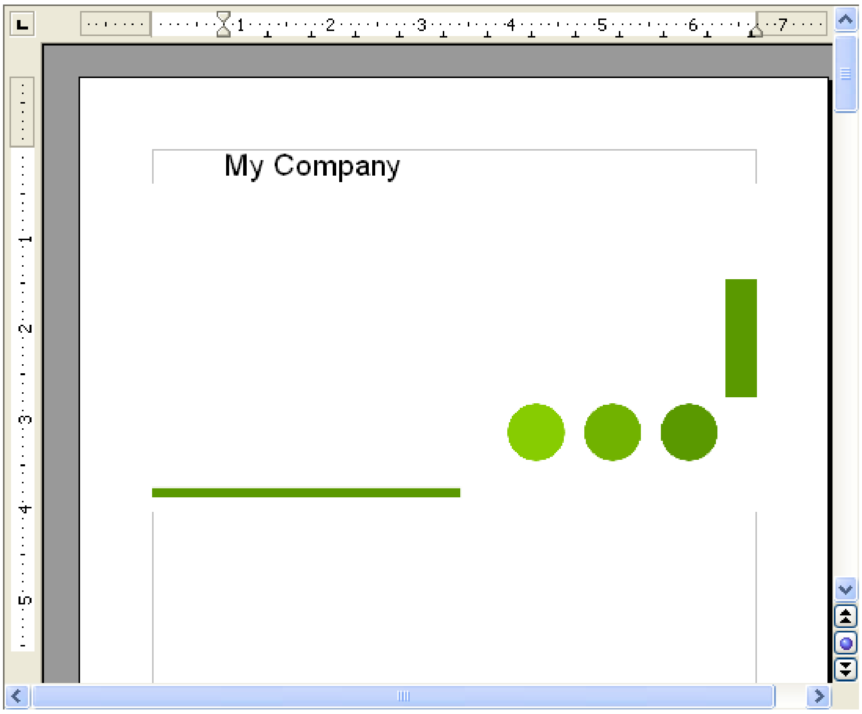


Figure 5-12 *Writer pane*

Toolbar

Toolbar is the horizontal bars placed below the menu bar. They store all the frequently used commands that appear as a row of buttons.

Button	Description
📄	To create a new report.
💾	To save modified reports.
📥	To import an external document (either an .sxw or a .doc file) as a report.
📤	To export the current report as an .sxw or .doc file.
📤	To export all the reports within the current project.
🔄	To update the content within the current report from the VP-UML models.
🖨	To print the current report by supplying the printer name.
↶	To undo the last action you performed.
↷	To redo the last action you performed.
Reports : Report 1 ⌄	To select a report from the current project for editing.
🗑	To remove the existing report(s).
📄	To display the stylist dialog box for modifying the style.
📋	To copy the style settings defined in another report.
🔖	To display bookmarks that outlines the boundary for each generated element.
🔗	To insert a hyperlink.
📑	To insert index or table
📝	To include documentation of model when generating content.
⏻	To close Report Writer and go back to VP-UML.

Table 5-4 *Details of toolbar*

Constructing Report

Creating a Generated Element

The term "Generated element" here means a block of report content generated by Report Writer and consists of details of a particular diagram or model element.

To create a generated element block:

1. Click to select the desired model element from either the Diagram Navigator or Mode Tree for content generation.

2. The supported templates for the selected model element are shown on the Template Pane. Each template represents a way in presenting the selected model element on the report. For example, "Children" template of a System represents a list of children placed inside a particular System.

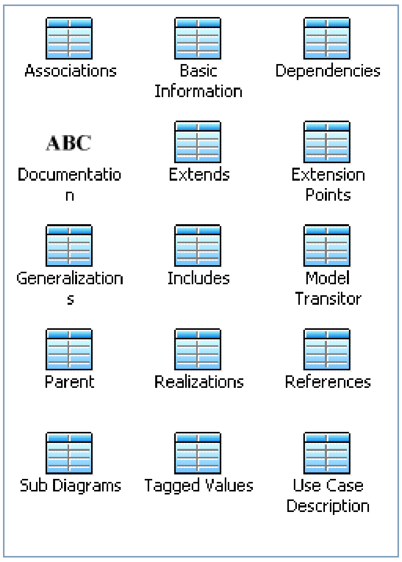


Figure 5-13 *Template pane*

3. Drag the desired template from the Template Pane and drop it onto the report.

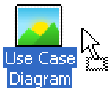
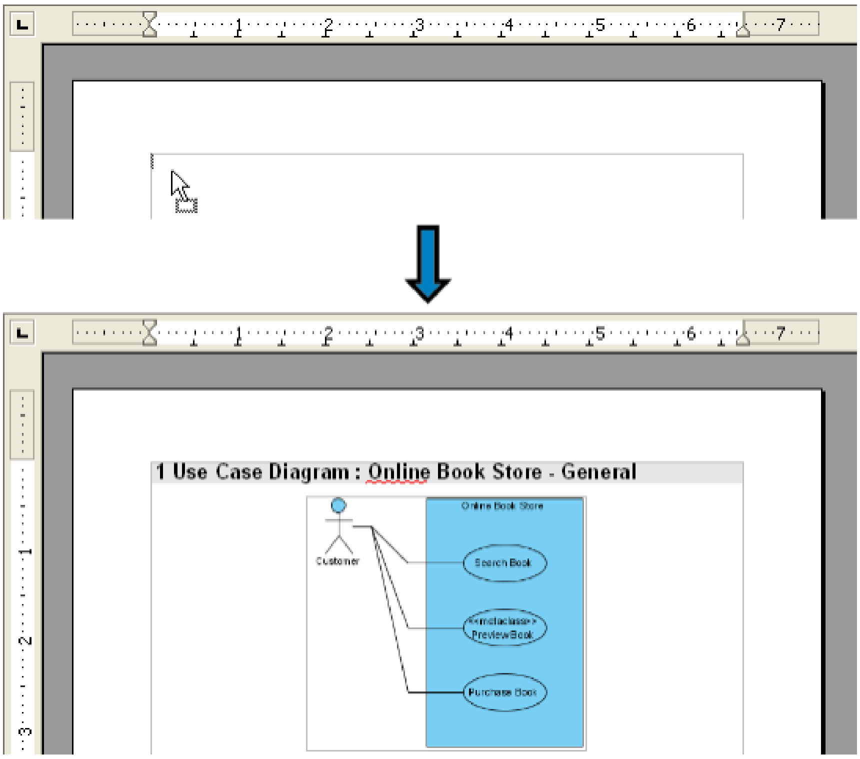


Figure 5-14 *Drag template from template pane*

4. When the cursor drags over the Writer Pane, a tiny straight line will appear in the report indicating the position of the expected position of the generated element. Once you've dropped the template onto the report, corresponding content will be generated element to the dropped position.



1 Use Case Diagram : Online Book Store - General

Figure 5-15 Drop template to report

5. Same as diagram, model element's template also able to drag to report.

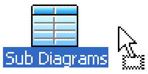
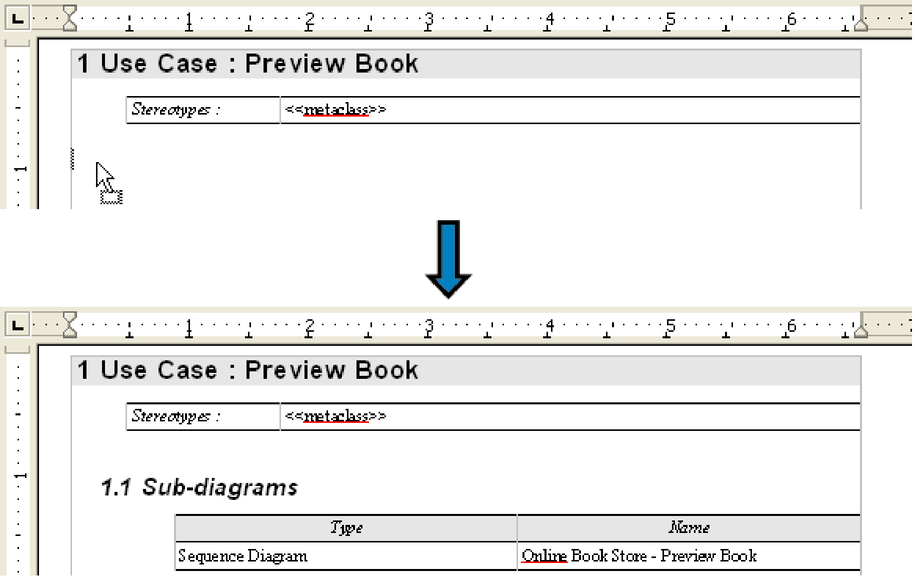


Figure 5-16 Drag template from template pane

6. When the cursor drags over the Writer Pane, a tiny straight line will appear in the report indicating the position of the expected position of the generated element. Once you've dropped the template onto the report, corresponding content will be generated element to the dropped position.



1 Use Case : Preview Book

Stereotypes :	<<metaclass>>

1.1 Sub-diagrams

Type	Name
Sequence Diagram	Online Book Store - Preview Book

Figure 5-17 Drop template to report

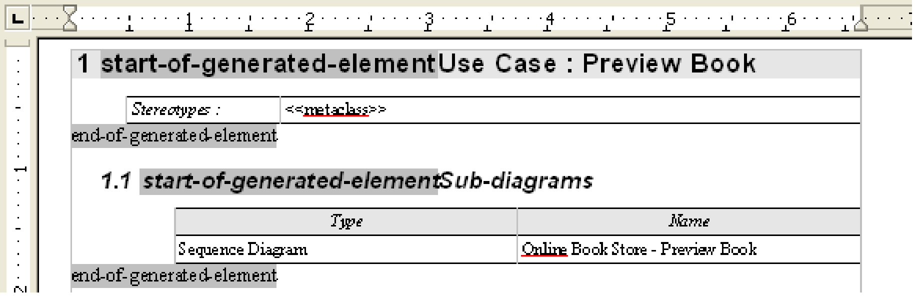
Showing the Bookmarks

Creating a new generated element within the boundary of an existing one is dangerous because the content may be messed up during a report update process. To avoid this, you can display bookmarks to indicate the start and end position of each generated element, and to prevent dropping a new one within the scope of the existing generated element.

To show/hide bookmarks:

Check/Uncheck Tool > Show Indicator from main menu to show/hide bookmarks.

Select/Deselect Click to Show Bookmarks/Click to Hide Bookmark on the toolbar to show/hide bookmarks.



1 start-of-generated-elementUse Case : Preview Book

Stereotypes :	<<metaclass>>

end-of-generated-element

1.1 start-of-generated-elementSub-diagrams

Type	*Name*
Sequence Diagram	Online Book Store - Preview Book

end-of-generated-element

Figure 5-18 *Showing bookmarks*

Applying Style to Report

A style in Report Writer is a collection of formatting attributes that describe the nature of paragraphs. The generated element highly adopts the predefined styles in Report Writer therefore users can customize the related styles to bring consistency to the whole document. There are two ways for applying style to report.

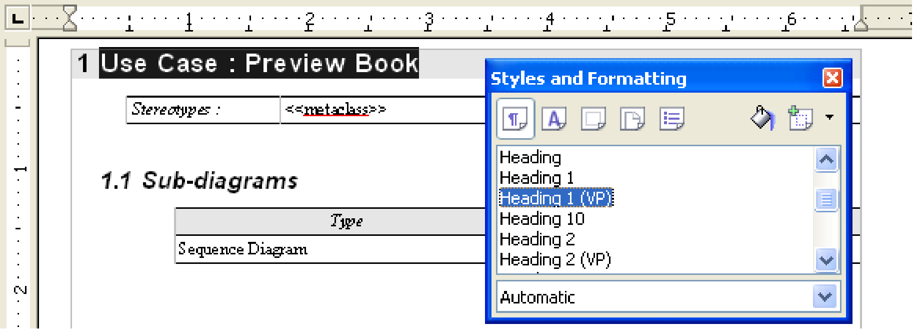
Style configuration

The Stylist dialog box allows you to configure the pre-defined styles. To display the Stylist dialog box:

Select Tool > Show Stylist from main menu.

Click the Click to Show Stylist button on the toolbar.

In both cases, the Stylist dialog box display.



1 Use Case : Preview Book

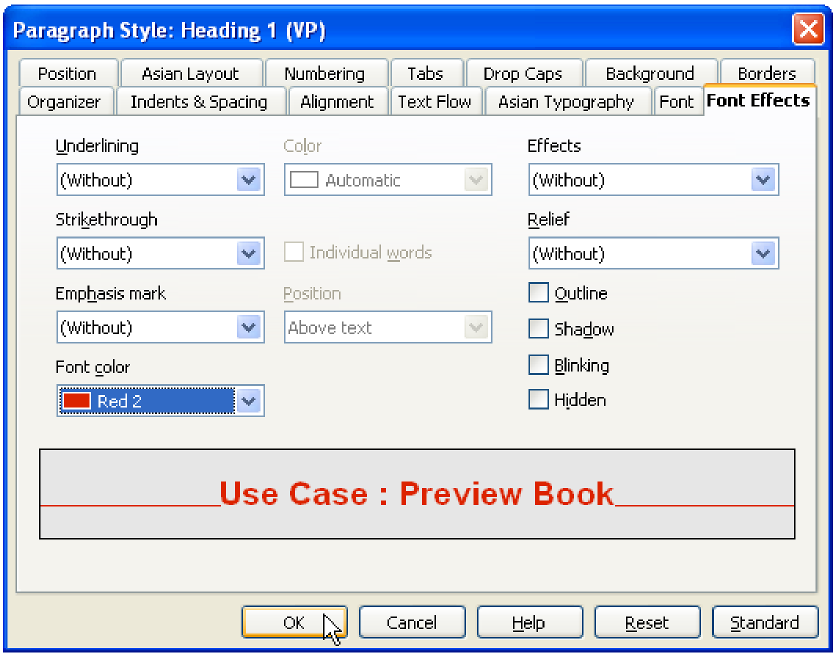
Stereotypes :	<<metaclass>>

1.1 Sub-diagrams

Type
Sequence Diagram

Figure 5-19 *Stylist*

To edit the style, right-click on the highlighted style and choose Modify...from the popup menu. This displays the dialog box for the selected style. You can now adjust it with your own preference. When everything is ready, please click OK to commit the settings and exit the dialog.

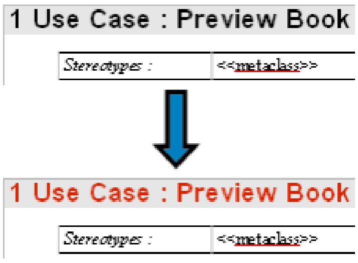


Underlining: (Without)
Color: Automatic
Effects: (Without)
Strikethrough: (Without)
□ Individual words
Relief: (Without)
Emphasis mark: (Without)
Position: Above text
□ Outline
□ Shadow
□ Blinking
□ Hidden
Font color: 🟥 Red 2

Use Case : Preview Book

Figure 5-20 *Edit style*

The changes will take effect immediately and you will notice the style is applied to those generated elements using the same style.



1 Use Case : Preview Book

Stereotypes :	<<metaclass>>

1 Use Case : Preview Book

Stereotypes :	<<metaclass>>

Figure 5-21 *Style change*

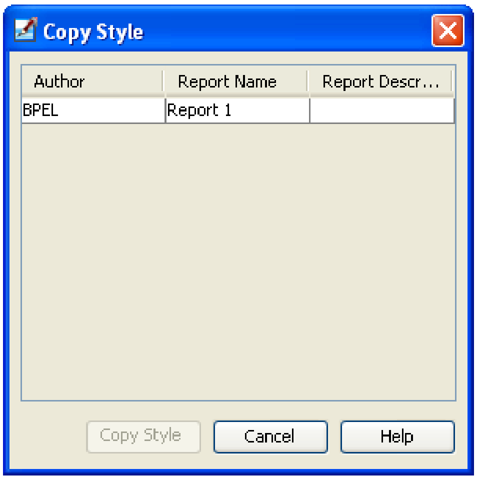
Loading Style from Other Report

The Copy Style dialog box allows you to copy the style from existing report. To display the Copy Style dialog box:

Select Tool > Copy Style from main menu.

Click the Copy Style to Current Report button on the toolbar.

In both cases, the Copy Style dialog box display.



Author	Report Name	Report Descr...
BPEL	Report 1	

Figure 5-22 *Copy style dialog*

Select a desired report for getting the style configuration and click Copy Style. The style configuration in the current report is replaced by the style configuration of the selected report. All the predefined styles will be overwritten.

Updating Table of Contents

There is a predefined Table of Contents in each of the report template. Here is the pre-built Table of Contents structure:

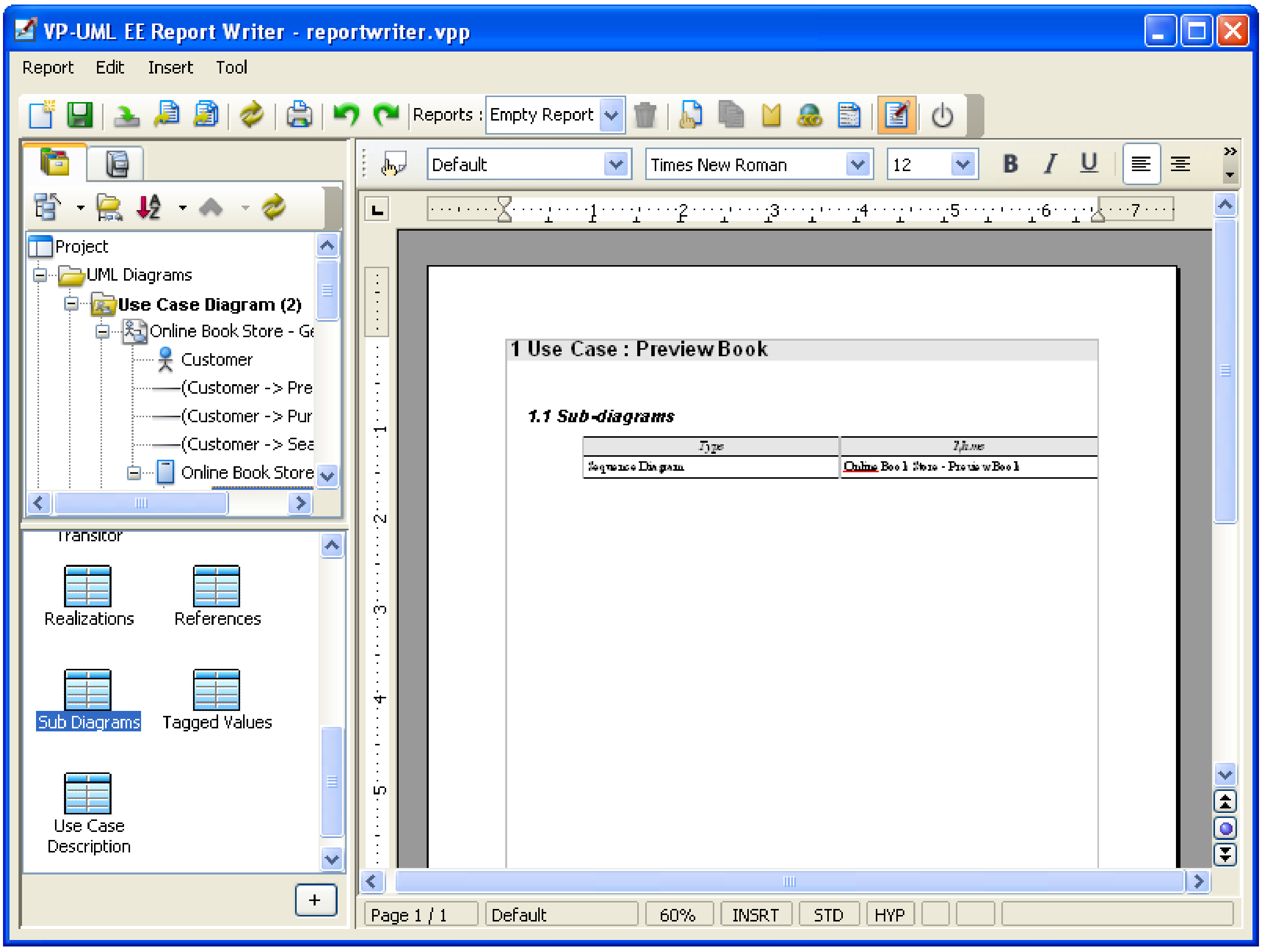
Level	Paragraph Style
0	Heading 1 (VP)
1	Heading 2 (VP)
2	Heading 3 (VP)
3-9	None

Table 5-5 *Details of predefined table of contents*

To update the Table of Contents, right-click on the caption Table of Contents and select Update Index/Table from popup menu.

Updating Report Content

In reality, software design keeps evolving from time to time. Originally, users needed to modify the related documents manually to ensure that it is fully conformed to the latest design. Report Writer binds closely with the VP-UML project, and hence generated elements can then be updated without affecting the user-defined content.



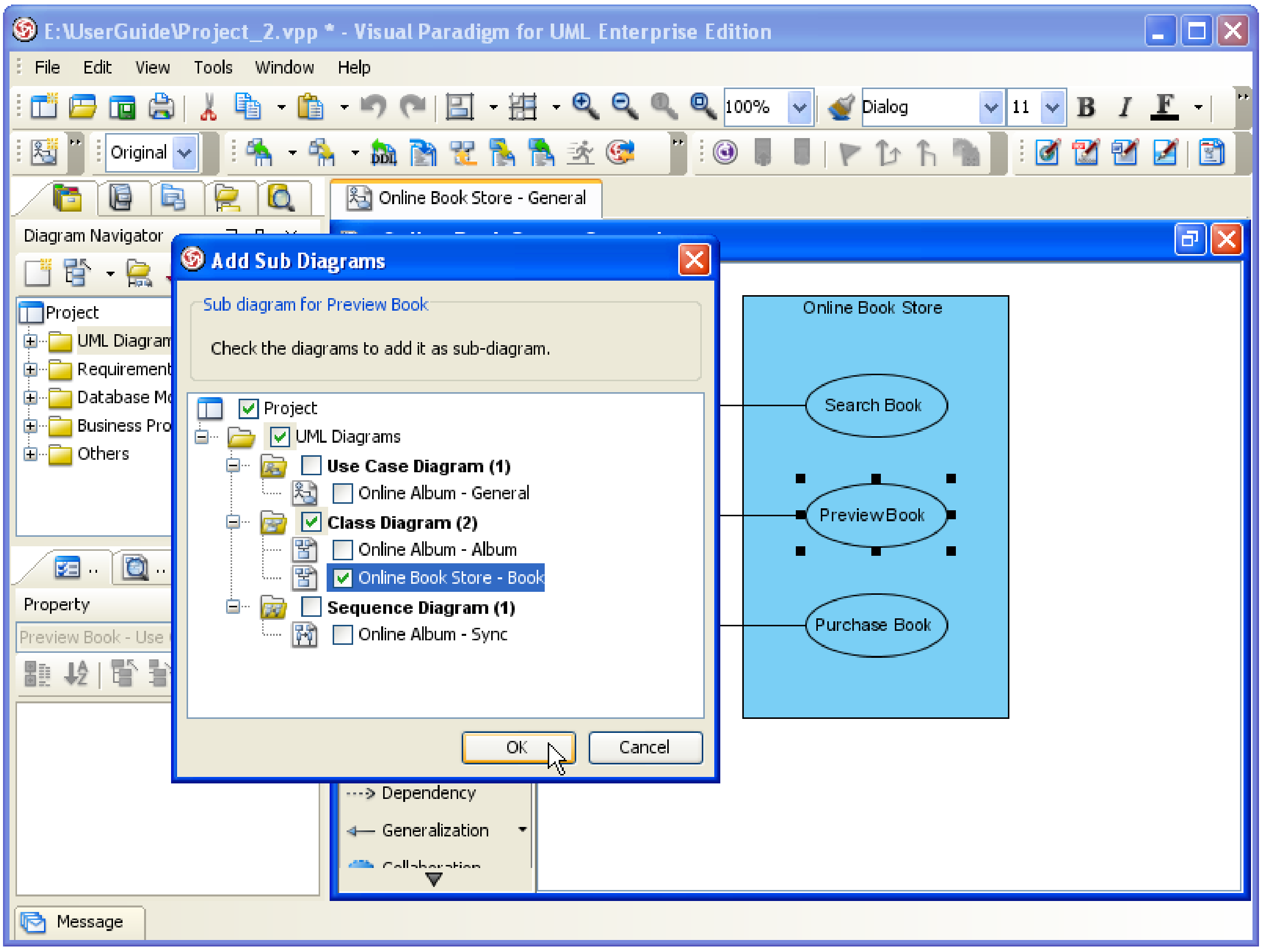
1 Use Case : Preview Book

1.1 Sub-diagrams

Type	Name
Sequence Diagram	Online Book Store - Preview Book

Figure 5-23 Report writer

Edit project's content which used by report.



Add Sub Diagrams

Sub diagram for Preview Book

Check the diagrams to add it as sub-diagram.

- ☑ Project
  - ☑ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Online Album - General
    - ☑ **Class Diagram (2)**
      - □ Online Album - Album
      - ☑ Online Book Store - Book
    - □ **Sequence Diagram (1)**
      - □ Online Album - Sync

Figure 5-24 Edit project

To update a report, choose the desired report for updating from the drop-down menu and click the Update from Model button from the toolbar or select Report > Update from Model from main menu.

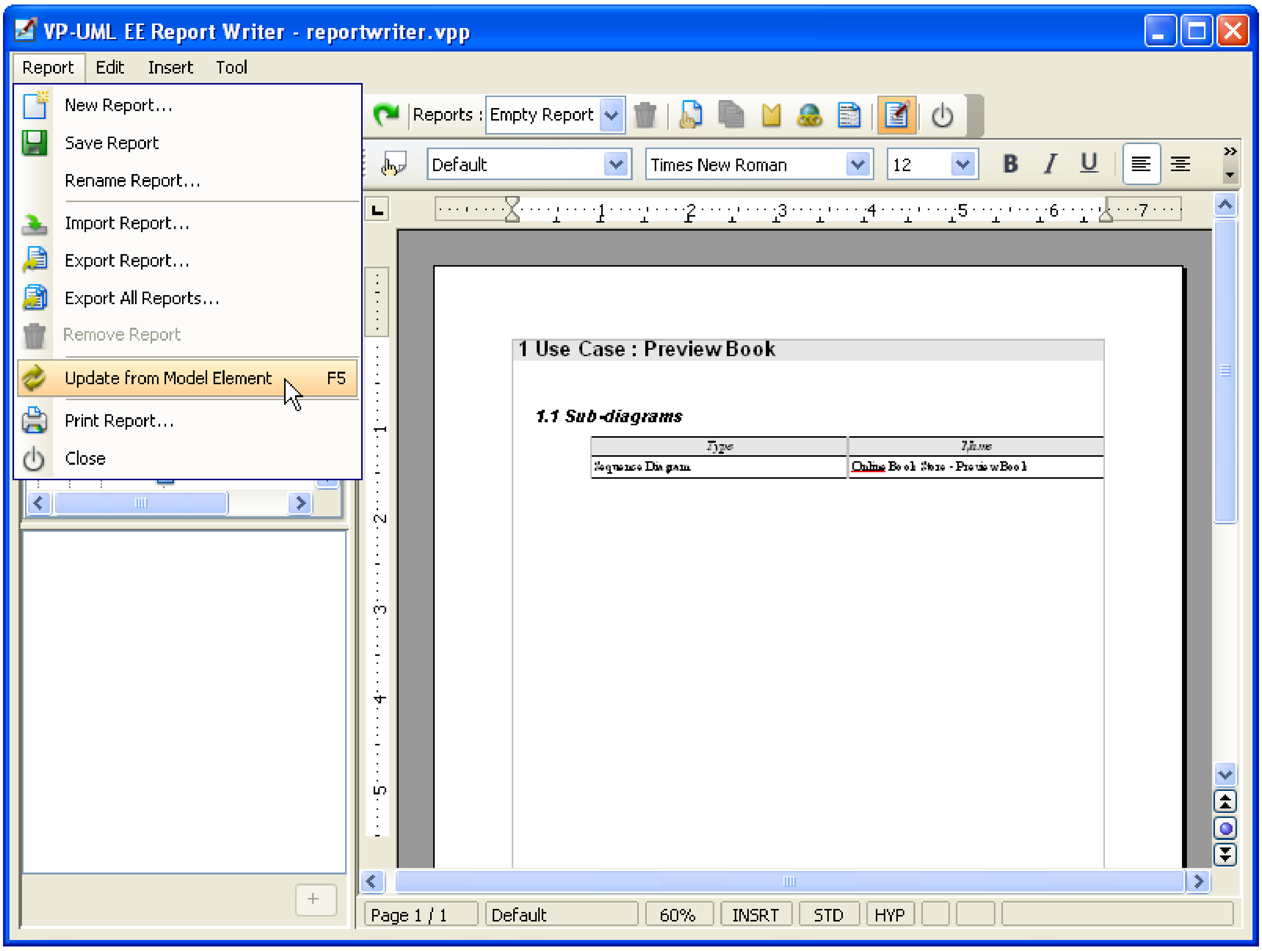
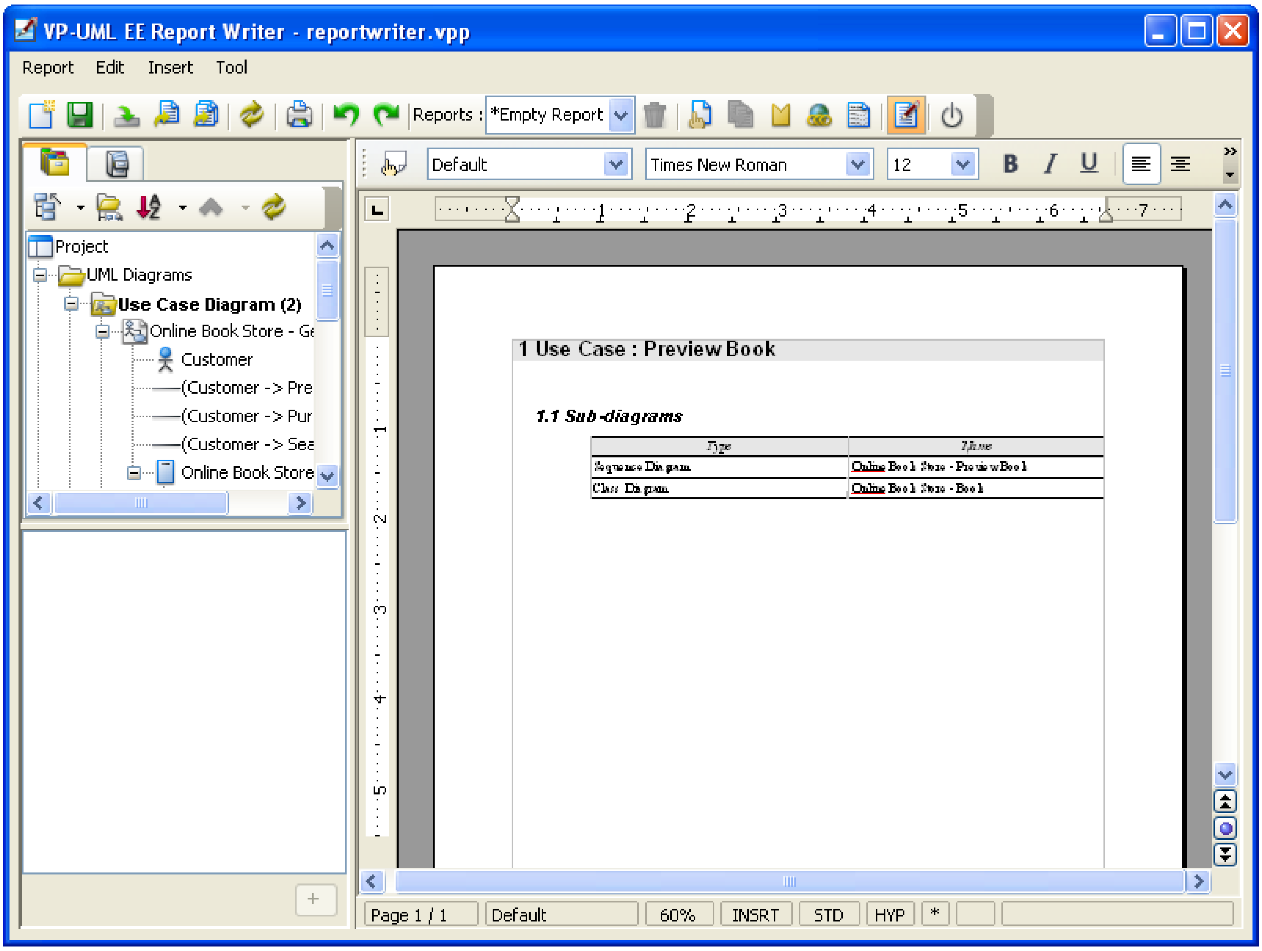


Figure 5-25 Update from model element

Update process will start automatically. Data will be extracted from project and replace with the content within the existing generated element.



1 Use Case : Preview Book

1.1 Sub-diagrams

Type	Name
Sequence Diagram	Online Book Store - Preview Book
Class Diagram	Online Book Store - Book

Figure 5-26 Updated report

NOTE: Please do not click on the Writer Pane while the update process is undergoing, as it may affect the accuracy of the content. It can also damage the generated element, so that updating cannot be performed anymore unless the damaged block is removed manually.

NOTE: The update process will replace ALL the contents within each generated element without notification. Therefore please insert the content carefully and ensure that it is not located inside the scope of any generated elements.

Export and Import Report

You can export report as file and edit it outside VP-UML. Supported format includes Microsoft Word 97/2000/XP document (with extension .doc) and OpenOffice.org 1.0 Text Document (with extension .sxw).

Exporting current report

1. Click on the Export Report... button on the toolbar or select Report > Export Report...from main menu. This display the Save dialog box.

2. In the Save dialog box, enter the file name and select OpenOffice.org 1.0 Text Document (.sxw) format or Microsoft Word 97/2000/XP (.doc) format for exporting.

3. When everything is ready, click Save to export the report.

Exporting all report(s)

1. Click the Export All Reports... button on the toolbar or select Report > Export All Reports... from main menu. This display the Save dialog box.

2. In the Save dialog box, enter the directory for storing the reports in the File name field and select either OpenOffice.org 1.0 Documents or Microsoft Word Documents for the Document Type.

3. When everything is ready, click Save to export the report(s).

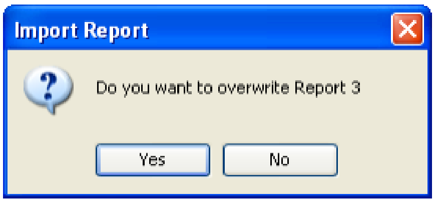
Importing a Report

You can import a document back into Report Writer for data updating. To import a report:

1. Click the Import Report... button on the toolbar or select Report > Import Report... from main menu. This displays the Open dialog box.

2. In the Open dialog box, select either OpenOffice.org 1.0 Text Document (.sxw) format or Microsoft Word 97/2000XP (.doc) format for importing. Select a file and click Open to import the selected document into Report Writer.

3. If the document has previously been exported from Report Writer, a dialog will appear and ask for overwriting the existing one or not.


Import Report

Do you want to overwrite Report 3

Figure 5-27 *Confirm overwrite existing report*

If you click Yes, the existing report will be replaced by the imported one. If you click No, the imported report will be stored into Report.

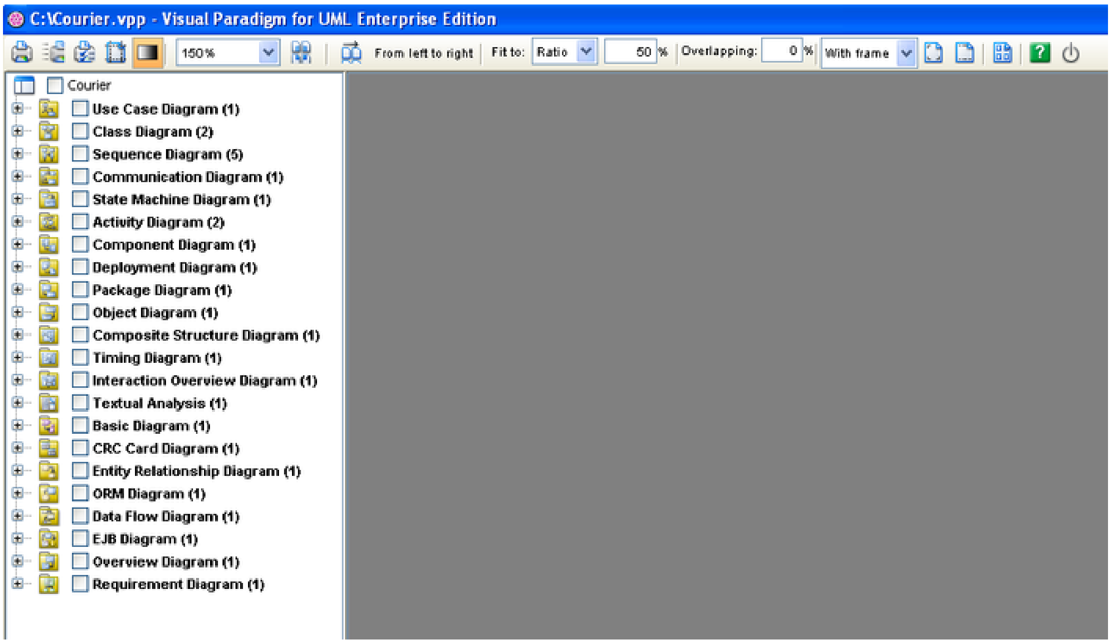
Printing Diagrams

Print Preview

The Print Preview dialog box allows you to preview the printout and provides a set of options for changing the printout style. To display the dialog box, perform one of the following actions:

Select File > Print...from main menu

Click on the Print button on the toolbar



C:\Courier.vpp - Visual Paradigm for UML Enterprise Edition

150 % | From left to right | Fit to: Ratio | 50 % | Overlapping: 0 % | With frame

- □ Courier
  - □ Use Case Diagram (1)
  - □ Class Diagram (2)
  - □ Sequence Diagram (5)
  - □ Communication Diagram (1)
  - □ State Machine Diagram (1)
  - □ Activity Diagram (2)
  - □ Component Diagram (1)
  - □ Deployment Diagram (1)
  - □ Package Diagram (1)
  - □ Object Diagram (1)
  - □ Composite Structure Diagram (1)
  - □ Timing Diagram (1)
  - □ Interaction Overview Diagram (1)
  - □ Textual Analysis (1)
  - □ Basic Diagram (1)
  - □ CRC Card Diagram (1)
  - □ Entity Relationship Diagram (1)
  - □ ORM Diagram (1)
  - □ Data Flow Diagram (1)
  - □ EJB Diagram (1)
  - □ Overview Diagram (1)
  - □ Requirement Diagram (1)

Figure 6-1 *Print preview dialog*

The toolbar of the print preview pane allows you to configure the print settings. The buttons and their descriptions are shown in the table below:

Button	Function
	Print the diagram(s). The Print dialog box will be opened.
	Print diagrams without previewing them. The Quick Print dialog will be opened.
	Set up the page properties such as paper size and orientation.
	Adjust the margins of the pages.
	Select to 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.
50%	Select the percentage to reduce/enlarge the print preview of diagrams.
/	If the Fit to Pages option is selected, and there are multiple pages in the printout, selecting Paper Base Layout will cause the distribution of pages to be paper-oriented (the diagram size is ignored in arranging the preview); while selecting 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 change the order of the printout. A large diagram is divided into many pages, selecting From left to right will arrange the printout order from the pages on the left to the pages on the right, while selecting 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 %	Set the diagram size to fit to the specified ratio.
Fit to: Pages 1 x 1	Set the diagram to be printed on the number of pages specified.
Overlapping: 0 %	Set the percentage of the margins to overlap among adjacent pages.
	Select/deselect to show/hide the clip marks on the printout.
	Edit the header and the footer of the printout.
	Switch to the Multiple Page Mode to set the multiple page options.
?	Calling the VP-UML help file
⏻	Close the print preview pane and return to the design area.

Table 6-1 *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 diagrams for printing. The selected diagrams will be shown at the preview area.

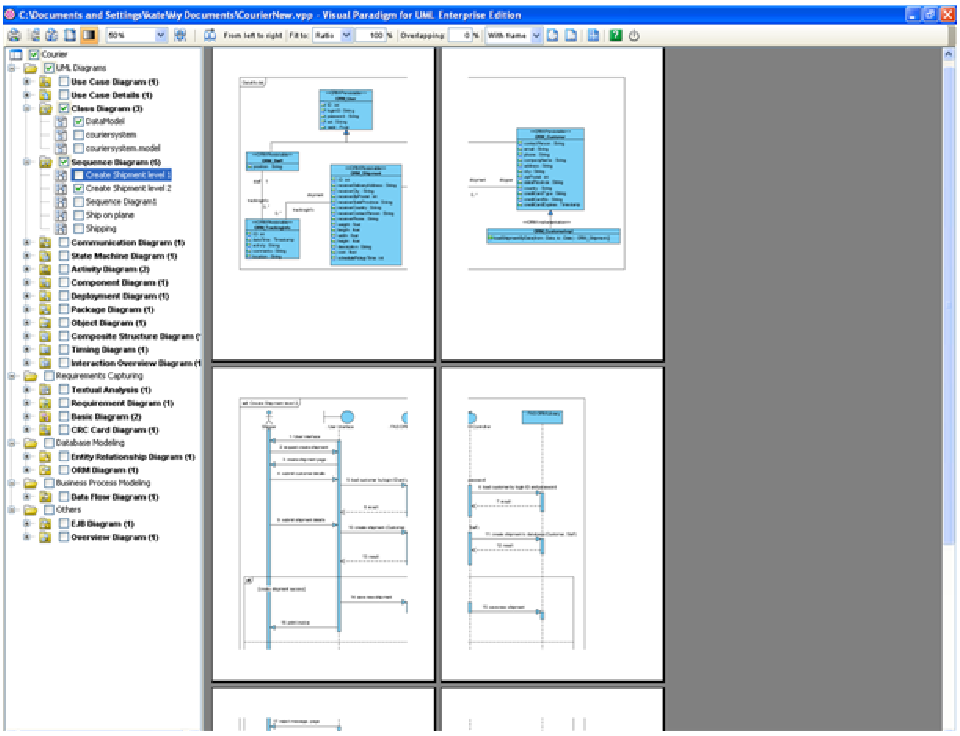
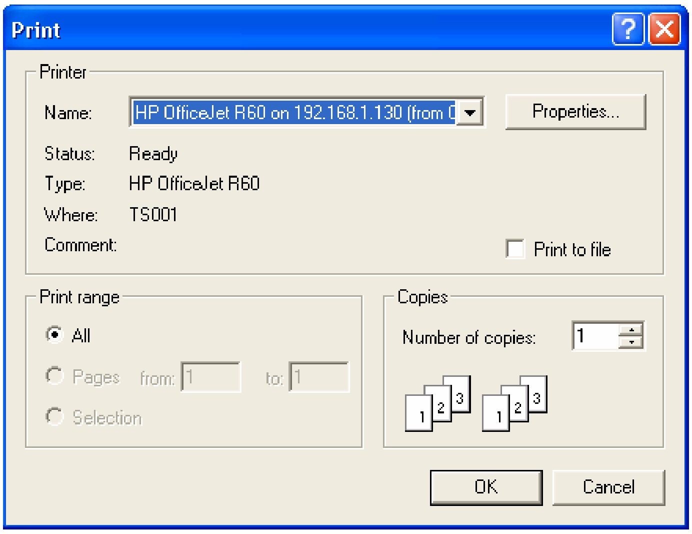


Figure 6-2 *Diagram preview*

2. Click on the Print button 🖨 on the Print Preview Toolbar. The Print dialog box appears.



Name: HP OfficeJet R60 on 192.168.1.130 (from C

Status: Ready

Type: HP OfficeJet R60

Where: TS001

Comment:

□ Print to file

● All

○ Pages from: 1 to: 1

○ Selection

Number of copies: 1

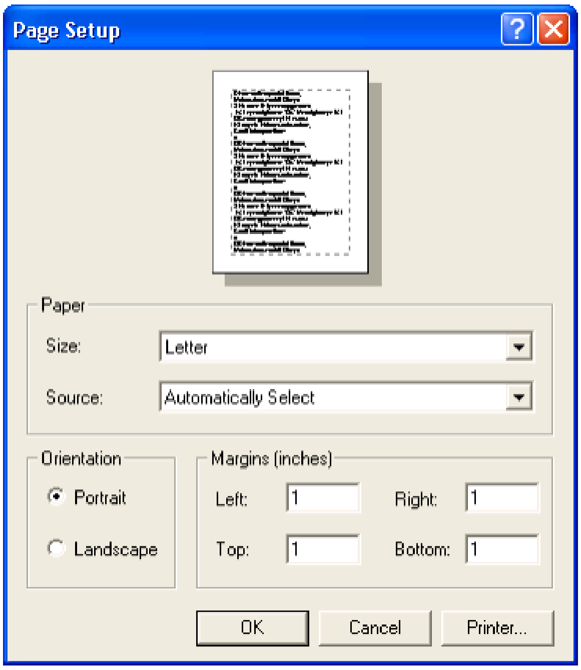
Figure 6-3 *Print the diagram*

3. Select the printer to use, the page range and the number of copies to be printed. You may click on the Properties... button to configure the printer-specific properties as well.

4. Click OK to start printing.

Page Setup

Page Setup allows the user to specify the page size, orientation as well as the margins of the pages.



Size: Letter

Source: Automatically Select

● Portrait

○ Landscape

Left: 1 Right: 1

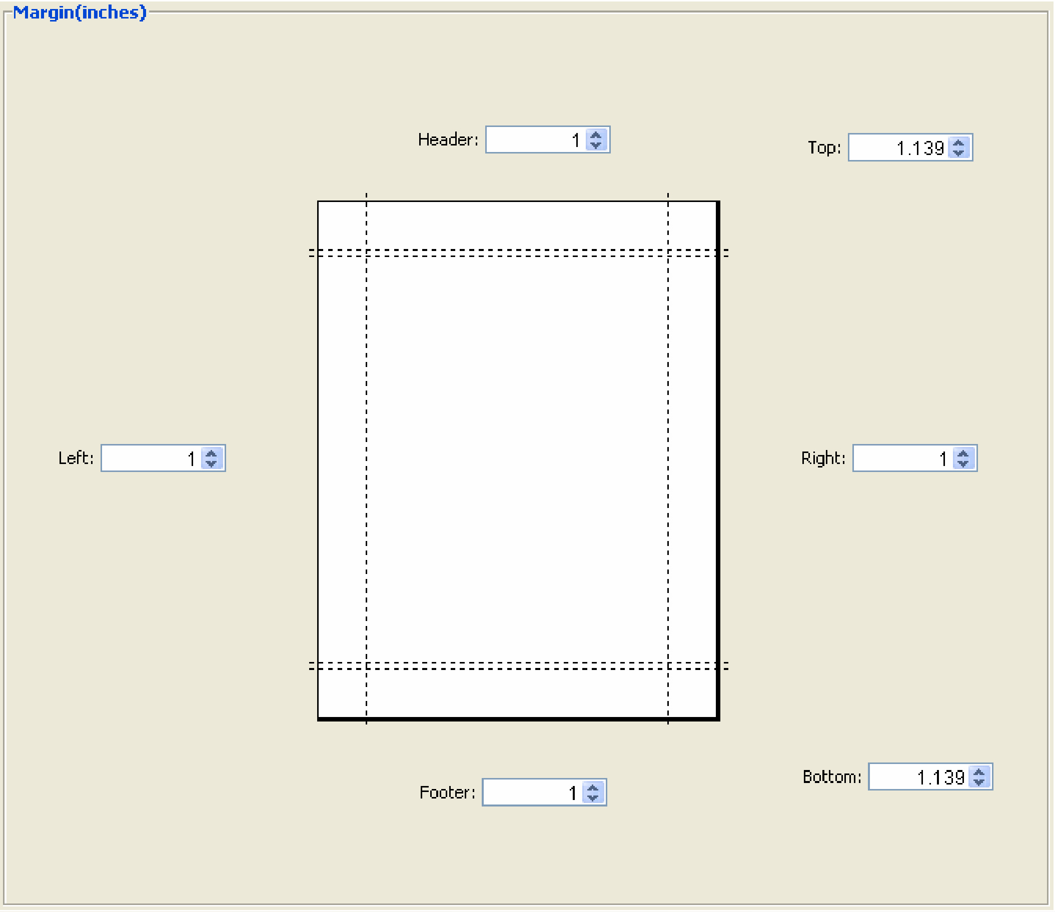
Top: 1 Bottom: 1

Figure 6-4 *Page setup*

1. Click on the Page Setup button on the toolbar. The Page Setup dialog box appears.
2. You can click on the Size drop-down menu to select the paper size to use.
3. You can select the orientation for the page(s) to be printed (either Portrait or Landscape) in the Orientation field.
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.



Margin(inches)

Header: 1

Top: 1.139

Left: 1

Right: 1

Footer: 1

Bottom: 1.139

Figure 6-5 Adjusting margins

1. Click on the Adjust Margins button on the Toolbar. The preview area shows the margin setting page.

2. You can edit the margins sizes by entering the sizes into the text fields. Alternatively, click on the spinner buttons to increase/decrease the margin sizes.

3. Click the Finish Adjust Margin button when you have finished configuring the margin settings. The margin sizes will then be updated.

Zooming Pages

Diagrams can be zoomed in or zoomed out according to user preference.

Click on the Zoom drop-down menu to select the desired zoom ratio.

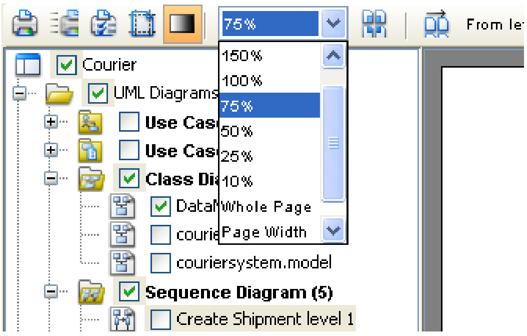


Figure 6-6 Set zoom ratio

The preview area will show the diagrams in the zoom ratio that you have selected.

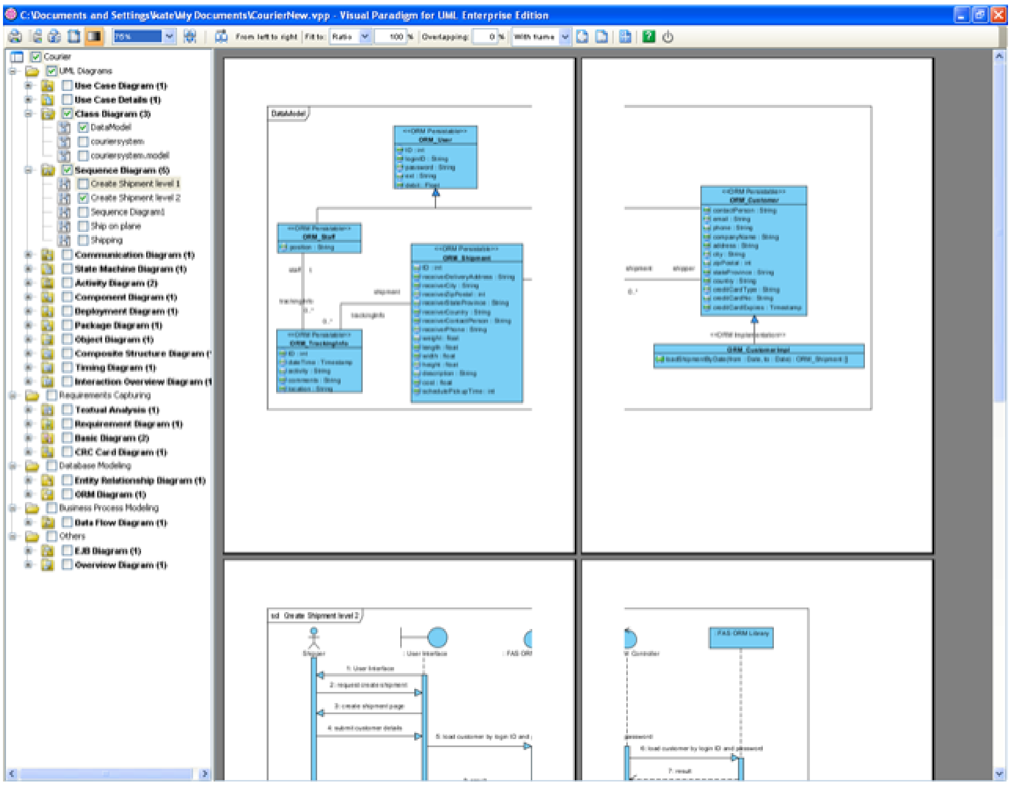


Figure 6-7 *Preview in preview dialog*

Selecting the Preview Layout

There are two layouts that you can select for the print preview, the Paper Base Layout and the Diagram Base Layout.

If the Fit to Pages option is selected and there are multiple pages in the printout, selecting Paper Base Layout will cause the distribution of pages to be paper-oriented (the diagram size is ignored in arranging the preview); while selecting 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 on the Paper Base Layout button or Diagram Base Layout button on the toolbar. A popup menu will appear where you can select the layout to use.

The preview after applying the Paper Base Layout:

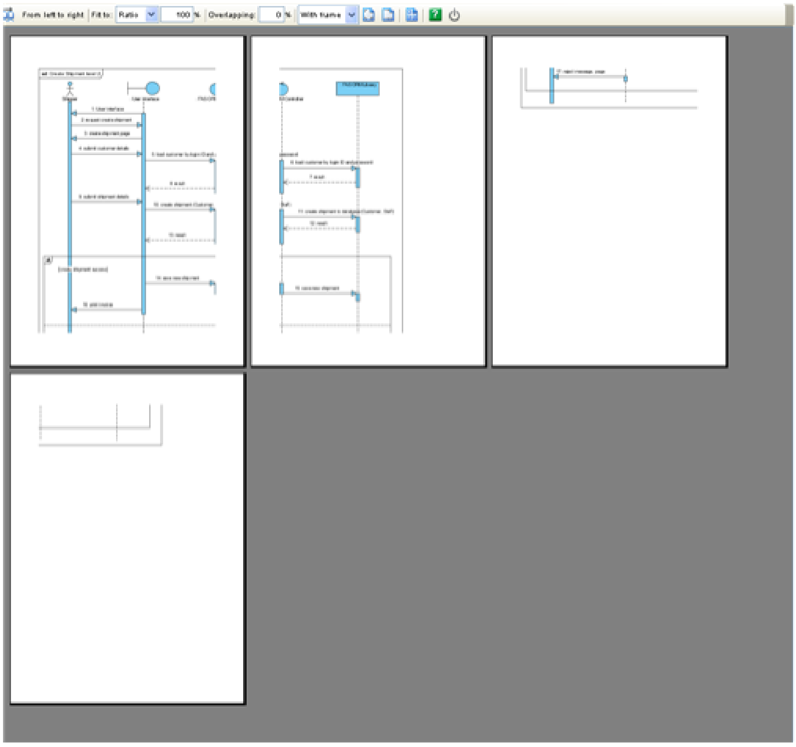


Figure 6-8 *Preview in paper base layout*

The preview after applying the Diagram Base Layout:

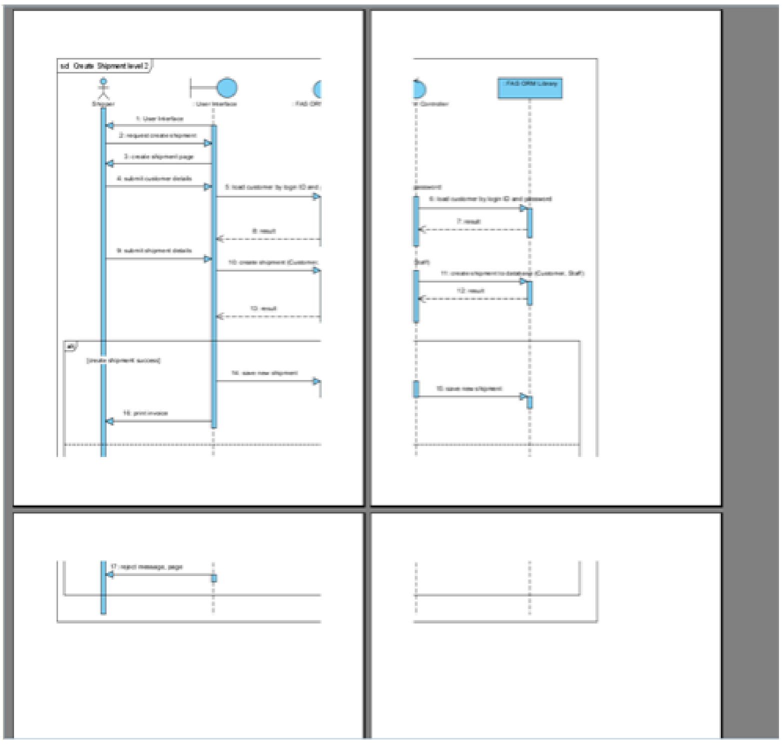


Figure 6-9 *Preview in diagram base layout*

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 on the Paper Place Style button on the toolbar. A popup menu appears where you can select a paper place style.

Consider a large diagram is divided into many pages, selecting 'From left to right' will arrange the printout order from the pages on the left to the pages on the right, while selecting '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 selecting From left to right.

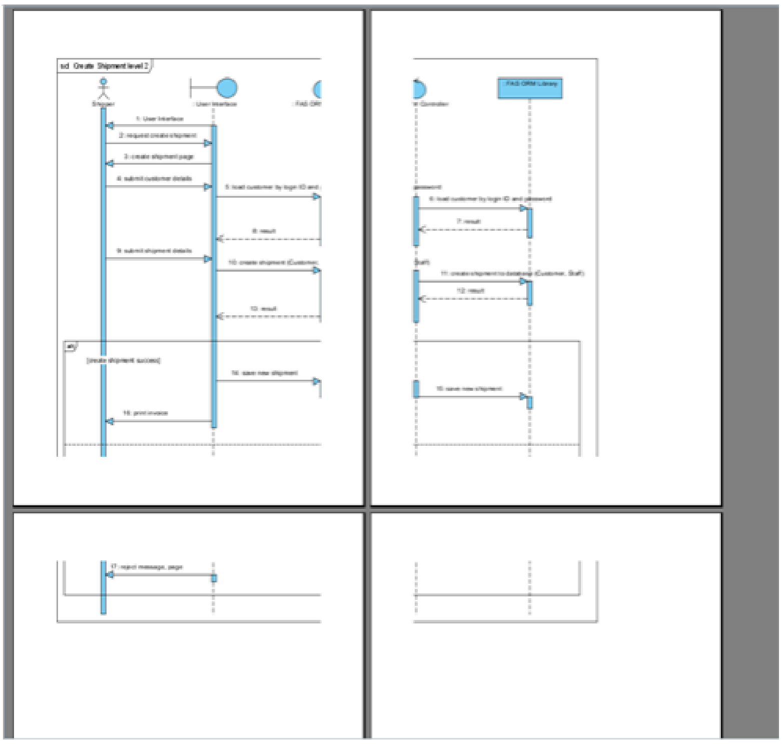


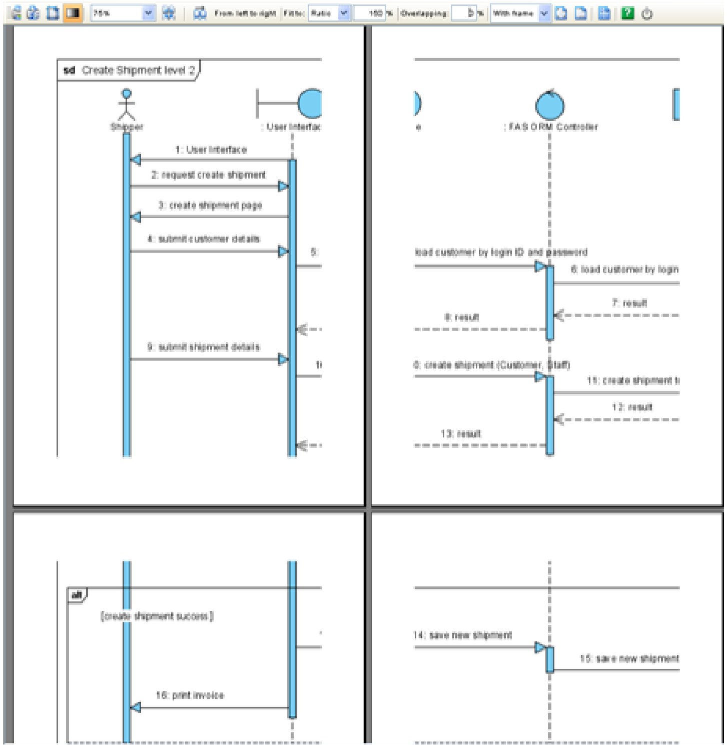
Figure 6-10 Printout order is left to right

Fit to Ratio

Fit to Ratio is used to resize the diagrams in the printout to a specific ratio.

Click on the Fit to drop-down menu and select Ratio.

You can enter the ratio into the textbox, e.g. enter 150 to set the ratio to 150%. After you have edited the ratio, the diagrams in the printout will be resized to the new ratio.


75%
From left to right
Fit to: Ratio
150 %
Overlapping: 5 %
With frame

sd Create Shipment level 2
Shipper
: User Interfac
: FAS ORM Controller
1: User Interface
2: request create shipment
3: create shipment page
4: submit customer details
load customer by login ID and password
6: load customer by login
7: result
8: result
9: submit shipment details
0: create shipment (Customer, Staff)
11: create shipment t
12: result
13: result
alt
[create shipment success]
14: save new shipment
15: save new shipment
16: print invoice

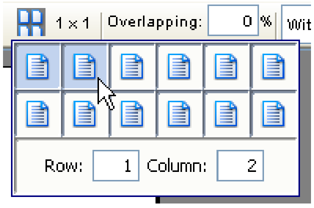
Figure 6-11 Fit to ratio

Fit to Pages

Fit to Pages is used to split the diagram to a desired number of pages when printing.

1. Click on the Fit to drop-down menu and select Pages.

2. Click on the Multiple Pages button on the toolbar. The page selector appears.



1 x 1 | Overlapping: 0 % | Wit

Row: 1 Column: 2

Figure 6-12 *Select multiple pages page*

3. Click on 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.

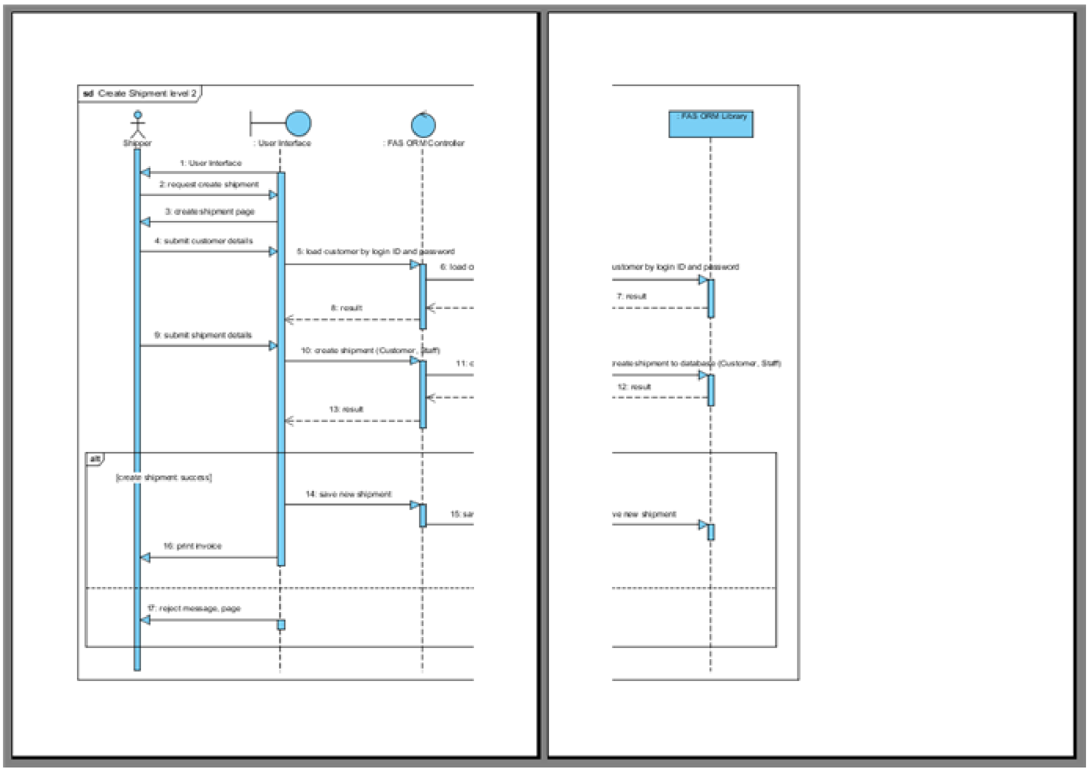

sd Create Shipment level 2
Shipper
: User Interface
: FAS ORM Controller
: FAS ORM Library
1: User Interface
2: request create shipment
3: create shipment page
4: submit customer details
5: load customer by login ID and password
6: load customer by login ID and password
7: result
8: result
9: submit shipment details
10: create shipment (Customer, Staff)
11: create shipment to database (Customer, Staff)
12: result
13: result
alt
[create shipment success]
14: save new shipment
15: save new shipment
16: print invoice
17: reject message, page

Figure 6-13 *Fit to page*

Setting the Diagram Overlap Percentage

Overlapping is used when users want the diagrams to have overlapping 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.

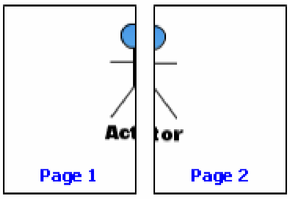


Figure 6-14 *Multiple page without overlap*

1. Click on the Overlapping textbox to input the overlapping percentage and press the Enter key.

2. The printing area near the boundaries of the pages will be duplicated by the overlapping percentage inputted.

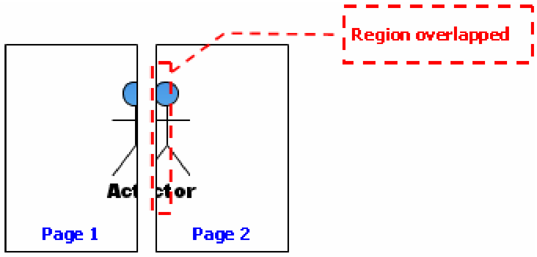


Figure 6-15 *Multiple page with overlap*

Printing with Frame/Border Option

You can print your diagram with a frame or border. There are three options:

- With frame
- With border
- No border

Select With frame/ With border/No border option from the drop-down menu.

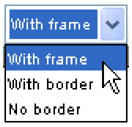


Figure 6-16 *Select option from drop-down menu*

Output of printing with frame

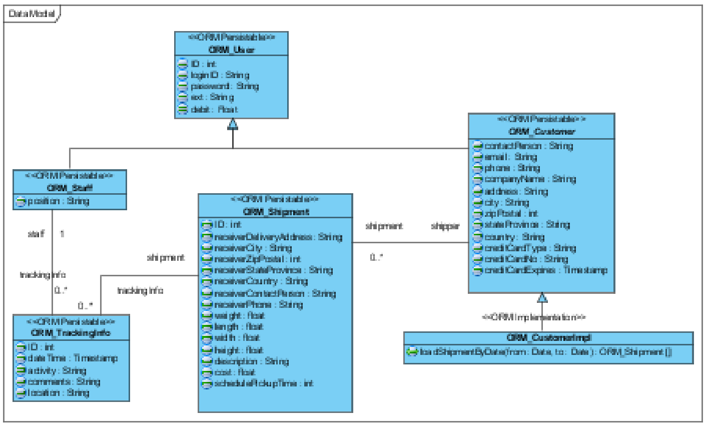


Figure 6-17 *Printing with frame*

Output of printing with border

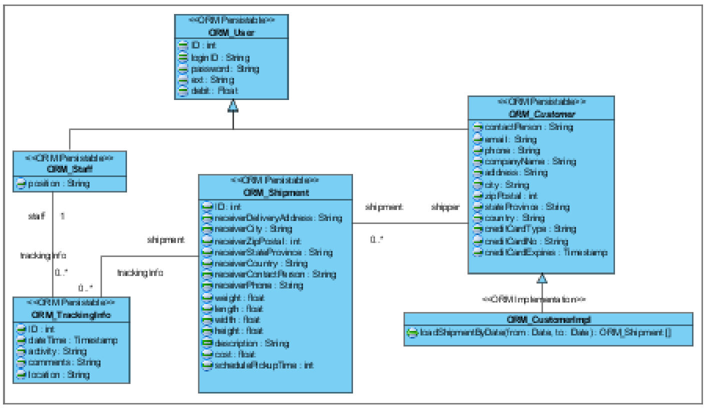


Figure 6-18 Printing with border

Output of printing with no border

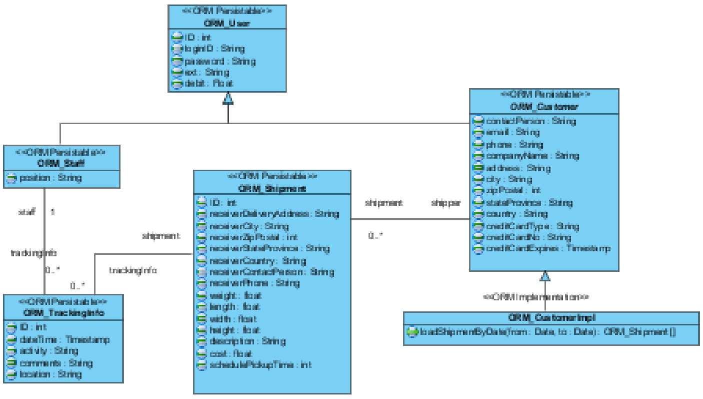


Figure 6-19 Printing with no border

Showing/Hiding Clip Marks on Page

Clip marks act as an indication of the boundary of a page.

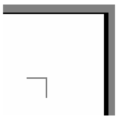


Figure 6-20 Clip marks

To show clip marks on the printout click on the Show Clip Marks on Page button. You will see the boundaries of the pages are surrounded by clip marks. To hide the clip marks click on the Hide Clip Marks on Page button again.

Editing Header/Footer of the Pages

To edit the header/footer of the printout click on the Edit Header/Footer button 🗎 on the toolbar. You will then switch to the edit header/footer pane.

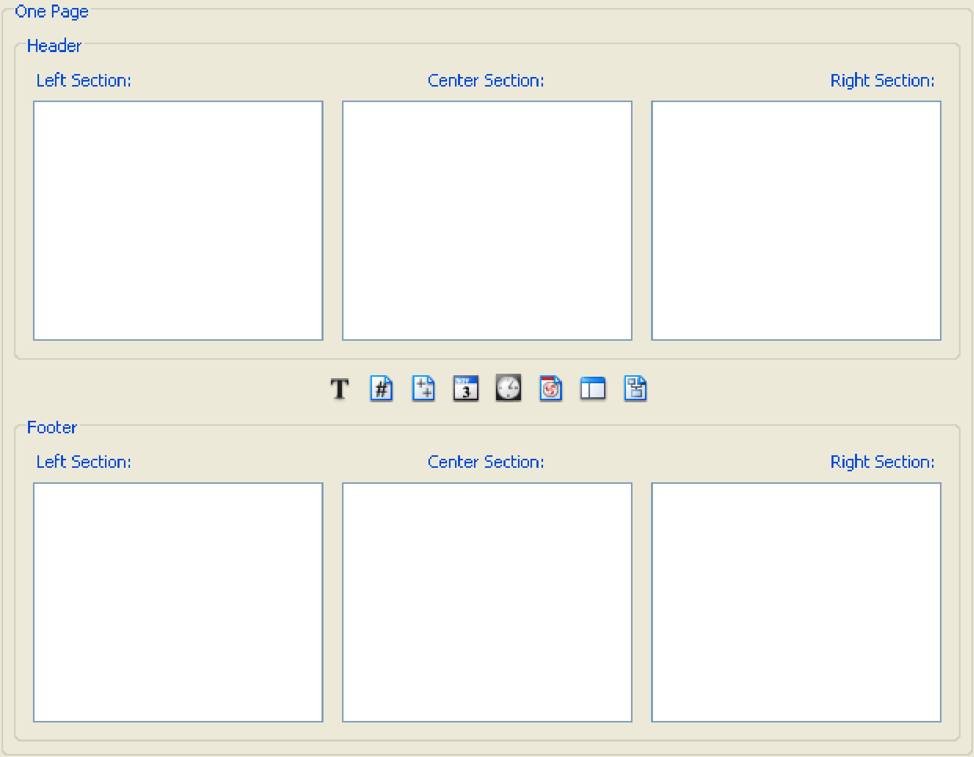


Figure 6-21 *Editing header/footer of pages*

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	Description
T	Select the font to use for the selected section. Note that you must click on the section once in order to select it.
#	Insert the page number.
⊞	Insert the total number of pages.
3	Insert the date that the printing starts.
🕒	Insert the time that the printing starts.
🗎	Insert the file name of the VP-UML project.
▭	Insert the name of the VP-UML project.
🗎	Insert the diagram name.

Table 6-2

After you have finished editing the header/footer, click on the Close Edit Header/Footer button to switch to the print preview mode. A sample page that has the header and footer formatted is shown in the picture below:

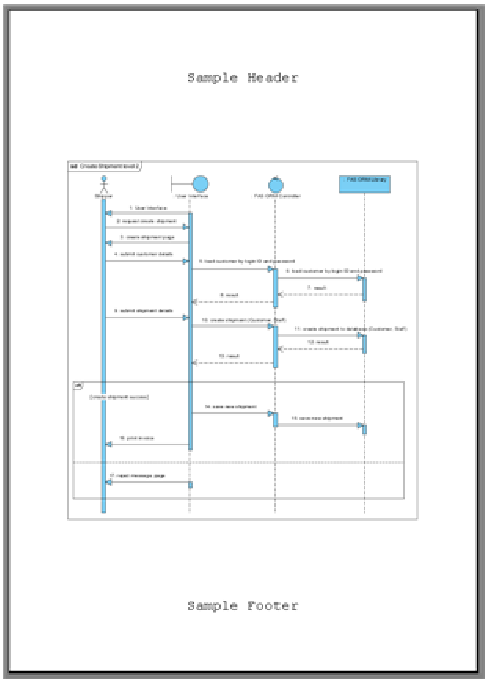
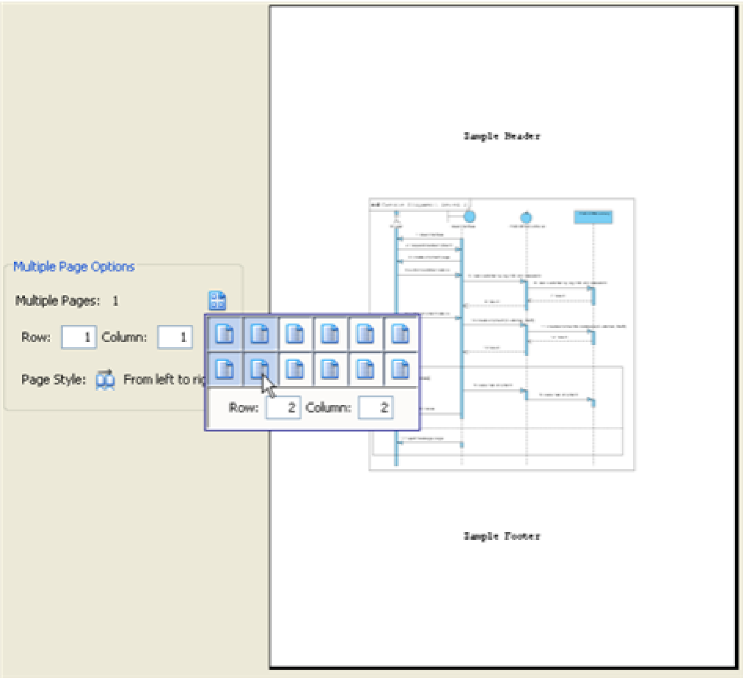


Figure 6-22 *Page with header and footer*

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 on the Multiple Page Mode button on the toolbar.

Clicking on the button beside the Multiple Pages field will invoke the page selector, where you can select the row-column combination for the printout. Alternatively, you can type in the Row and Column text field directly.



Multiple Page Options

Multiple Pages: 1

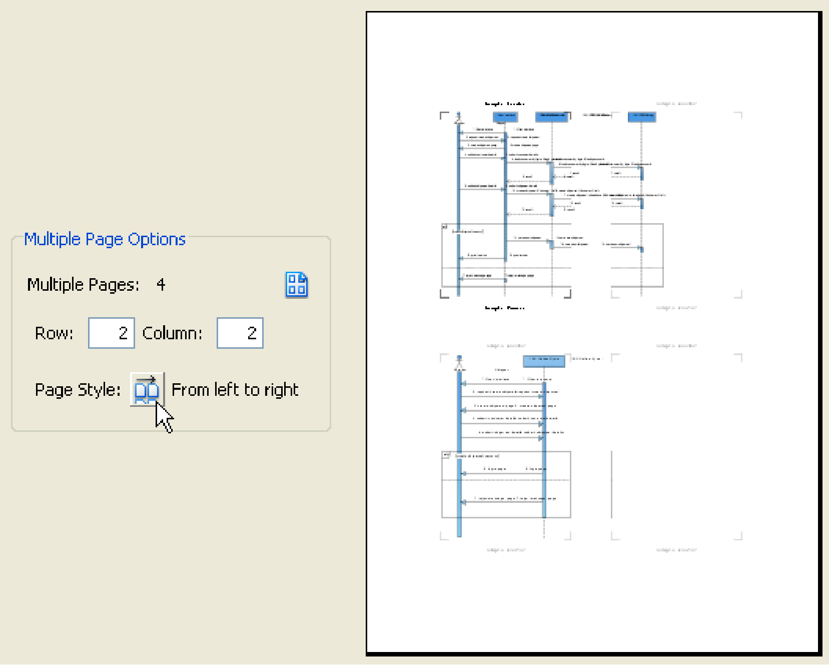
Row: 1 Column: 1

Page Style: From left to ri

Row: 2 Column: 2

Figure 6-23 *Select multiple page*

Click on the button beside the Page Style field to change the printout order. Consider a large diagram is divided into many pages, selecting 'From left to right' will arrange the printout order from the pages on the left to the pages on the right, while selecting 'From top to bottom' will arrange the print order from the pages on the top to the pages on the bottom.



Multiple Page Options

Multiple Pages: 4

Row: 2 Column: 2

Page Style: From left to right

__Figure 6-24__ Distributes diagram in multiple page

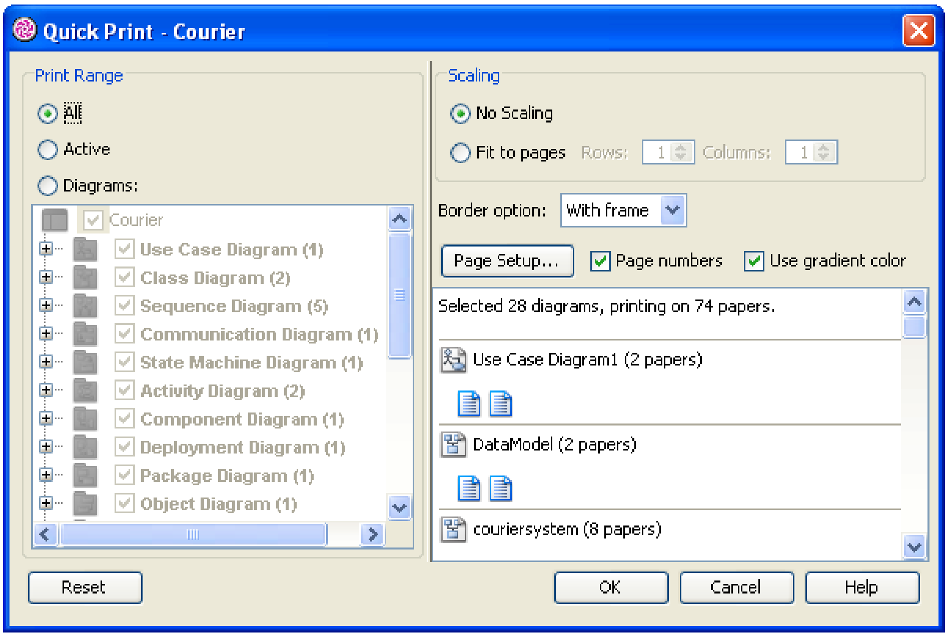
After you have finished configuring the multiple page settings click on the Close Multiple Page Mode button to close the Multiple Page Mode.

Printing a Diagram with Quick Print

The Quick Print feature allows you to print diagrams without previewing them, hence speeding up the print job. To quick print, perform one of the following actions:

- Select File > Quick Print...from main menu
- Select File > Print... from main menu. This displays the Print Preview dialog box. Click on the toolbar of the Print Preview dialog box.

In both cases, the Quick Print dialog box will show.



Quick Print - Courier

Print Range
● All
○ Active
○ Diagrams:

- ☑ Courier
  - ☑ Use Case Diagram (1)
  - ☑ Class Diagram (2)
  - ☑ Sequence Diagram (5)
  - ☑ Communication Diagram (1)
  - ☑ State Machine Diagram (1)
  - ☑ Activity Diagram (2)
  - ☑ Component Diagram (1)
  - ☑ Deployment Diagram (1)
  - ☑ Package Diagram (1)
  - ☑ Object Diagram (1)

Scaling
● No Scaling
○ Fit to pages Rows: 1 Columns: 1

Border option: With frame

☑ Page numbers ☑ Use gradient color

Selected 28 diagrams, printing on 74 papers.

Use Case Diagram1 (2 papers)

DataModel (2 papers)

couriersystem (8 papers)

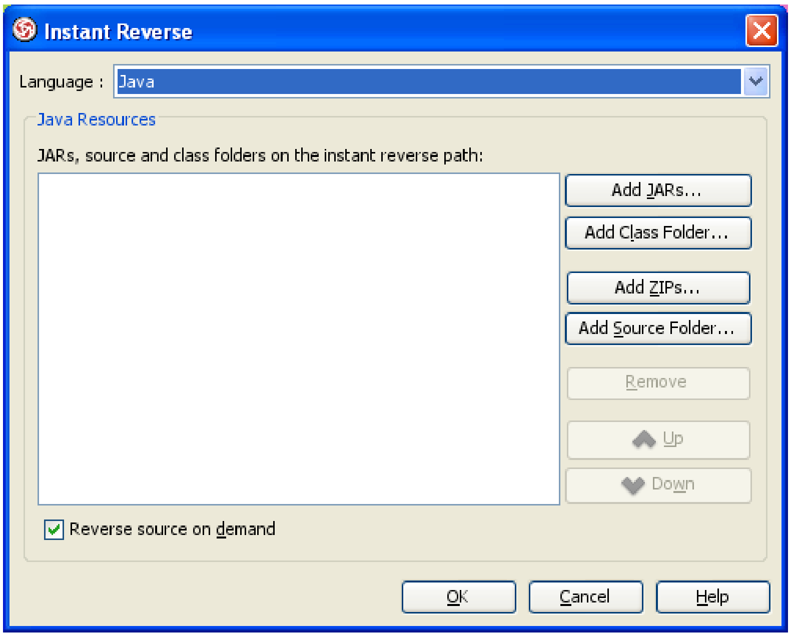
Figure 6-25 *Quick print dialog*

Field	Description
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
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.
Border option	Select border option of printout.
Page Setup...	Page Setup allows you to specify the page size, the orientation as well as the margins of the pages.
Page numbers	Select to print diagrams with page number on it.
Use gradient color	Select to use gradient color in printout.

Table 6-3 *Details of quick print dialog*

Instant Reverse Java Sources and Classes

Open Instant Reverse dialog box



Language : Java

JARs, source and class folders on the instant reverse path:

☑ Reverse source on demand

Figure 1-2 Instant reverse dialog

Add JARS

Click on **Add JARs...** button and select JAR files to add to the instant reverse paths.

Add Classes Folder

Click on **Add Class Folder...** button and select class folders to add to the instant reverse paths.

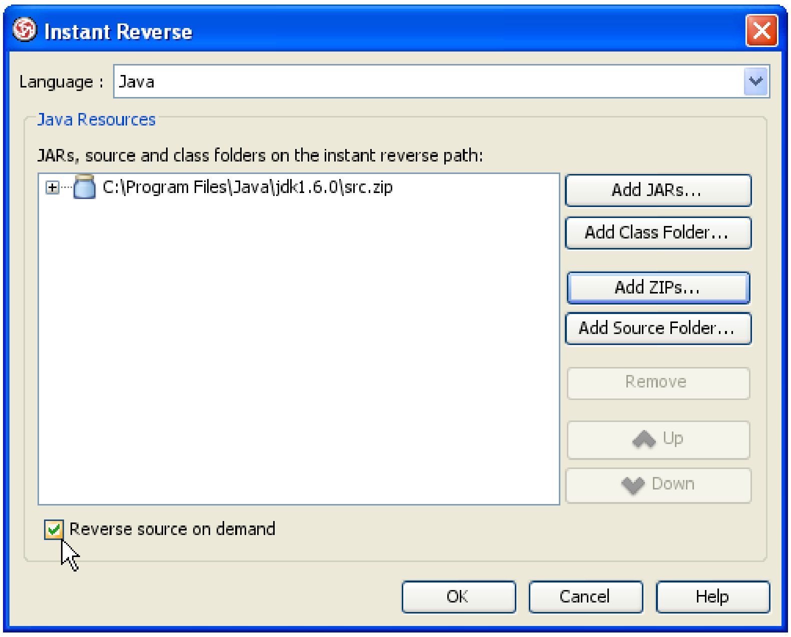
Add ZIPs

Click on **Add ZIPs...** button and select ZIP files to add to the instant reverse paths.

Add Source Folder

Click on **Add Source Folder...** button and select source folders to add to the instant reverse paths.

Reverse source on demand



Language : Java

JARs, source and class folders on the instant reverse path:

C:\Program Files\Java\jdk1.6.0\src.zip

☑ Reverse source on demand

Figure 1-7 Reverse source on demand option

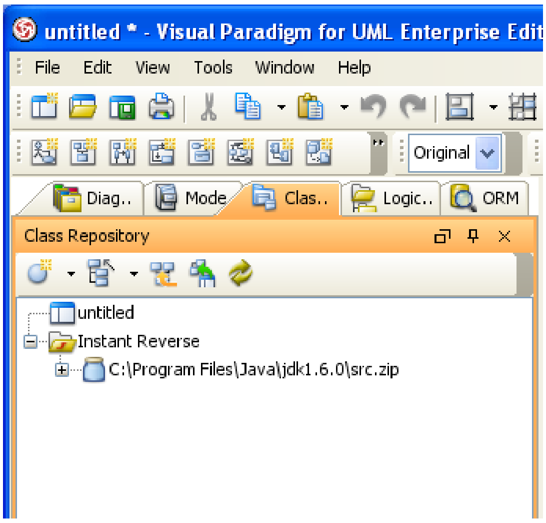


Figure 1-8 Instant reverse node

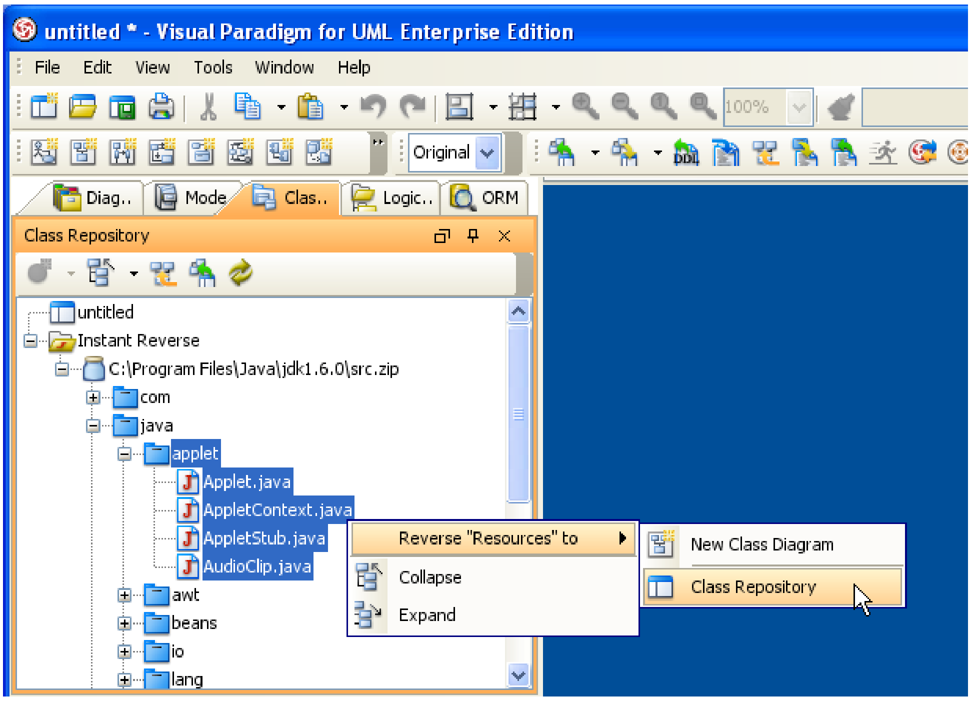


Figure 1-9 Reverse to class repository

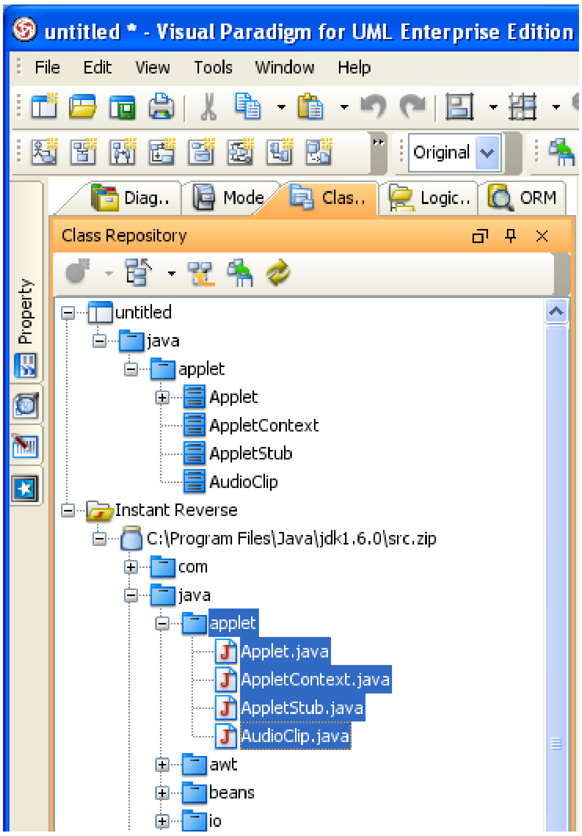


Figure 1-10 Classes reversed

Create Class Diagram

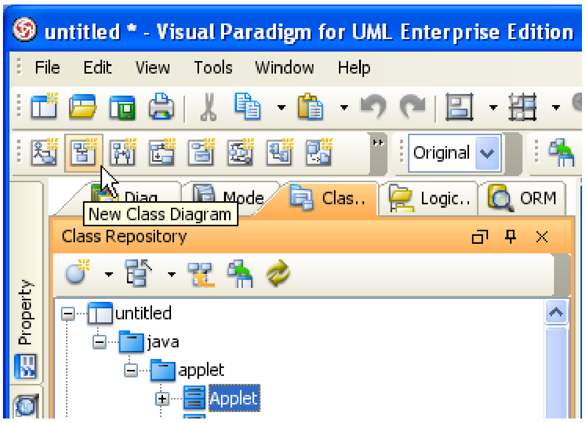


Figure 1-11 New diagram

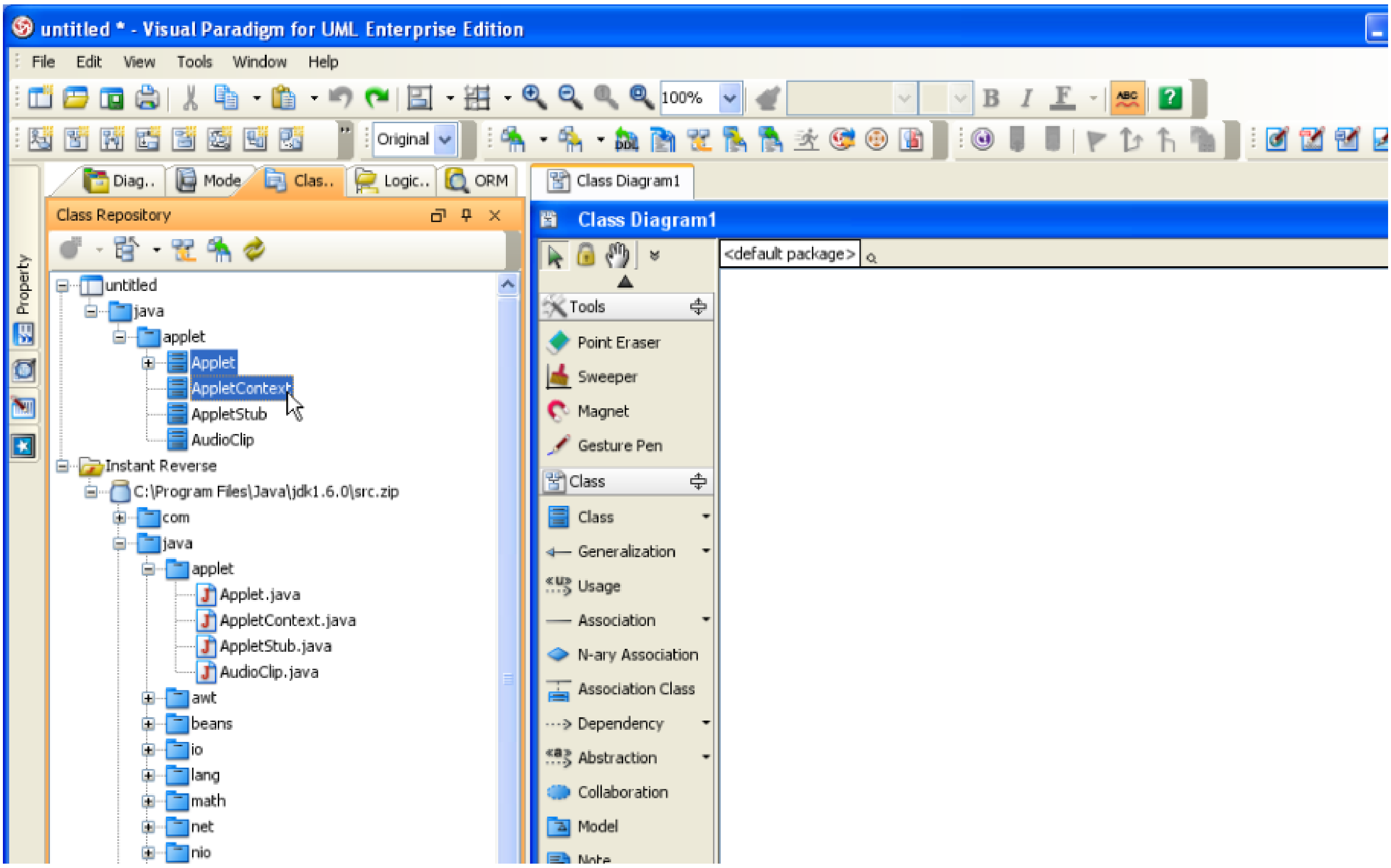


Figure 1-12 Dragging classes

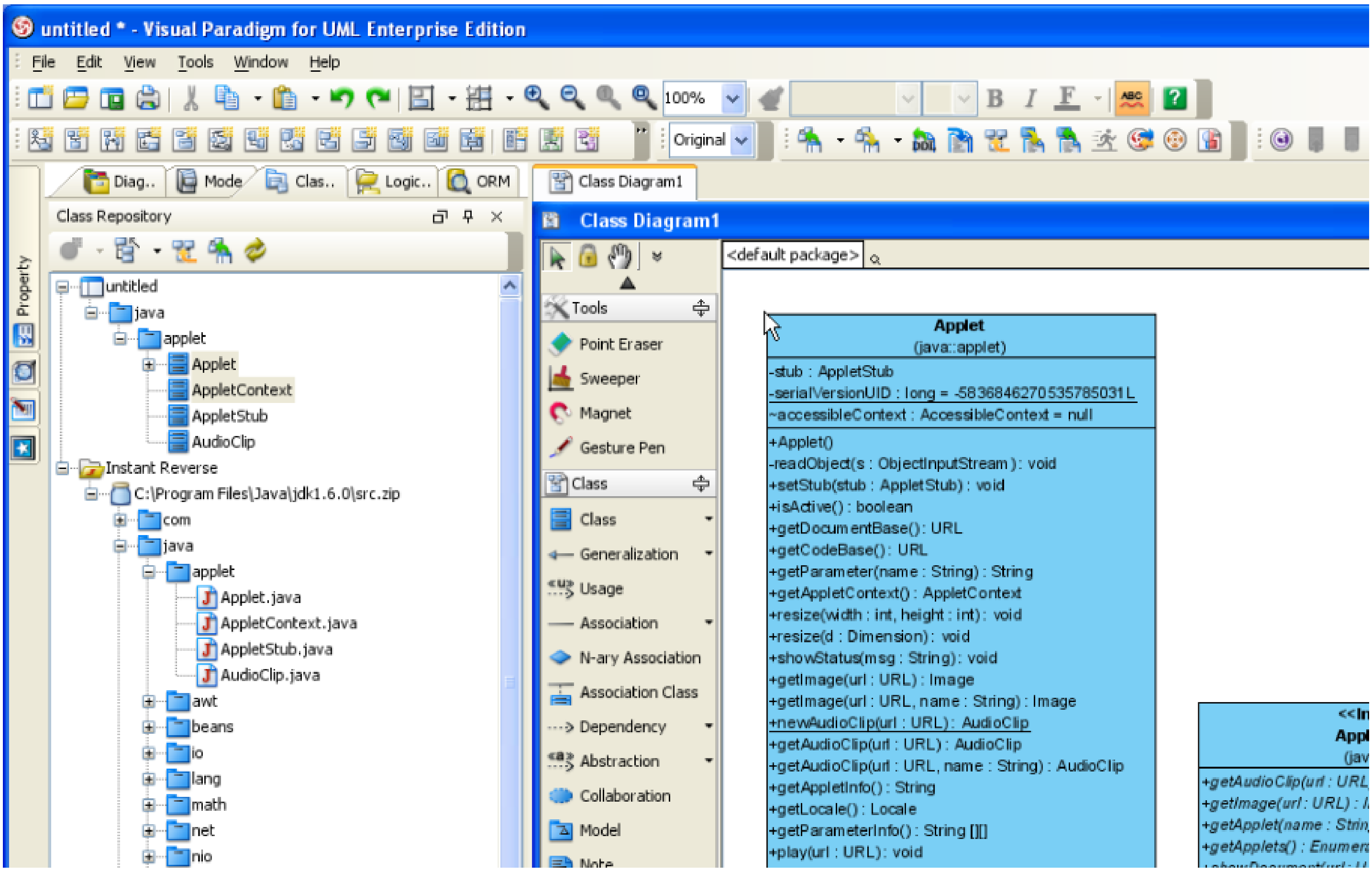

untitled * - Visual Paradigm for UML Enterprise Edition
File Edit View Tools Window Help
100%
Original
Diag.. Mode Clas.. Logic.. ORM
Class Diagram1
Class Repository
Property
untitled
java
applet
Applet
AppletContext
AppletStub
AudioClip
Instant Reverse
C:\Program Files\Java\jdk1.6.0\src.zip
com
java
applet
Applet.java
AppletContext.java
AppletStub.java
AudioClip.java
awt
beans
io
lang
math
net
nio
Class Diagram1
<default package>
Tools
Point Eraser
Sweeper
Magnet
Gesture Pen
Class
Class
Generalization
Usage
Association
N-ary Association
Association Class
Dependency
Abstraction
Collaboration
Model
Note
Applet
(java::applet)
-stub : AppletStub
-serialVersionUID : long = -5836846270535785031L
~accessibleContext : AccessibleContext = null
+Applet()
-readObject(s : ObjectInputStream) : void
+setStub(stub : AppletStub) : void
+isActive() : boolean
+getDocumentBase() : URL
+getCodeBase() : URL
+getParameter(name : String) : String
+getAppletContext() : AppletContext
+resize(width : int, height : int) : void
+resize(d : Dimension) : void
+showStatus(msg : String) : void
+getImage(url : URL) : Image
+getImage(url : URL, name : String) : Image
+newAudioClip(url : URL) : AudioClip
+getAudioClip(url : URL) : AudioClip
+getAudioClip(url : URL, name : String) : AudioClip
+getAppletInfo() : String
+getLocale() : Locale
+getParameterInfo() : String [][]
+play(url : URL) : void
<<In
Appl
(jav
+getAudioClip(url : URL
+getImage(url : URL) : I
+getApplet(name : Strin
+getApplets() : Enumera

Figure 1-13 Classes created

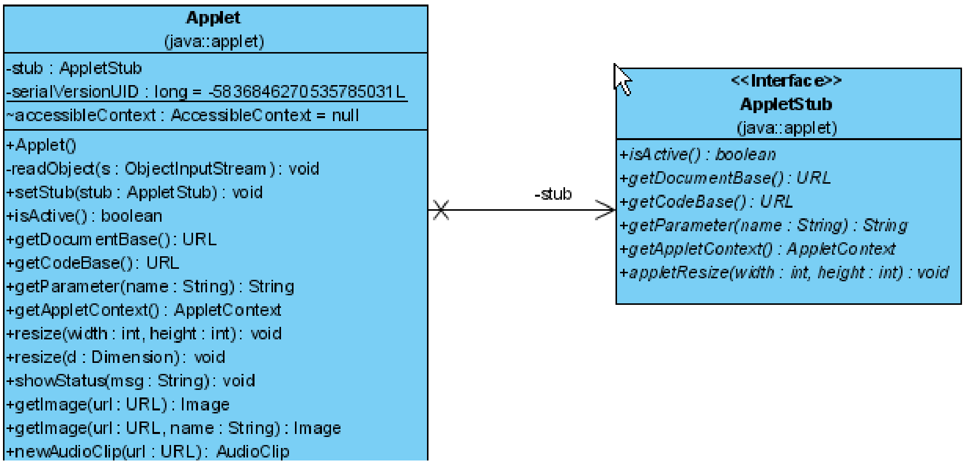

Applet
(java::applet)
-stub : AppletStub
-serialVersionUID : long = -5836846270535785031L
~accessibleContext : AccessibleContext = null
+Applet()
-readObject(s : ObjectInputStream) : void
+setStub(stub : AppletStub) : void
+isActive() : boolean
+getDocumentBase() : URL
+getCodeBase() : URL
+getParameter(name : String) : String
+getAppletContext() : AppletContext
+resize(width : int, height : int) : void
+resize(d : Dimension) : void
+showStatus(msg : String) : void
+getImage(url : URL) : Image
+getImage(url : URL, name : String) : Image
+newAudioClip(url : URL) : AudioClip
-stub
<<Interface>>
AppletStub
(java::applet)
+isActive() : boolean
+getDocumentBase() : URL
+getCodeBase() : URL
+getParameter(name : String) : String
+getAppletContext() : AppletContext
+appletResize(width : int, height : int) : void

Figure 1-14 *Relationship created*

Instant Reverse C++ header files

1. Select menu **Tools > Instant Reverse > C++ Source...**,

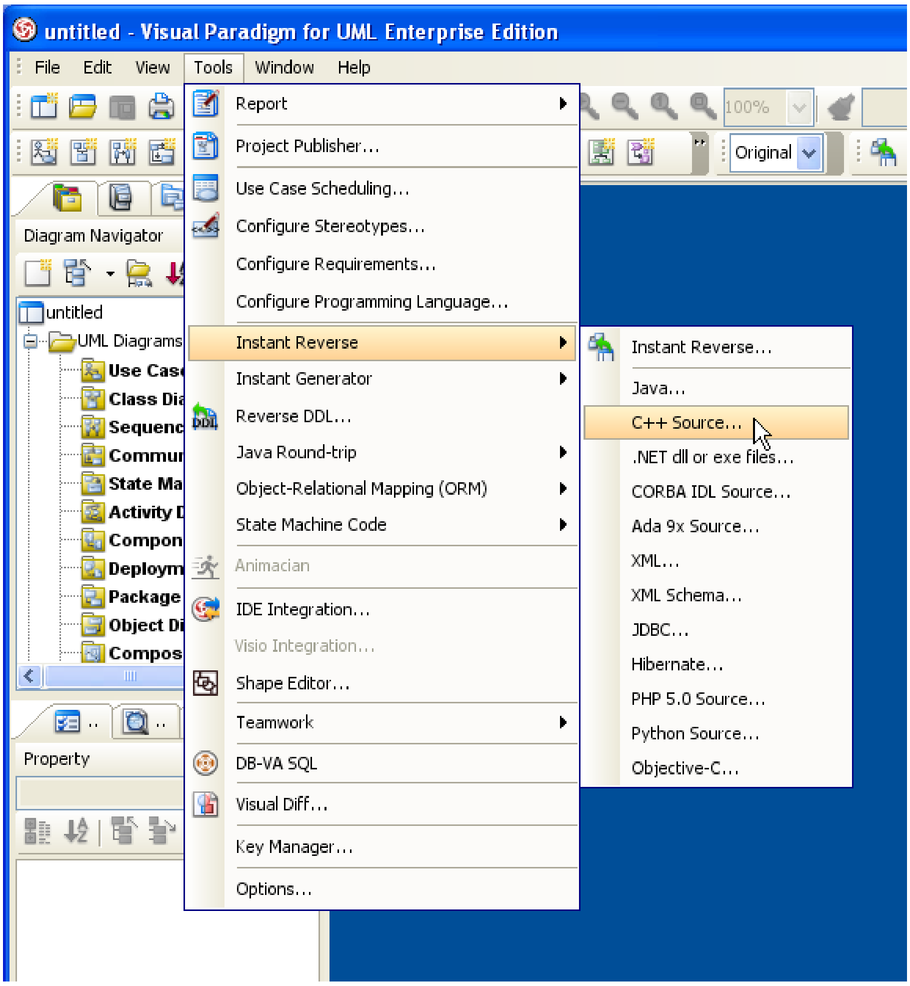
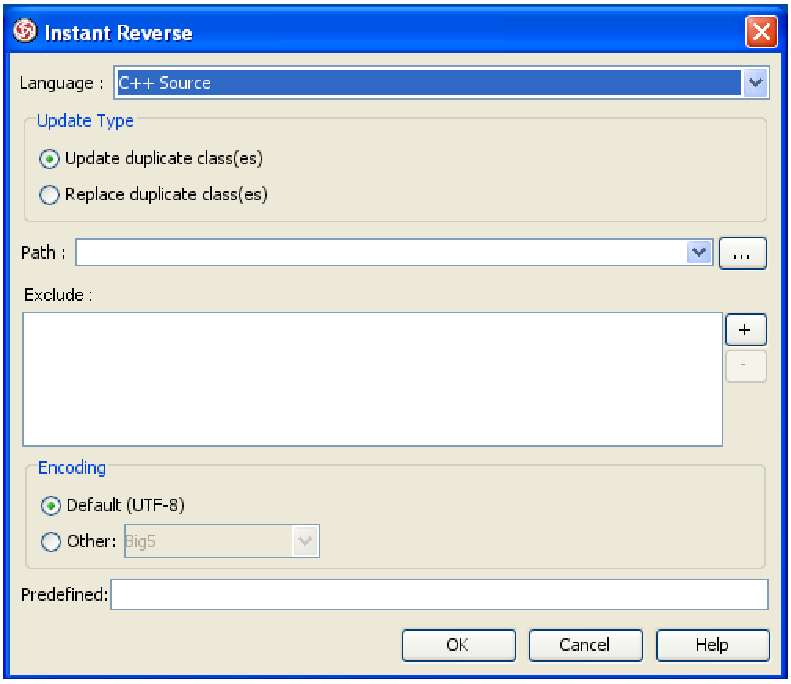


Figure 1-15 Instant reverse menu

2. the Instant Reverse dialog box shows.


Instant Reverse

Language : C++ Source

Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path :

Exclude :

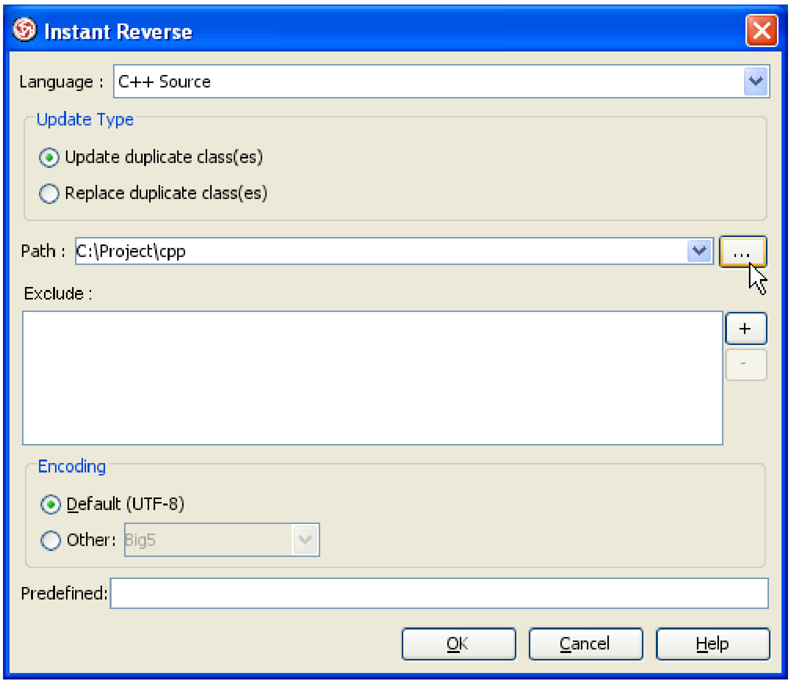
Encoding

● Default (UTF-8)

○ Other: Big5

Predefined:

Figure 1-16 Instant reverse dialog


Instant Reverse

Language : C++ Source

Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : C:\Project\cpp

Exclude :

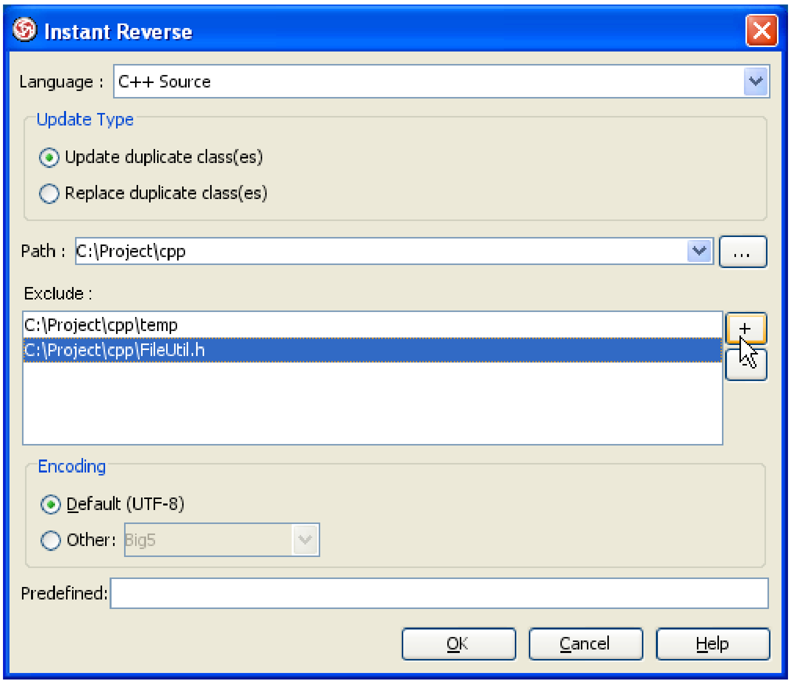
Encoding

● Default (UTF-8)

○ Other: Big5

Predefined:

Figure 1-17 *Select path*


Instant Reverse

Language : C++ Source

Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : C:\Project\cpp

Exclude :

C:\Project\cpp\temp
C:\Project\cpp\FileUtil.h

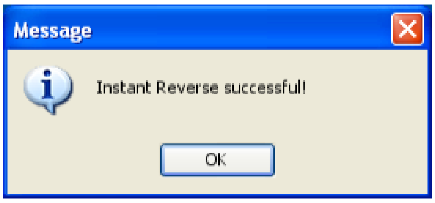
Encoding

● Default (UTF-8)

○ Other: Big5

Predefined:

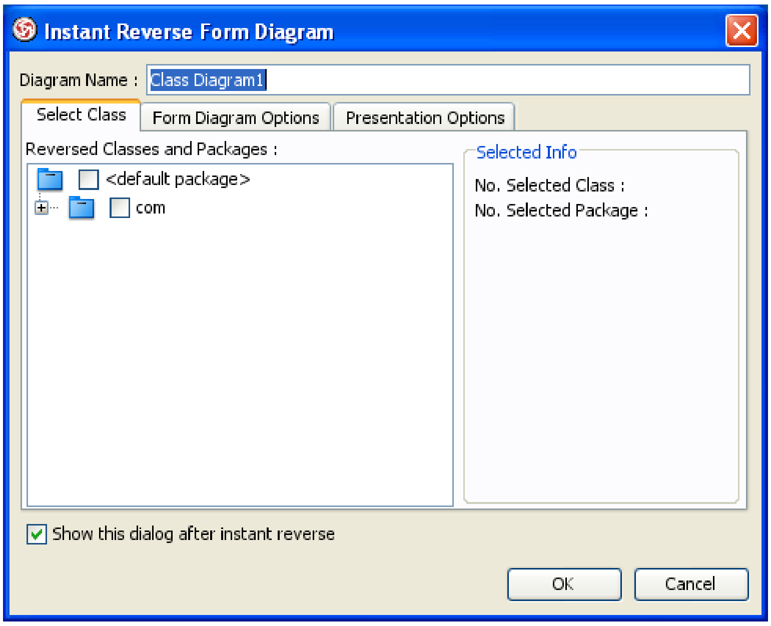
Figure 1-18 *Select exclude*


Message

Instant Reverse successful!

Figure 1-19 *Success message*

6. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.



Instant Reverse Form Diagram

Diagram Name : Class Diagram1

Reversed Classes and Packages :

- □ <default package>
- □ com

Selected Info

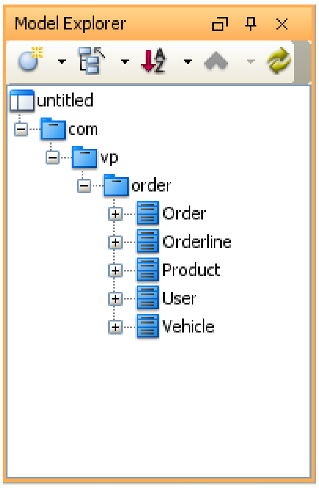
No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

Figure 1-20 *Instant Reverse Form Diagram dialog*

7. Reversed classes appear in **Model Exploer**



Model Explorer

- untitled
- com
  - vp
    - order
      - Order
      - Orderline
      - Product
      - User
      - Vehicle

Figure 1-21 *Reversed classes in model exploer*

Instant Reverse .NET dll and exec files

1. Select menu **Tools > Instant Reverse > .NET dll or exe files...**,

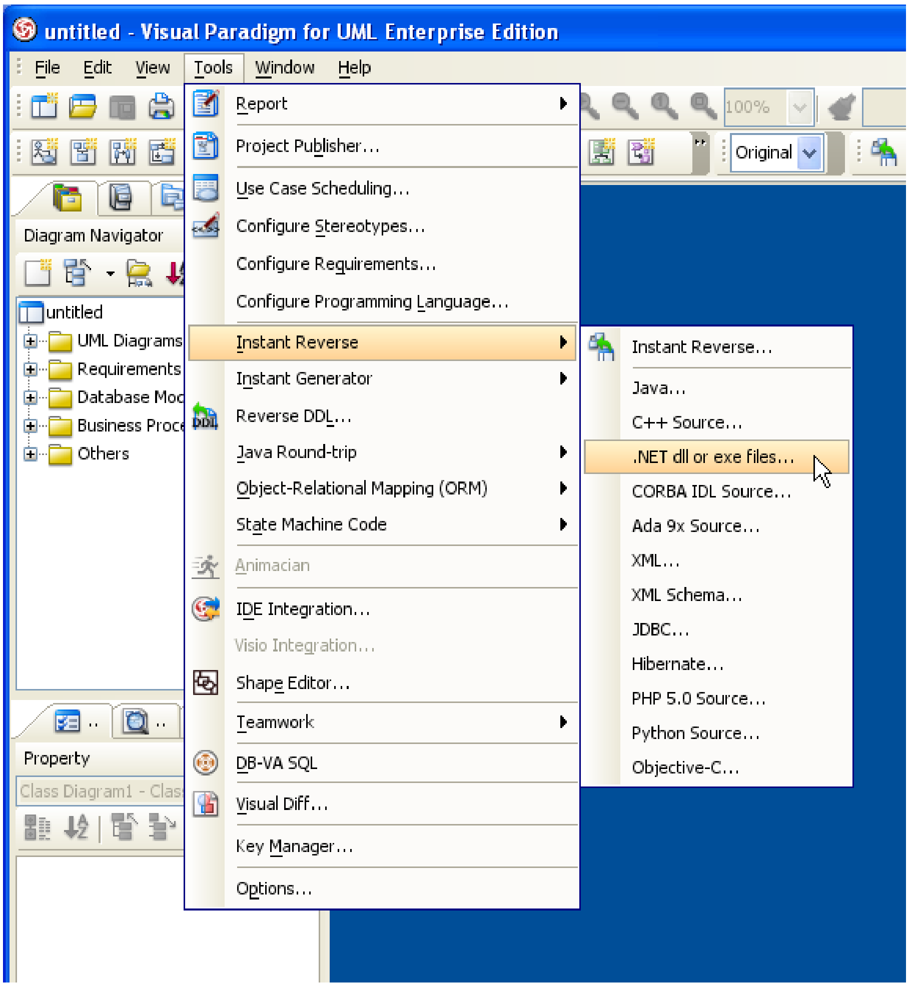
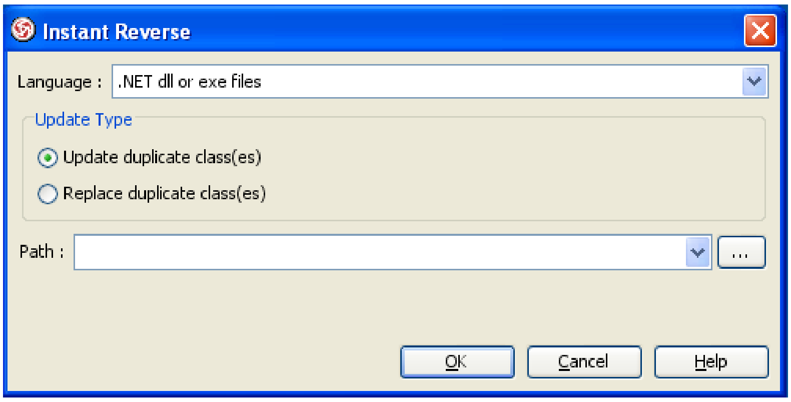


Figure 1-22 *Instant reverse menu*

2. the Instant Reverse dialog box shows.


Instant Reverse

Language : .NET dll or exe files

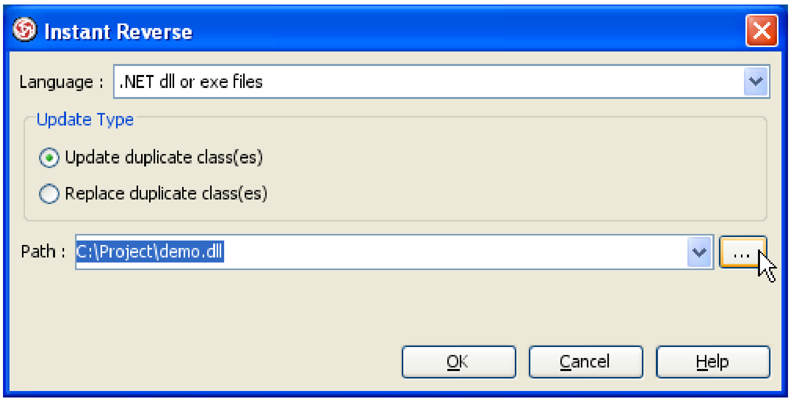
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path :

Figure 1-23 *Instant reverse dialog*



Instant Reverse

Language : .NET dll or exe files

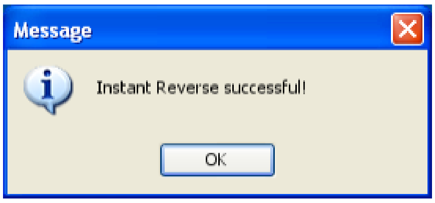
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : C:\Project\demo.dll

Figure 1-24 *Select path*

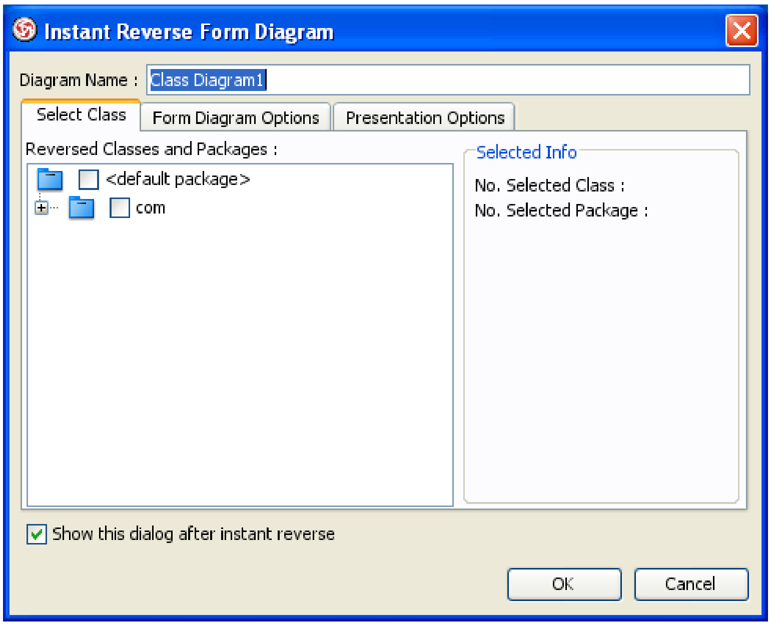


Message

Instant Reverse successful!

Figure 1-25 *Success message*

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.



Instant Reverse Form Diagram

Diagram Name : Class Diagram1

Reversed Classes and Packages :

□ <default package>

□ com

Selected Info

No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

Figure 1-26 *Instant Reverse Form Diagram dialog*

6. Reversed classes appear in **Model Exploer**

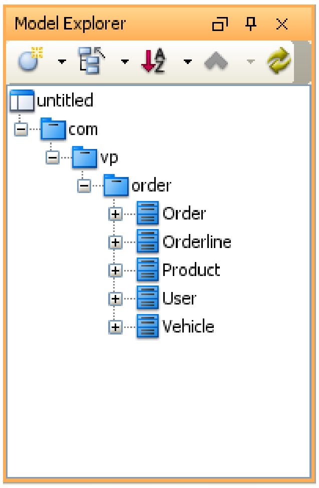


Figure 1-27 *Reversed classes in model exploer*

Instant Reverse CORBA IDL Source file

1. Select menu **Tools > Instant Reverse > CORBA IDL Source...,**

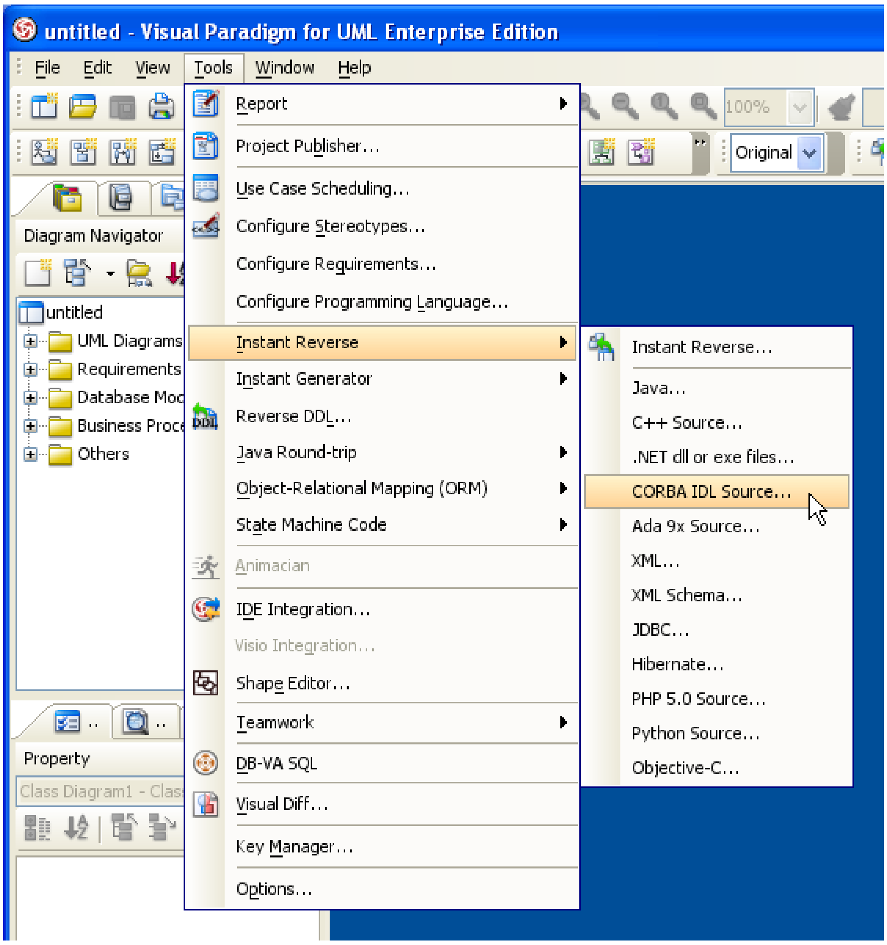
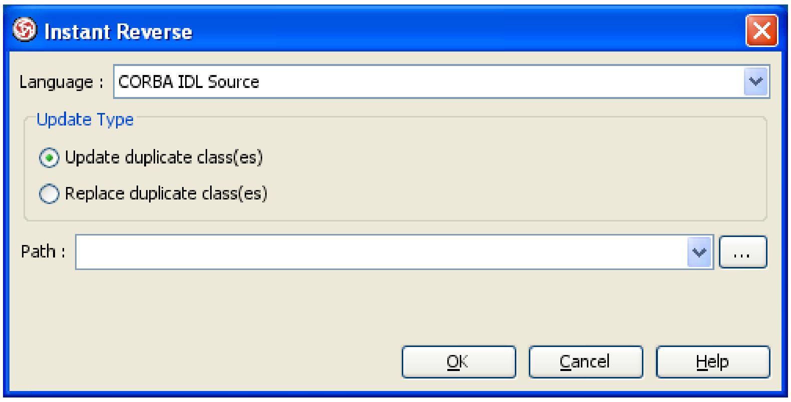


Figure 1-28 *Instant reverse menu*

2. the Instant Reverse dialog box shows.



Language : CORBA IDL Source

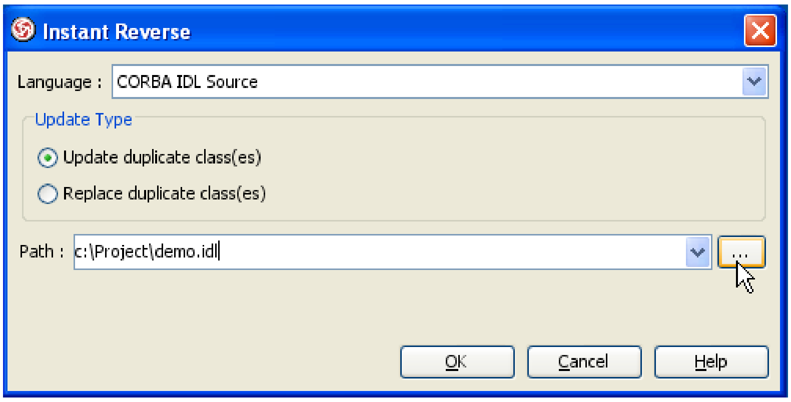
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path :

Figure 1-29 *Instant reverse dialog*


Instant Reverse

Language : CORBA IDL Source

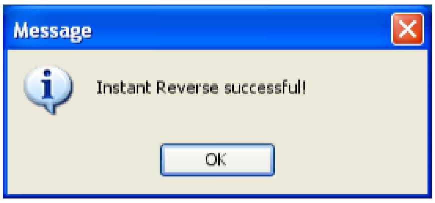
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : c:\Project\demo.idl

__Figure 1-30__ Select path


Message

Instant Reverse successful!

__Figure 1-31__ Success message

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.

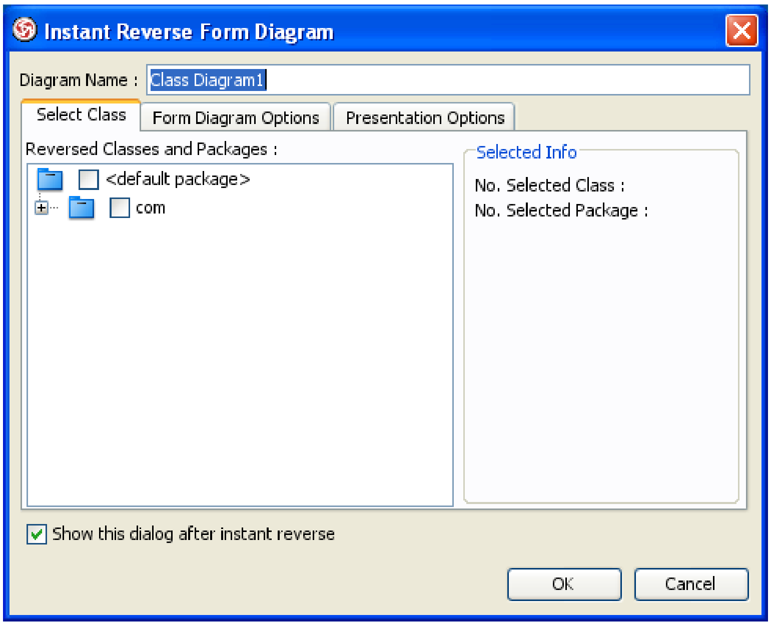

Instant Reverse Form Diagram

Diagram Name : Class Diagram1

Reversed Classes and Packages :

□ <default package>

□ com

Selected Info

No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

__Figure 1-32__ Instant Reverse Form Diagram dialog

6. Reversed classes appear in **Model Exploer**

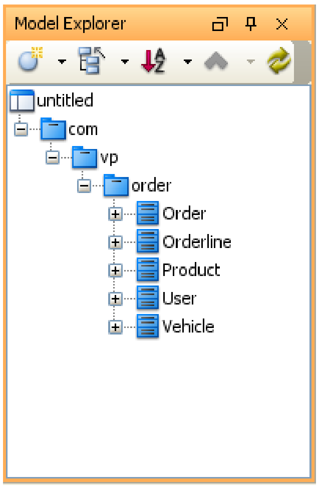


Figure 1-33 *Reversed classes in model exploer*

Instant Reverse Ada 9x Source files

1. Select menu **Tools > Instant Reverse > Ada 9x Source...,**

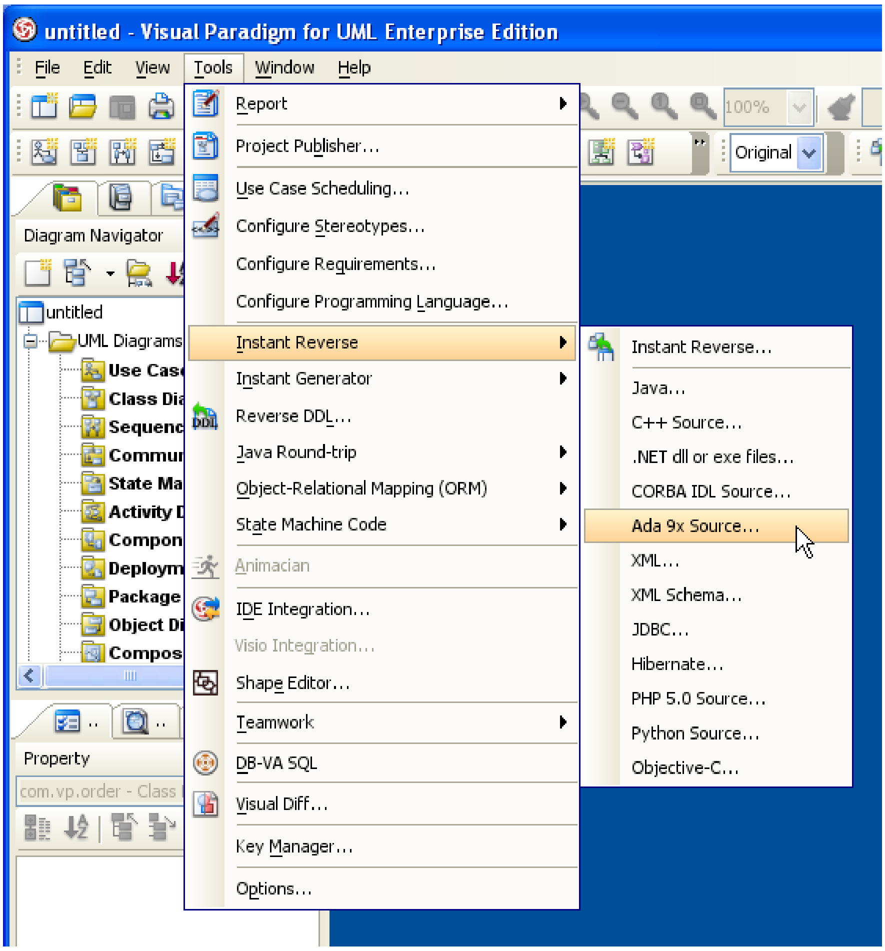
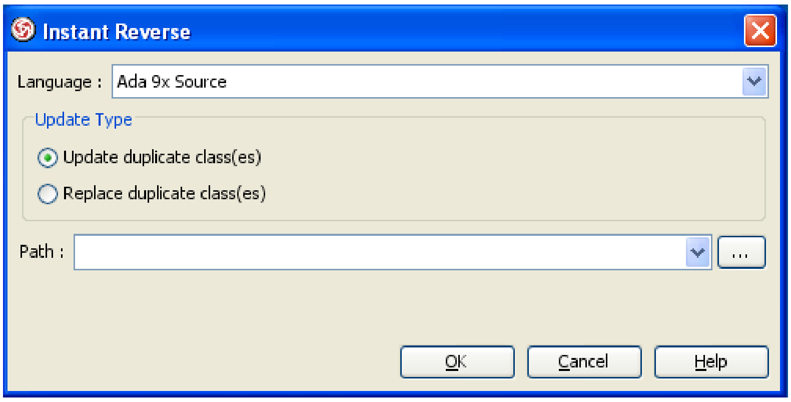


Figure 1-34 Instant reverse menu

2. the Instant Reverse dialog box shows.



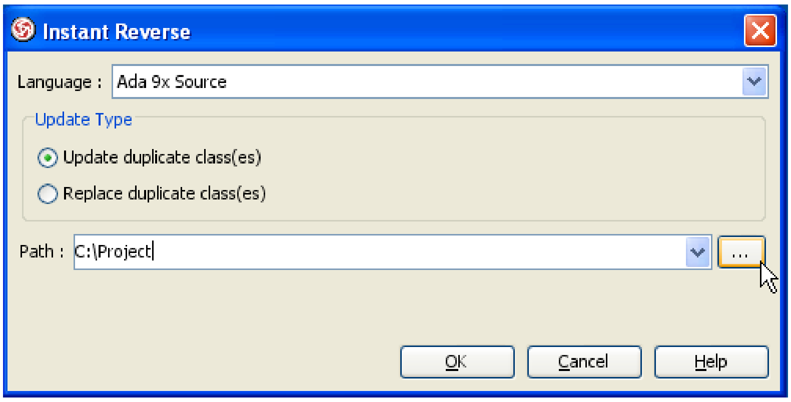
Language : Ada 9x Source

● Update duplicate class(es)

○ Replace duplicate class(es)

Path :

Figure 1-35 Instant reverse dialog


Instant Reverse

Language : Ada 9x Source

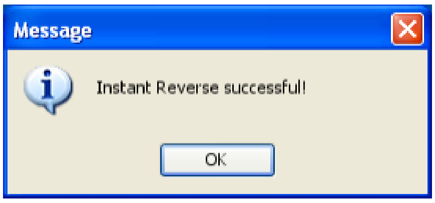
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : C:\Project

Figure 1-36 *Select path*


Message

Instant Reverse successful!

Figure 1-37 *Success message*

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.

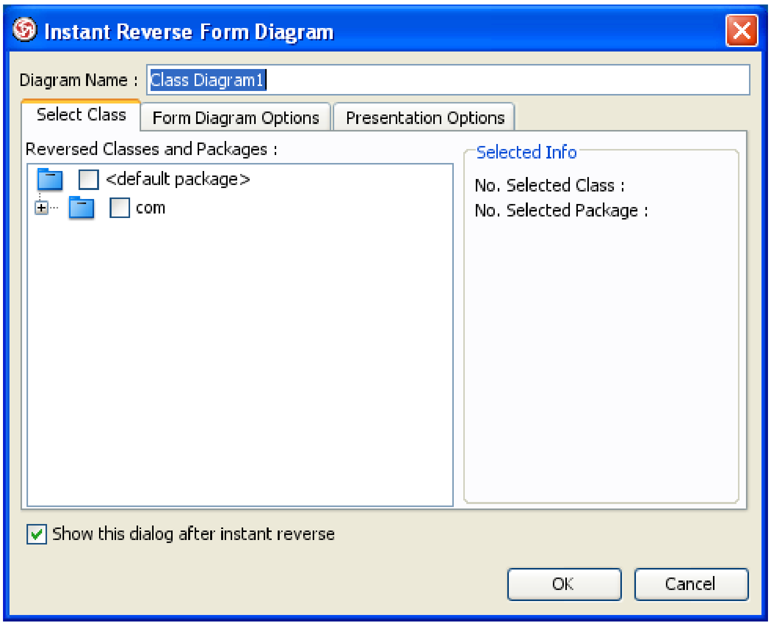

Instant Reverse Form Diagram

Diagram Name : Class Diagram1

Reversed Classes and Packages :

□ <default package>

□ com

Selected Info

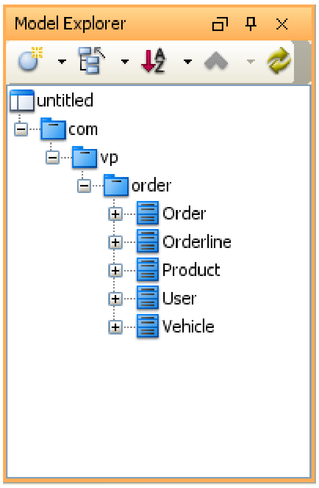
No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

Figure 1-38 *Instant Reverse Form Diagram dialog*

6. Reversed classes appear in **Model Exploer**



Model Explorer

untitled

com

vp

order

Order

Orderline

Product

User

Vehicle

Figure 1-39 *Reversed classes in model exploer*

Instant Reverse XML

1. Select menu **Tools > Instant Reverse > XML...**,

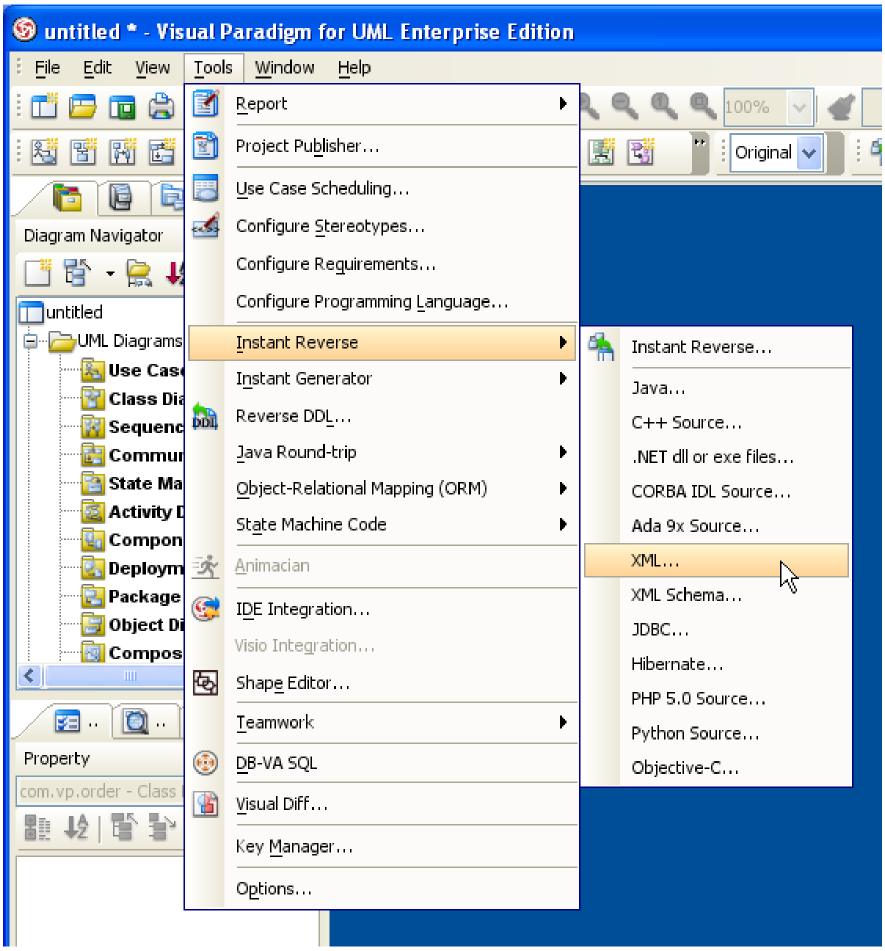
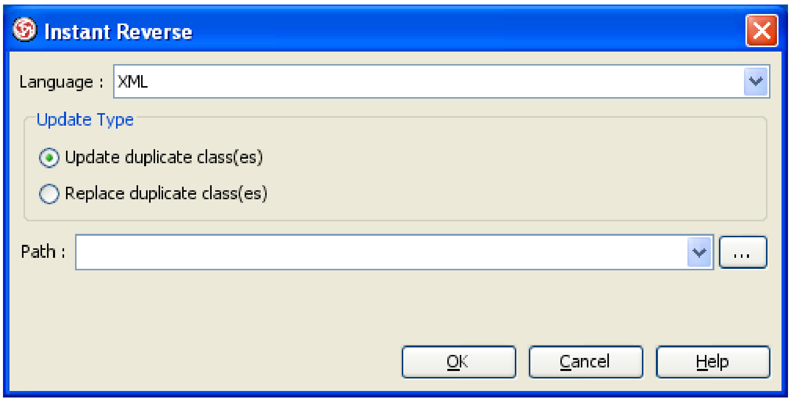


***Figure 1-40** Instant reverse menu*

2. the Instant Reverse dialog box shows.



Language : XML

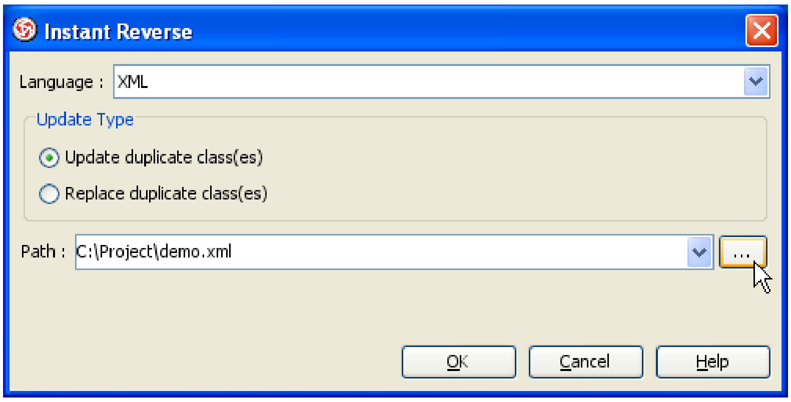
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path :

***Figure 1-41** Instant reverse dialog*



Instant Reverse

Language : XML

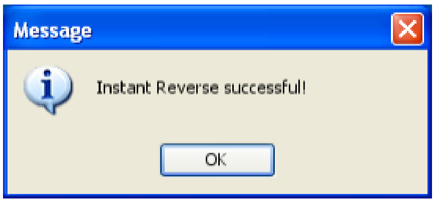
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : C:\Project\demo.xml

Figure 1-42 *Select path*

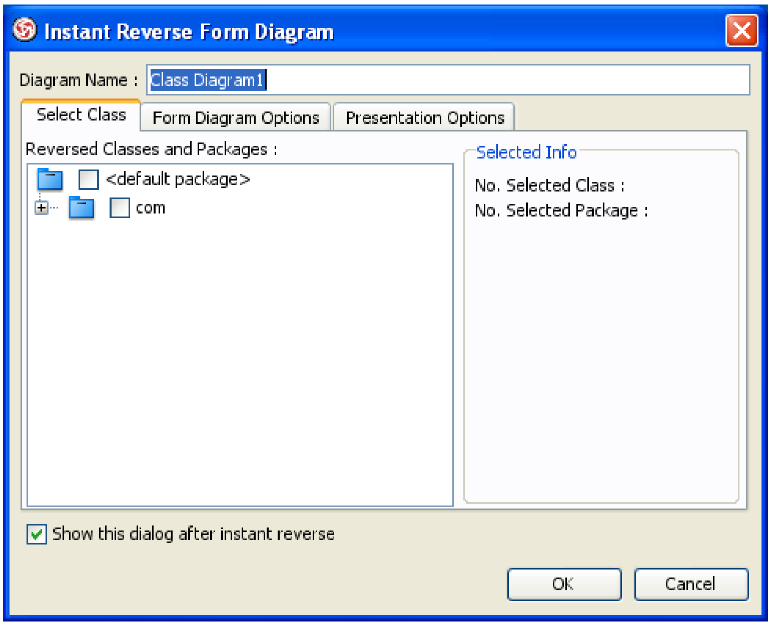


Message

Instant Reverse successful!

Figure 1-43 *Success message*

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.



Instant Reverse Form Diagram

Diagram Name : Class Diagram1

Reversed Classes and Packages :

□ <default package>

□ com

Selected Info

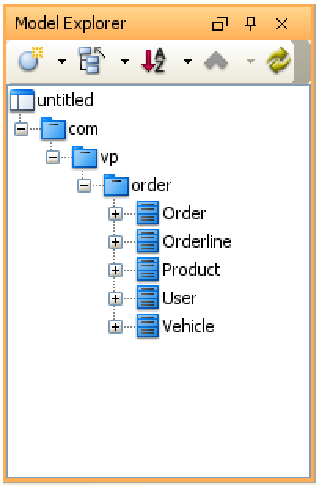
No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

Figure 1-44 *Instant Reverse Form Diagram dialog*

6. Reversed classes appear in **Model Exploer**


Model Explorer

untitled

com

vp

order

Order

Orderline

Product

User

Vehicle

Figure 1-45 *Reversed classes in model exploer*

Instant Reverse XML Schema

1. Select menu **Tools > Instant Reverse > XML Schema...,**

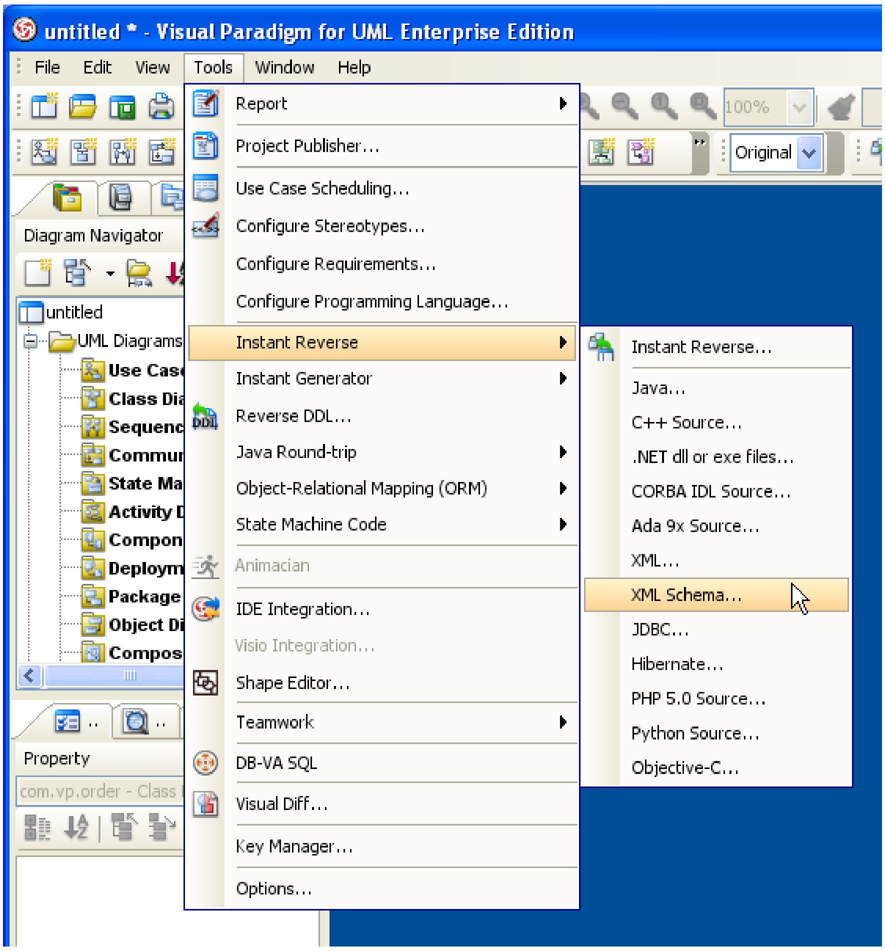
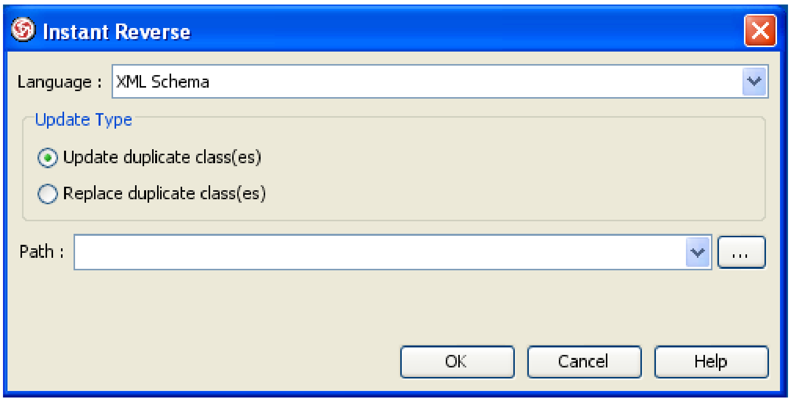


Figure 1-46 *Instant reverse menu*

2. the Instant Reverse dialog box shows.



Language : XML Schema

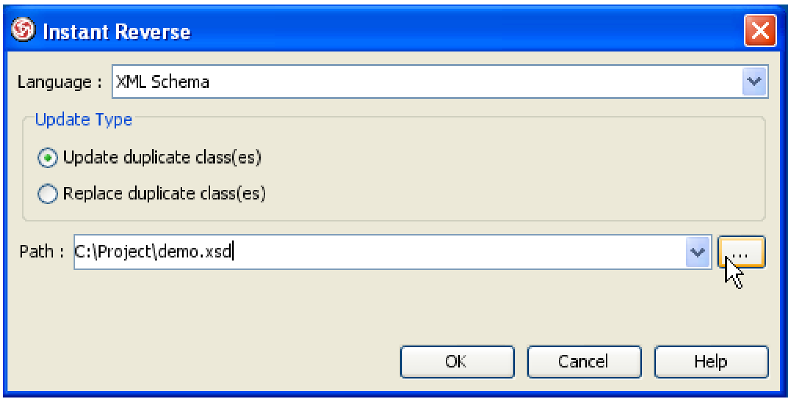
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path :

Figure 1-47 *Instant reverse dialog*



Instant Reverse

Language : XML Schema

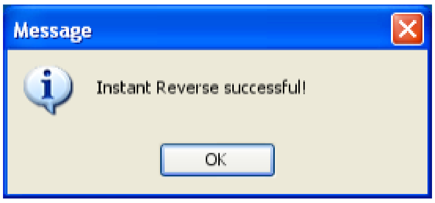
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : C:\Project\demo.xsd

Figure 1-48 *Select path*

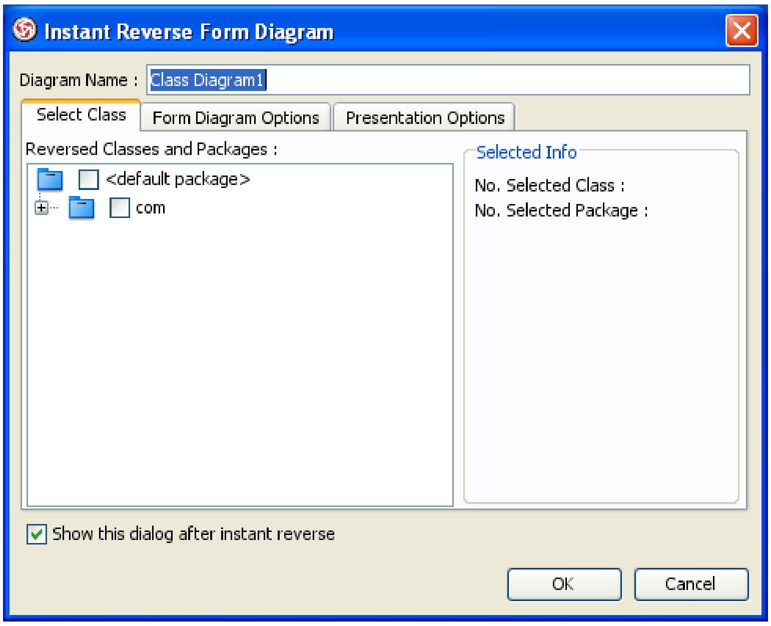


Message

Instant Reverse successful!

Figure 1-49 *Success message*

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.



Instant Reverse Form Diagram

Diagram Name : Class Diagram1

Reversed Classes and Packages :

□ <default package>

□ com

Selected Info

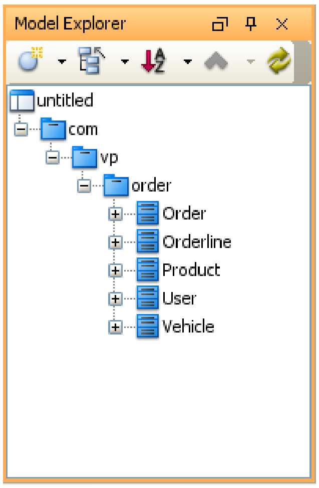
No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

Figure 1-50 *Instant Reverse Form Diagram dialog*

6. Reversed classes appear in **Model Exploer**



Model Explorer

- untitled
  - com
    - vp
      - order
        - Order
        - Orderline
        - Product
        - User
        - Vehicle

__Figure 1-51__ Reversed classes in model exploer

Instant Reverse Database through JDBC

1. Select menu **Tools > Instant Reverse > JDBC...,**

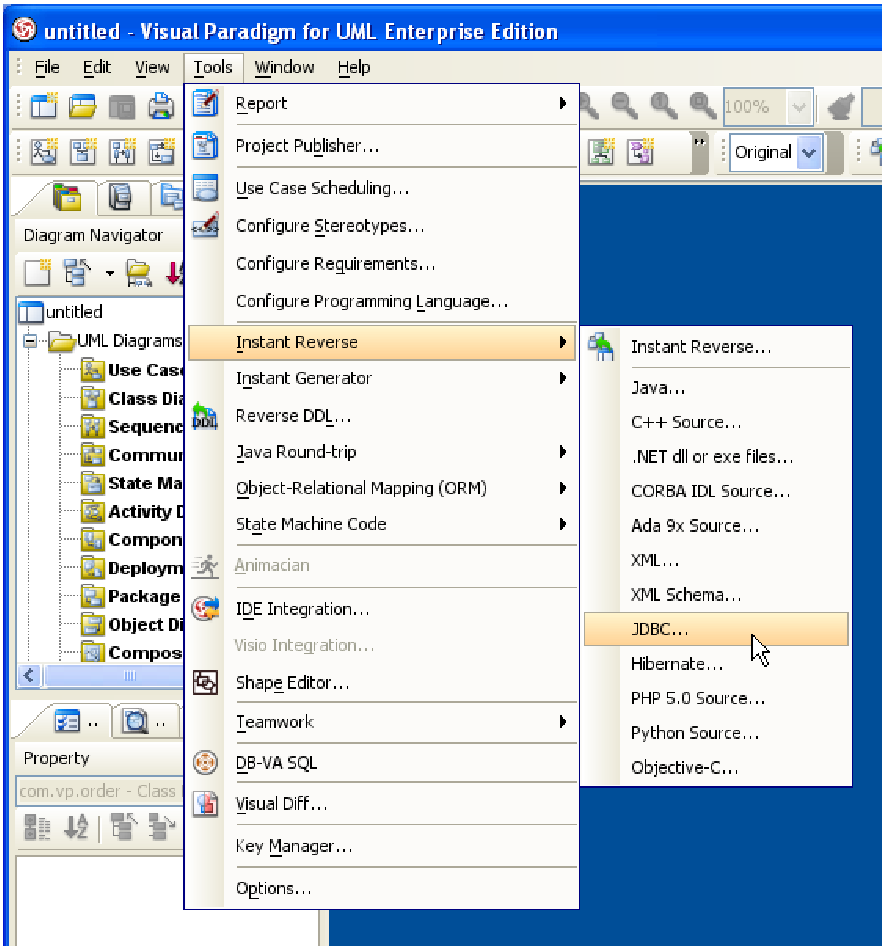
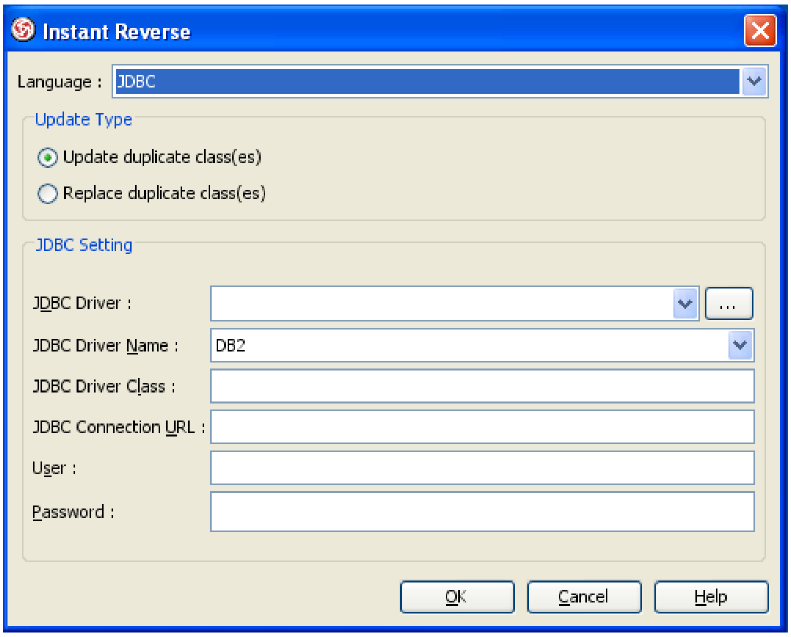


Figure 1-52 *Instant reverse menu*

2. the Instant Reverse dialog box shows.


Instant Reverse

Language : JDBC

Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

JDBC Setting

JDBC Driver :

JDBC Driver Name : DB2

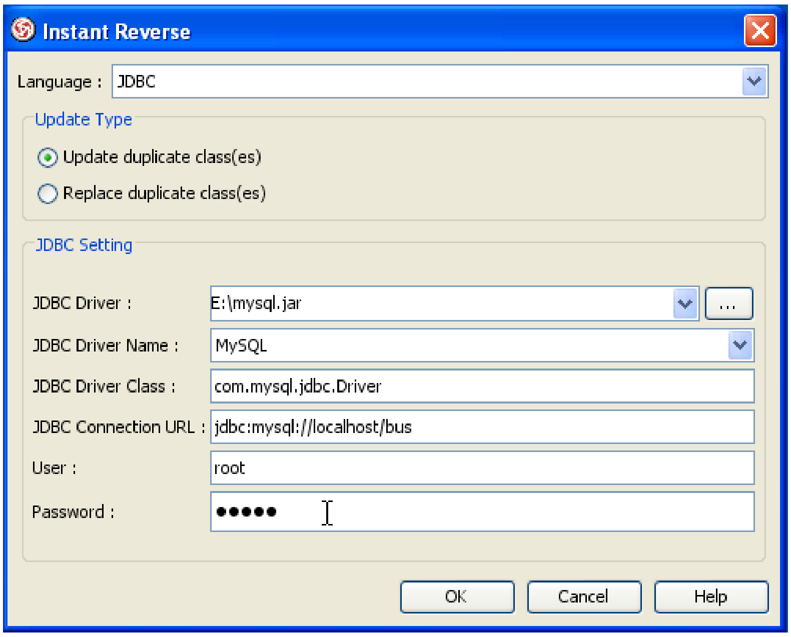
JDBC Driver Class :

JDBC Connection URL :

User :

Password :

Figure 1-53 *Instant reverse dialog*



Language : JDBC

Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

JDBC Setting

JDBC Driver : E:\mysql.jar

JDBC Driver Name : MySQL

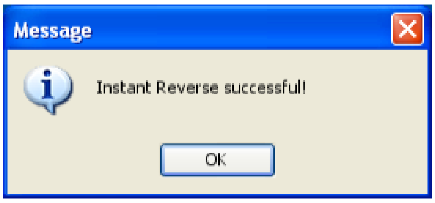
JDBC Driver Class : com.mysql.jdbc.Driver

JDBC Connection URL : jdbc:mysql://localhost/bus

User : root

Password : •••••

***Figure 1-54** JDBC setting*



Instant Reverse successful!

***Figure 1-55** Success message*

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.

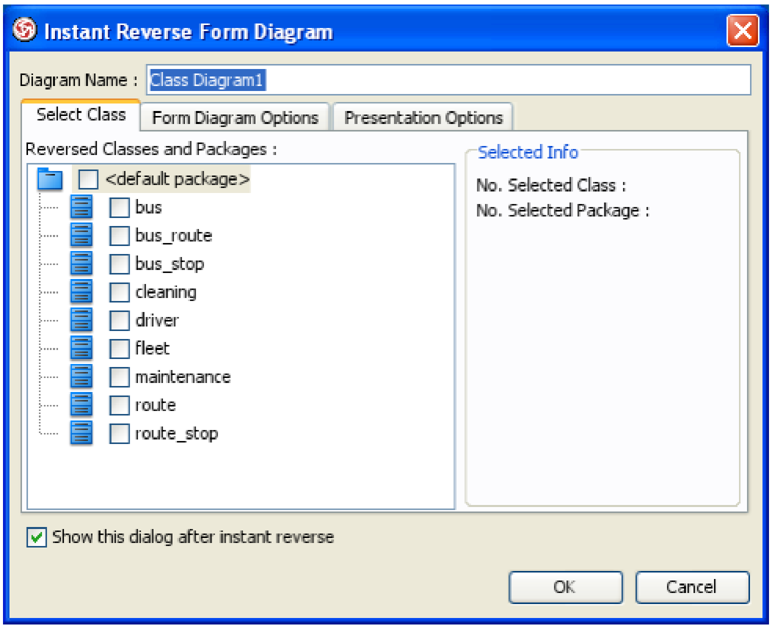


Diagram Name : Class Diagram1

Reversed Classes and Packages :

□ <default package>
- □ bus
- □ bus_route
- □ bus_stop
- □ cleaning
- □ driver
- □ fleet
- □ maintenance
- □ route
- □ route_stop

Selected Info

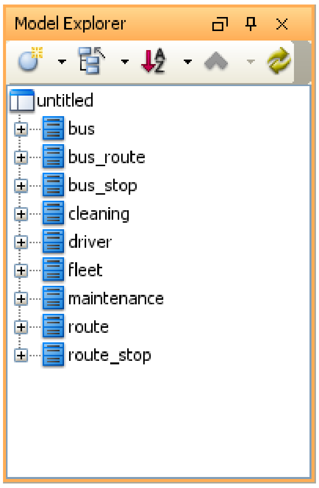
No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

***Figure 1-56** Instant Reverse Form Diagram dialog*

6. Reversed classes appear in **Model Exploer**


Model Explorer

untitled
- bus
- bus_route
- bus_stop
- cleaning
- driver
- fleet
- maintenance
- route
- route_stop

Figure 1-57 *Reversed classes in model exploer*

Instant Reverse Hibernate mapping file

1. Select menu **Tools > Instant Reverse > Hibernate...**,

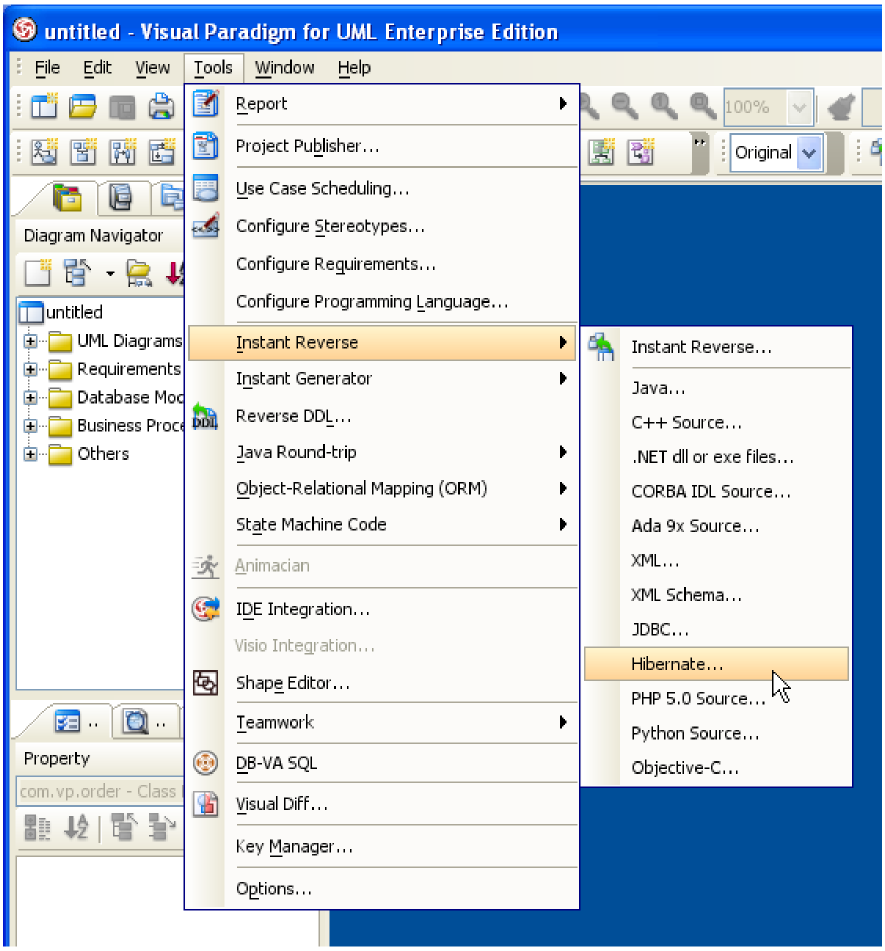
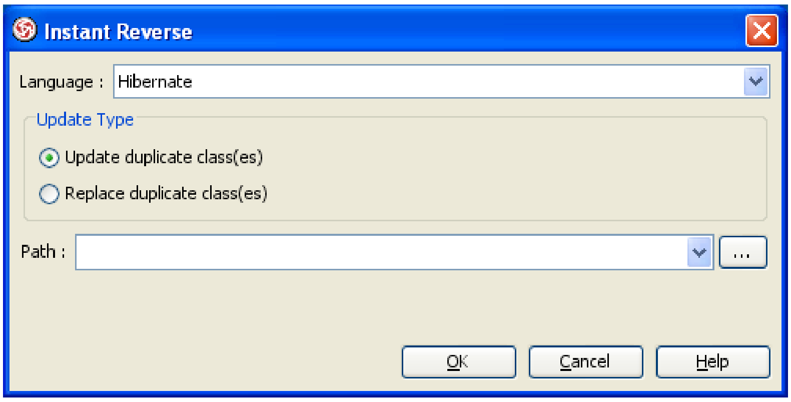


Figure 1-58 Instant reverse menu

2. the Instant Reverse dialog box shows.



Instant Reverse

Language : Hibernate

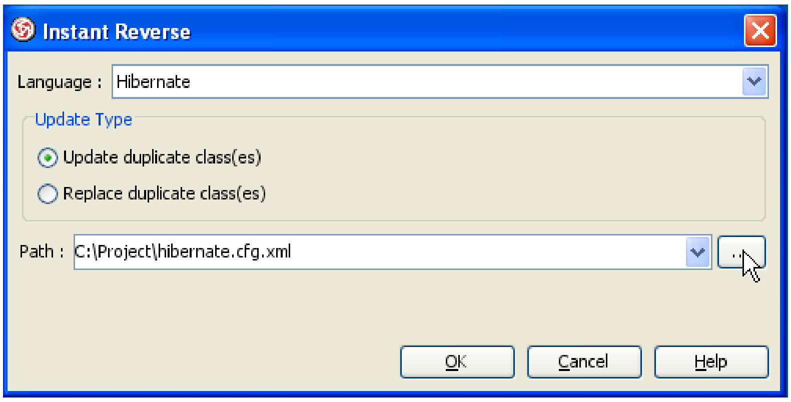
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path :

Figure 1-59 Instant reverse dialog



Instant Reverse

Language : Hibernate

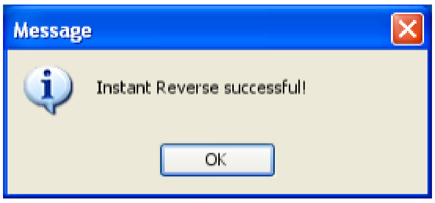
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : C:\Project\hibernate.cfg.xml

Figure 1-60 *Select path*

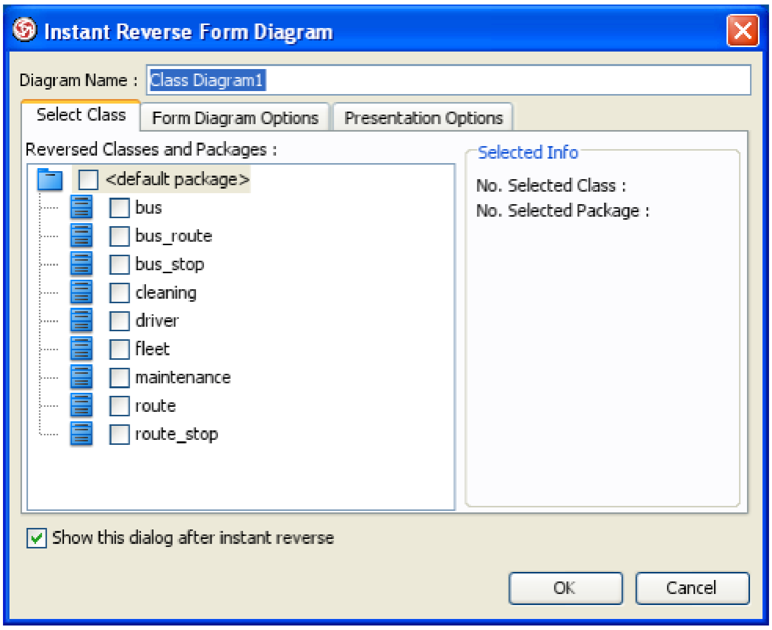


Message

Instant Reverse successful!

Figure 1-61 *Success message*

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.



Instant Reverse Form Diagram

Diagram Name : Class Diagram1

Reversed Classes and Packages :

- □ <default package>
  - □ bus
  - □ bus_route
  - □ bus_stop
  - □ cleaning
  - □ driver
  - □ fleet
  - □ maintenance
  - □ route
  - □ route_stop

Selected Info

No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

Figure 1-62 *Instant Reverse Form Diagram dialog*

6. Reversed classes appear in **Model Exploer**

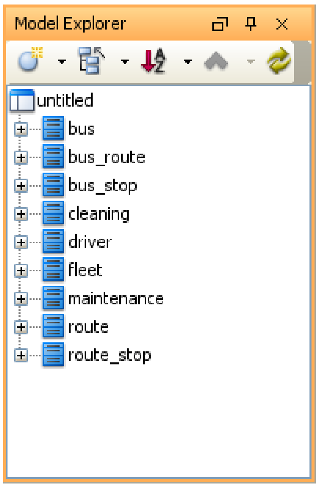


Figure 1-63 *Reversed classes in model exploer*

Instant Reverse PHP 5.0 Source files

1. Select menu **Tools > Instant Reverse > PHP 5.0 Source...,**

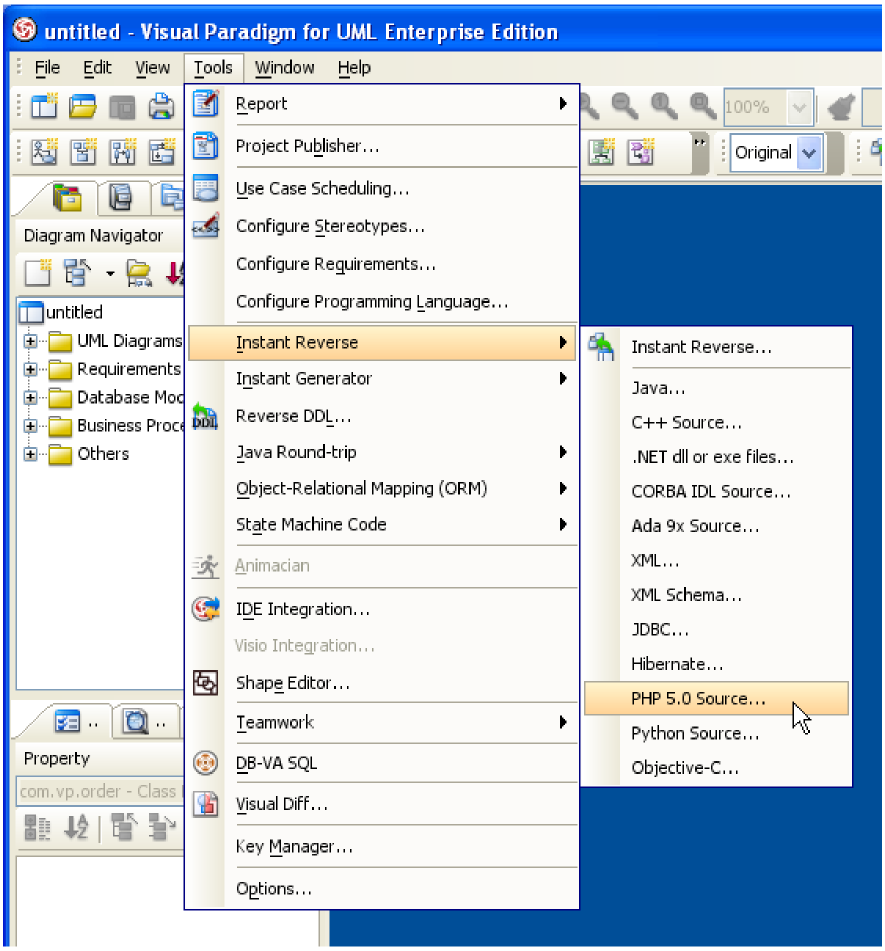
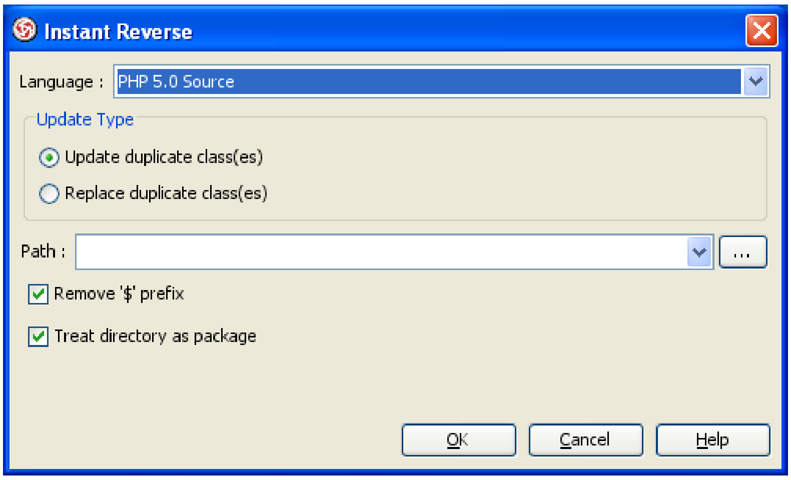


Figure 1-64 Instant reverse menu

2. the Instant Reverse dialog box shows.


Instant Reverse

Language : PHP 5.0 Source

Update Type

● Update duplicate class(es)

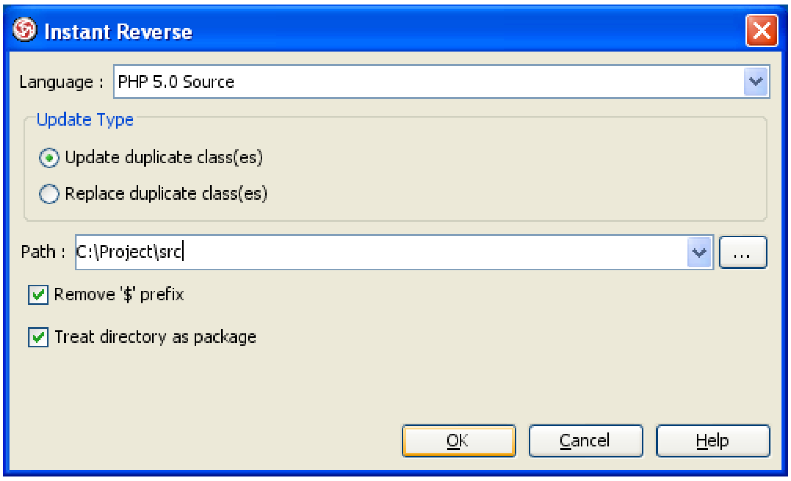
○ Replace duplicate class(es)

Path :

☑ Remove '$' prefix

☑ Treat directory as package

Figure 1-65 Instant reverse dialog



Language : PHP 5.0 Source

Update Type

● Update duplicate class(es)

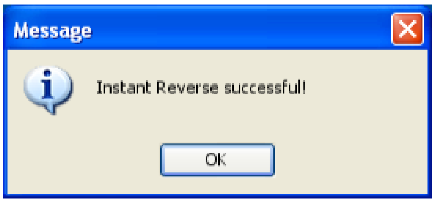
○ Replace duplicate class(es)

Path : C:\Project\src

☑ Remove '$' prefix

☑ Treat directory as package

Figure 1-66 Select path



Instant Reverse successful!

Figure 1-67 Success message

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.

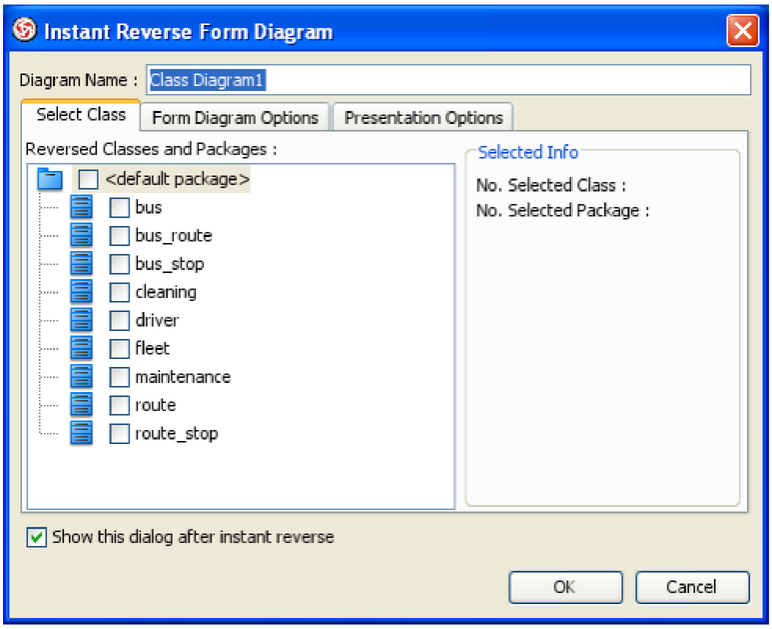


Diagram Name : Class Diagram1

Reversed Classes and Packages :

- □ <default package>
  - □ bus
  - □ bus_route
  - □ bus_stop
  - □ cleaning
  - □ driver
  - □ fleet
  - □ maintenance
  - □ route
  - □ route_stop

Selected Info

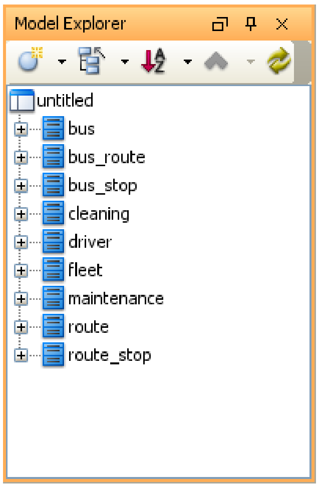
No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

Figure 1-68 Instant Reverse Form Diagram dialog

6. Reversed classes appear in **Model Exploer**



Model Explorer

untitled
- bus
- bus_route
- bus_stop
- cleaning
- driver
- fleet
- maintenance
- route
- route_stop

Figure 1-69 *Reversed classes in model exploer*

Instant Reverse Objective-C

1. Select menu **Tools > Instant Reverse > Objective-C...,**

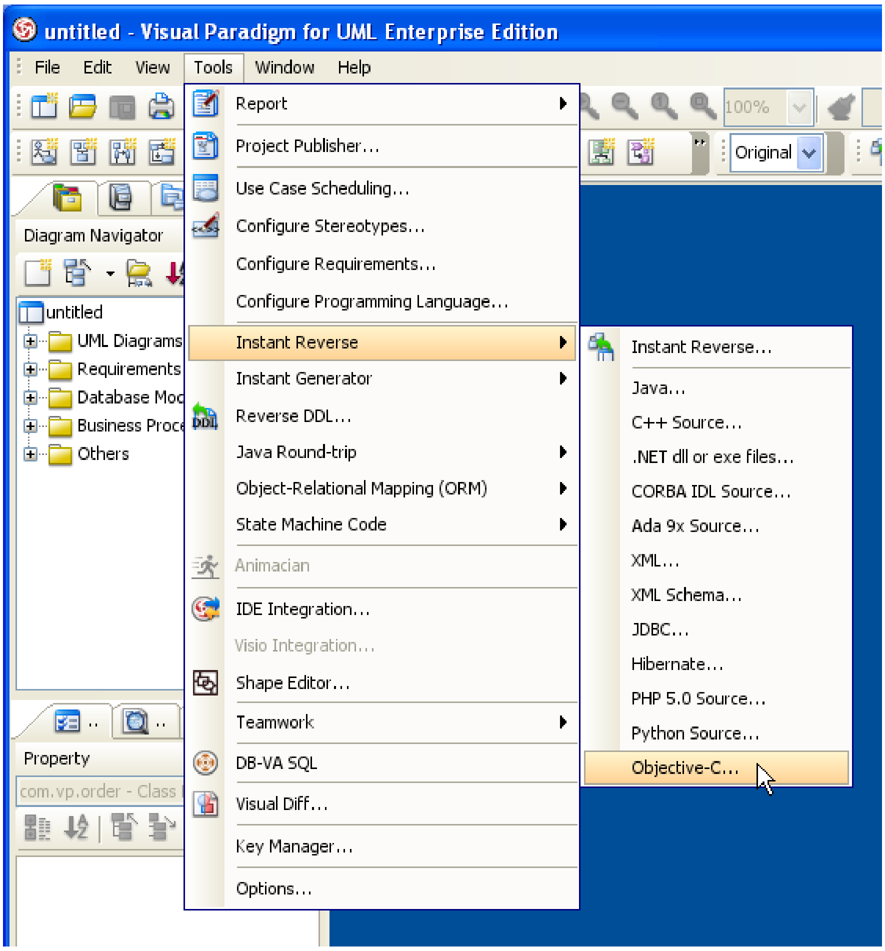
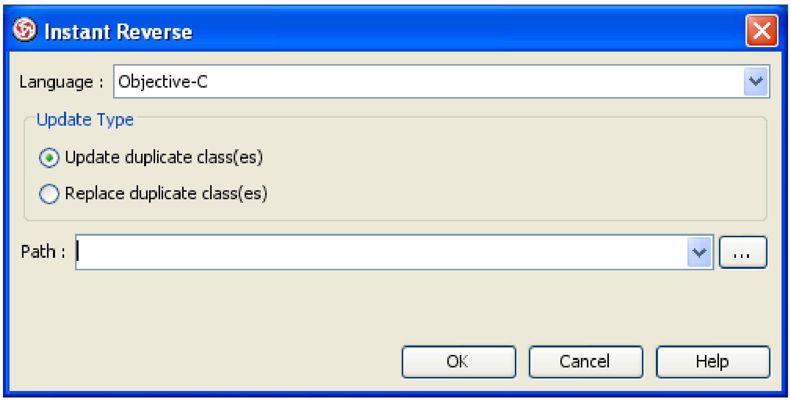


***Figure 1-70** Instant reverse menu*

2. the Instant Reverse dialog box shows.



Language : Objective-C

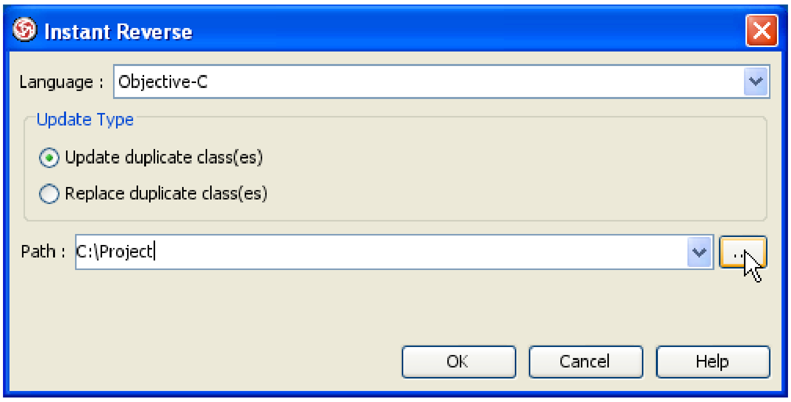
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path :

***Figure 1-71** Instant reverse dialog*



Instant Reverse

Language : Objective-C

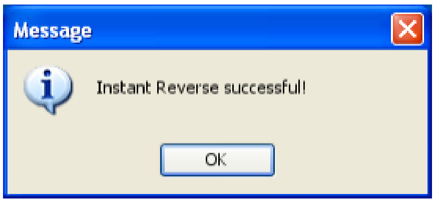
Update Type

● Update duplicate class(es)

○ Replace duplicate class(es)

Path : C:\Project

Figure 1-72 *Select path*

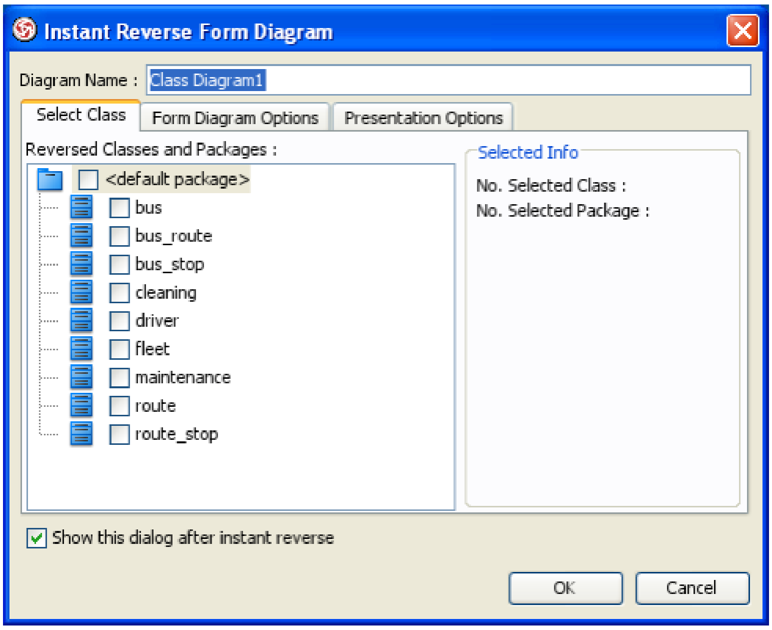


Message

Instant Reverse successful!

Figure 1-73 *Success message*

5. An **Instant Reverse Form Diagram** dialog box then appears. If you want to form new diagram now, select packages and classes then click **OK**, otherwise click **Cancel**. You can still form diagram from model explorer later.



Instant Reverse Form Diagram

Diagram Name : Class Diagram1

Reversed Classes and Packages :

- □ <default package>
  - □ bus
  - □ bus_route
  - □ bus_stop
  - □ cleaning
  - □ driver
  - □ fleet
  - □ maintenance
  - □ route
  - □ route_stop

Selected Info

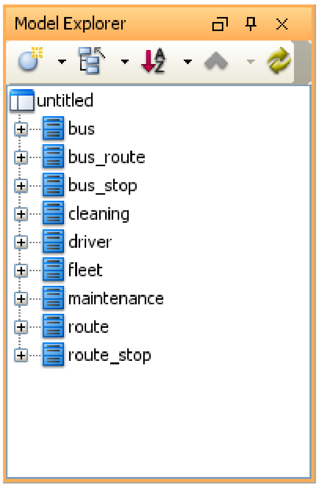
No. Selected Class :

No. Selected Package :

☑ Show this dialog after instant reverse

Figure 1-74 *Instant Reverse Form Diagram dialog*

6. Reversed classes appear in **Model Exploer**



Model Explorer

untitled

- bus
- bus_route
- bus_stop
- cleaning
- driver
- fleet
- maintenance
- route
- route_stop

Figure 1-75 *Reversed classes in model exploer*

Instant Generate C# source code

Instant Generate PHP source code

Instant Generate ODL source code

Instant Generate IDL source code

Instant Generate C++ source code

Instant Generate Delphi source code

Instant Generate Perl source code

Instant Generate Python source code

Instant Generate Ruby

Customizing Code Generation

Instant generator allows you to generate programming source code from class models. Basically, the content of the generated code follows the common coding convention of the programming language. There are also advanced options for you to configure some of the specific settings in forming the code, like the use of prefix for attributes and parameters.

Although the built-in way of generating source code can satisfy most of the general needs, you may want to define something more specific. For example, you may need to print a copyright statement at the beginning of the code file, which is not a kind of customization being supported by Instant generator.

Fortunately, the way of how source code will be generated is handled by Apache Velocity engine, a templating engine, and the templates being used are fully opened for customization. In the following sections, we will explain how to customize a template to make the generated code follow your requirement.

Preparation

Text Editor

The customization of template require the use of a text editor. A suggestion of text editor would be JEdit, a powerful, yet free of charge text editor. More important, it provides syntax highlighting, which helps you read the template content easier by styling different parts with different colors. You can download JEdit from its official site at:

http://www.jedit.org/

To install JEdit:

1. Run the downloaded setup program.
2. Press **Next >** in the Welcome screen.
3. Accept the license agreement and press **Next >**.
4. Select the installation folder and press **Next >**.
5. Select the components to be installed. The editing of template does not require the API documentation, macros and batch files. Depending on your interest, you may decide install them or not.
6. Select the Start Menu folder and press **Next >**.
7. Select whether to create desktop icons and quick launch icon and press **Next >**.
8. Confirm by pressing **Install**.

Development Environment

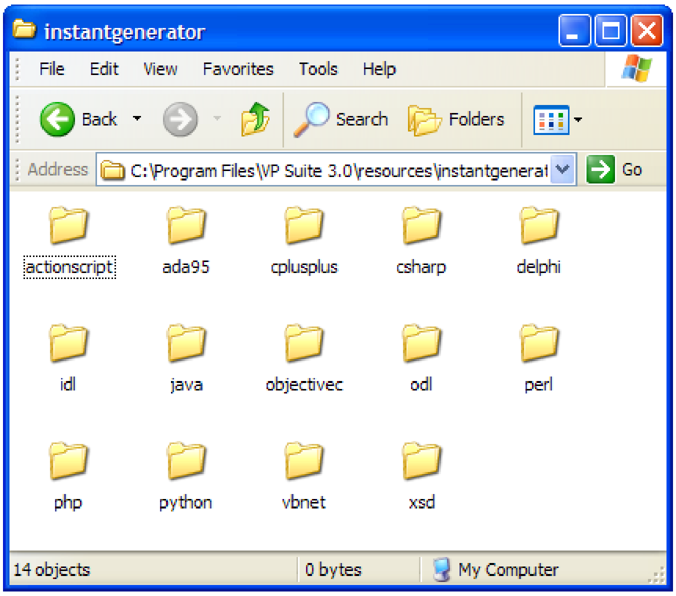
The template files are put under the **resources/instantgenerator** folder of VP Suite installation directory. It is absolutely alright to edit those files directly. However, it is recommended to setup your own development environment, copy the template files to there to perform further editing. There are two reasons for separating the development environment from VP Suite:

- Avoid the unexpected template removal by un-installing the VP Suite.
- Avoid accidental file replacement by running product updates.

To setup your development environment:

1.Create a folder as working directory.

2.Explore **%VP-Suite-Installation-Directory%/resources/instantgenerator**.



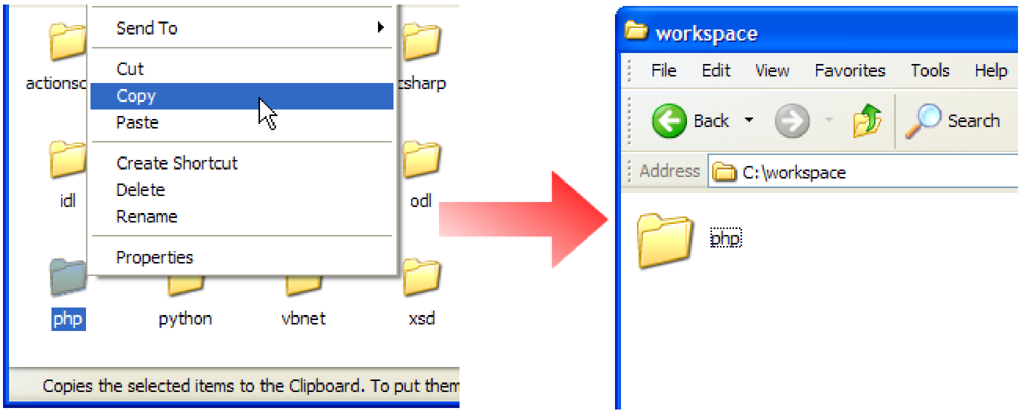
Address: C:\Program Files\VP Suite 3.0\resources\instantgenerat

actionscript, ada95, cplusplus, csharp, delphi, idl, java, objectivec, odl, perl, php, python, vbnet, xsd

14 objects | 0 bytes | My Computer

***Figure 2-164** Expore instantgenerator folder*

You will see numerous sub-folders having the programming language as its name. Each of them contains the templates that we need to edit.
3.Copy the folder(s) that contain the template files, and paste it to the working directory.

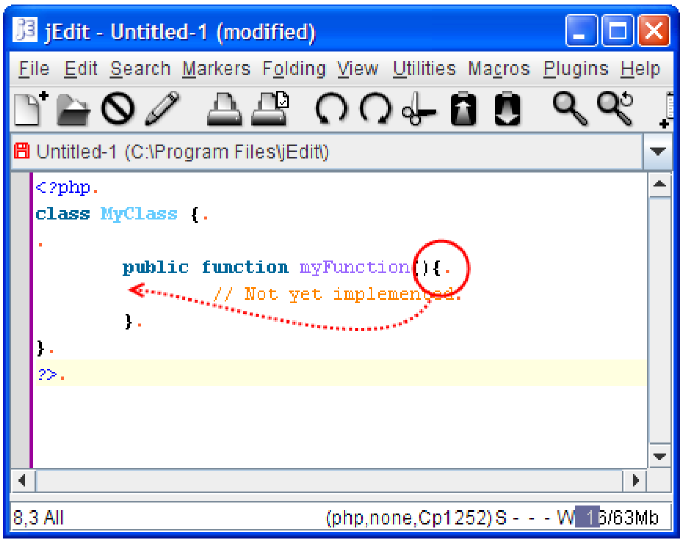


Copies the selected items to the Clipboard. To put them

Figure 2-165 *Copy folder*

Customizing Template

By having the text editor and the development environment ready, it&rsquo;s time to get your hand dirty with editing the template. As mentioned before, Instant generator adopted the [Apache Velocity](#) engine in generating source code. For those who are interested in knowing how to write templates, please read Velocity&rsquo;s Users&rsquo; guide at:
http://velocity.apache.org/engine/releases/velocity-1.5/user-guide.html .
The following example demonstrate how to edit the PHP code generation template to reposition the brace of operation blocks to a new line.

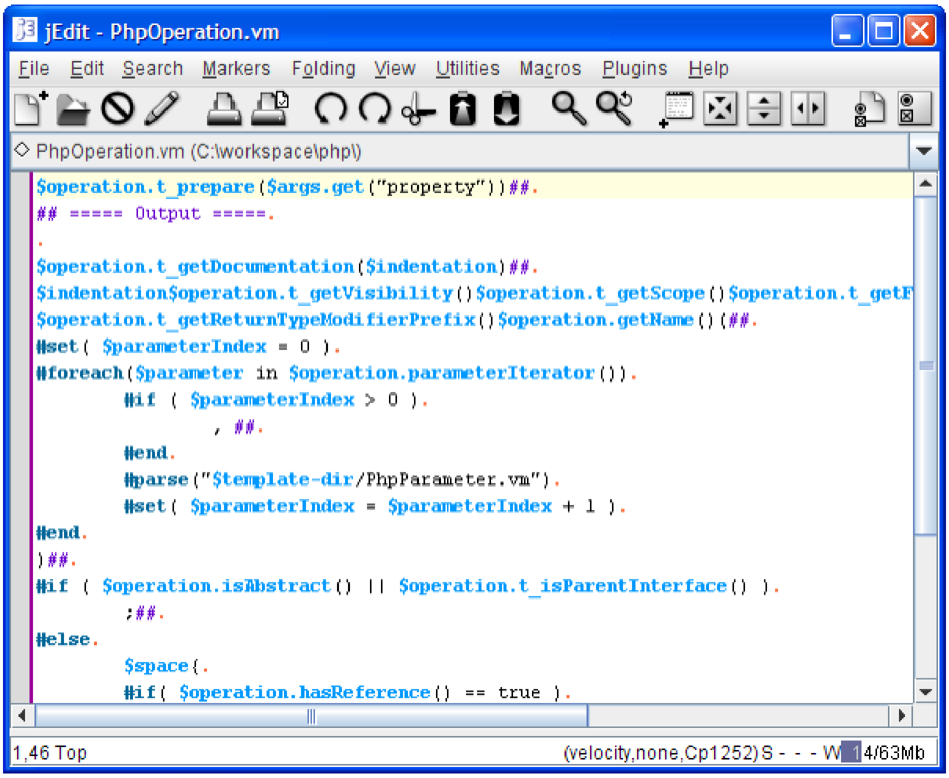

jEdit - Untitled-1 (modified)
File
Edit
Search
Markers
Folding
View
Utilities
Macros
Plugins
Help
Untitled-1 (C:\Program Files\jEdit\)

```
<?php
class MyClass {

        public function myFunction(){
            // Not yet implemented
        }
}
?>
```

8,3 All (php,none,Cp1252) S - - - W 16/63Mb

Figure 2-166 *Example*

1. Start JEdit.
2. Open **%working-directory%/php/PhpOperation.vm** in JEdit.

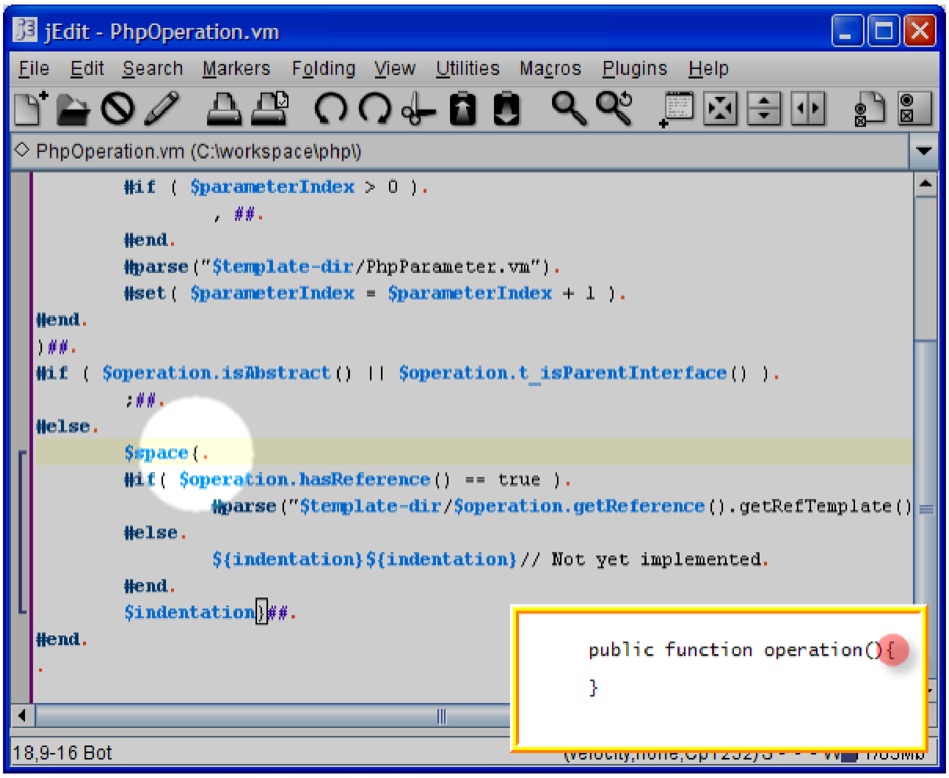

jEdit - PhpOperation.vm
File Edit Search Markers Folding View Utilities Macros Plugins Help
PhpOperation.vm (C:\workspace\php\)
1,46 Top
(velocity,none,Cp1252) S - - - W 14/63Mb

```
$operation.t_prepare($args.get("property"))##
## ===== Output =====

$operation.t_getDocumentation($indentation)##
$indentation$operation.t_getVisibility()$operation.t_getScope()$operation.t_getF
$operation.t_getReturnTypeModifierPrefix()$operation.getName()(##
#set( $parameterIndex = 0 )
#foreach($parameter in $operation.parameterIterator())
        #if ( $parameterIndex > 0 )
                , ##
        #end
        #parse("$template-dir/PhpParameter.vm")
        #set( $parameterIndex = $parameterIndex + 1 )
#end
)##
#if ( $operation.isAbstract() || $operation.t_isParentInterface() )
        ;##
#else
        $space{
        #if( $operation.hasReference() == true )
```

Figure 2-167 Open PhpOperation.vm in JEdit

At the beginning, you may find the template a bit complex. But once you start working on it for a while, you&rsquo;ll find the syntax easy to understanding. In fact, it just compose of common programming construct like if-then-else statements, foreach and variables that programmers should find intuitive.

3. Search for the open brace *{*

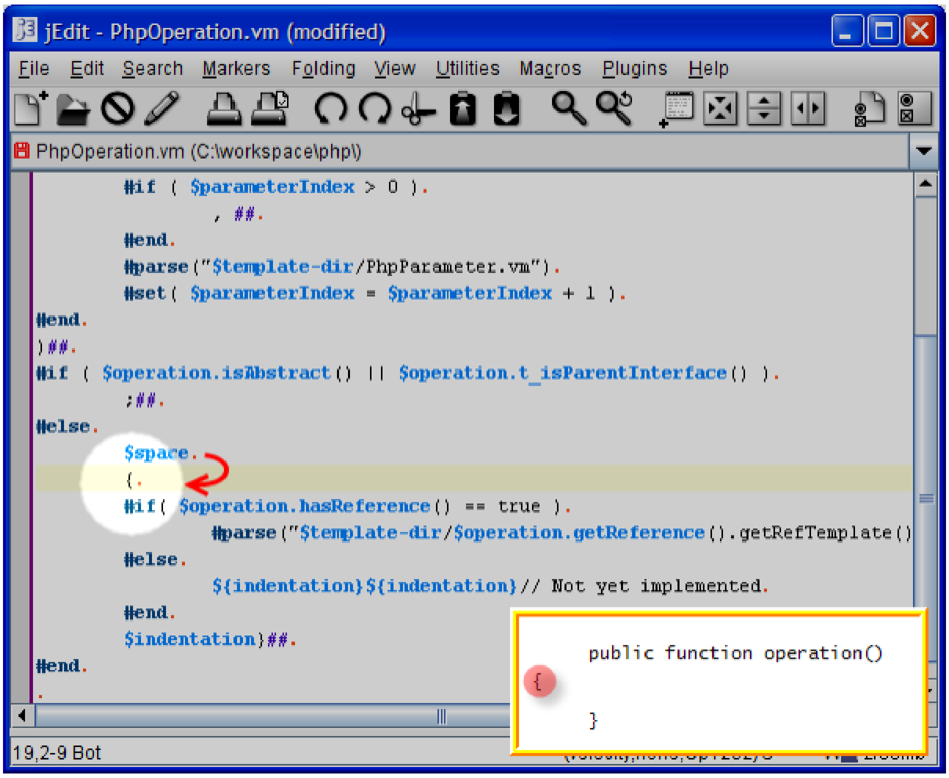

jEdit - PhpOperation.vm
File Edit Search Markers Folding View Utilities Macros Plugins Help
PhpOperation.vm (C:\workspace\php\)
18,9-16 Bot

```
        #if ( $parameterIndex > 0 )
                , ##
        #end
        #parse("$template-dir/PhpParameter.vm")
        #set( $parameterIndex = $parameterIndex + 1 )
#end
)##
#if ( $operation.isAbstract() || $operation.t_isParentInterface() )
        ;##
#else
        $space{
        #if( $operation.hasReference() == true )
                #parse("$template-dir/$operation.getReference().getRefTemplate()
        #else
                ${indentation}${indentation}// Not yet implemented
        #end
        $indentation}##
#end
```

```
public function operation(){
}
```

Figure 2-168 Search for the open branch

4. Insert line breaks in front of { to cause Instant generator insert line breaks in front of {.

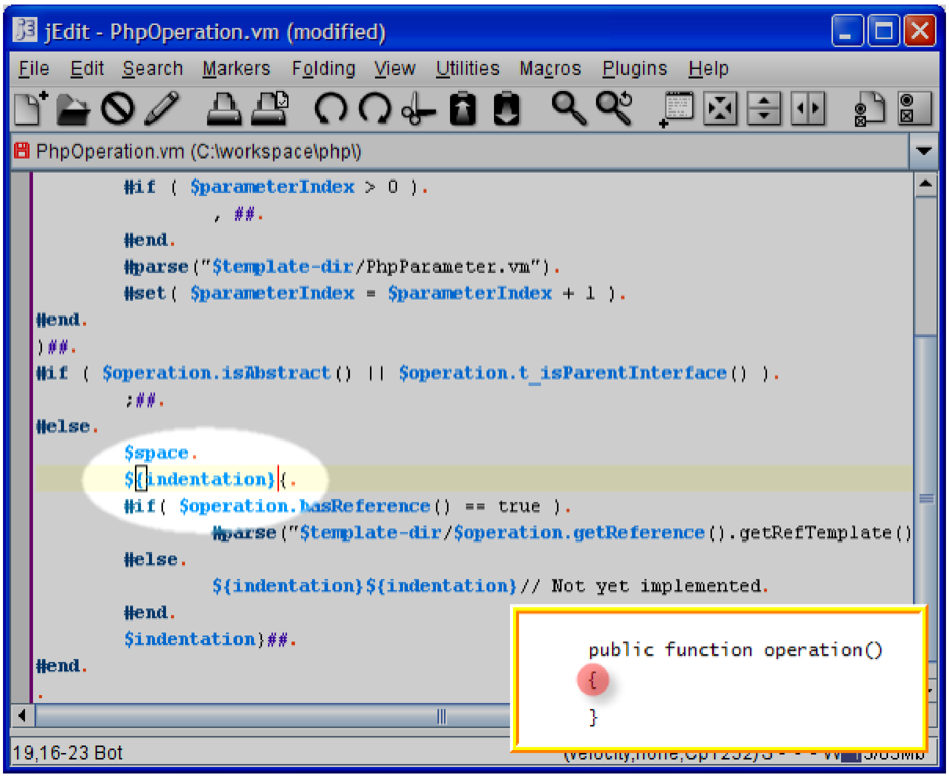

jEdit - PhpOperation.vm (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
PhpOperation.vm (C:\workspace\php\)
19,2-9 Bot

```
        #if ( $parameterIndex > 0 ).
                , ##.
        #end.
        #parse("$template-dir/PhpParameter.vm").
        #set( $parameterIndex = $parameterIndex + 1 ).
#end.
)##.
#if ( $operation.isAbstract() || $operation.t_isParentInterface() ).
        ;##.
#else.
        $space.
        {.
        #if( $operation.hasReference() == true ).
                #parse("$template-dir/$operation.getReference().getRefTemplate()
        #else.
                ${indentation}${indentation}// Not yet implemented.
        #end.
        $indentation}##.
#end.
.
```

```
public function operation()
{

}
```

Figure 2-169 Insert line breaks

5. Add a variable $indentation to indicate the need of printing indentation before the open brace.


jEdit - PhpOperation.vm (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
PhpOperation.vm (C:\workspace\php\)
19,16-23 Bot

```
        #if ( $parameterIndex > 0 ).
                , ##.
        #end.
        #parse("$template-dir/PhpParameter.vm").
        #set( $parameterIndex = $parameterIndex + 1 ).
#end.
)##.
#if ( $operation.isAbstract() || $operation.t_isParentInterface() ).
        ;##.
#else.
        $space.
        ${indentation}{.
        #if( $operation.hasReference() == true ).
                #parse("$template-dir/$operation.getReference().getRefTemplate()
        #else.
                ${indentation}${indentation}// Not yet implemented.
        #end.
        $indentation}##.
#end.
.
```

```
public function operation()
{
}
```

Figure 2-170 Add variable

6. Save the file.

Generate Code with the Customized Template

This is perhaps the most excited moment - Run the Instant generator with the customized template. Things start from running the Instant generator.

1. From the application (e.g. VP-UML), select **Tools > Instant Generator > PHP**

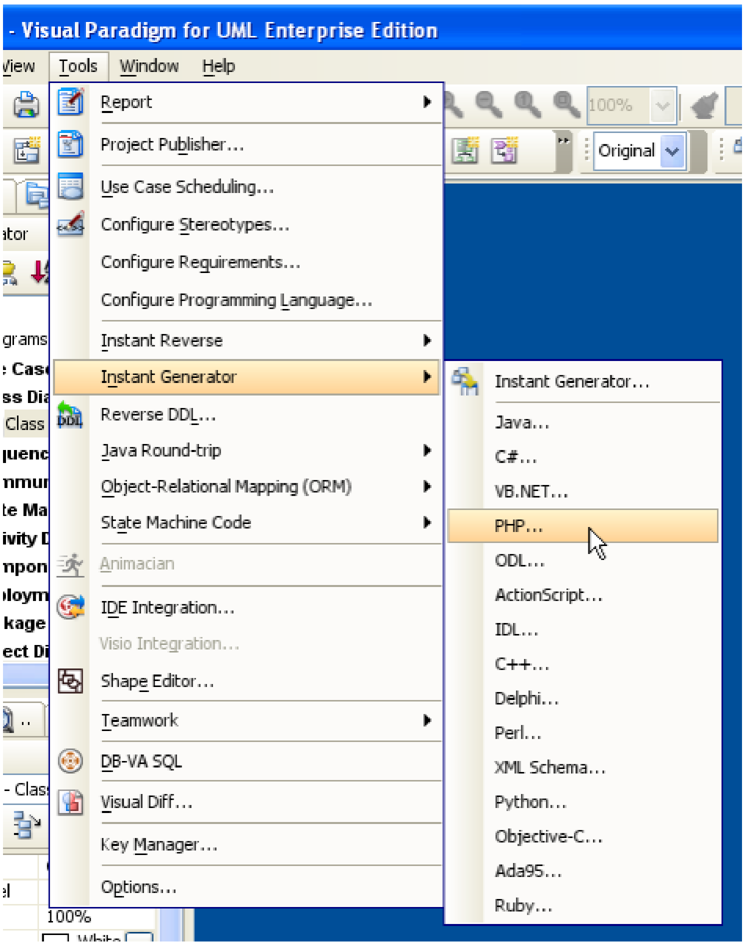
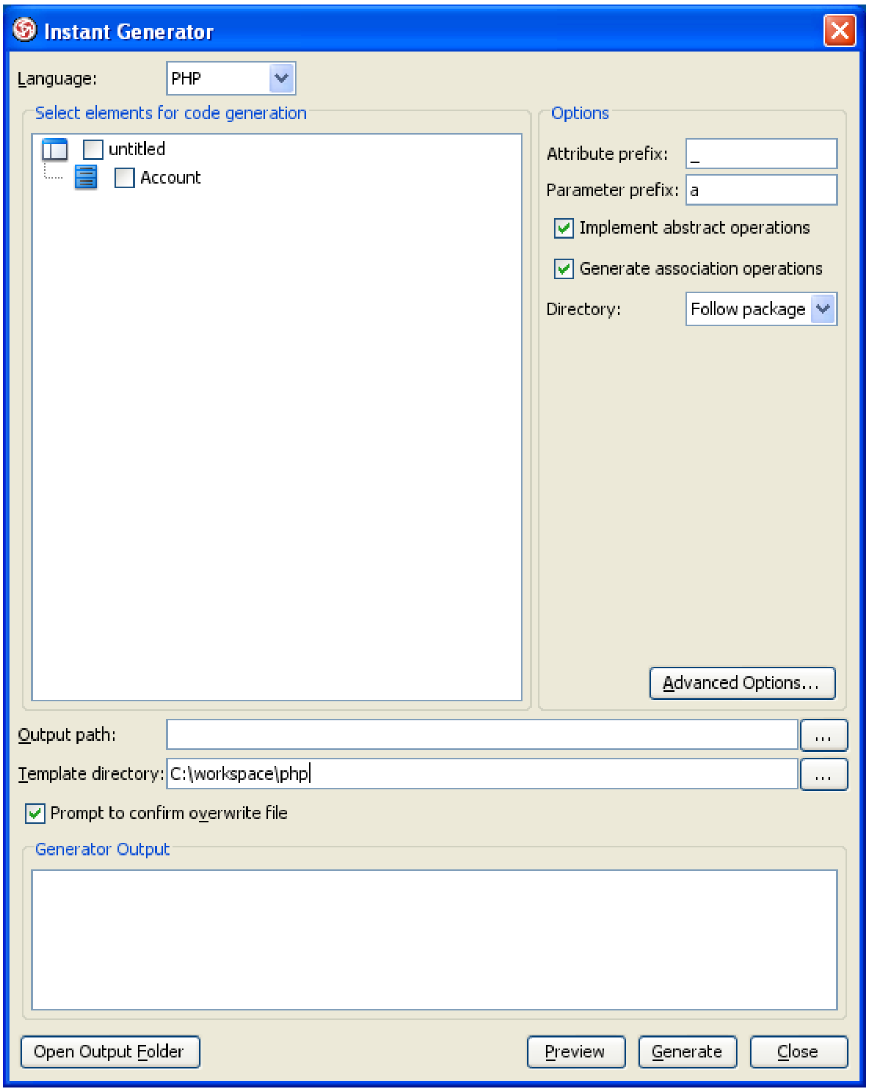


Figure 2-171 *Instant generator menu*

2. From the **Instant Generator** dialog box, specify the Template directory. In this example, it is C:\workspace\php


Instant Generator

Language: PHP

Select elements for code generation

- □ untitled
  - □ Account

Options

Attribute prefix: _

Parameter prefix: a

☑ Implement abstract operations

☑ Generate association operations

Directory: Follow package

Output path:

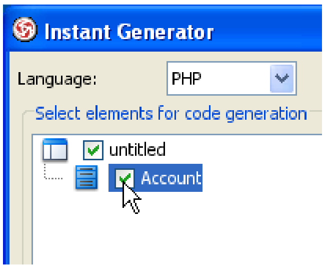
Template directory: C:\workspace\php

☑ Prompt to confirm overwrite file

Generator Output

Figure 2-172 *Instant generator dialog box*

3. Select classes to generate code.


Instant Generator

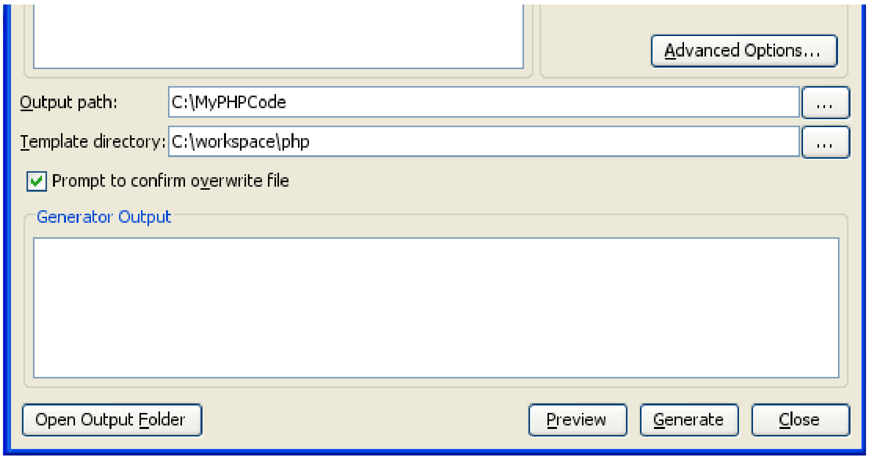
Language: PHP

Select elements for code generation

- ☑ untitled
  - ☑ Account

Figure 2-173 *Select class*

4. Enter the Output path.



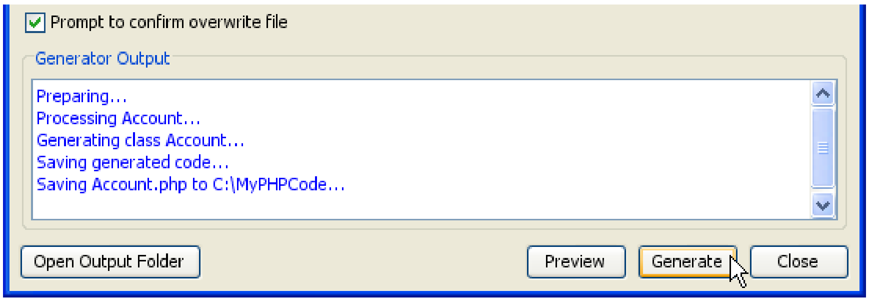
Output path: C:\MyPHPCode

Template directory: C:\workspace\php

☑ Prompt to confirm overwrite file

Figure 2-174 Enter output path

5. Click **Generate** to start generating the PHP code from the class diagram.

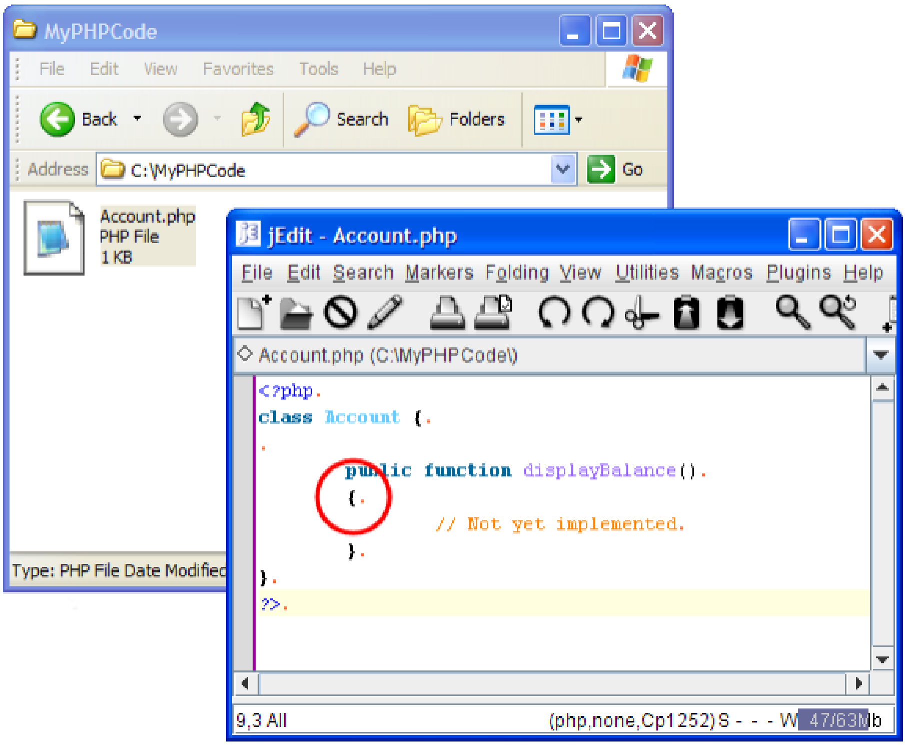


☑ Prompt to confirm overwrite file

```
Preparing...
Processing Account...
Generating class Account...
Saving generated code...
Saving Account.php to C:\MyPHPCode...
```

Figure 2-175 Generate

6. The code is generated to the output path. As you can see, the brace is reposition-ed based on the updated template.


MyPHPCode
File Edit View Favorites Tools Help
Back
Search
Folders
Address
Go
Account.php
PHP File
1 KB
Type: PHP File Date Modified
jEdit - Account.php
File Edit Search Markers Folding View Utilities Macros Plugins Help
Account.php (C:\MyPHPCode\)
9,3 All
(php,none,Cp1252) S - - - W 47/63Mb

Address: C:\MyPHPCode

```
<?php
class Account {

        public function displayBalance()
        {
                // Not yet implemented.
        }
}
?>
```

Figure 2-176 Generated result

Appendix A - API

The following table lists the available API calls for retrieving data from models.

Class	API	Return Value
Object	getDocumentation()	String
	getName()	String
Association	associationClassArray()	AssociationClass[]
	associationClassAt(int)	AssociationClass
	associationClassCount()	int
	associationClassIterator()	Iterator
	getFromElement()	Object
	getFromEnd()	AssociationEnd
	getToElement()	Object
	getToEnd()	AssociationEnd
	isAbstract()	boolean
	isDerived()	boolean
	isLeaf()	boolean
AssociationClass	getFromElement()	Object
	getToElement()	Object
AssociationEnd	getAggregationKind()	String
	getMultiplicity()	String
	getNavigable()	int
	getReferencedAttribute()	Attribute
	getTypeModifier()	String
	getVisibility()	String
	isOrdered()	boolean
	isProvideGetterMethod()	boolean
	isProvideSetterMethod()	boolean
	isUnique()	boolean
Attribute	getInitialValue()	String
	getMultiplicity()	String
	getScope()	String
	getTemplateTypeBindInfo()	TemplateTypeBindInfo
	getType()	Object
	getTypeModifier()	String
	getVisibility()	String
	isAbstract()	boolean
	isHasGetter()	boolean
	isHasSetter()	boolean
	isOrdered()	boolean
	isUnique()	boolean

	isVisible()	boolean
	getDeclarativeAttribute()	String
	getStorage()	int
	isConst()	boolean
	isFinal()	boolean
	isNew()	boolean
	isReadonly()	boolean
	isShadow()	boolean
	isTransient()	boolean
	isVolatile()	boolean
	isWithEvent()	boolean
Class	associationArray()	Association[]
	associationAt(int)	Association
	associationClassArray()	AssociationClass[]
	associationClassAt(int)	AssociationClass
	associationClassCount()	int
	associationClassIterator()	Iterator
	associationCount()	int
	associationIterator()	Iterator
	attributeArray()	Attribute[]
	attributeAt(int)	Attribute
	attributeCount()	int
	attributeIterator()	Iterator
	containmentClassArray()	Class[]
	containmentClassAt(int)	Class
	containmentClassCount()	int
	containmentClassIterator()	Iterator
	generalizationArray()	Generalization[]
	generalizationAt(int)	Generalization
	generalizationCount()	int
	generalizationIterator()	Iterator
	getVisibility()	String
	isAbstract()	boolean
	isActive()	boolean
	isLeaf()	boolean
	isRoot()	boolean
	operationArray()	Operation[]
	operationAt(int)	Operation

	operationCount()	int
	operationIterator()	Iterator
	realizationArray()	Realization[]
	realizationAt(int)	Realization
	realizationCount()	int
	realizationIterator()	Iterator
	templateParameterArray()	TemplateParameter[]
	templateParameterAt(int)	TemplateParameter
	templateParameterCount()	int
	templateParameterIterator()	Iterator
	getDeclarativeAttribute()	String
	getManageType()	int
	isFinal()	boolean
	isInterface()	boolean
	isNew()	boolean
	isNotInheritable()	boolean
	isSealed()	boolean
	isShadow()	boolean
	isStatic()	boolean
	isStereotypeInterface()	boolean
DataType	templateParameterArray()	TemplateParameter[]
	templateParameterAt(int)	TemplateParameter
	templateParameterCount()	int
	templateParameterIterator()	Iterator
Generalization	getFromElement()	Object
	getTemplateTypeBindInfo()	TemplateTypeBindInfo
	getToElement()	Object
	isSubstitutable()	boolean
Operation	getReturnType()	Object
	getReturnTypeModifier()	String
	getScope()	String
	getTemplateTypeBindInfo()	TemplateTypeBindInfo
	getVisibility()	String
	isAbstract()	boolean
	isQuery()	boolean
	isVisible()	boolean
	parameterArray()	Parameter[]
	parameterAt(int)	Parameter

	parameterCount()	int
	parameterIterator()	Iterator
	raisedExceptionArray()	Object[]
	raisedExceptionAt(int)	Object
	raisedExceptionCount()	int
	raisedExceptionIterator()	Iterator
	templateParameterArray()	TemplateParameter[]
	templateParameterAt(int)	TemplateParameter
	templateParameterCount()	int
	templateParameterIterator()	Iterator
	getAlias()	String
	getCharset()	int
	getDeclarativeAttribute()	String
	getDllName()	String
	getMethodKind()	int
	getOperatorType()	int
	getProcedureName()	String
	isConst()	
	isDeclare()	boolean
	isDelegate()	boolean
	isExtern()	
	isFinal()	boolean
	isFriend()	boolean
	isInline()	boolean
	isNative()	boolean
	isNew()	boolean
	isNotOverridable()	boolean
	isOverload()	boolean
	isOverridable()	boolean
	isOverride()	boolean
	isSealed()	boolean
	isShadow()	boolean
	isSynchronized()	boolean
	isUnsafe()	boolean
	isVirtual()	boolean
Package	classArray()	Class[]
	classAt(int)	Class
	classCount()	int

	classIterator()	Iterator
	packageArray()	Package[]
	packageAt(int)	Package
	packageCount()	int
	packageIterator()	Iterator
	templateParameterArray()	TemplateParameter[]
	templateParameterAt(int)	TemplateParameter
	templateParameterCount()	int
	templateParameterIterator()	Iterator
Parameter	getDefaultValue()	String
	getDirection()	String
	getTemplateTypeBindInfo()	TemplateTypeBindInfo
	getType()	Object
	getTypeModifier()	String
	getDeclarativeAttribute()	String
	isConst()	boolean
	isFinal()	boolean
	isOptional()	boolean
	isParamArray()	boolean
	isParams()	boolean
Realization	getFromElement()	Object
	getMapping()	String
	getTemplateTypeBindInfo()	TemplateTypeBindInfo
	getToElement()	Object
Stereotype		
TaggedValue	getType()	int
	getValue()	Object
TemplateParameter	getDefaultValue()	String
	templateTypeBindInfoArray()	TemplateTypeBindInfo[]
	templateTypeBindInfoAt(int)	TemplateTypeBindInfo
	templateTypeBindInfoCount()	int
	templateTypeBindInfoIterator()	Iterator
	typeArray()	Object[]
	typeAt(int)	Object
	typeCount()	int
	typeIterator()	Iterator
	typeModifierArray()	String[]
	typeModifierAt(int)	String

	typeModifierCount()	int
	typeModifierIterator()	Iterator
TemplateTypeBindDetails	getArguments()	TemplateTypeBindInfo
	getWildcard()	int
TemplateTypeBindInfo	detailsArray()	TemplateTypeBindDetails[]
	detailsAt(int)	TemplateTypeBindDetails
	detailsCount()	int
	detailsIterator()	Iterator
	getBindedType()	Object
	getTypeModifier()	String

Appendix B - Velocity Syntax

The following lists the syntax that of statements that can be used in the template.

```
## ===== If =====
#if(...)
...
#end
## ===== If-then-Else =====
#if(...)
...
#else
...
#end
## ===== For-each =====
#foreach
...
#end
## ===== Continue with the template defined in (...) at the point where the call is made =====
#parse(...)
#set(...)
## ===== Comment =====
## ...
## ===== Comment =====
#* ... *#
## ===== Variable=====
${...}
```

Generate Java Code for active diagram

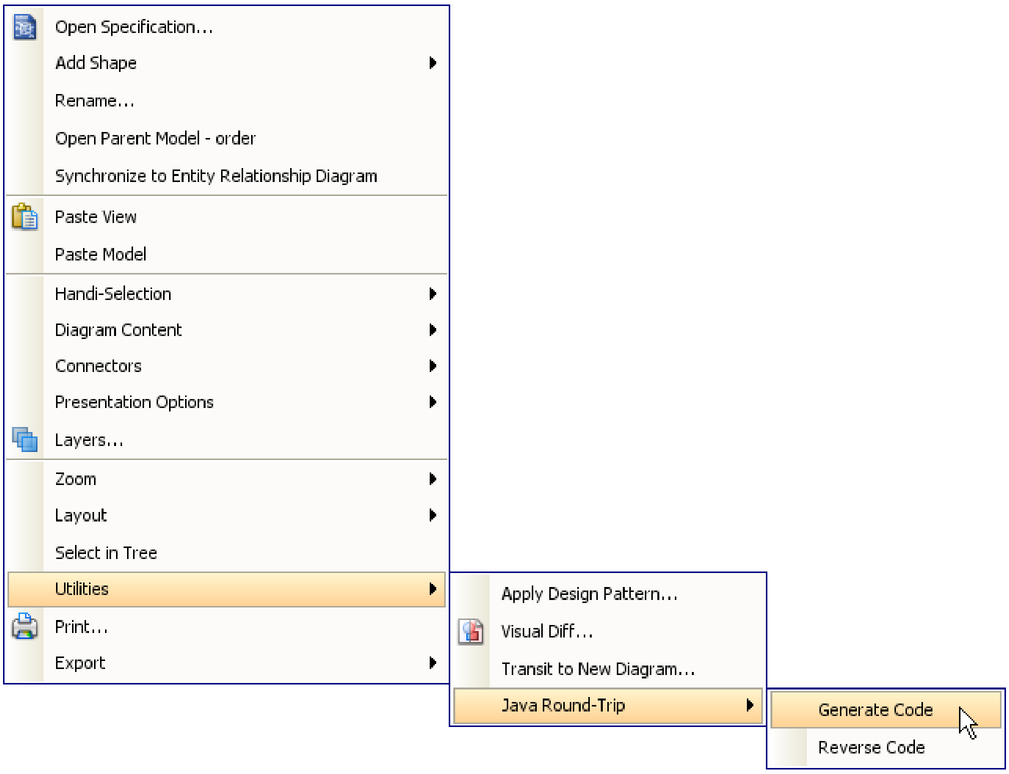
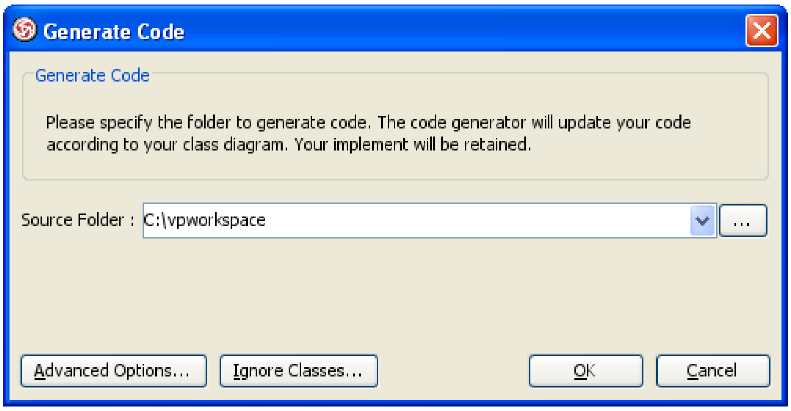


Figure 3-2 *Popup menu*

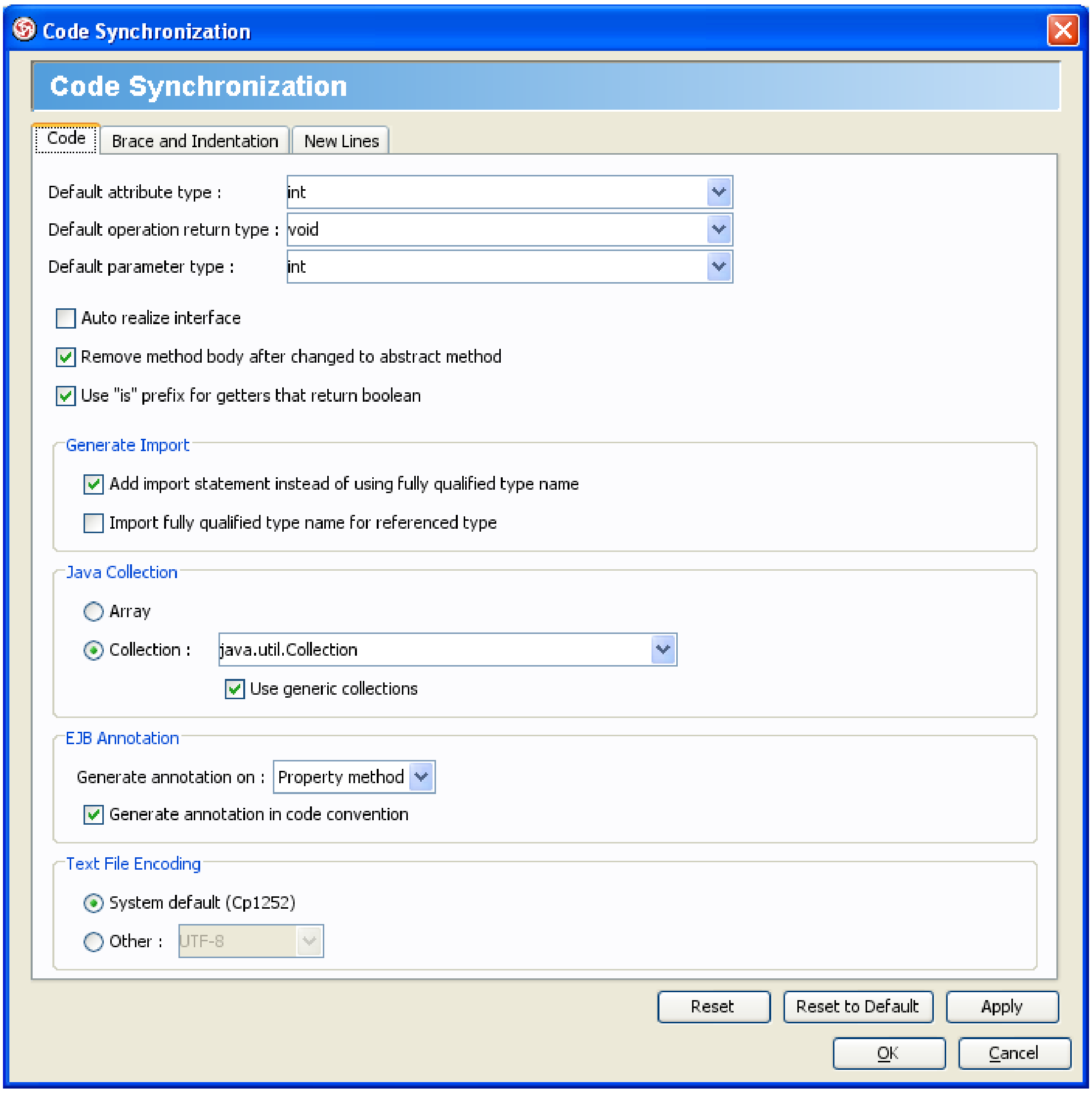

Generate Code

Generate Code

Please specify the folder to generate code. The code generator will update your code according to your class diagram. Your implement will be retained.

Source Folder : C:\vpworkspace

Figure 3-3 *Generate code dialog*


Code Synchronization

Code Synchronization

Default attribute type : int

Default operation return type : void

Default parameter type : int

☐ Auto realize interface

☑ Remove method body after changed to abstract method

☑ Use "is" prefix for getters that return boolean

Generate Import

☑ Add import statement instead of using fully qualified type name

☐ Import fully qualified type name for referenced type

Java Collection

○ Array

● Collection : java.util.Collection

☑ Use generic collections

EJB Annotation

Generate annotation on : Property method

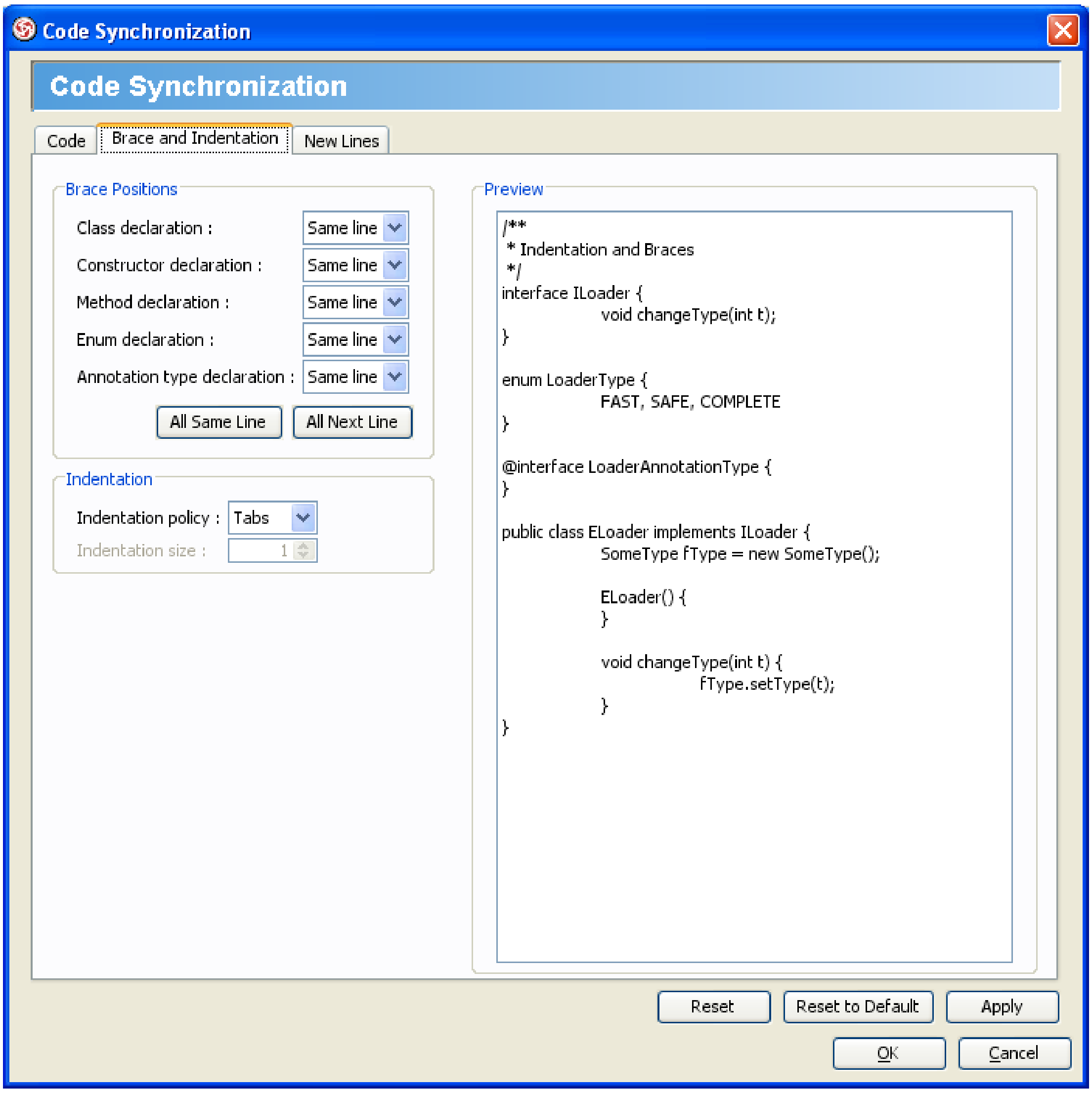
☑ Generate annotation in code convention

Text File Encoding

● System default (Cp1252)

○ Other : UTF-8

Figure 3-4 *Advanced option Code tab*

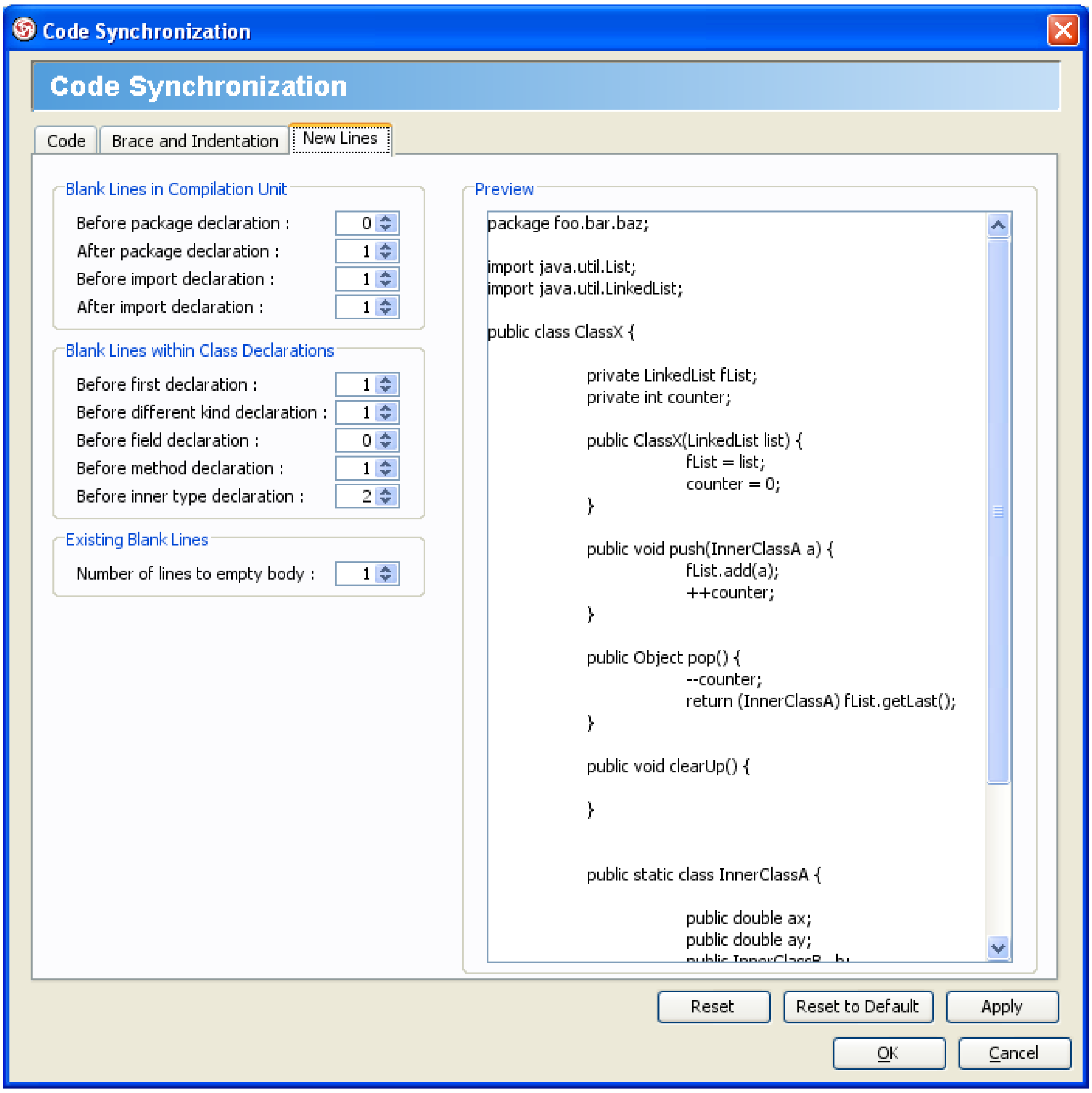


Class declaration : Same line
Constructor declaration : Same line
Method declaration : Same line
Enum declaration : Same line
Annotation type declaration : Same line

Indentation policy : Tabs
Indentation size : 1

```
/**
 * Indentation and Braces
 */
interface ILoader {
		void changeType(int t);
}

enum LoaderType {
		FAST, SAFE, COMPLETE
}

@interface LoaderAnnotationType {
}

public class ELoader implements ILoader {
		SomeType fType = new SomeType();

		ELoader() {
		}

		void changeType(int t) {
				fType.setType(t);
		}
}
```

Figure 3-5 *Advanced option Brace and Indentation tab*



Blank Lines in Compilation Unit

Before package declaration : 0
After package declaration : 1
Before import declaration : 1
After import declaration : 1

Blank Lines within Class Declarations

Before first declaration : 1
Before different kind declaration : 1
Before field declaration : 0
Before method declaration : 1
Before inner type declaration : 2

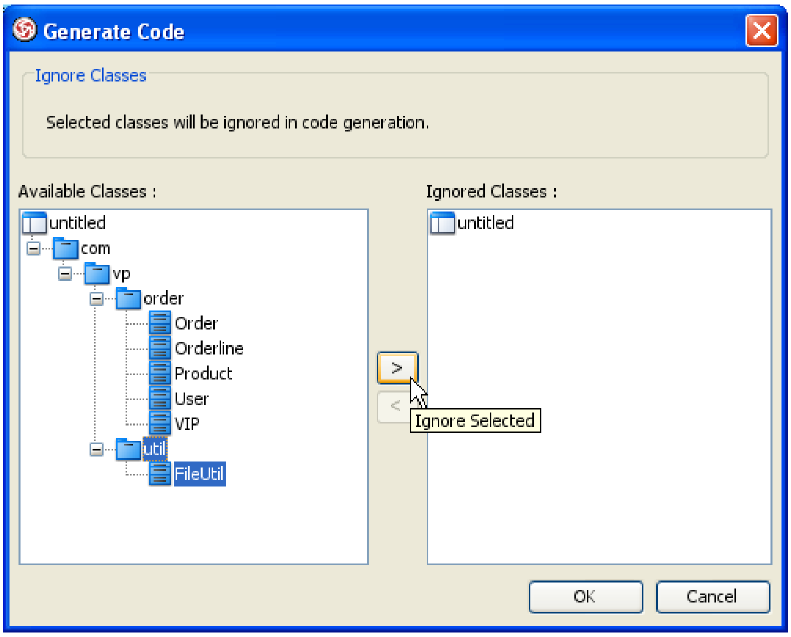
Existing Blank Lines

Number of lines to empty body : 1

Preview

```
package foo.bar.baz;

import java.util.List;
import java.util.LinkedList;

public class ClassX {

        private LinkedList fList;
        private int counter;

        public ClassX(LinkedList list) {
                fList = list;
                counter = 0;
        }

        public void push(InnerClassA a) {
                fList.add(a);
                ++counter;
        }

        public Object pop() {
                --counter;
                return (InnerClassA) fList.getLast();
        }

        public void clearUp() {

        }


        public static class InnerClassA {

                public double ax;
                public double ay;
```

Figure 3-6 Advanced option New Lines tab



Ignore Classes

Selected classes will be ignored in code generation.

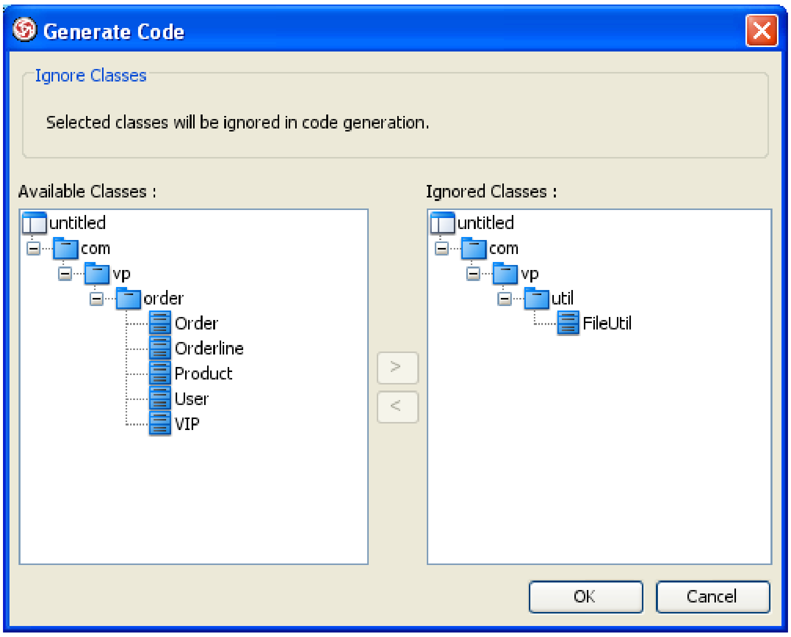
Available Classes :

- untitled
  - com
    - vp
      - order
        - Order
        - Orderline
        - Product
        - User
        - VIP
      - util
        - FileUtil

Ignored Classes :

- untitled

Figure 3-7 Selecting ignore classes



Selected classes will be ignored in code generation.

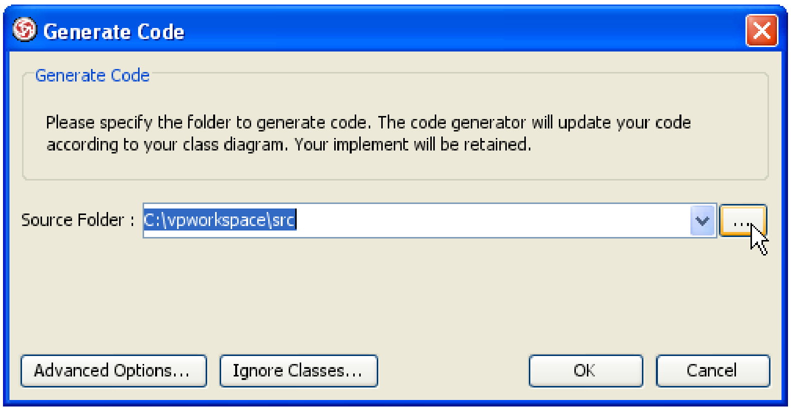
Available Classes :

- untitled
  - com
    - vp
      - order
        - Order
        - Orderline
        - Product
        - User
        - VIP

Ignored Classes :

- untitled
  - com
    - vp
      - util
        - FileUtil

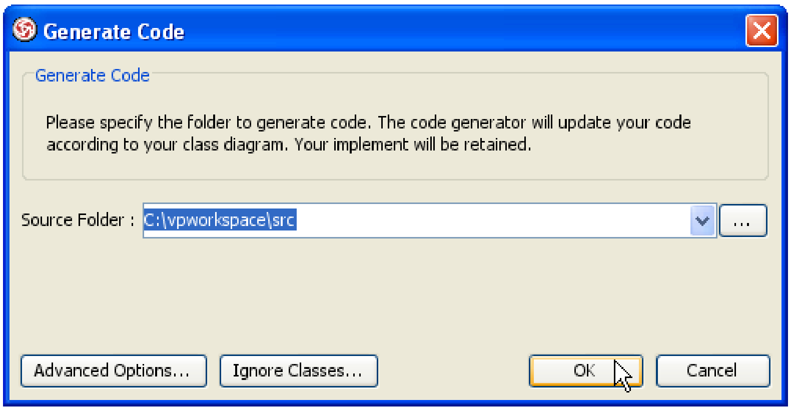
Figure 3-8 *Selected ignore classes*



Please specify the folder to generate code. The code generator will update your code according to your class diagram. Your implement will be retained.

Source Folder : C:\vpworkspace\src

Figure 3-9 *Select source folder*



Please specify the folder to generate code. The code generator will update your code according to your class diagram. Your implement will be retained.

Source Folder : C:\vpworkspace\src

Figure 3-10 *Generate*

Synchronize from source code to diagram

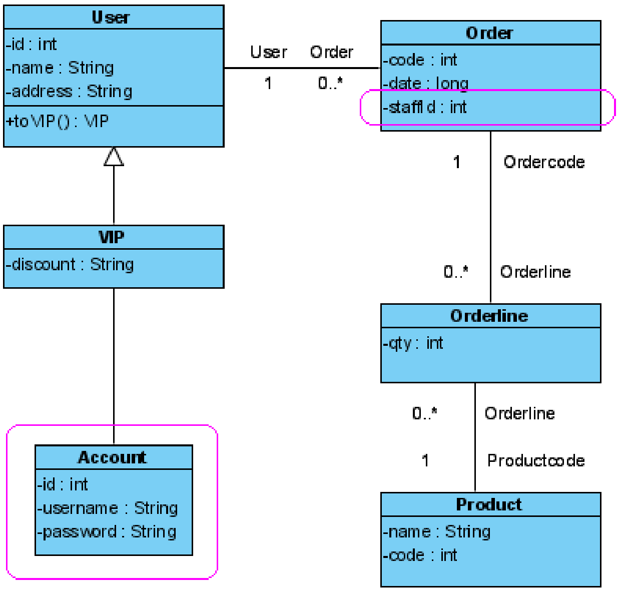


Figure 3-12 Add new class and attribute to diagram

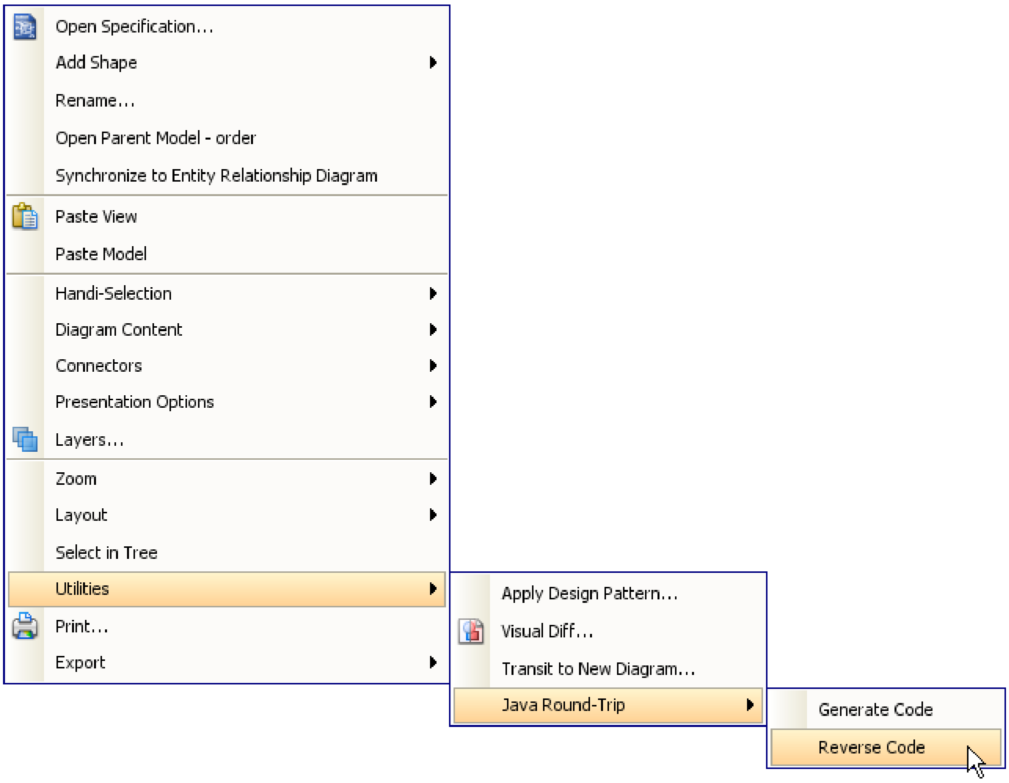


Figure 3-13 Reverse code menu

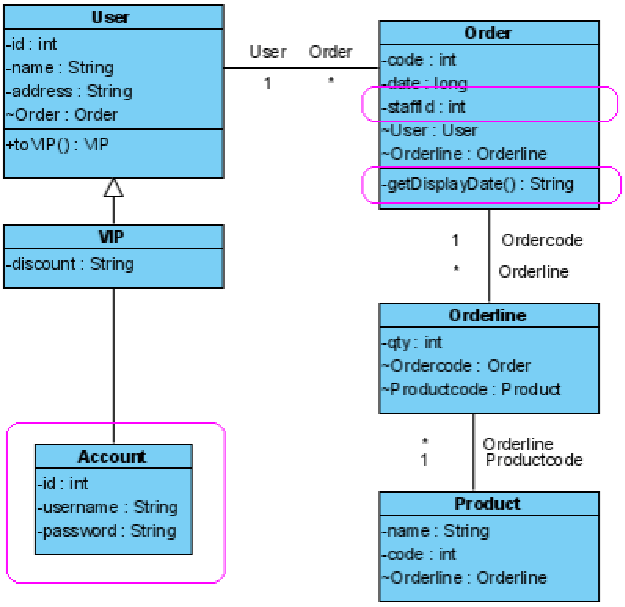
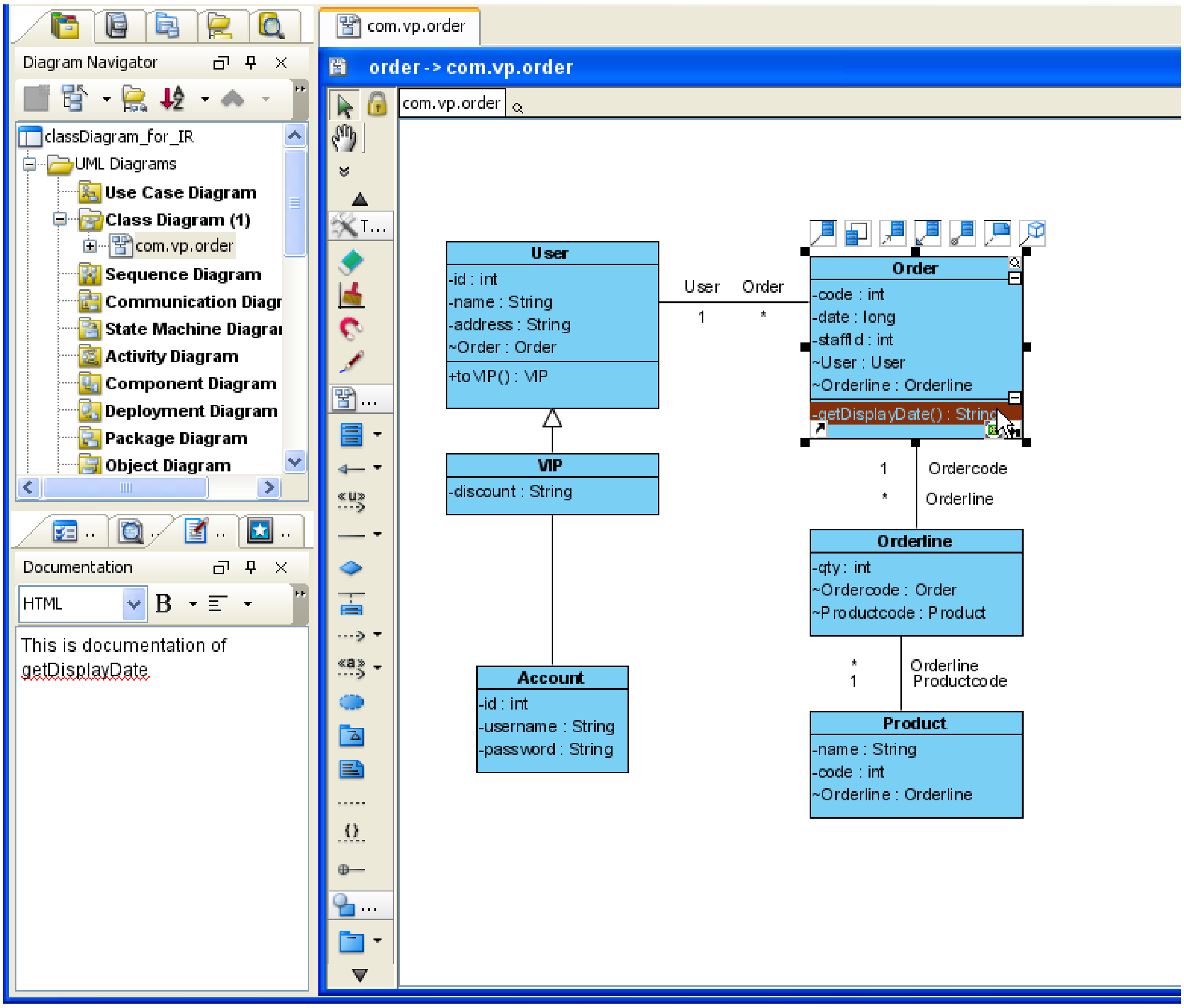


Figure 3-14 Reversed to diagram



This is documentation of getDisplayDate

Figure 3-15 Documentation reversed

Synchronize from diagram to code

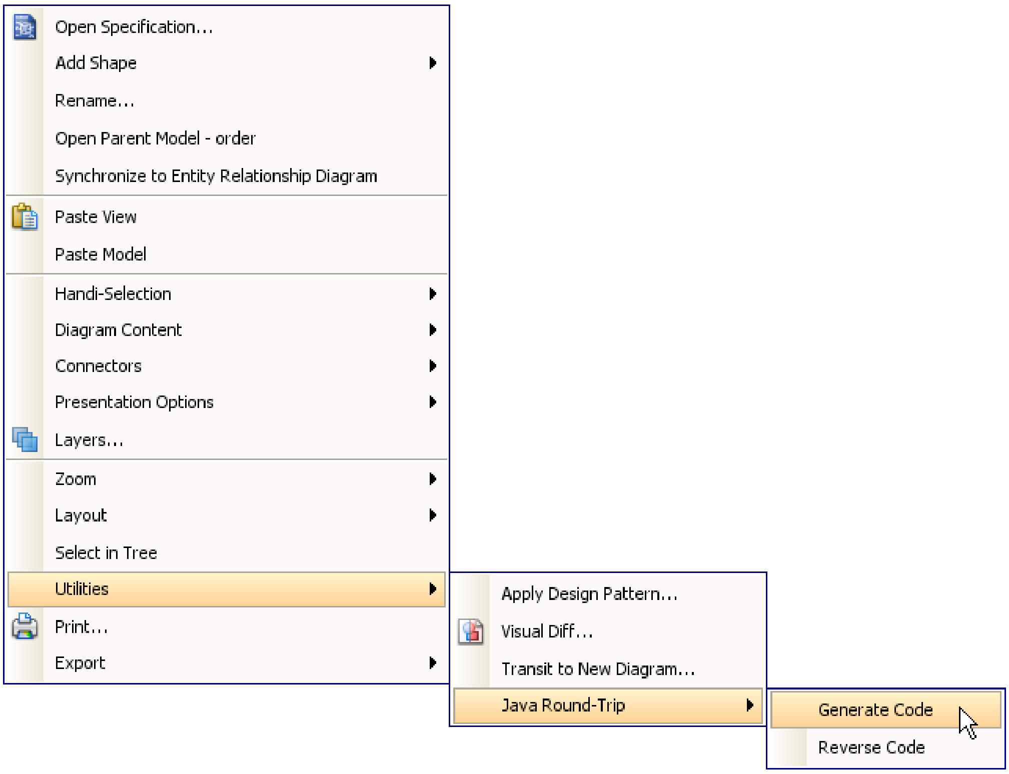
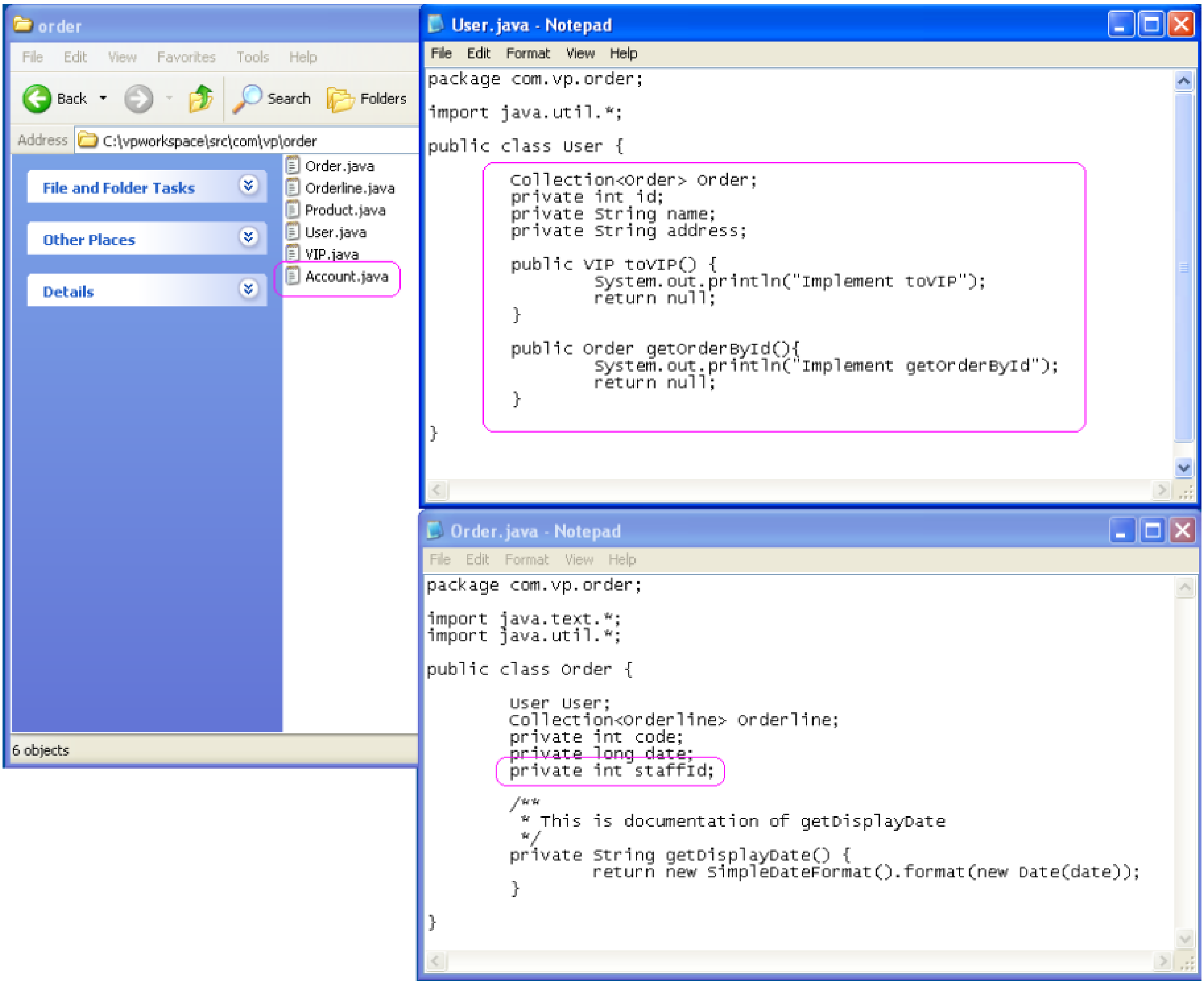


Figure 3-17 *Generate code menu*



```
package com.vp.order;

import java.util.*;

public class User {

        Collection<Order> order;
        private int id;
        private String name;
        private String address;

        public VIP toVIP() {
                System.out.println("Implement toVIP");
                return null;
        }

        public Order getOrderById(){
                System.out.println("Implement getOrderById");
                return null;
        }

}
```

```
package com.vp.order;

import java.text.*;
import java.util.*;

public class Order {

        User user;
        Collection<Orderline> orderline;
        private int code;
        private long date;
        private int staffId;

        /**
         * This is documentation of getDisplayDate
         */
        private String getDisplayDate() {
                return new SimpleDateFormat().format(new Date(date));
        }

}
```

Figure 3-18 *Synchronized*

Java Round-trip engineering for whole project

1. To generate code for whole project, select **Tools > Java Round-trip > Generate code...** from main menu.

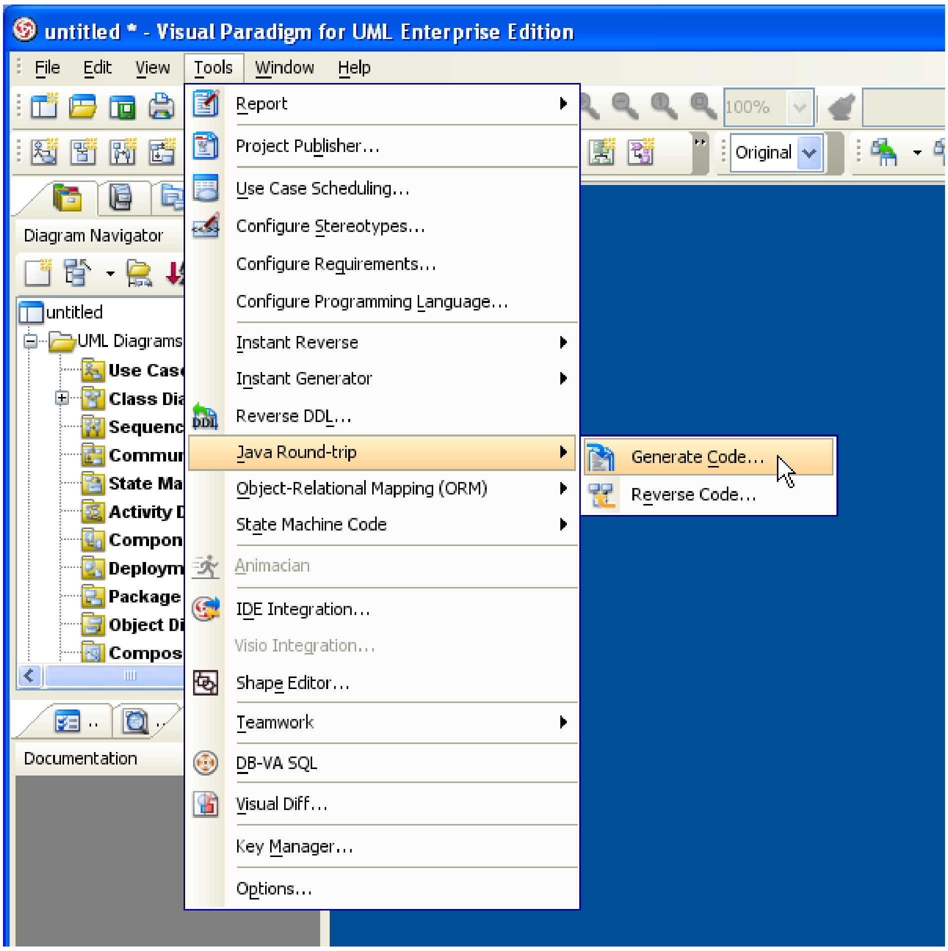
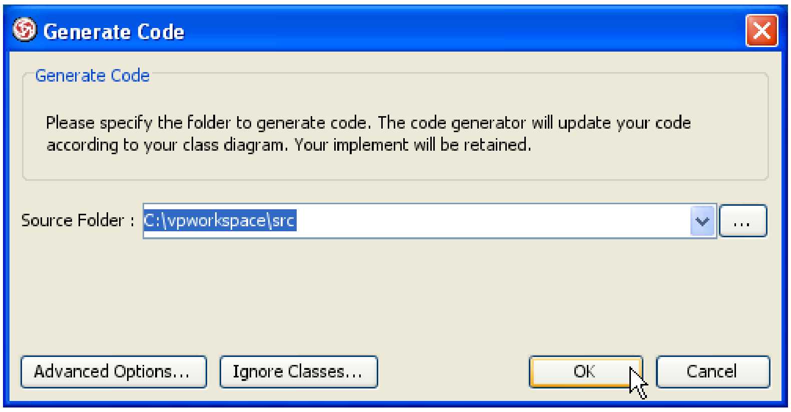


Figure 3-19 *Generate code menu*

2. Generate code dialog show, select path and click **OK** to generate.



Please specify the folder to generate code. The code generator will update your code according to your class diagram. Your implement will be retained.

Source Folder : C:\vpworkspace\src

Figure 3-20 *Generate code dialog*

3. To reverse code for whole project, select **Tools > Java Round-trip > Reverse code...** from main menu.

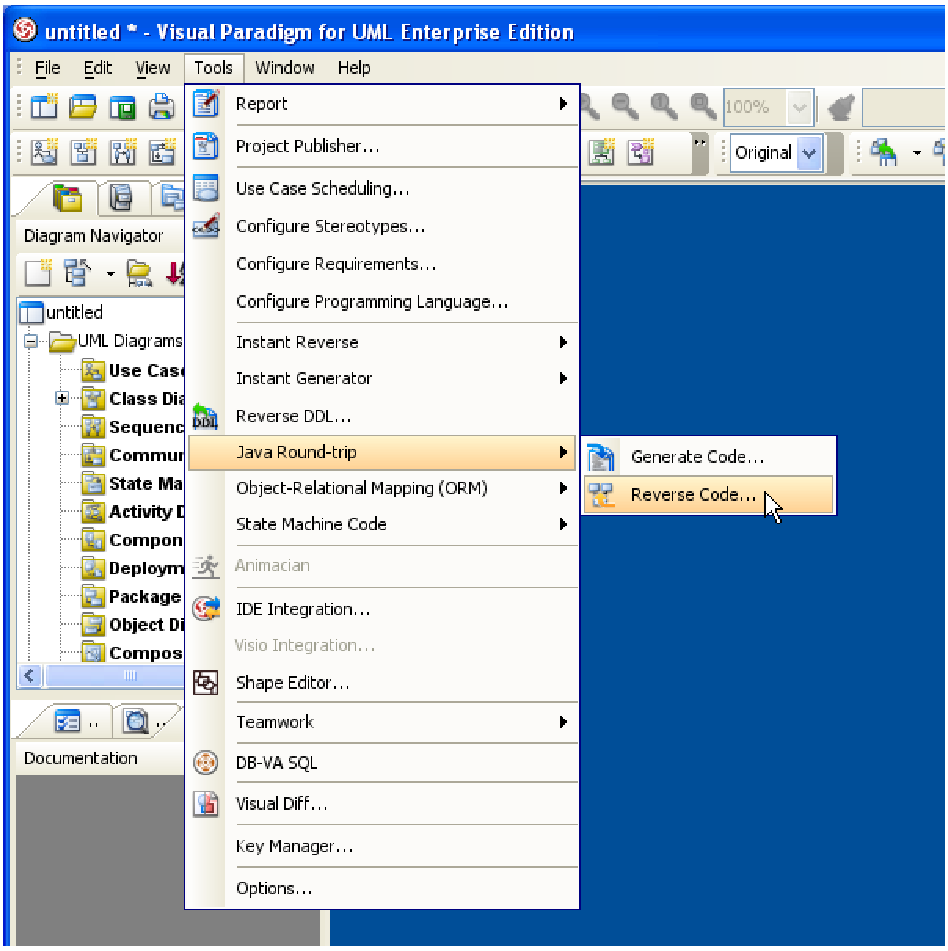
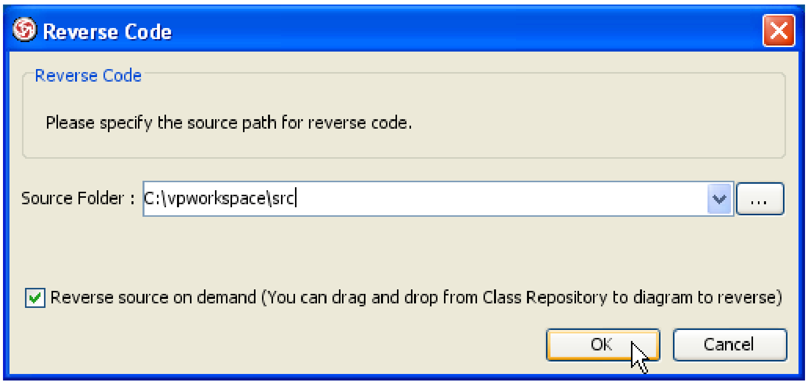


Figure 3-21 *Reverse code menu*

4. Reverse code dialog show, select path and click **OK** to reverse.



Please specify the source path for reverse code.

Source Folder : C:\vpworkspace\src

☑ Reverse source on demand (You can drag and drop from Class Repository to diagram to reverse)

Figure 3-22 *Reverse code dialog*

Drawing State Machine Diagram

1. To generate a state machine diagram, you should first have a Class on the class diagram. Select **Sub Diagrams** > **State Machine Diagram** > **Create State Machine Diagram** from the pop-down menu of the class to create a sub diagram.

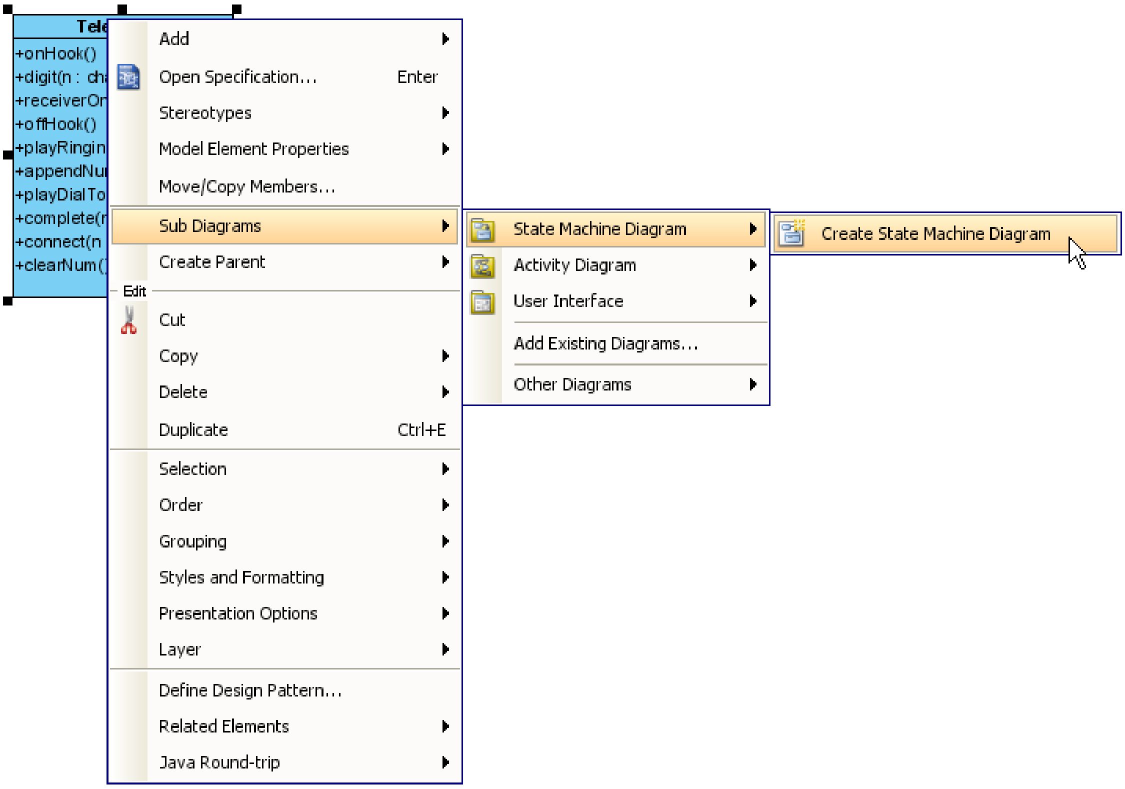


***Figure 4-1** Select Create State Machine Diagram*

2. You will see an initial pseudo state on the state machine diagram.

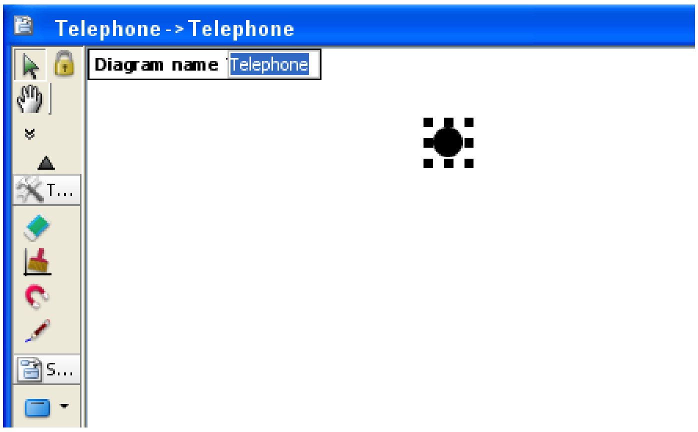


***Figure 4-2** State machine diagram with an initial pseudo state*

3. Drag a State from the diagram toolbar and drop the diagram.

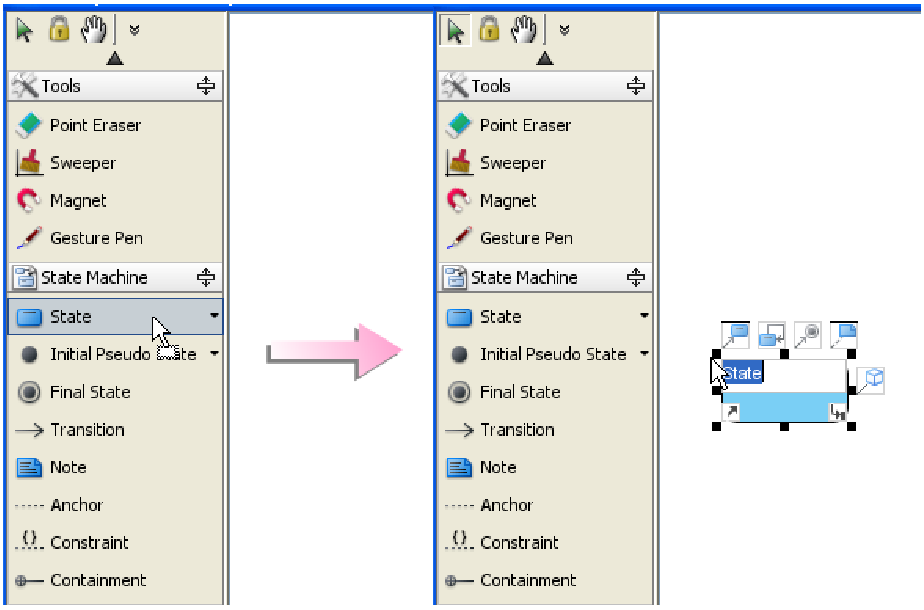


Figure 4-3 Drag and drop a State

4. Alternatively, you can use the resources of the initial pseudo state.

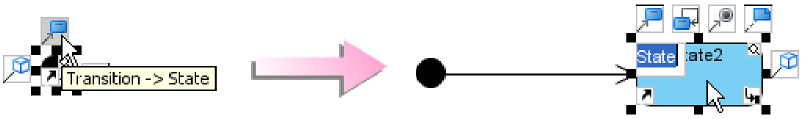


Figure 4-4 Using resources

5. A diagram can then be created.

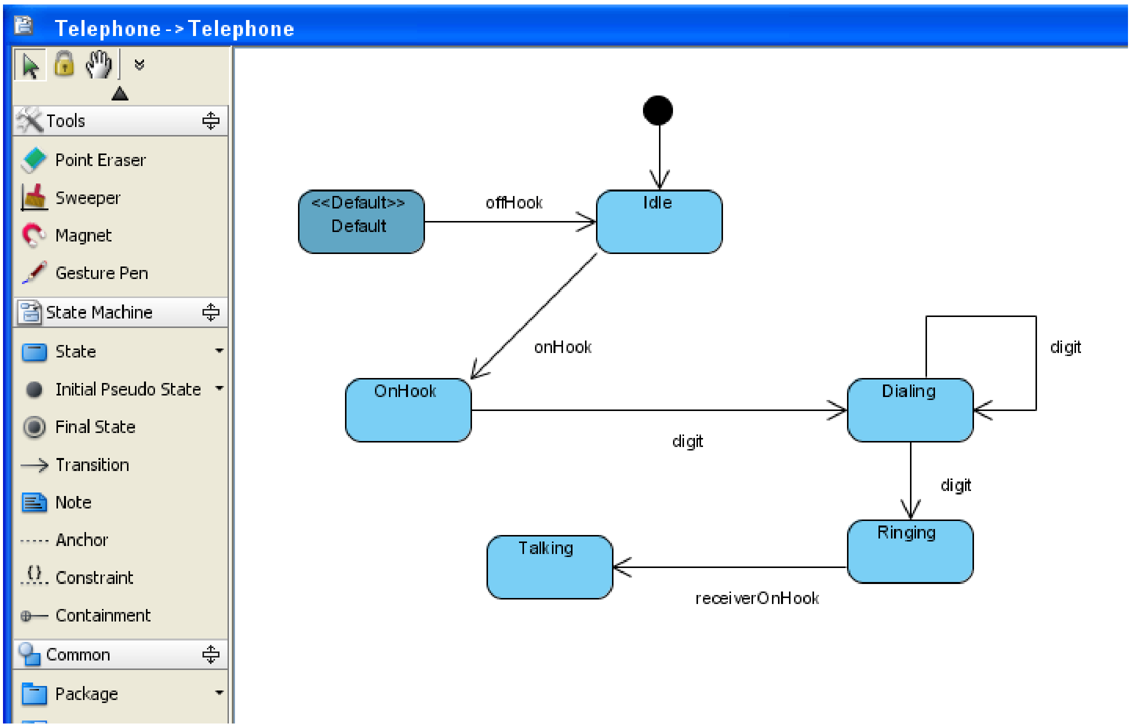


Figure 4-5 Diagram created

Generate State Machine Code

The process of creating a State Machine Code is simple.
To generate a state machine code:

1. Select **Tools > State Machine Code >Generate Code**... from the main menu.

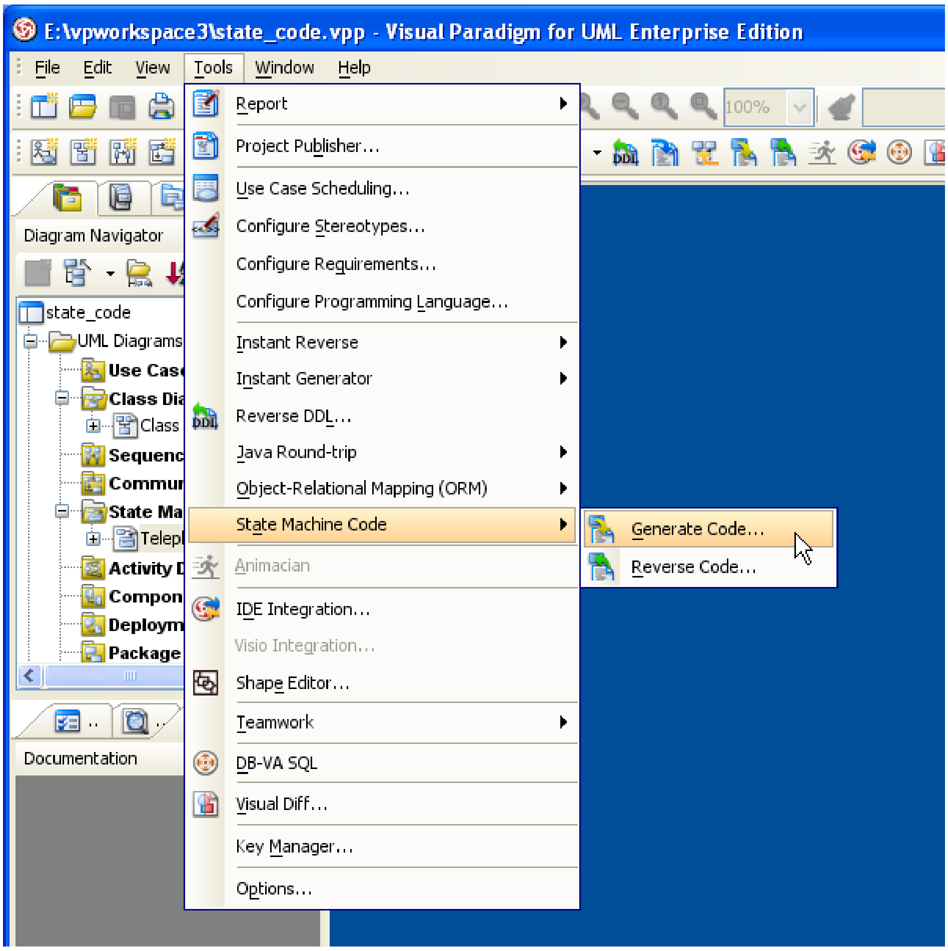
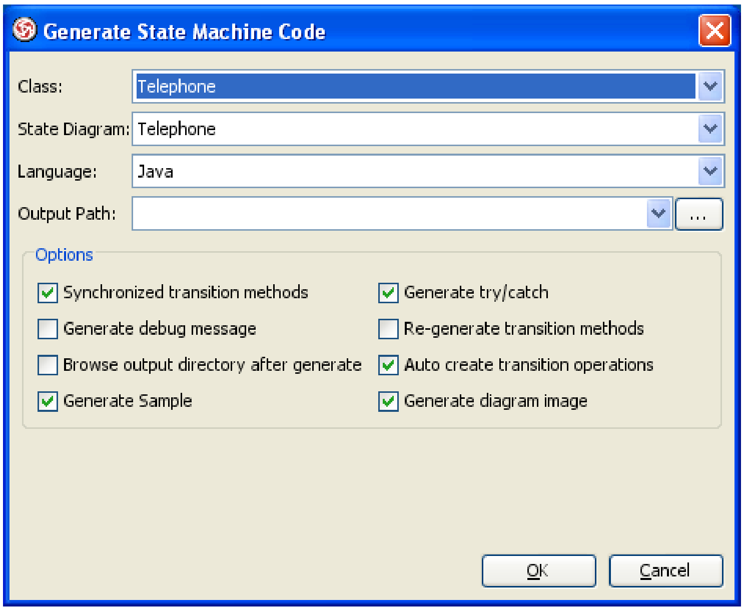


Figure 4-6 Select Generate Code...

2. The Generate state machine code dialog is displayed.


Generate State Machine Code

Class: Telephone

State Diagram: Telephone

Language: Java

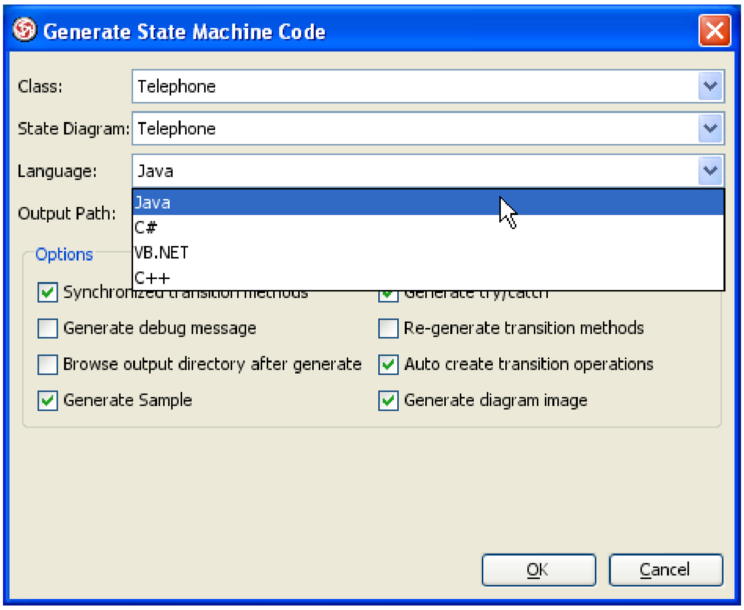
Output Path:

Options

☑ Synchronized transition methods
☑ Generate try/catch
☐ Generate debug message
☐ Re-generate transition methods
☐ Browse output directory after generate
☑ Auto create transition operations
☑ Generate Sample
☑ Generate diagram image

Figure 4-7 Generate state machine code dialog

3. Select a language to generate the code in from the drop-down menu.



Class: Telephone

State Diagram: Telephone

Language: Java

Output Path:

Options

☑ Synchronized transition methods ☑ Generate try/catch

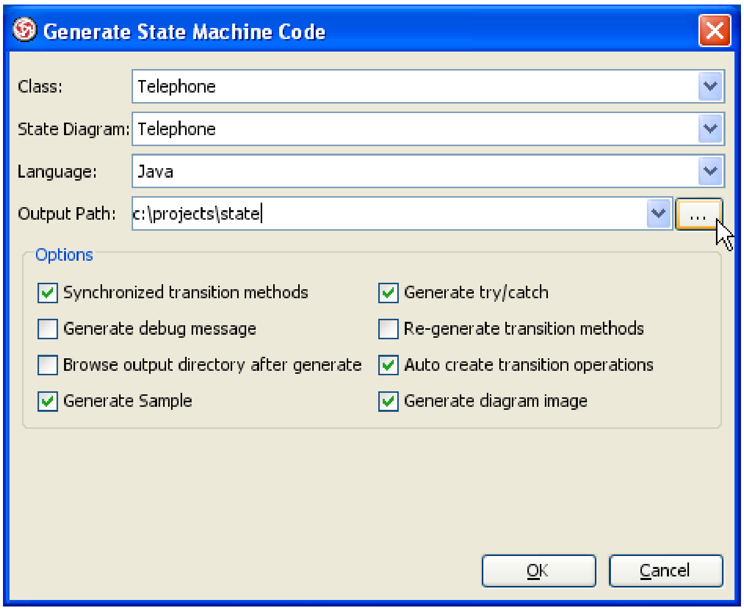
□ Generate debug message □ Re-generate transition methods

□ Browse output directory after generate ☑ Auto create transition operations

☑ Generate Sample ☑ Generate diagram image

Figure 4-8 Select Language

4. Configure an output path by selecting **...** or type in the path in the text box, and configure the options in the dialog.



Class: Telephone

State Diagram: Telephone

Language: Java

Output Path: c:\projects\state

Options

☑ Synchronized transition methods ☑ Generate try/catch

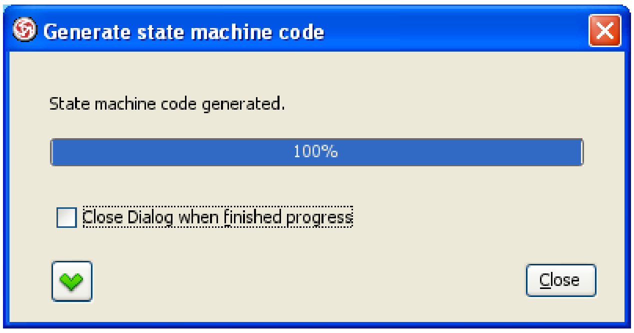
□ Generate debug message □ Re-generate transition methods

□ Browse output directory after generate ☑ Auto create transition operations

☑ Generate Sample ☑ Generate diagram image

Figure 4-9 Configure an output path

5. The process of generation is shown. You may choose to Close Dialog when finished progress by checking the check box.



State machine code generated.

100%

□ Close Dialog when finished progress

Figure 4-10 Process of generation

6. The code is generated.

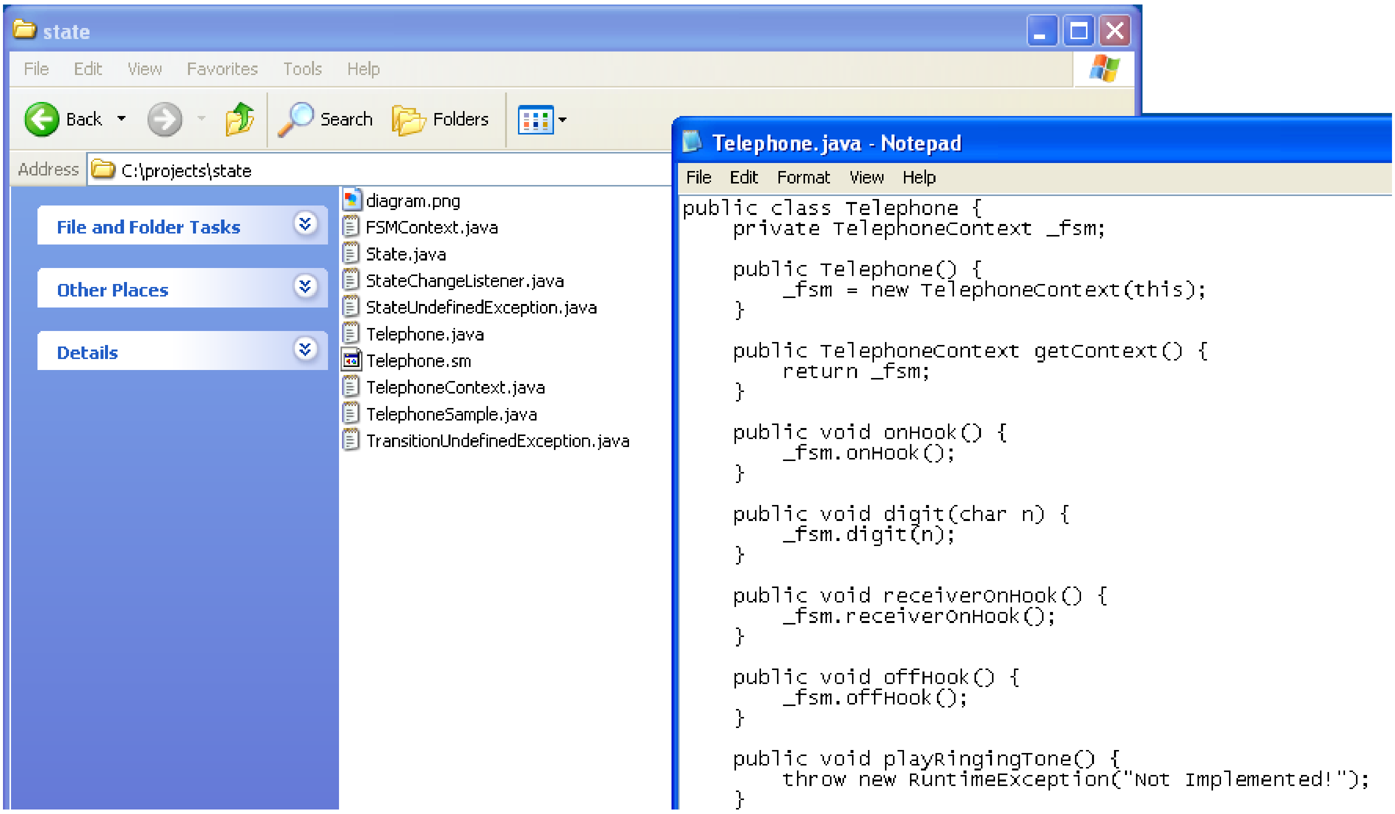

state
File Edit View Favorites Tools Help
Back
Search
Folders
Address C:\projects\state
File and Folder Tasks
Other Places
Details
Telephone.java - Notepad
File Edit Format View Help

diagram.png
FSMContext.java
State.java
StateChangeListener.java
StateUndefinedException.java
Telephone.java
Telephone.sm
TelephoneContext.java
TelephoneSample.java
TransitionUndefinedException.java

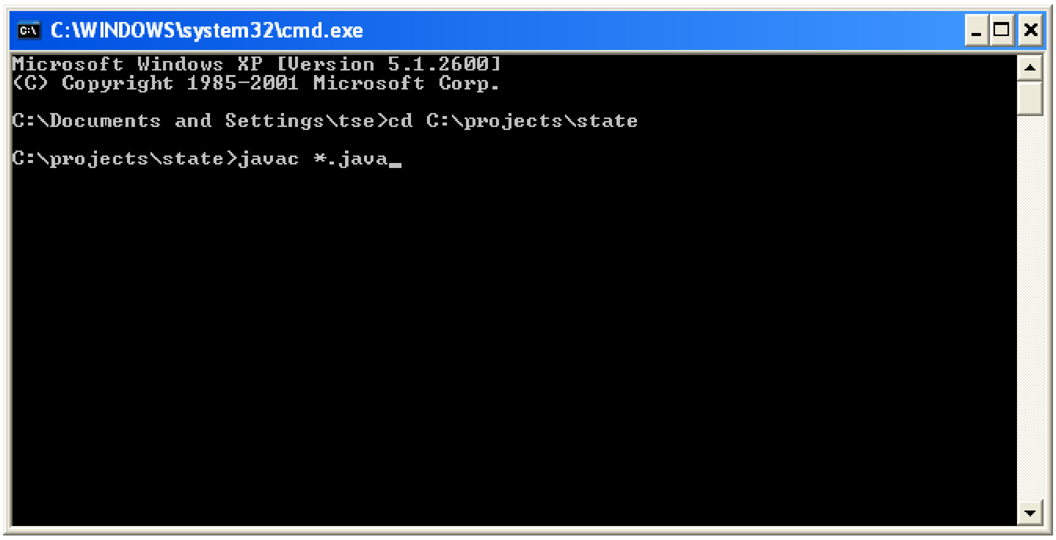
```
public class Telephone {
    private TelephoneContext _fsm;

    public Telephone() {
        _fsm = new TelephoneContext(this);
    }

    public TelephoneContext getContext() {
        return _fsm;
    }

    public void onHook() {
        _fsm.onHook();
    }

    public void digit(char n) {
        _fsm.digit(n);
    }

    public void receiverOnHook() {
        _fsm.receiverOnHook();
    }

    public void offHook() {
        _fsm.offHook();
    }

    public void playRingingTone() {
        throw new RuntimeException("Not Implemented!");
    }
```

Figure 4-11 *Code generated*

Running State Machine Sample Code

After generating Java files, you can compile and run them. Command Prompt in windows is used as an example to demonstrate the actions.

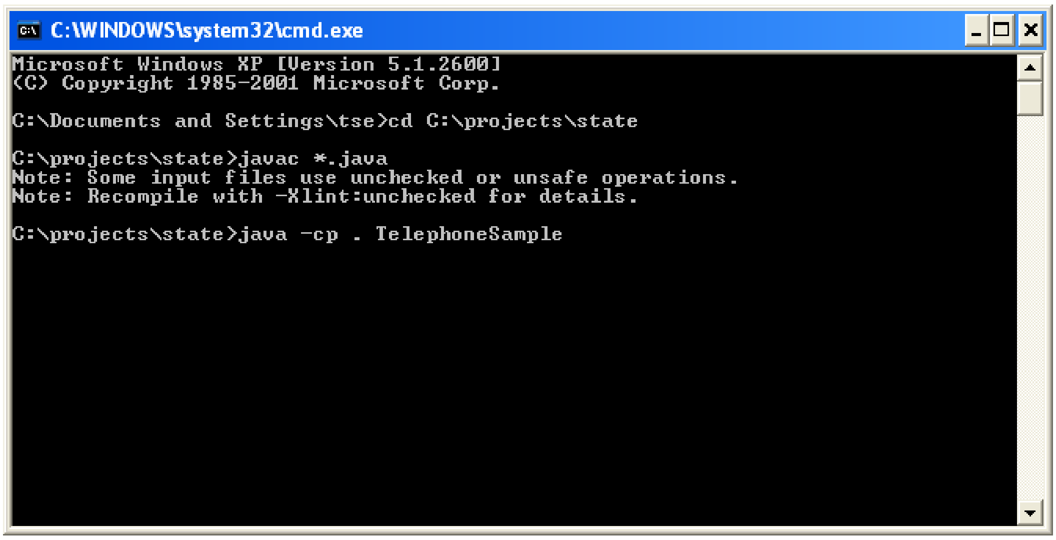
1. Run the compiler. Change the directory to the directory where the code file is generated. Type in the command for compiling. In this example, the command is javac *java .



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\tse>cd C:\projects\state

C:\projects\state>javac *.java_
```

Figure 4-12 *Compile Java*

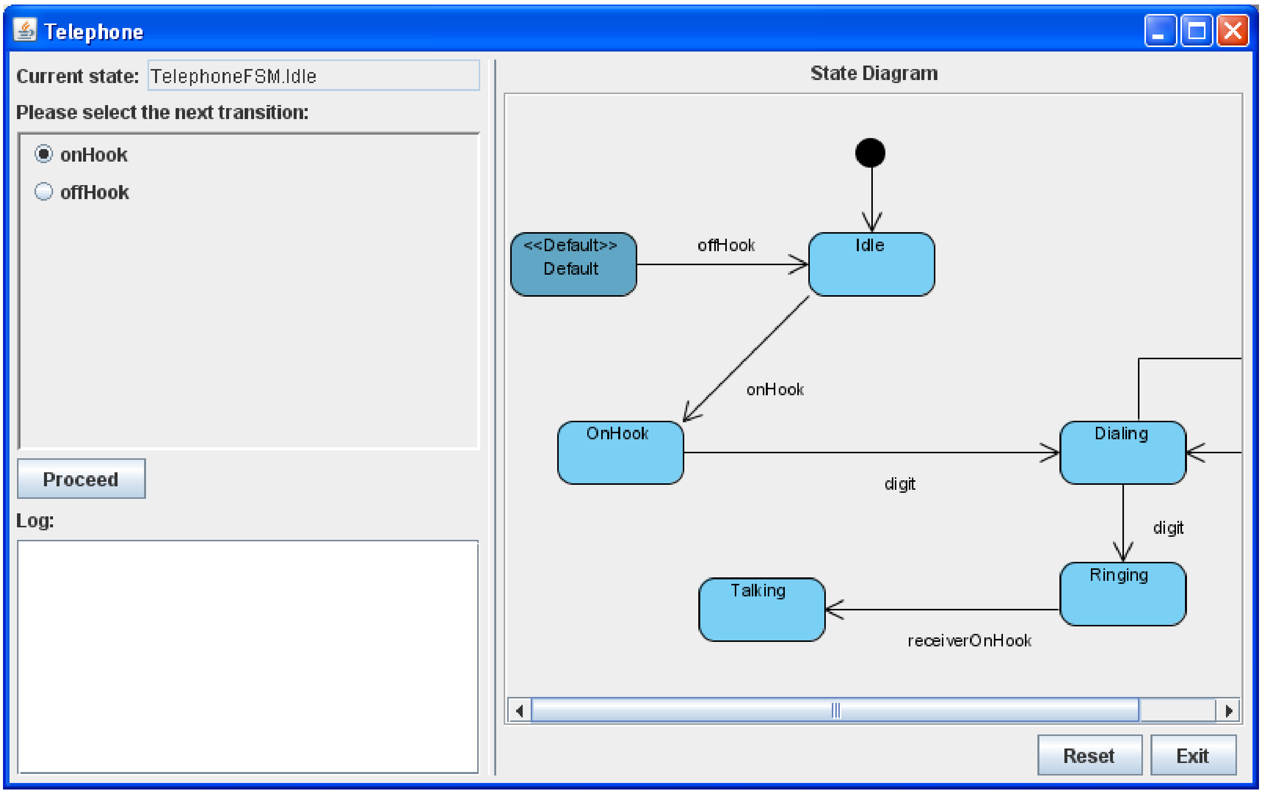
2. After compiling, you can enter another command for running Java. In this example, the command is java -cp . TelephoneSample .



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\tse>cd C:\projects\state

C:\projects\state>javac *.java
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\projects\state>java -cp . TelephoneSample
```

Figure 4-13 *Run Java*

3. The Java is run.



Telephone

Current state: TelephoneFSM.Idle

Please select the next transition:

● onHook

○ offHook

Proceed

Log:

Figure 4-14 *Sample Java*

Programming generated State Machine Code

VP-UML supports the generation of different types of state machine code. Since the steps for generation of codes in different languages are similar, Java will be used as an example to illustrate the steps.

Before generating the code, you may want to configure the properties of states and transitions.

For state, configure by selecting Open Specification...from the pop-up menu.

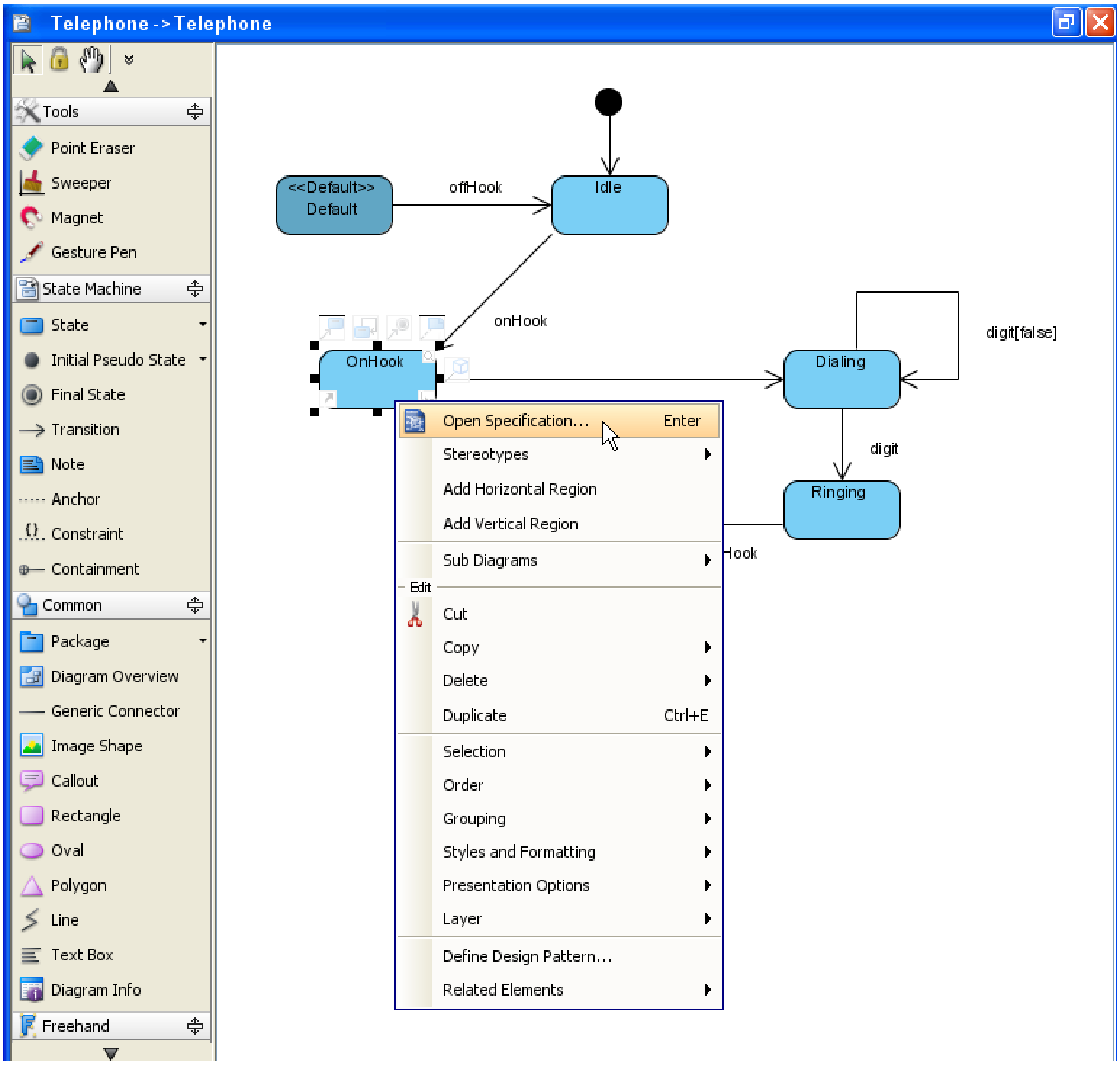
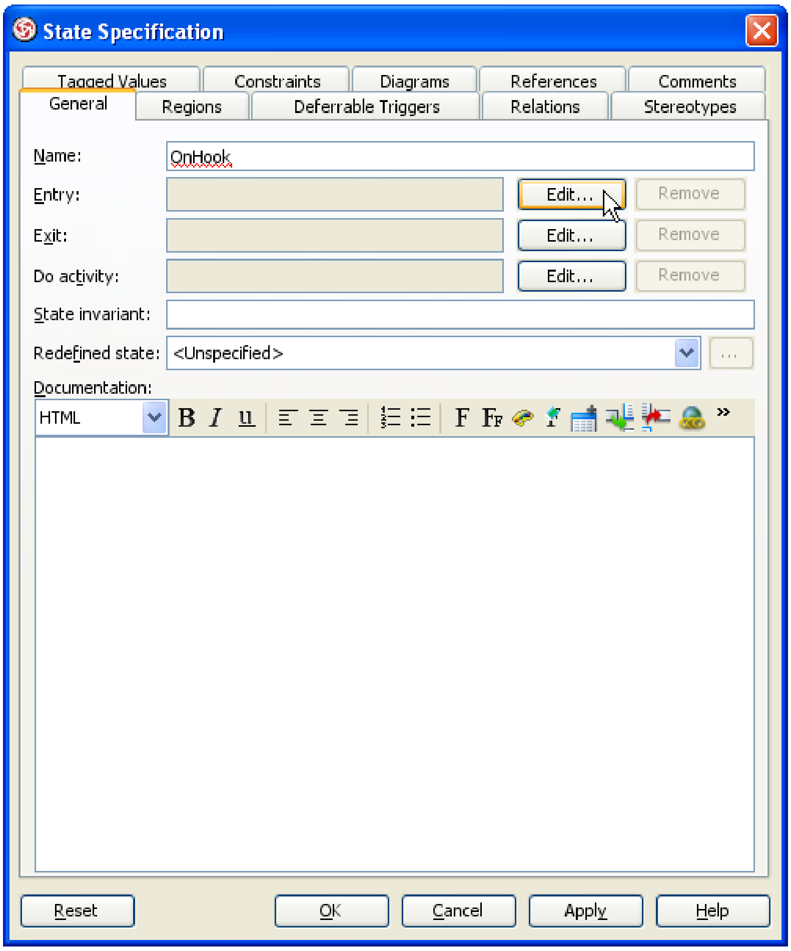


Figure 4-15 *Select Open Specification...*

The State Specification dialog box is now open, and you can configure the state properties here. You may edit the Entry property by clicking Edit... .



Name: OnHook

Entry:

Exit:

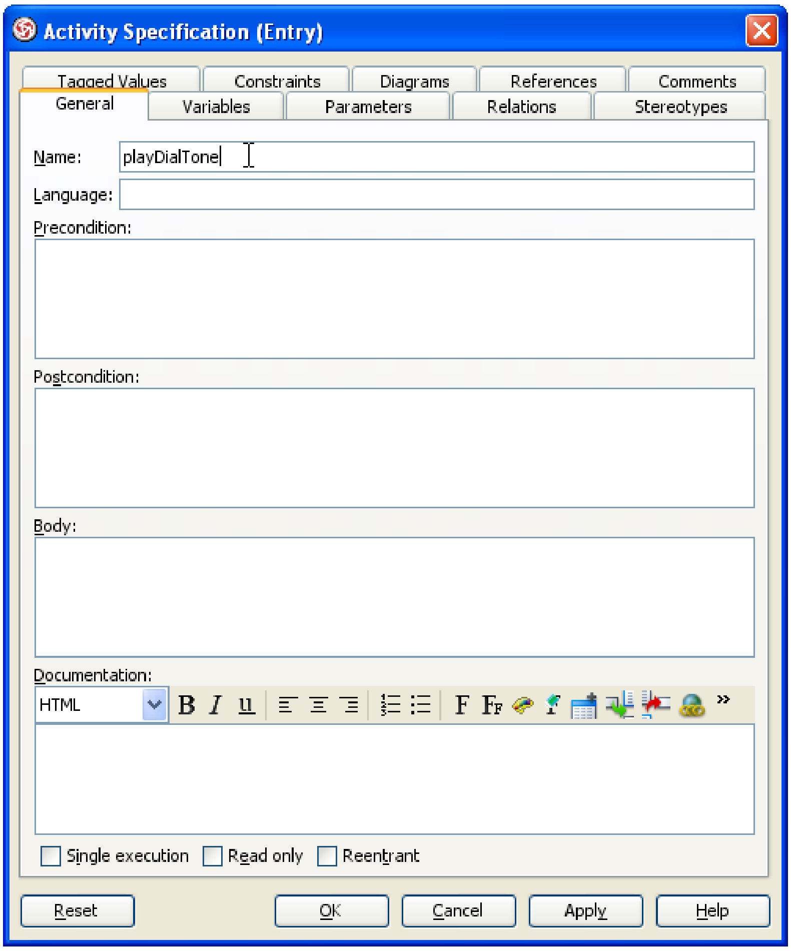
Do activity:

State invariant:

Redefined state: <Unspecified>

Figure 4-16 *State Specification dialog box*

This is the Activity Specification(Entry) dialog box. After editing to suit your needs, click OK to confirm.



Name: playDialTone

Language:

HTML

□ Single execution □ Read only □ Reentrant

Figure 4-17 Activity Specification(Entry) dialog box

State's property has been edited.

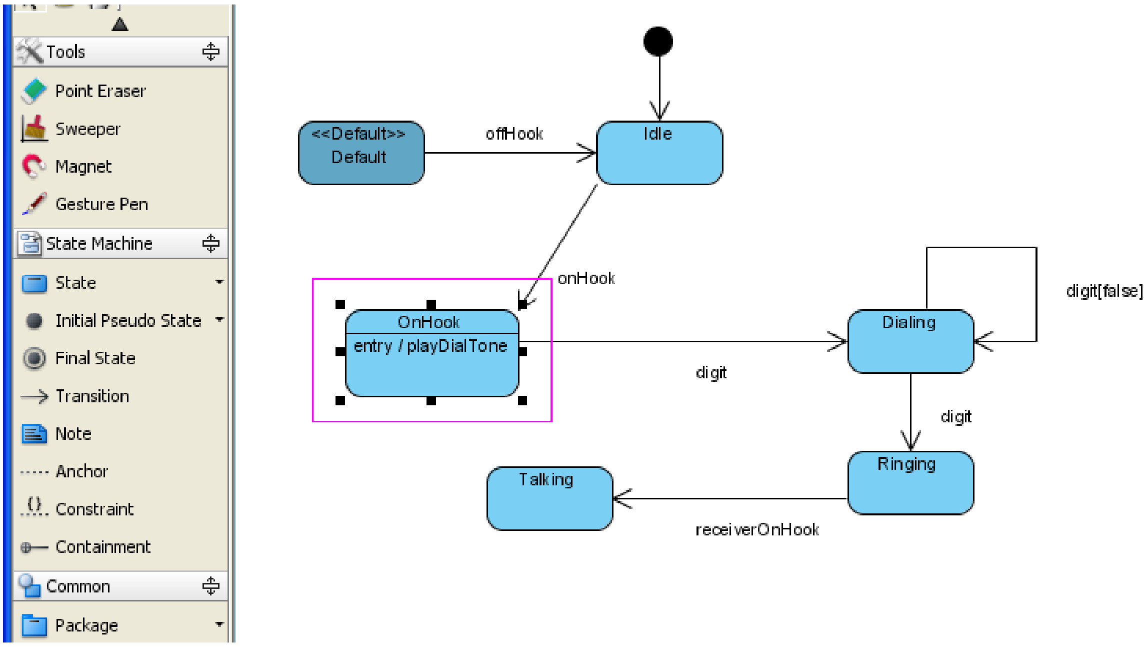


Figure 4-18 State's property edited

Similarly, you can edit the transition by selecting the Open Specification... from the pop-up menu.

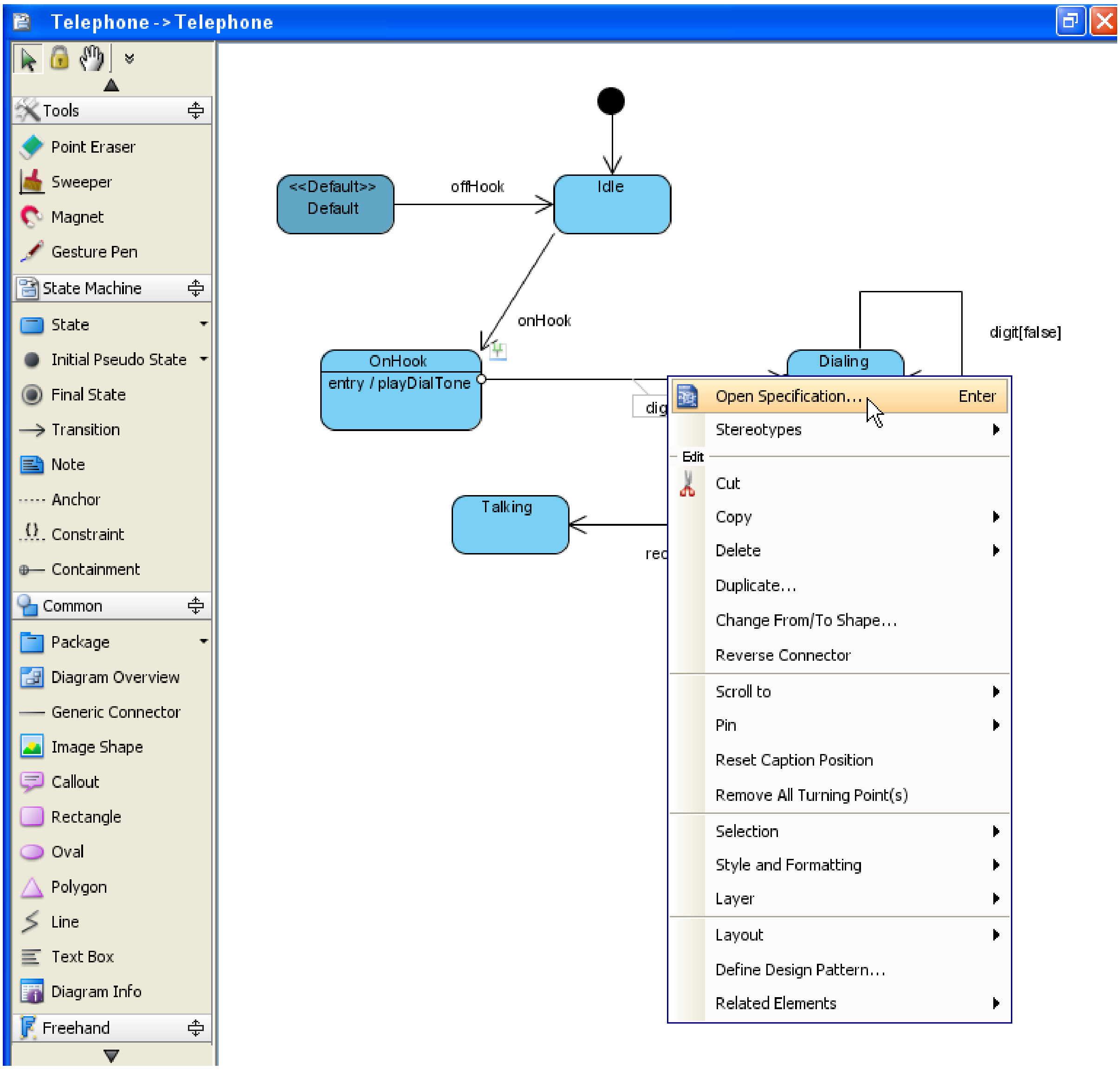
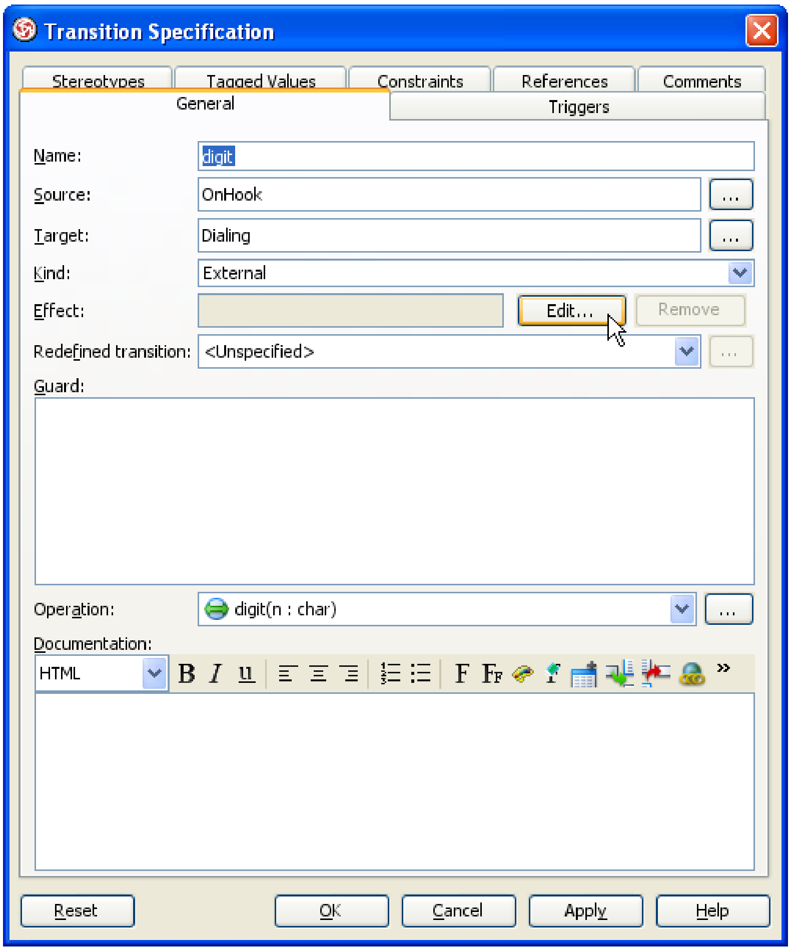


Figure 4-19 *Open Specification...*

Then, edit the properties of transition in the Transition Specification dialog box.



Transition Specification

Name: digit

Source: OnHook

Target: Dialing

Kind: External

Effect:

Redefined transition: <Unspecified>

Guard:

Operation: digit(n : char)

Documentation:

Figure 4-20 Transition Specification dialog box

The states and transitions properties are configured.

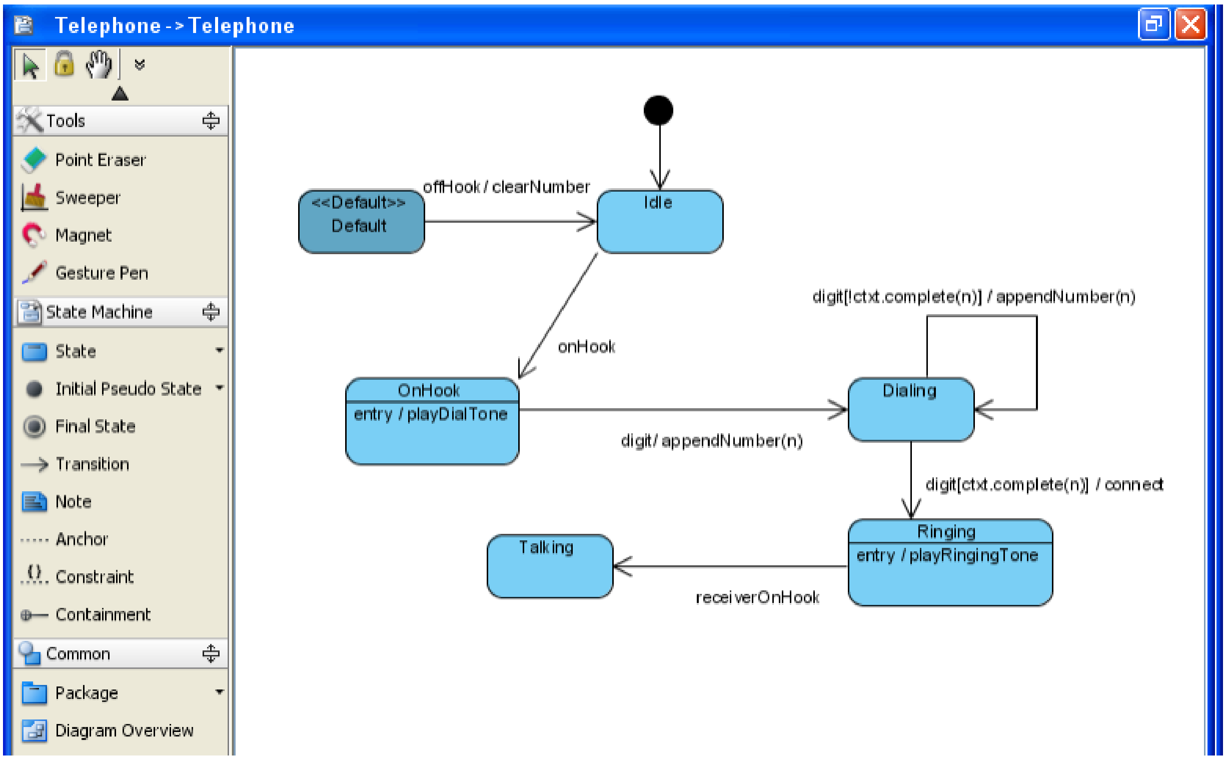
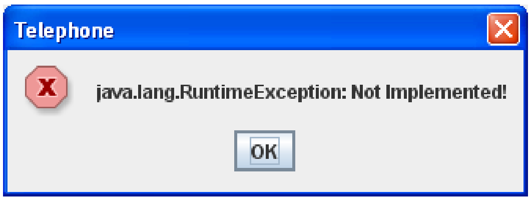


Figure 4-21 State Machine Diagram

Insert Implementation

If there is no implementation, the Java cannot be run because there is an error.

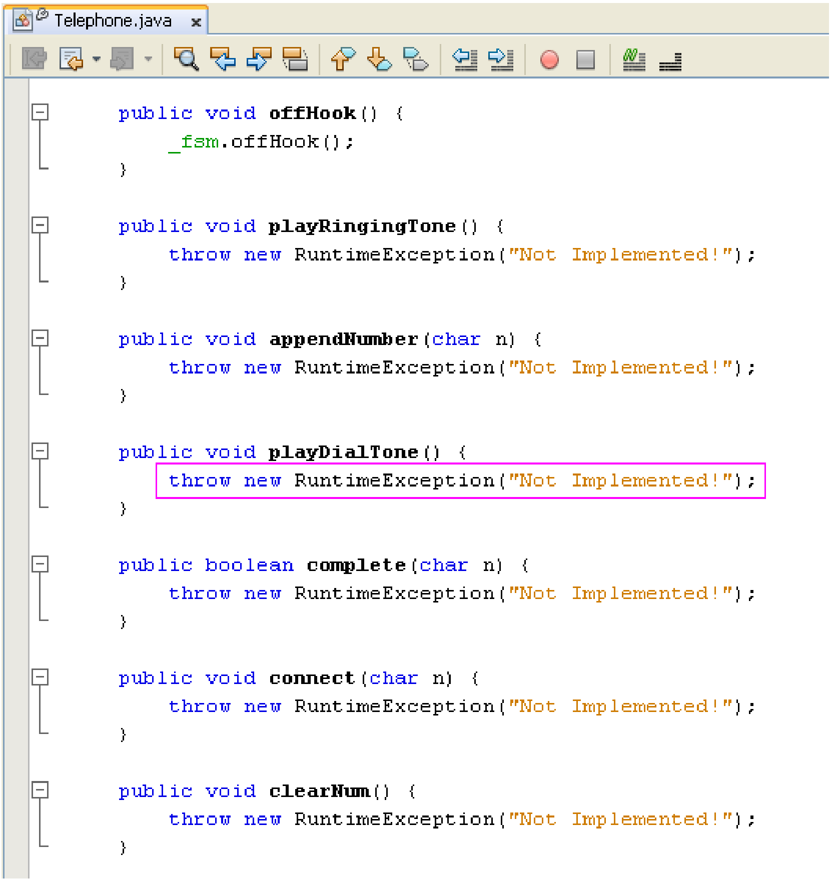


java.lang.RuntimeException: Not Implemented!

Figure 4-22 *Error message box*

.

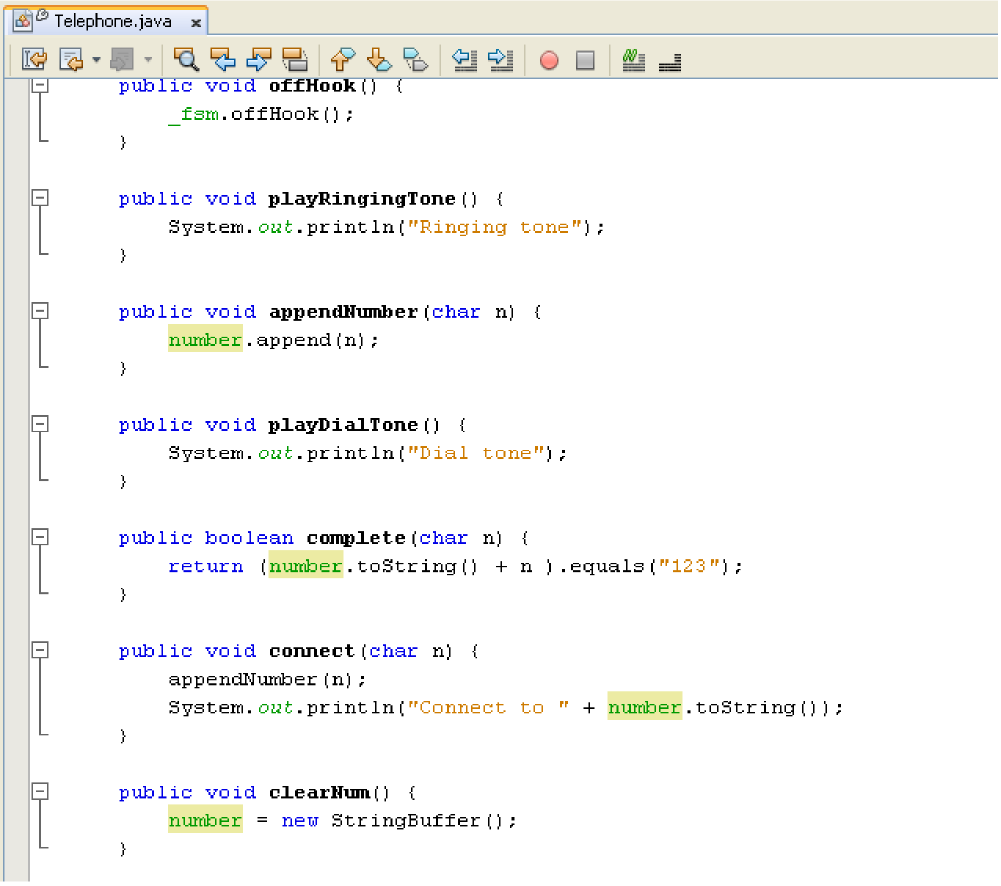
The original state machine diagram code generated did not have any implementation.



```
    public void offHook() {
        _fsm.offHook();
    }

    public void playRingingTone() {
        throw new RuntimeException("Not Implemented!");
    }

    public void appendNumber(char n) {
        throw new RuntimeException("Not Implemented!");
    }

    public void playDialTone() {
        throw new RuntimeException("Not Implemented!");
    }

    public boolean complete(char n) {
        throw new RuntimeException("Not Implemented!");
    }

    public void connect(char n) {
        throw new RuntimeException("Not Implemented!");
    }

    public void clearNum() {
        throw new RuntimeException("Not Implemented!");
    }
```

Figure 4-23 *No implementation*

You can add the implementation in the source.

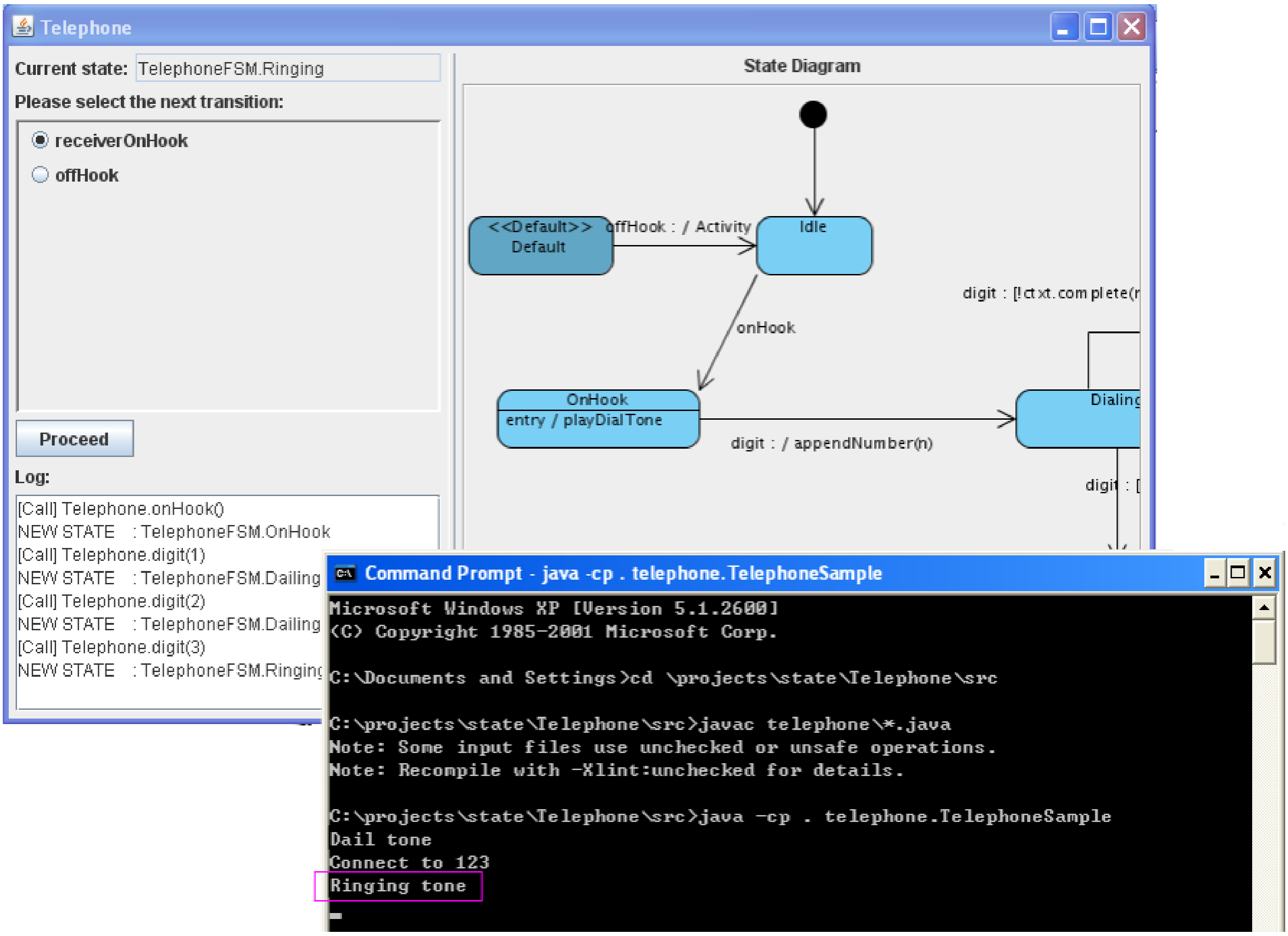


Telephone.java

```
    public void offHook() {
        _fsm.offHook();
    }

    public void playRingingTone() {
        System.out.println("Ringing tone");
    }

    public void appendNumber(char n) {
        number.append(n);
    }

    public void playDialTone() {
        System.out.println("Dial tone");
    }

    public boolean complete(char n) {
        return (number.toString() + n ).equals("123");
    }

    public void connect(char n) {
        appendNumber(n);
        System.out.println("Connect to " + number.toString());
    }

    public void clearNum() {
        number = new StringBuffer();
    }
```

Figure 4-24 *With implementation*

The diagram sample is run with the implementation.



Current state: TelephoneFSM.Ringing

Please select the next transition:

● receiverOnHook

○ offHook

Log:

```
[Call] Telephone.onHook()
NEW STATE    : TelephoneFSM.OnHook
[Call] Telephone.digit(1)
NEW STATE    : TelephoneFSM.Dailing
[Call] Telephone.digit(2)
NEW STATE    : TelephoneFSM.Dailing
[Call] Telephone.digit(3)
NEW STATE    : TelephoneFSM.Ringing
```

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings>cd \projects\state\Telephone\src

C:\projects\state\Telephone\src>javac telephone\*.java
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\projects\state\Telephone\src>java -cp . telephone.TelephoneSample
Dail tone
Connect to 123
Ringing tone
_
```

Figure 4-25 *Java sample*

Exporting XML

Exporting whole project to XML

1. Mouse over main menu.

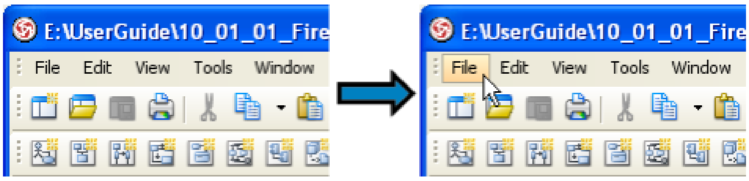


Figure 1-1 *Mouse over main menu*

2. Click on File of main menu.

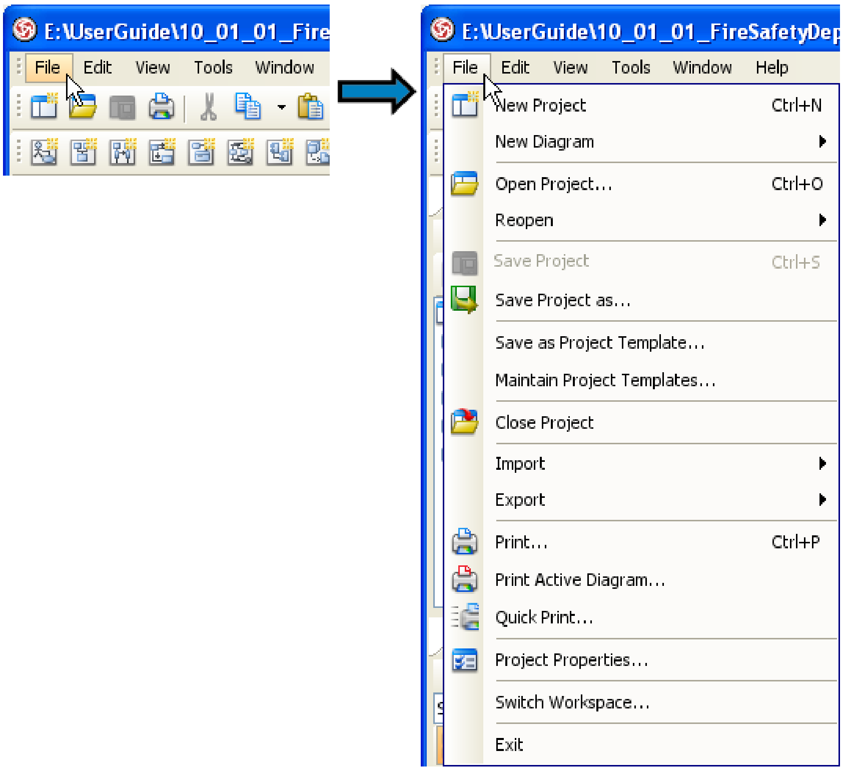


Figure 1-2 *Showing popup of main menu*

3. Click on Export > XML... menu item.

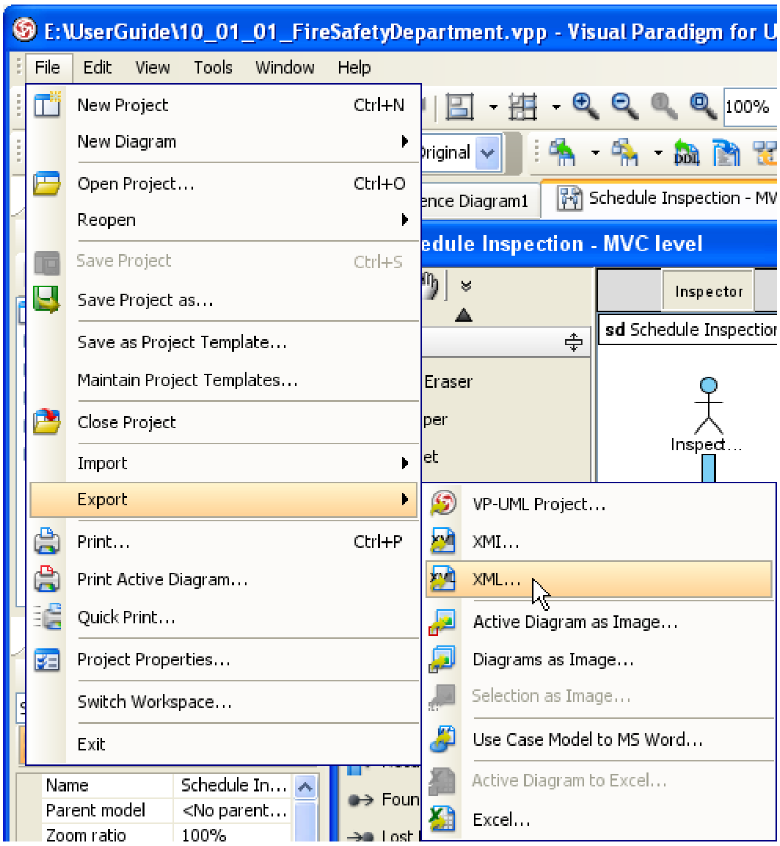
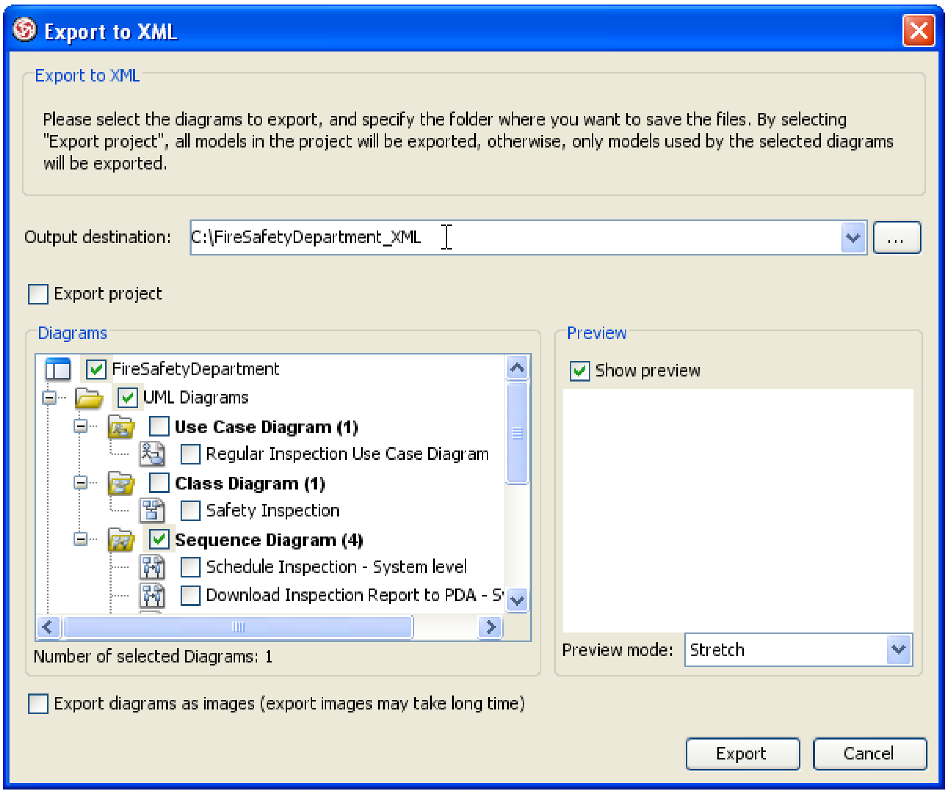


Figure 1-3 *Export XML*

4. Input Output destination.



Export to XML

Export to XML

Please select the diagrams to export, and specify the folder where you want to save the files. By selecting "Export project", all models in the project will be exported, otherwise, only models used by the selected diagrams will be exported.

Output destination: C:\FireSafetyDepartment_XML ...

□ Export project

Diagrams

- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Regular Inspection Use Case Diagram
    - □ **Class Diagram (1)**
      - □ Safety Inspection
    - ☑ **Sequence Diagram (4)**
      - □ Schedule Inspection - System level
      - □ Download Inspection Report to PDA - S

Number of selected Diagrams: 1

Preview

☑ Show preview

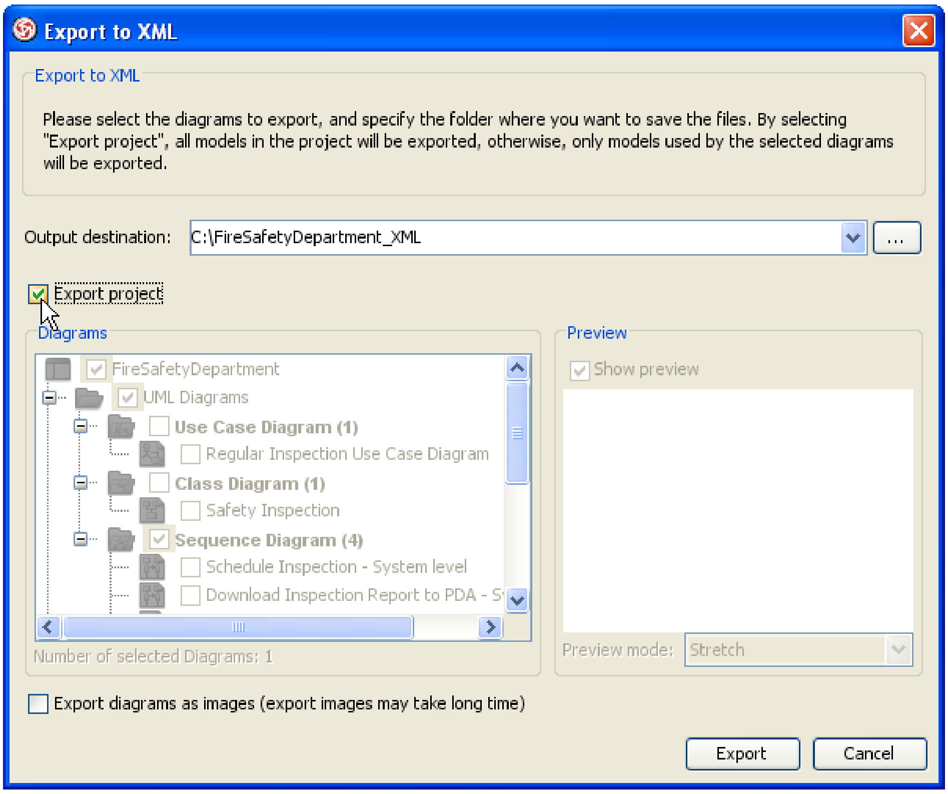
Preview mode: Stretch

□ Export diagrams as images (export images may take long time)

Export Cancel

Figure 1-4 *Inputting output destination*

5. Check Export project checkbox.



Export to XML

Please select the diagrams to export, and specify the folder where you want to save the files. By selecting "Export project", all models in the project will be exported, otherwise, only models used by the selected diagrams will be exported.

Output destination: C:\FireSafetyDepartment_XML

☑ Export project

Diagrams

- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - ☐ Use Case Diagram (1)
      - ☐ Regular Inspection Use Case Diagram
    - ☐ Class Diagram (1)
      - ☐ Safety Inspection
    - ☑ Sequence Diagram (4)
      - ☐ Schedule Inspection - System level
      - ☐ Download Inspection Report to PDA - S

Number of selected Diagrams: 1

Preview

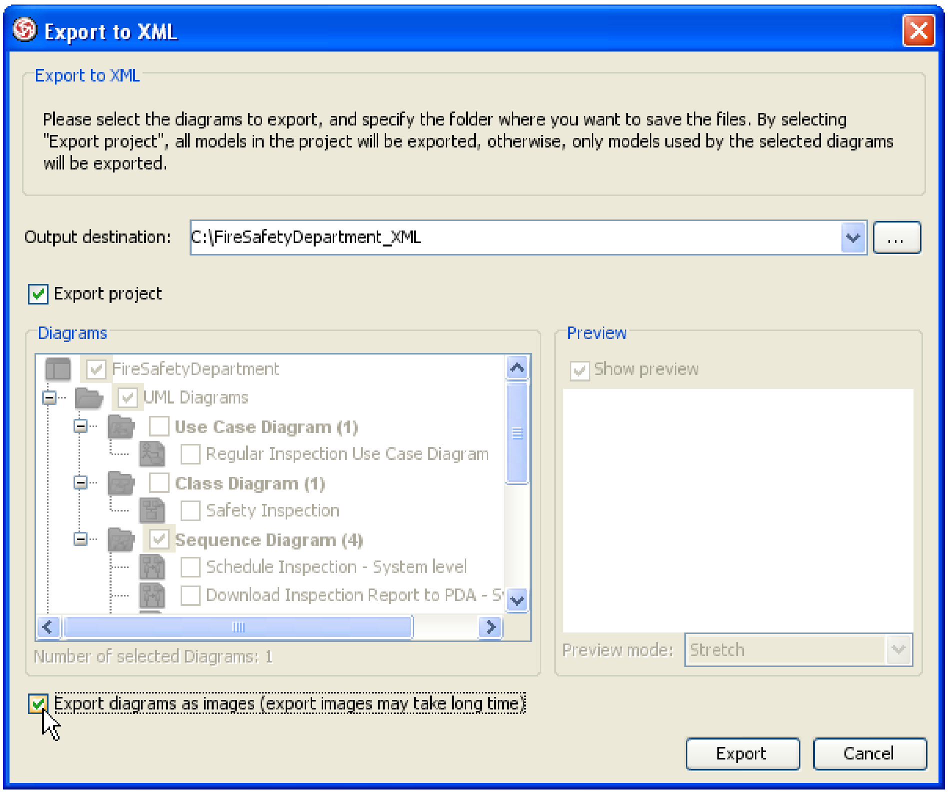
☑ Show preview

Preview mode: Stretch

☐ Export diagrams as images (export images may take long time)

Figure 1-5 Check Export project checkbox

6. Check Export diagrams as images (export images may take long time) checkbox.



Export to XML

Please select the diagrams to export, and specify the folder where you want to save the files. By selecting "Export project", all models in the project will be exported, otherwise, only models used by the selected diagrams will be exported.

Output destination: C:\FireSafetyDepartment_XML

☑ Export project

Diagrams

- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - ☐ Use Case Diagram (1)
      - ☐ Regular Inspection Use Case Diagram
    - ☐ Class Diagram (1)
      - ☐ Safety Inspection
    - ☑ Sequence Diagram (4)
      - ☐ Schedule Inspection - System level
      - ☐ Download Inspection Report to PDA - S

Number of selected Diagrams: 1

Preview

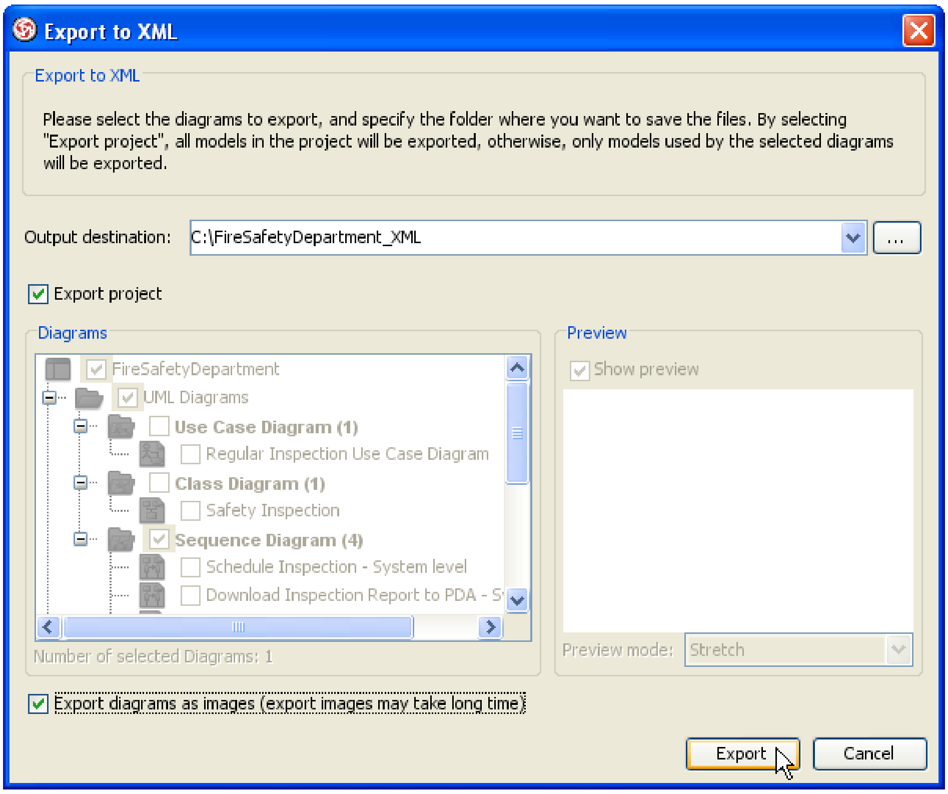
☑ Show preview

Preview mode: Stretch

☑ Export diagrams as images (export images may take long time)

Figure 1-6 Check export diagrams as images checkbox

7. Press Export button.



Please select the diagrams to export, and specify the folder where you want to save the files. By selecting "Export project", all models in the project will be exported, otherwise, only models used by the selected diagrams will be exported.

Output destination: C:\FireSafetyDepartment_XML

☑ Export project

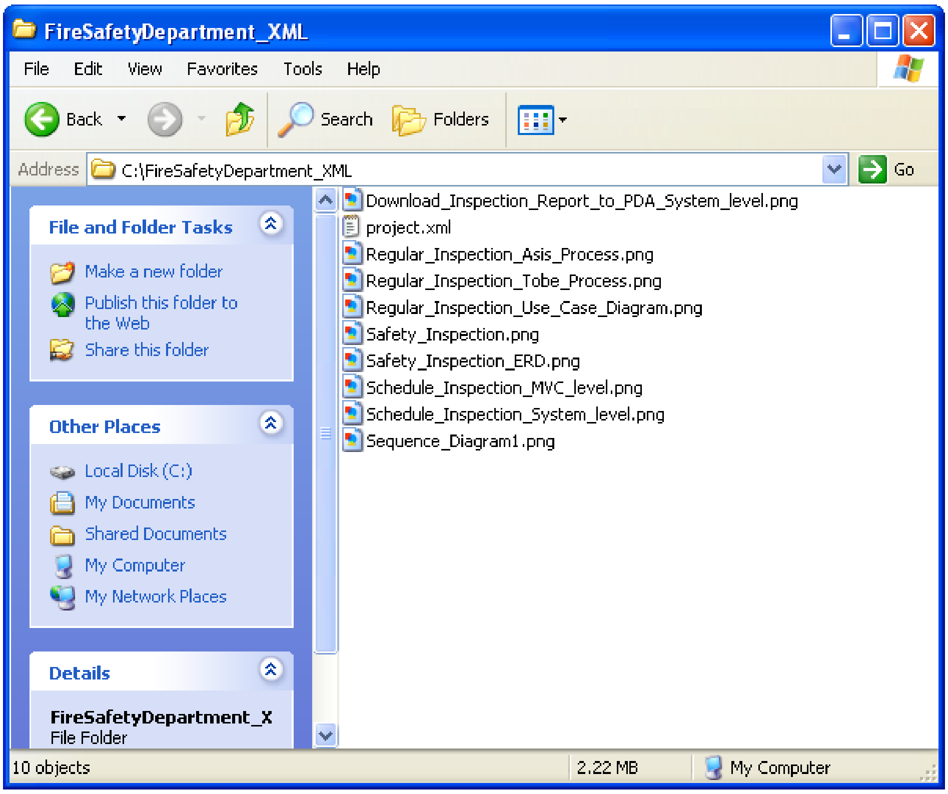
Number of selected Diagrams: 1

Preview mode: Stretch

☑ Export diagrams as images (export images may take long time)

Figure 1-7 *Export to XML*

8. Review exported content.



Address: C:\FireSafetyDepartment_XML

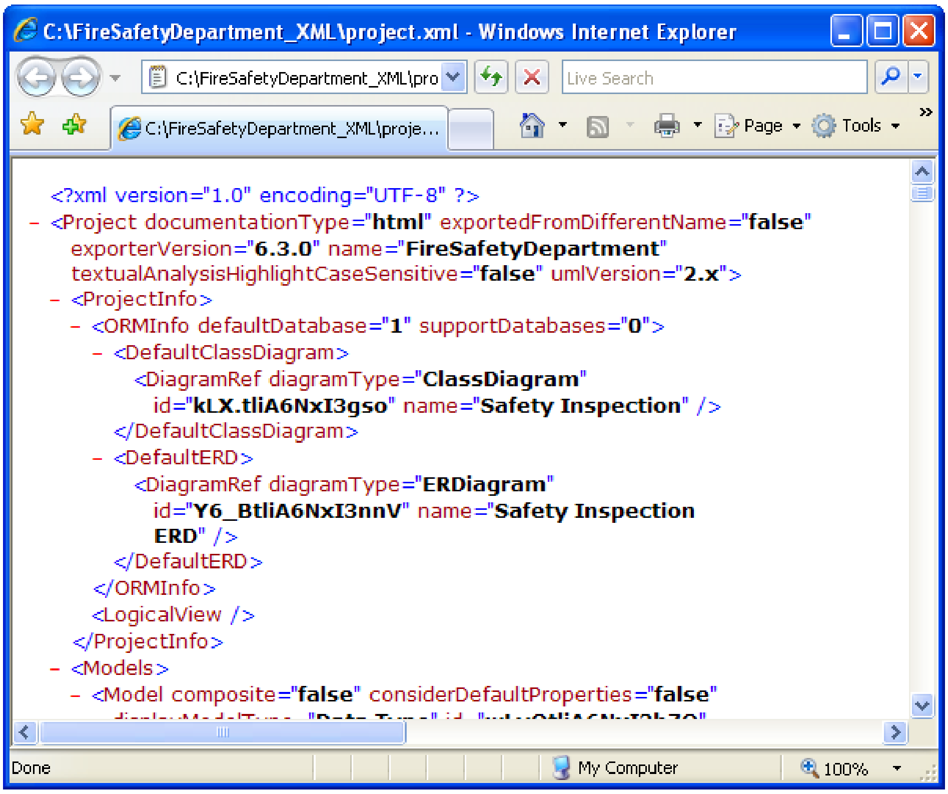
Download_Inspection_Report_to_PDA_System_level.png
project.xml
Regular_Inspection_Asis_Process.png
Regular_Inspection_Tobe_Process.png
Regular_Inspection_Use_Case_Diagram.png
Safety_Inspection.png
Safety_Inspection_ERD.png
Schedule_Inspection_MVC_level.png
Schedule_Inspection_System_level.png
Sequence_Diagram1.png

FireSafetyDepartment_X
File Folder

10 objects 2.22 MB My Computer

Figure 1-8 *Review exported content*

9. Review exported XML.



C:\FireSafetyDepartment_XML\project.xml - Windows Internet Explorer
C:\FireSafetyDepartment_XML\pro
Live Search
C:\FireSafetyDepartment_XML\proje...
Page
Tools
Done
My Computer
100%

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Project documentationType="html" exportedFromDifferentName="false"
    exporterVersion="6.3.0" name="FireSafetyDepartment"
    textualAnalysisHighlightCaseSensitive="false" umlVersion="2.x">
  - <ProjectInfo>
    - <ORMInfo defaultDatabase="1" supportDatabases="0">
      - <DefaultClassDiagram>
          <DiagramRef diagramType="ClassDiagram"
            id="kLX.tliA6NxI3gso" name="Safety Inspection" />
        </DefaultClassDiagram>
      - <DefaultERD>
          <DiagramRef diagramType="ERDiagram"
            id="Y6_BtliA6NxI3nnV" name="Safety Inspection
            ERD" />
        </DefaultERD>
      </ORMInfo>
      <LogicalView />
    </ProjectInfo>
  - <Models>
    - <Model composite="false" considerDefaultProperties="false"
```

Figure 1-9 Review exported XML

Exporting active diagram to XML

1. Right click on diagram and click on Export > Export XML... menu item.

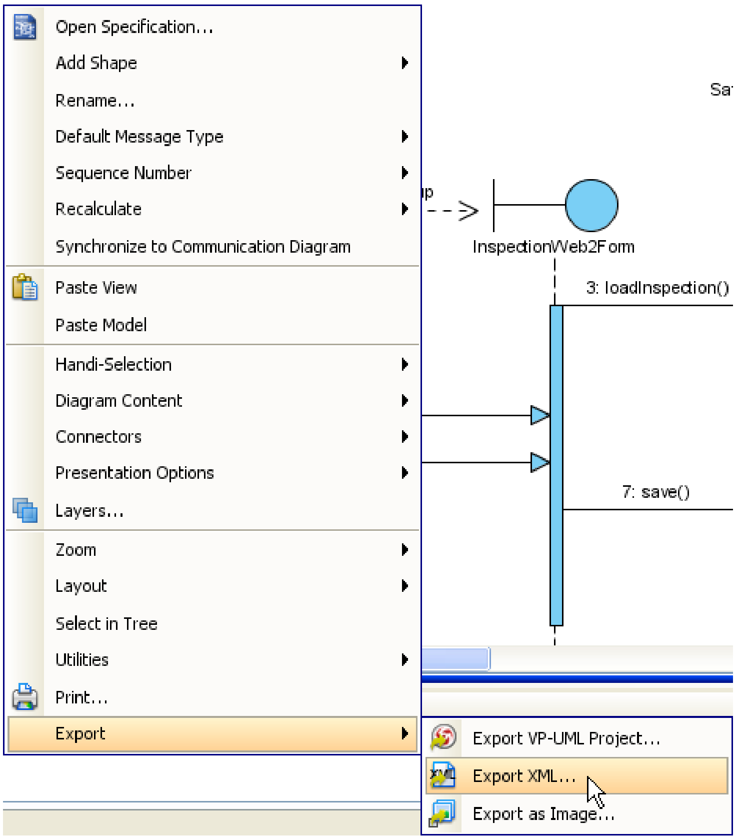
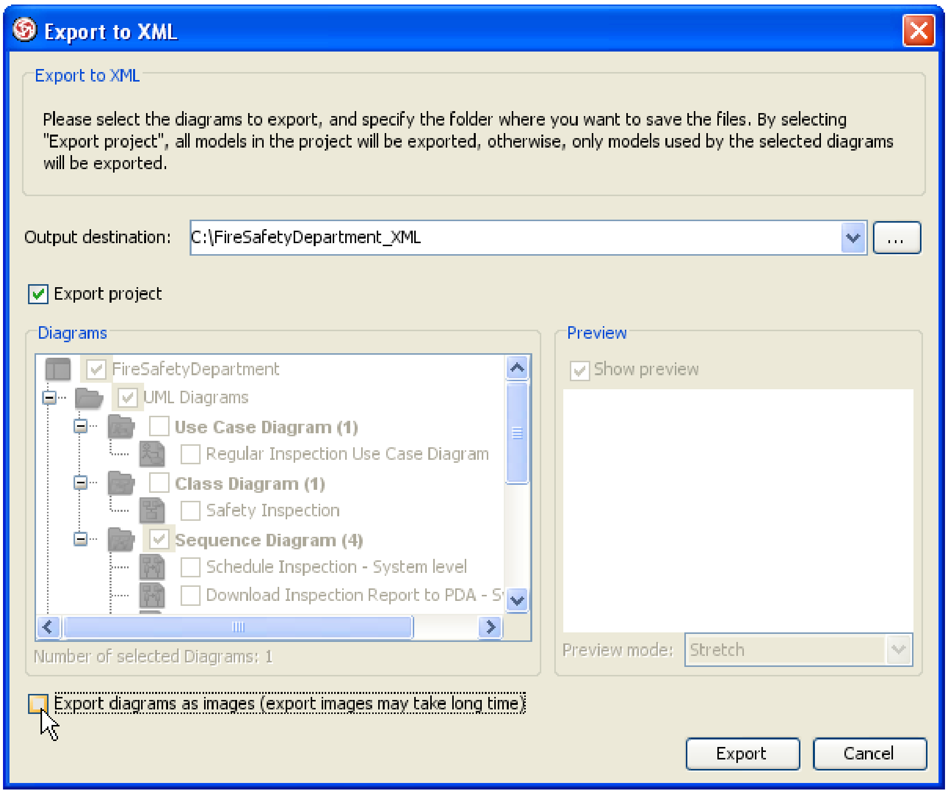


Figure 1-10 Export XML from diagram popup

2. Uncheck Export diagrams as images (export images may take long time) checkbox.



Export to XML

Export to XML

Please select the diagrams to export, and specify the folder where you want to save the files. By selecting "Export project", all models in the project will be exported, otherwise, only models used by the selected diagrams will be exported.

Output destination: C:\FireSafetyDepartment_XML

☑ Export project

Diagrams

- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - □ Use Case Diagram (1)
      - □ Regular Inspection Use Case Diagram
    - □ Class Diagram (1)
      - □ Safety Inspection
    - ☑ Sequence Diagram (4)
      - □ Schedule Inspection - System level
      - □ Download Inspection Report to PDA - S

Number of selected Diagrams: 1

Preview

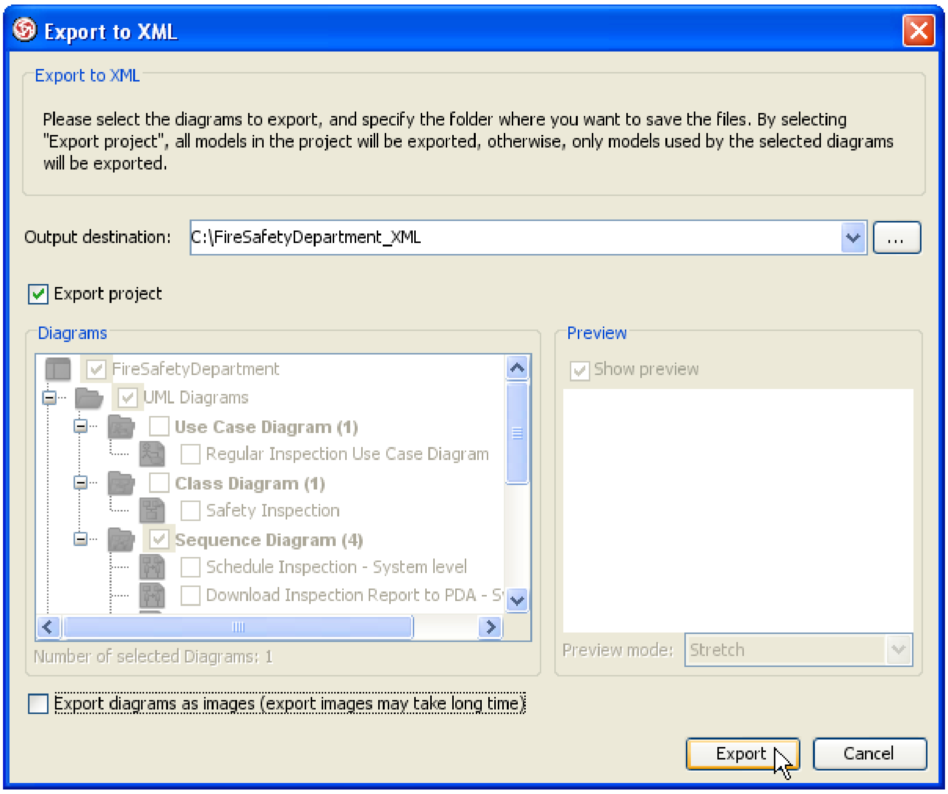
☑ Show preview

Preview mode: Stretch

□ Export diagrams as images (export images may take long time)

Figure 1-11 Uncheck export diagrams as images checkbox

3. Press Export button.



Export to XML

Export to XML

Please select the diagrams to export, and specify the folder where you want to save the files. By selecting "Export project", all models in the project will be exported, otherwise, only models used by the selected diagrams will be exported.

Output destination: C:\FireSafetyDepartment_XML

☑ Export project

Diagrams

- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - □ Use Case Diagram (1)
      - □ Regular Inspection Use Case Diagram
    - □ Class Diagram (1)
      - □ Safety Inspection
    - ☑ Sequence Diagram (4)
      - □ Schedule Inspection - System level
      - □ Download Inspection Report to PDA - S

Number of selected Diagrams: 1

Preview

☑ Show preview

Preview mode: Stretch

□ Export diagrams as images (export images may take long time)

Figure 1-12 Export to XML

4. Review exported content.

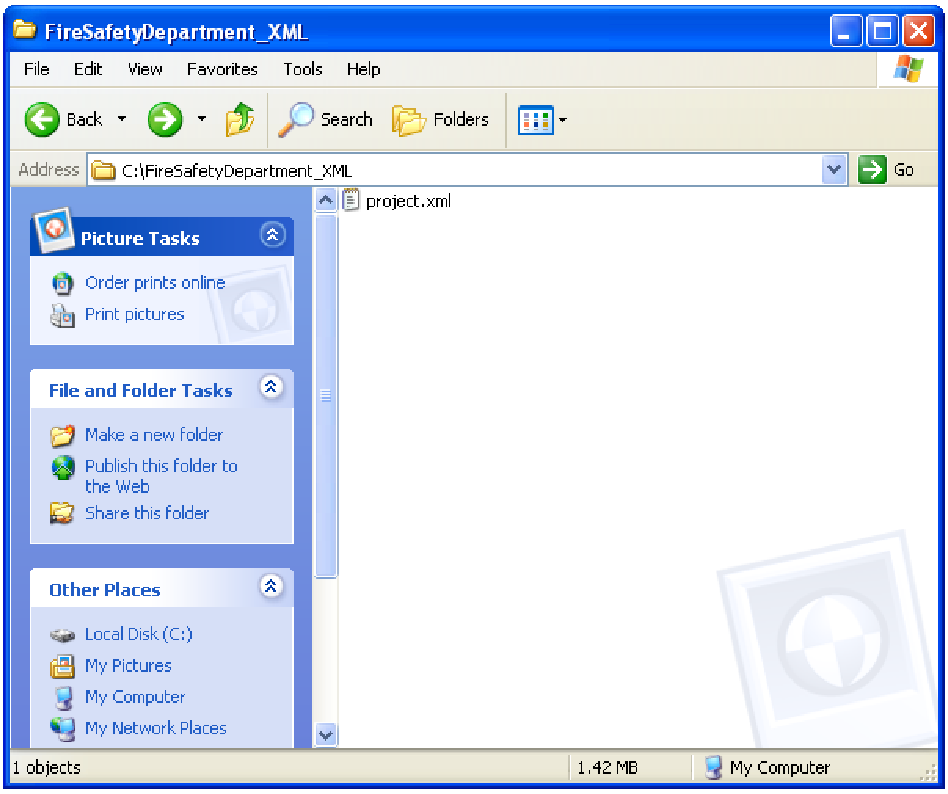
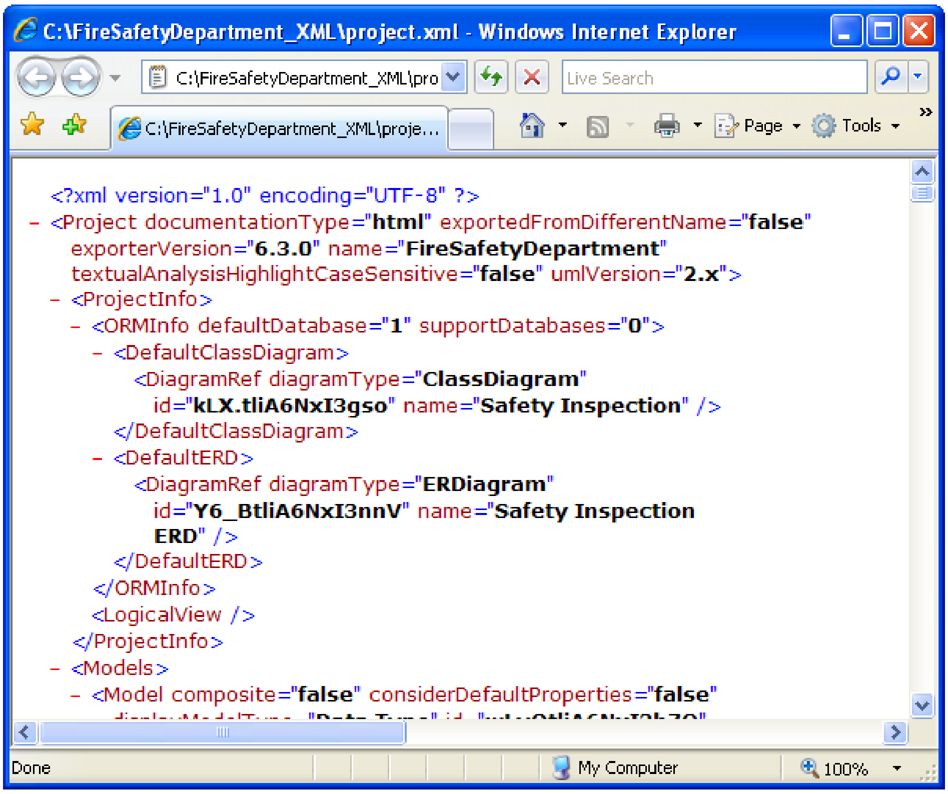


Figure 1-13 Review exported content

5. Review exported XML.


C:\FireSafetyDepartment_XML\project.xml - Windows Internet Explorer
C:\FireSafetyDepartment_XML\pro
Live Search
C:\FireSafetyDepartment_XML\proje...
Page
Tools
Done
My Computer
100%

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Project documentationType="html" exportedFromDifferentName="false"
    exporterVersion="6.3.0" name="FireSafetyDepartment"
    textualAnalysisHighlightCaseSensitive="false" umlVersion="2.x">
  - <ProjectInfo>
    - <ORMInfo defaultDatabase="1" supportDatabases="0">
      - <DefaultClassDiagram>
          <DiagramRef diagramType="ClassDiagram"
            id="kLX.tliA6NxI3gso" name="Safety Inspection" />
        </DefaultClassDiagram>
      - <DefaultERD>
          <DiagramRef diagramType="ERDiagram"
            id="Y6_BtliA6NxI3nnV" name="Safety Inspection
            ERD" />
        </DefaultERD>
      </ORMInfo>
      <LogicalView />
    </ProjectInfo>
  - <Models>
    - <Model composite="false" considerDefaultProperties="false"
```

Figure 1-14 Review exported XML

Exporting diagrams to XML with command line interface

1. Mouse over Start button.

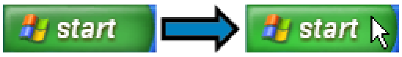


Figure 1-15 Mouse over start button

2. Press on Start button.

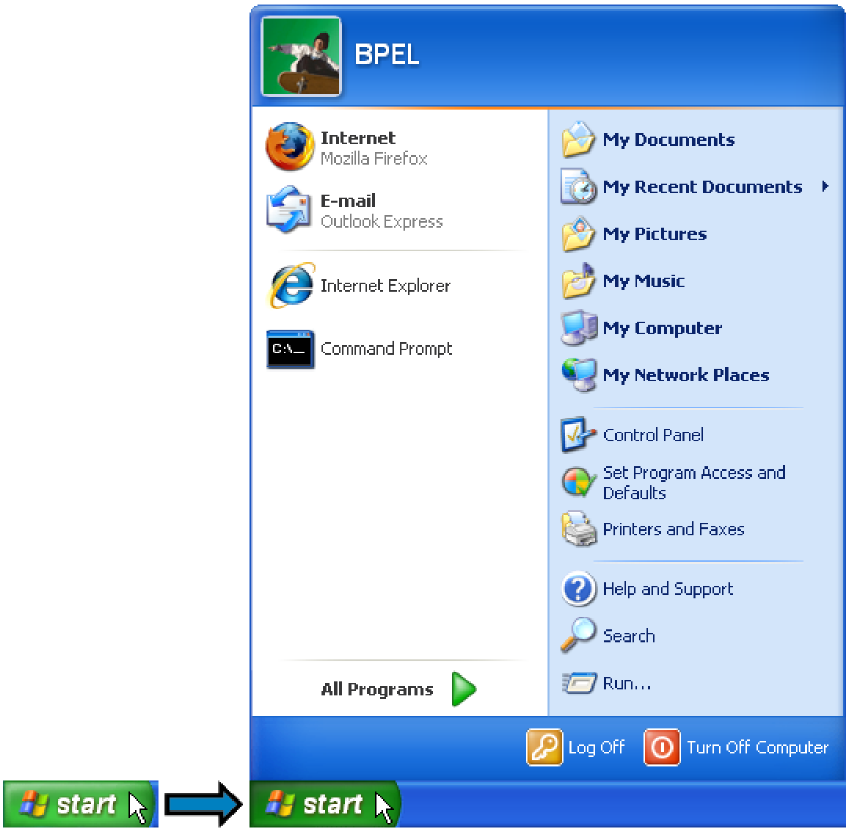


Figure 1-16 Showing start menu

3. Click on Run... button.

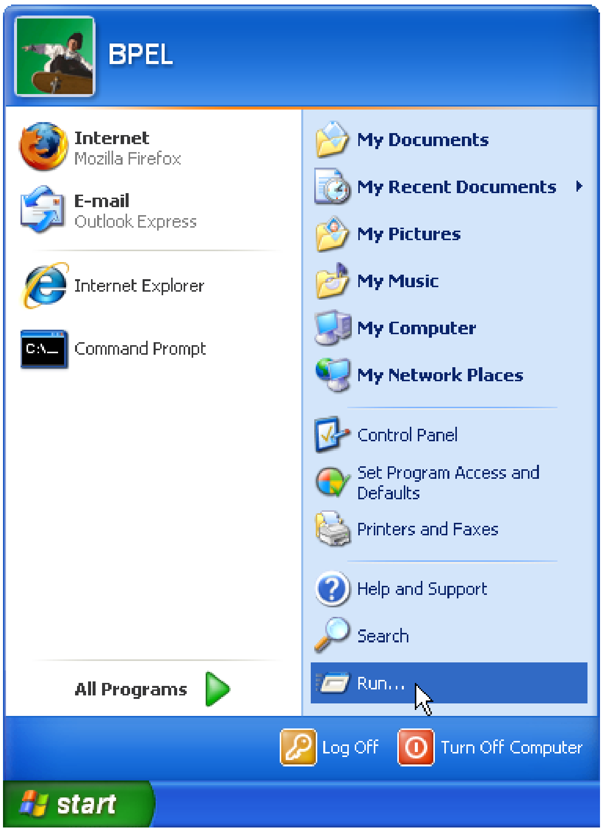
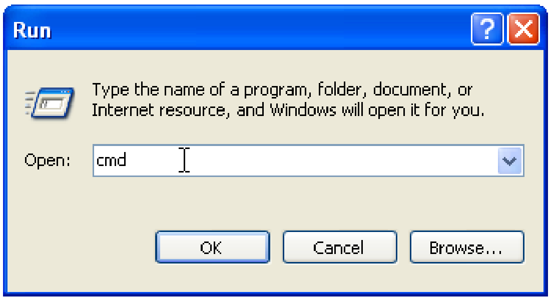


Figure 1-17 *Run from start menu*

4. Input cmd as command.

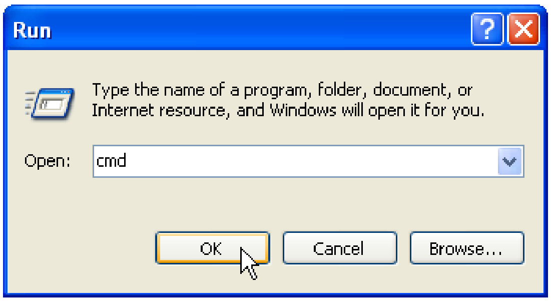


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Figure 1-18 *Inputting command to run*

5. Press OK button.

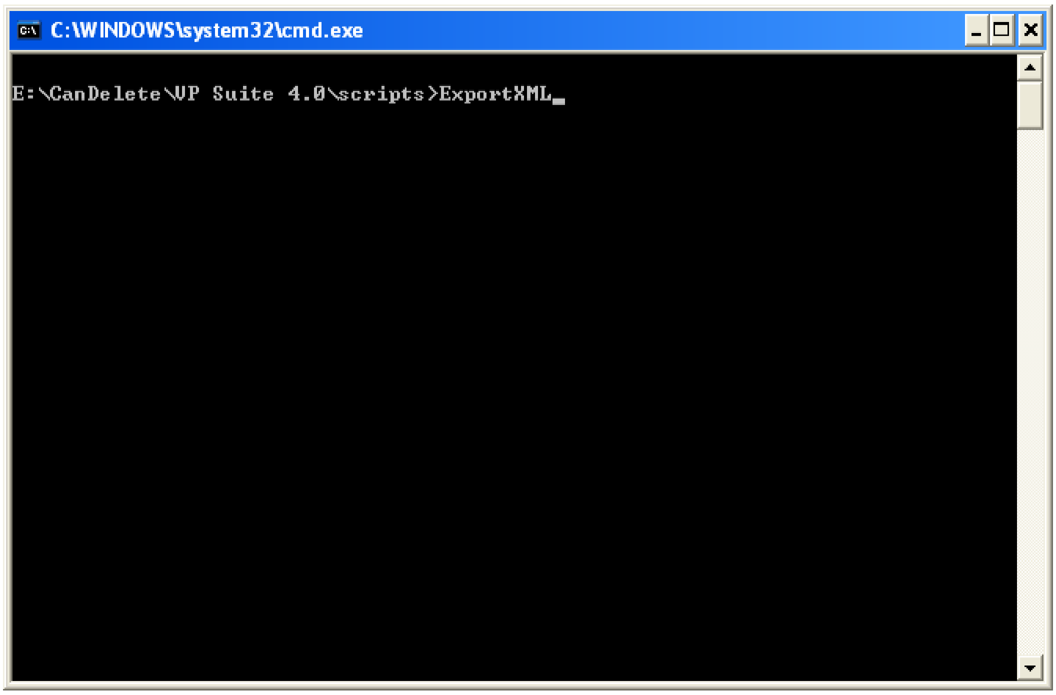


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Figure 1-19 *Run command*

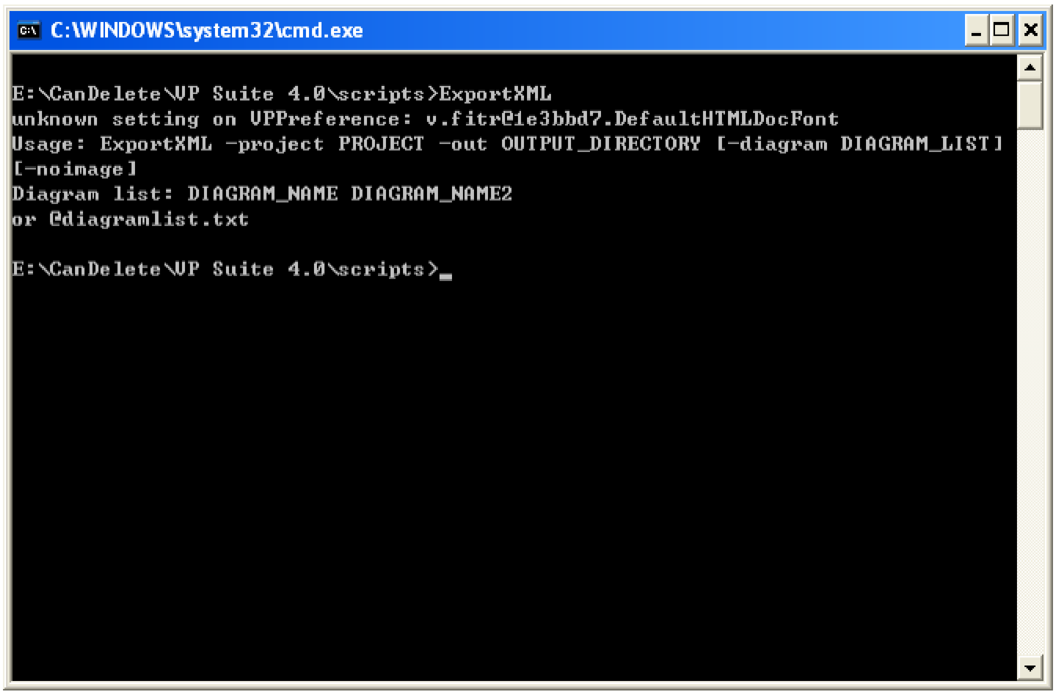
6. Change directory to scripts folder and execute ExportXML



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXML_
```

Figure 1-20 *Inputting command*

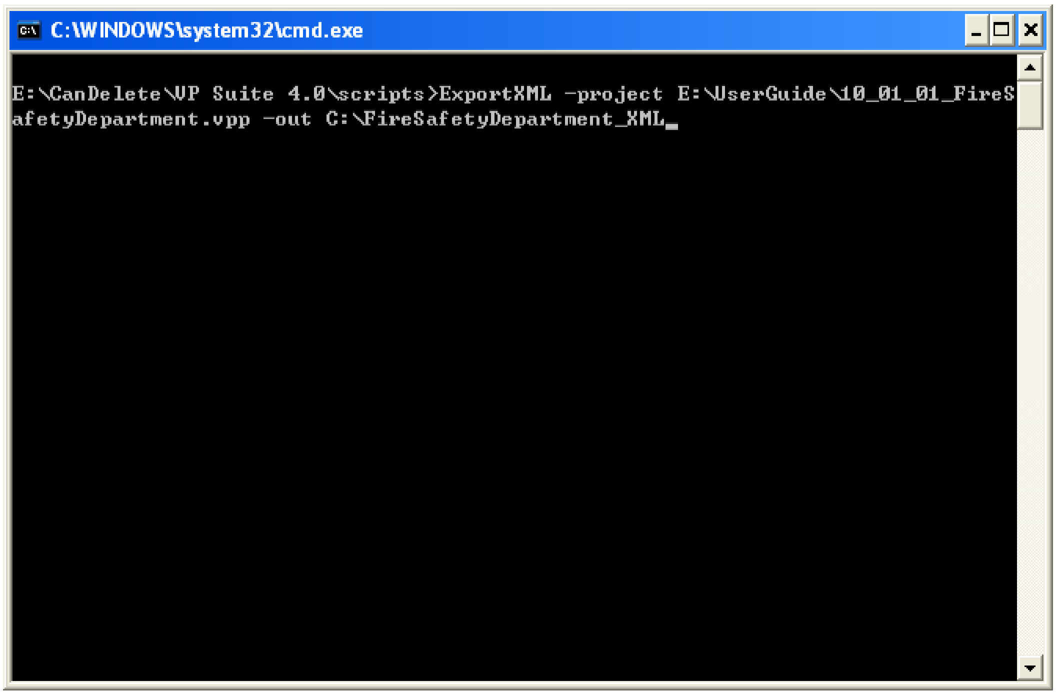
7. Usage of ExportXML will be shown.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXML
unknown setting on VPPreference: v.fitr@1e3bbd7.DefaultHTMLDocFont
Usage: ExportXML -project PROJECT -out OUTPUT_DIRECTORY [-diagram DIAGRAM_LIST]
[-noimage]
Diagram list: DIAGRAM_NAME DIAGRAM_NAME2
or @diagramlist.txt

E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 1-21 *Execute result of command*

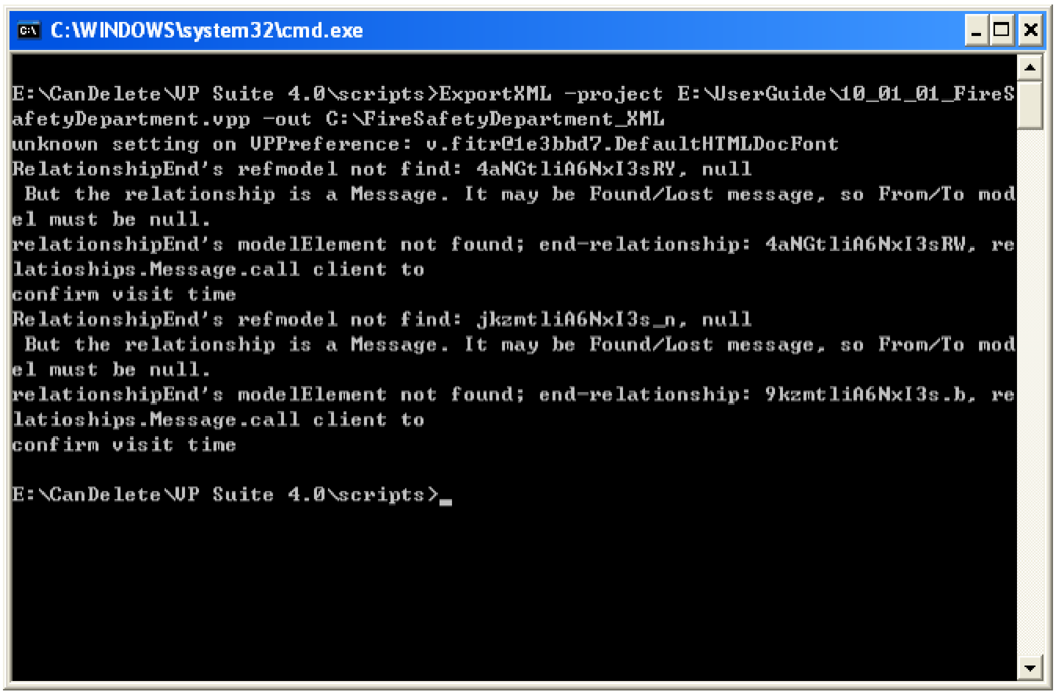
8. Example of exporting XML for whole project.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXML -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XML_
```

Figure 1-22 Inputting command

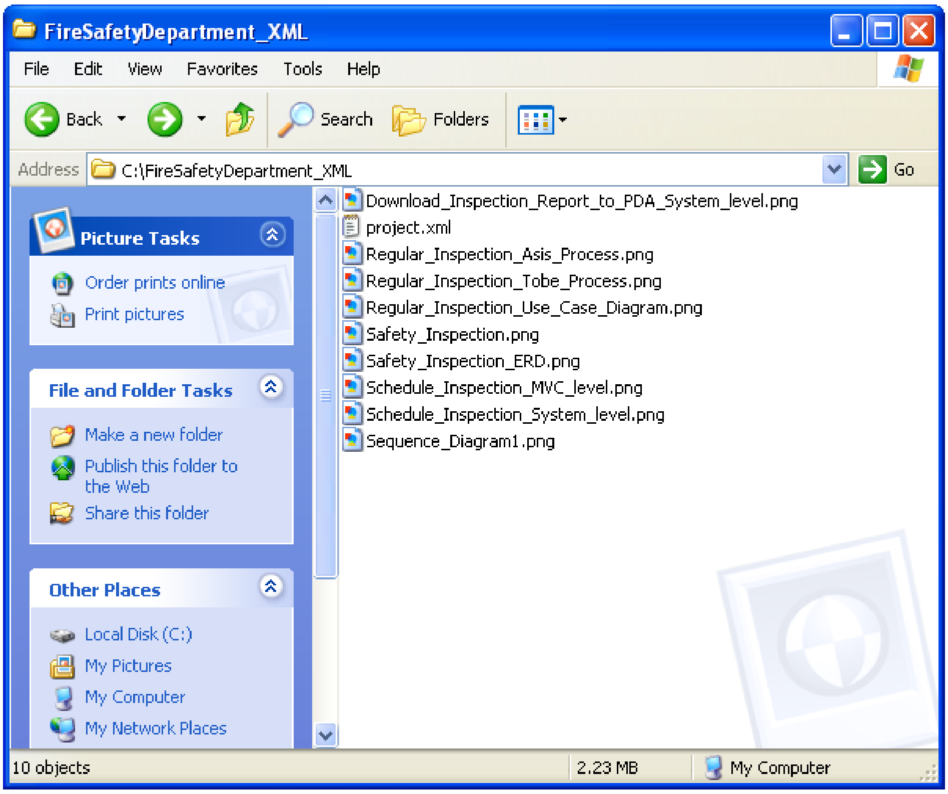
9. Execute result of exporting XML.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXML -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XML
unknown setting on VPPreference: v.fitr@1e3bbd7.DefaultHTMLDocFont
RelationshipEnd's refmodel not find: 4aNGtliA6NxI3sRY, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 4aNGtliA6NxI3sRW, re
latioships.Message.call client to
confirm visit time
RelationshipEnd's refmodel not find: jkzmtliA6NxI3s_n, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 9kzmtliA6NxI3s.b, re
latioships.Message.call client to
confirm visit time

E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 1-23 Execute result

10. Review exported content.

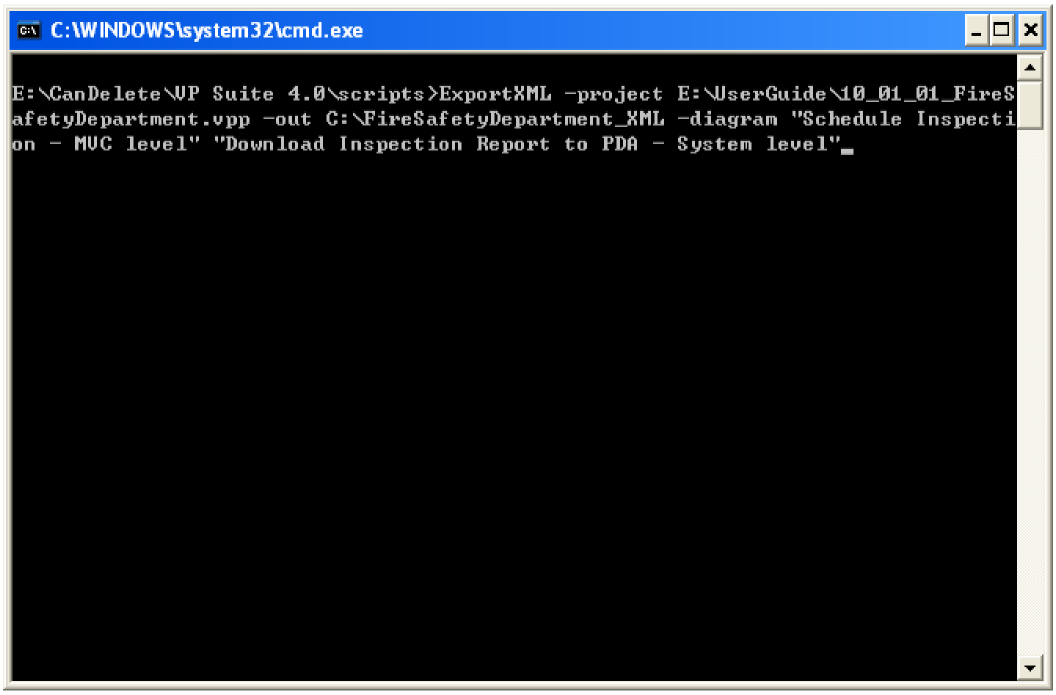


Download_Inspection_Report_to_PDA_System_level.png
project.xml
Regular_Inspection_Asis_Process.png
Regular_Inspection_Tobe_Process.png
Regular_Inspection_Use_Case_Diagram.png
Safety_Inspection.png
Safety_Inspection_ERD.png
Schedule_Inspection_MVC_level.png
Schedule_Inspection_System_level.png
Sequence_Diagram1.png

10 objects 2.23 MB My Computer

Figure 1-24 *Review exported content*

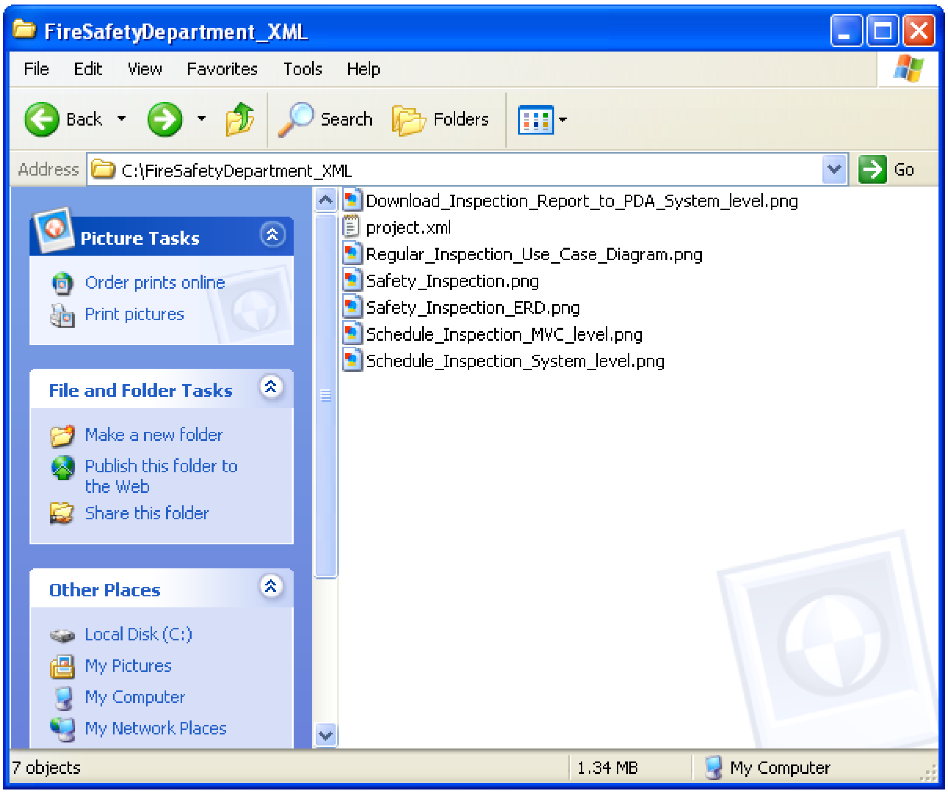
11. Example of exporting XML with specified diagrams.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXML -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XML -diagram "Schedule Inspecti
on - MVC level" "Download Inspection Report to PDA - System level"_
```

Figure 1-25 *Inputting command*

12. Review exported content.



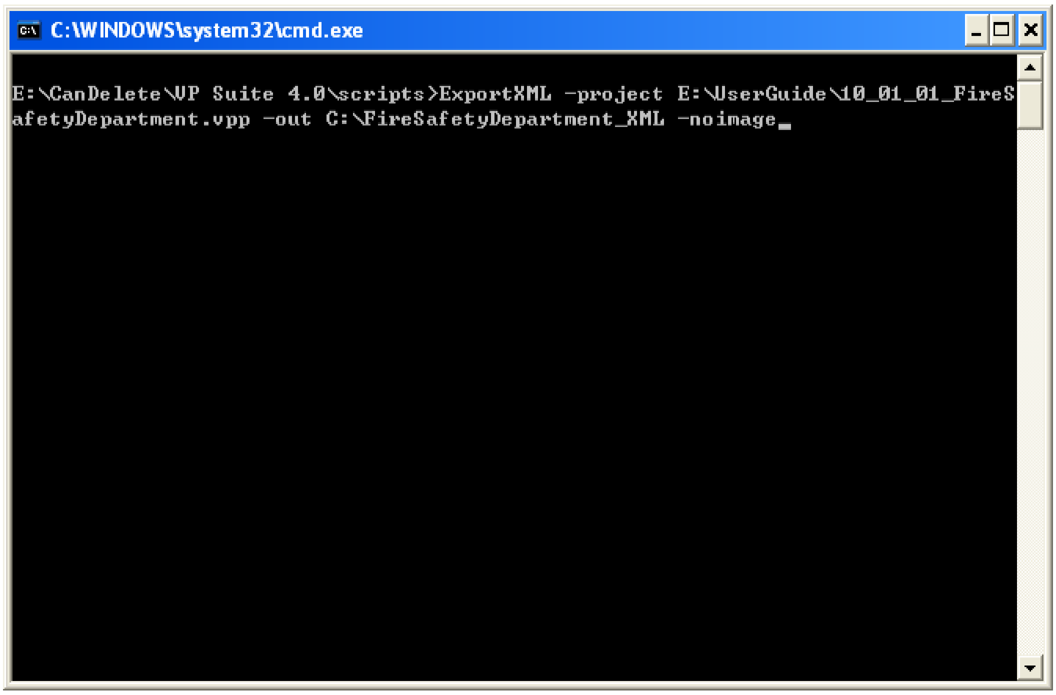
Address: C:\FireSafetyDepartment_XML

- Download_Inspection_Report_to_PDA_System_level.png
- project.xml
- Regular_Inspection_Use_Case_Diagram.png
- Safety_Inspection.png
- Safety_Inspection_ERD.png
- Schedule_Inspection_MVC_level.png
- Schedule_Inspection_System_level.png

7 objects 1.34 MB My Computer

Figure 1-26 Review exported content

13. Example of exporting XML without images.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXML -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XML -noimage_
```

Figure 1-27 Inputting command

14. Review exported content.

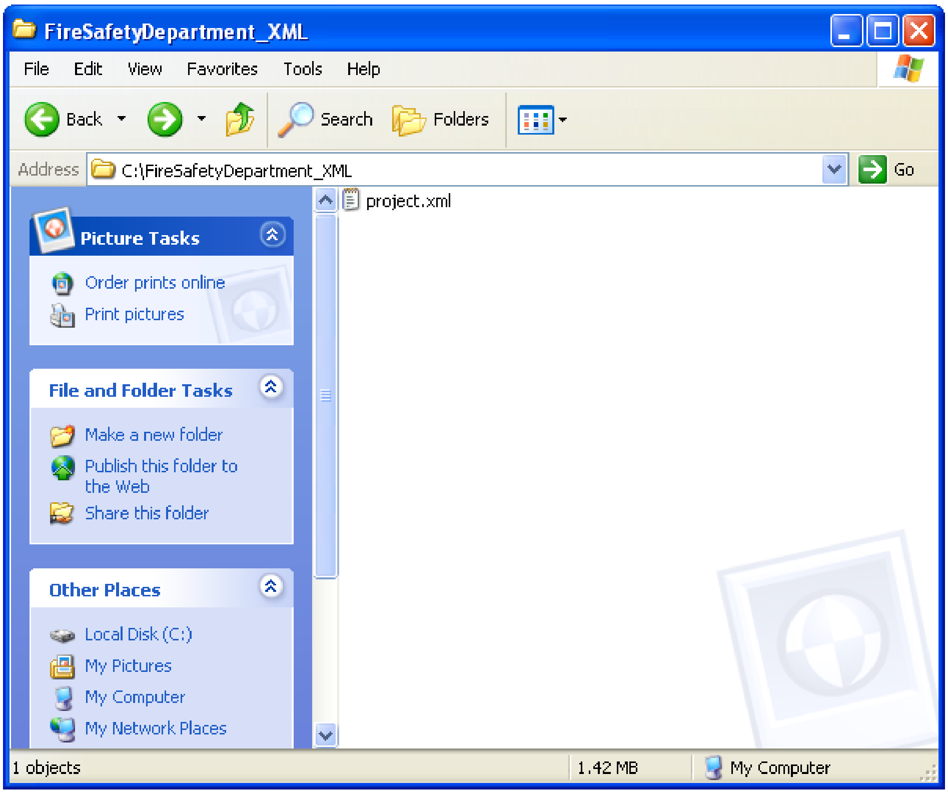
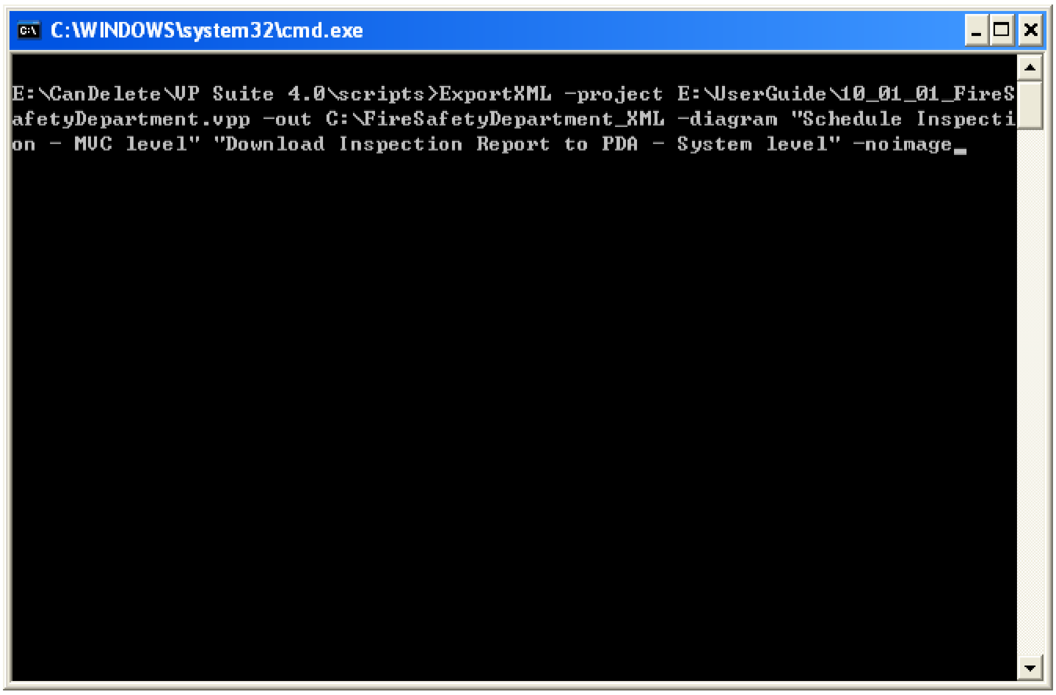


Figure 1-28 *Review exported content*

15. Example of exporting XML with specified diagrams and without images.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXML -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XML -diagram "Schedule Inspecti
on - MVC level" "Download Inspection Report to PDA - System level" -noimage_
```

Figure 1-29 *Inputting command*

16. Review exported content.

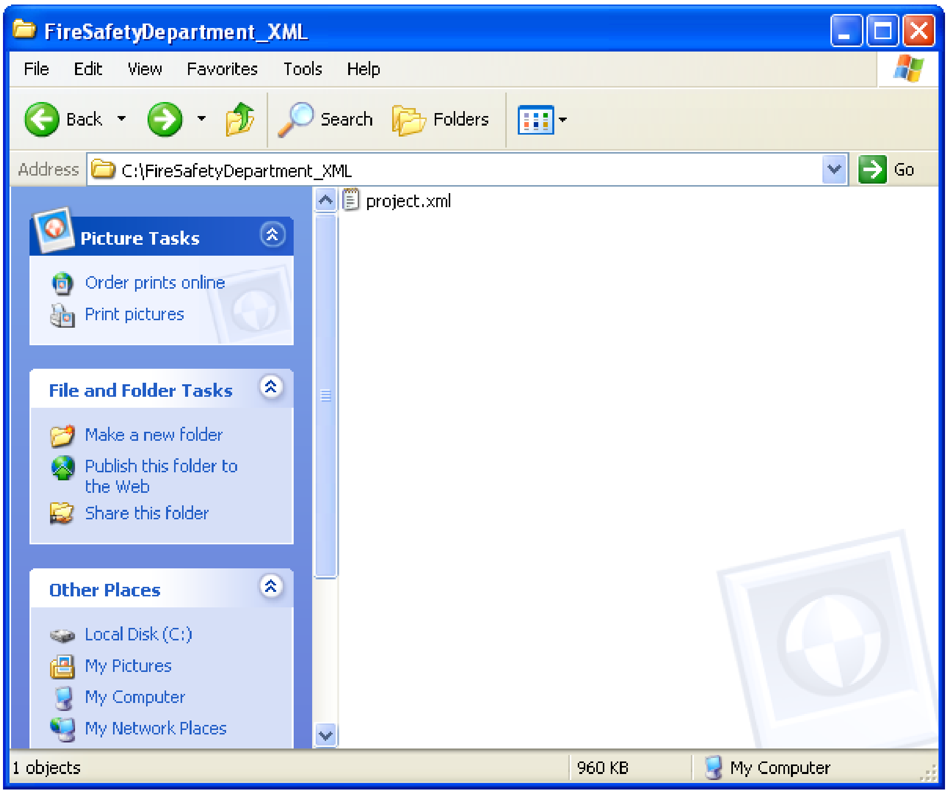


Figure 1-30 *Review exported content*

Importing XML

Importing XML to current project

1. Mouse over main menu.

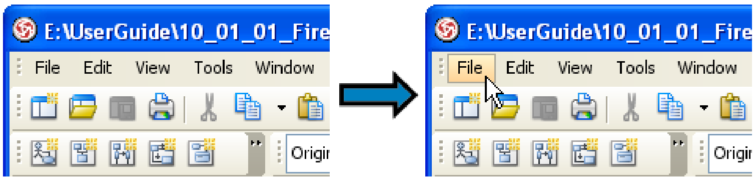

E:\UserGuide\10_01_01_Fire
File Edit View Tools Window
Origir
E:\UserGuide\10_01_01_Fire
File Edit View Tools Window
Origir

Figure 1-31 *Mouse over main menu*

2. Click on File of main menu.

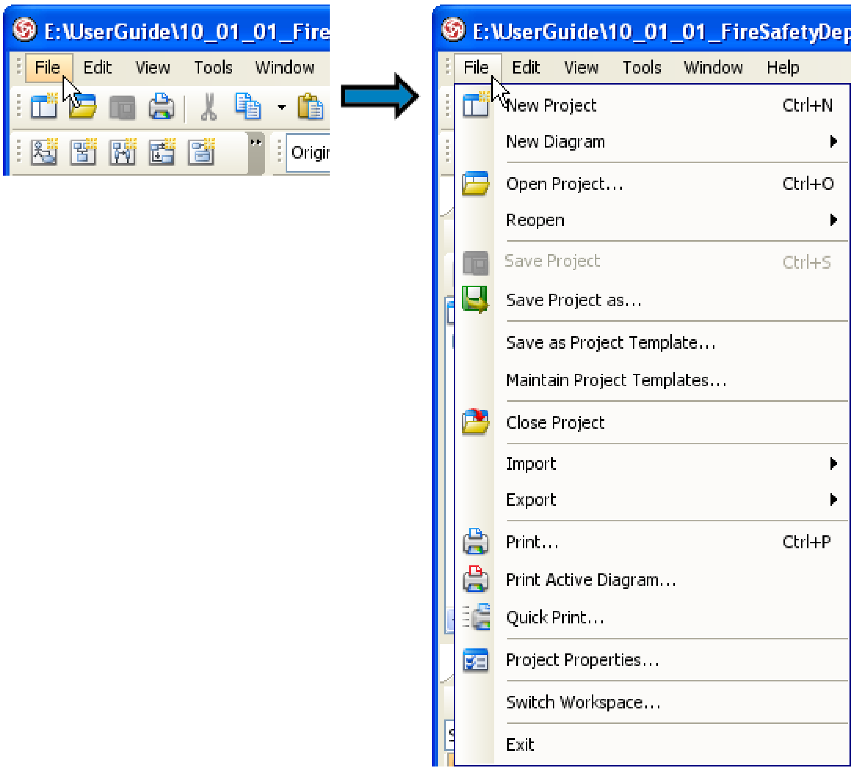


Figure 1-32 *Showing popup of main menu*

3. Click on Import > XML... menu item.

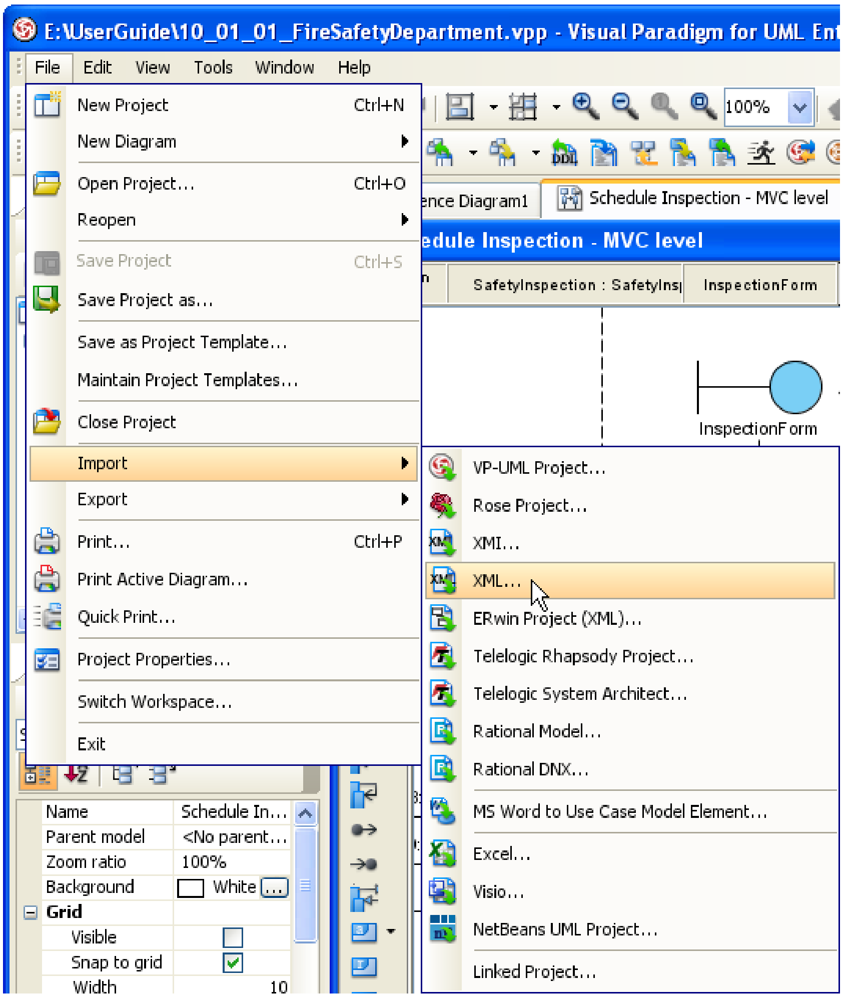
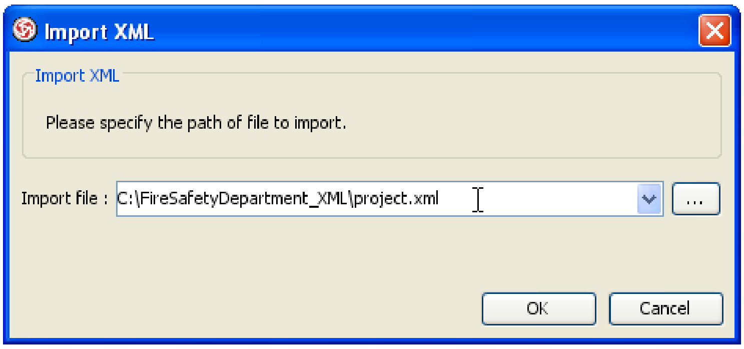


Figure 1-33 Import XML

4. Input Import file.



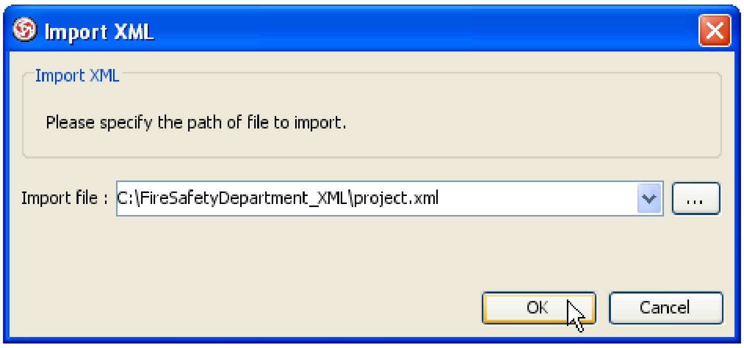
Import XML

Please specify the path of file to import.

Import file : C:\FireSafetyDepartment_XML\project.xml

Figure 1-34 Inputting import file

5. Press OK button.


Import XML

Import XML

Please specify the path of file to import.

Import file : C:\FireSafetyDepartment_XML\project.xml

Figure 1-35 Import from XML

Importing XML to project with command line interface

1. Mouse over Start button.

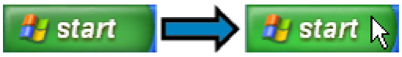


Figure 1-36 Mouse over start button

2. Press on Start button.

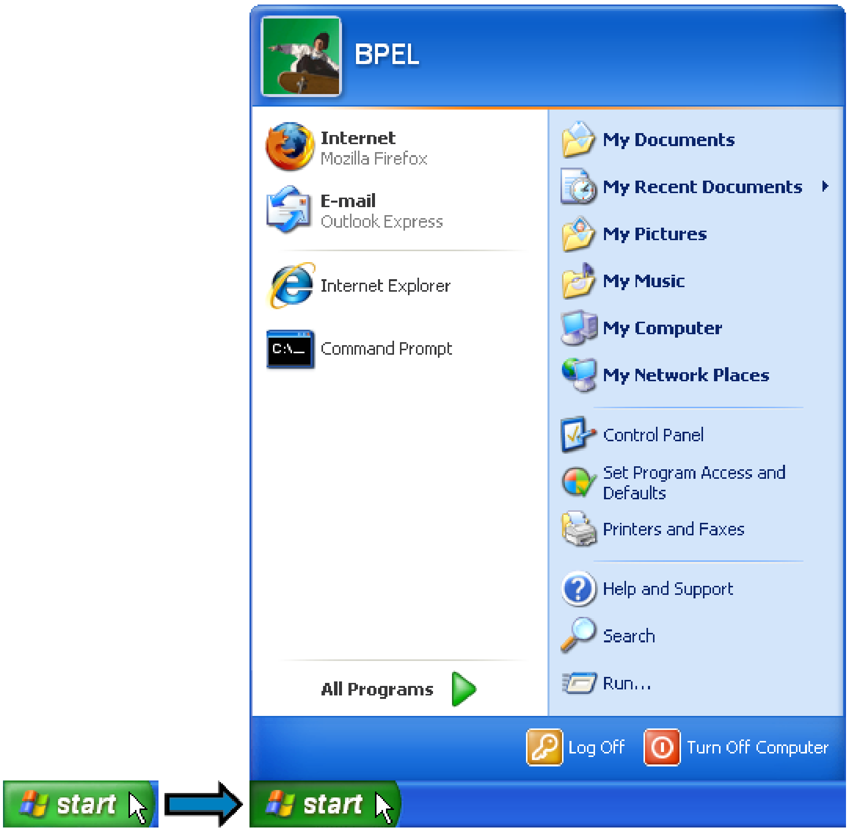


Figure 1-37 Showing start menu

3. Click on Run... button.

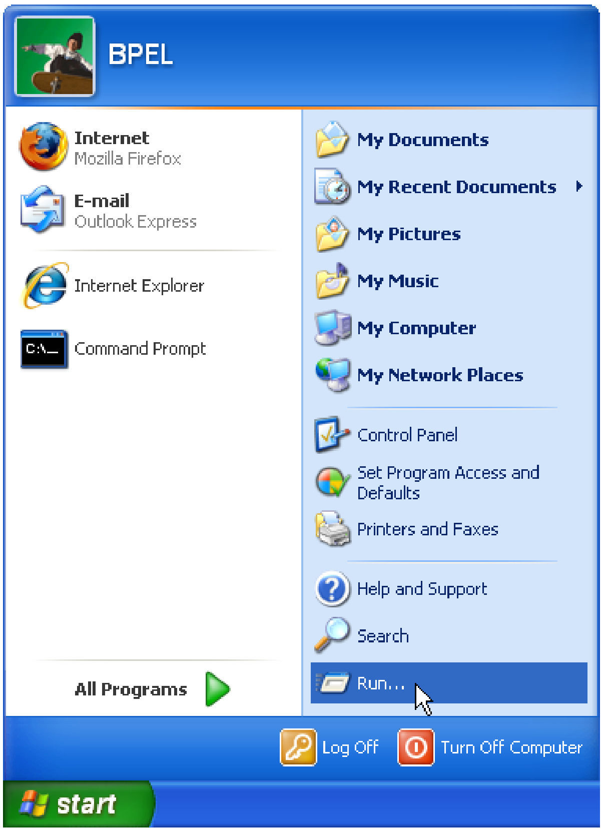
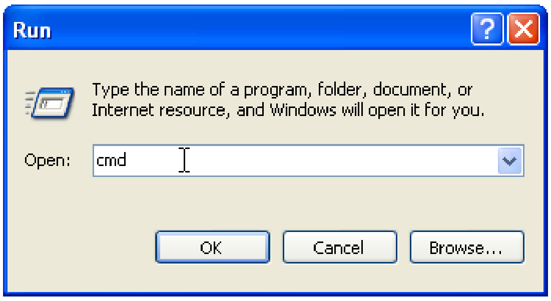


Figure 1-38 *Run from start menu*

4. Input cmd as command.

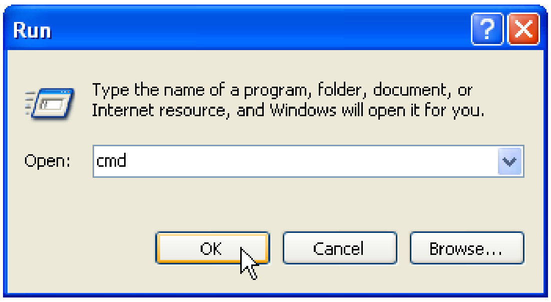


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Figure 1-39 *Inputting command to run*

5. Press OK button.

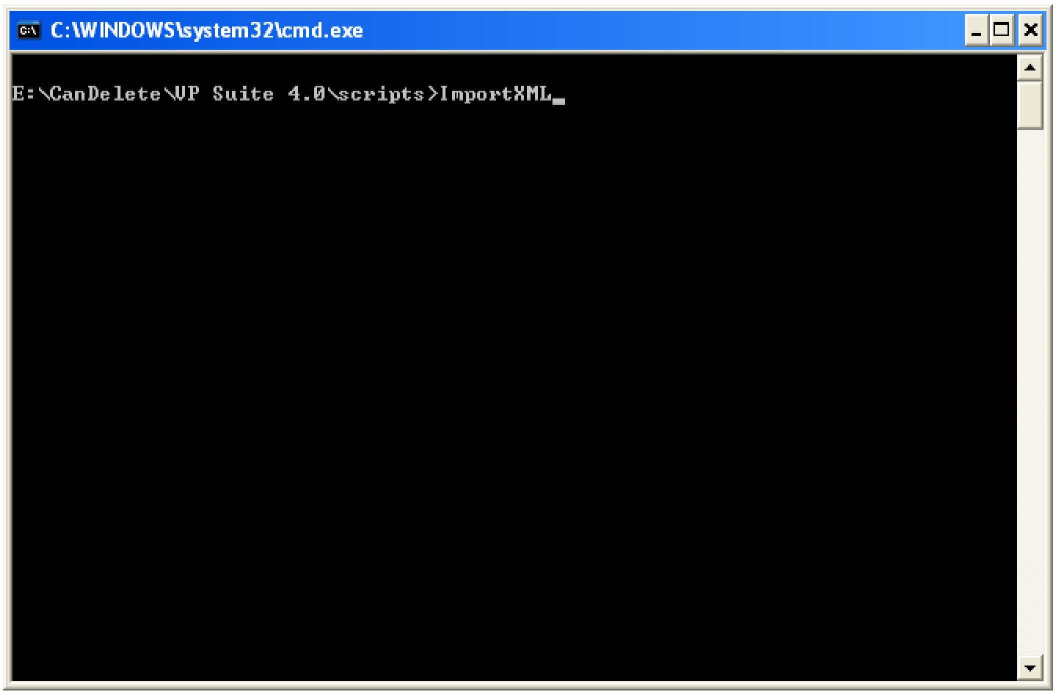


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Figure 1-40 *Run command*

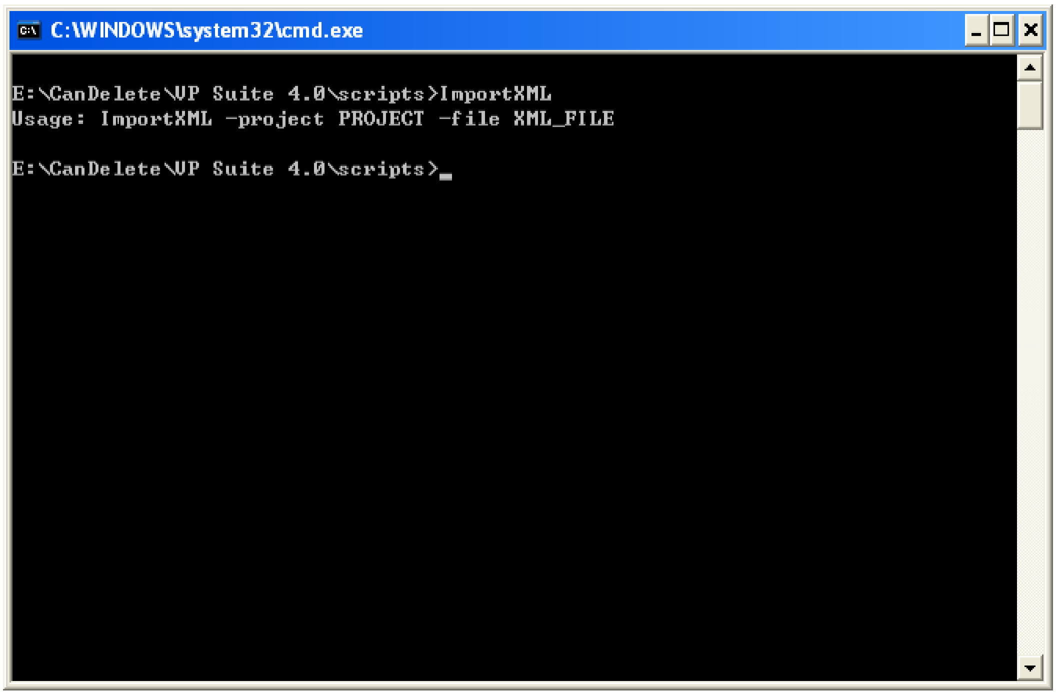
6. Change directory to scripts folder and execute ImportXML



```
E:\CanDelete\VP Suite 4.0\scripts>ImportXML_
```

Figure 1-41 *Inputting command*

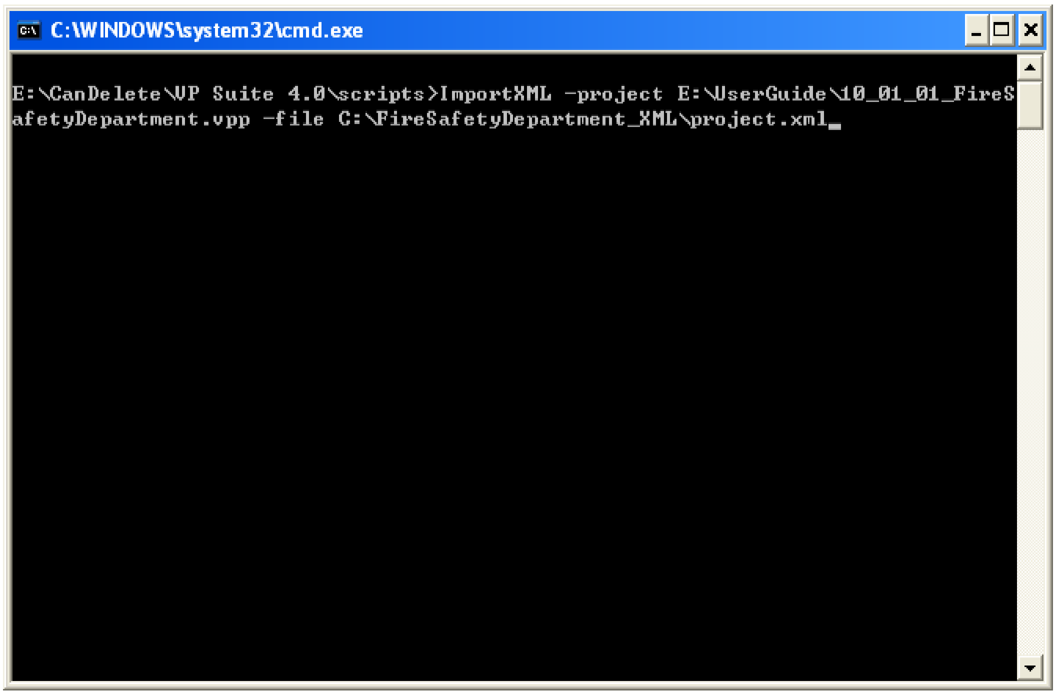
7. Usage of ImportXML will be shown.



```
E:\CanDelete\VP Suite 4.0\scripts>ImportXML
Usage: ImportXML -project PROJECT -file XML_FILE

E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 1-42 *Execute result of command*

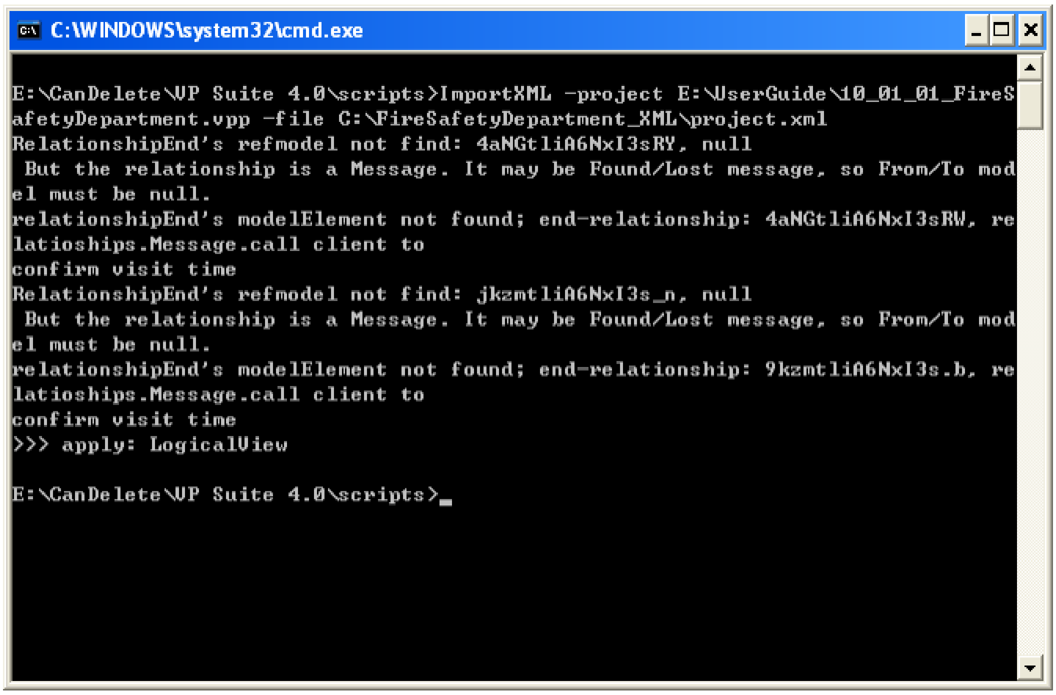
8. Example of importing XML



```
E:\CanDelete\VP Suite 4.0\scripts>ImportXML -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -file C:\FireSafetyDepartment_XML\project.xml_
```

Figure 1-43 *Inputting command*

9. Execute result of importing XML.



```
E:\CanDelete\VP Suite 4.0\scripts>ImportXML -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -file C:\FireSafetyDepartment_XML\project.xml
RelationshipEnd's refmodel not find: 4aNGtliA6NxI3sRY, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 4aNGtliA6NxI3sRW, re
latioships.Message.call client to
confirm visit time
RelationshipEnd's refmodel not find: jkzmtliA6NxI3s_n, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 9kzmtliA6NxI3s.b, re
latioships.Message.call client to
confirm visit time
>>> apply: LogicalView

E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 1-44 *Execute result*

Import overwrite mechanism

Importing XML will overwrite all covered models from XML

1. Export XML from main menu.

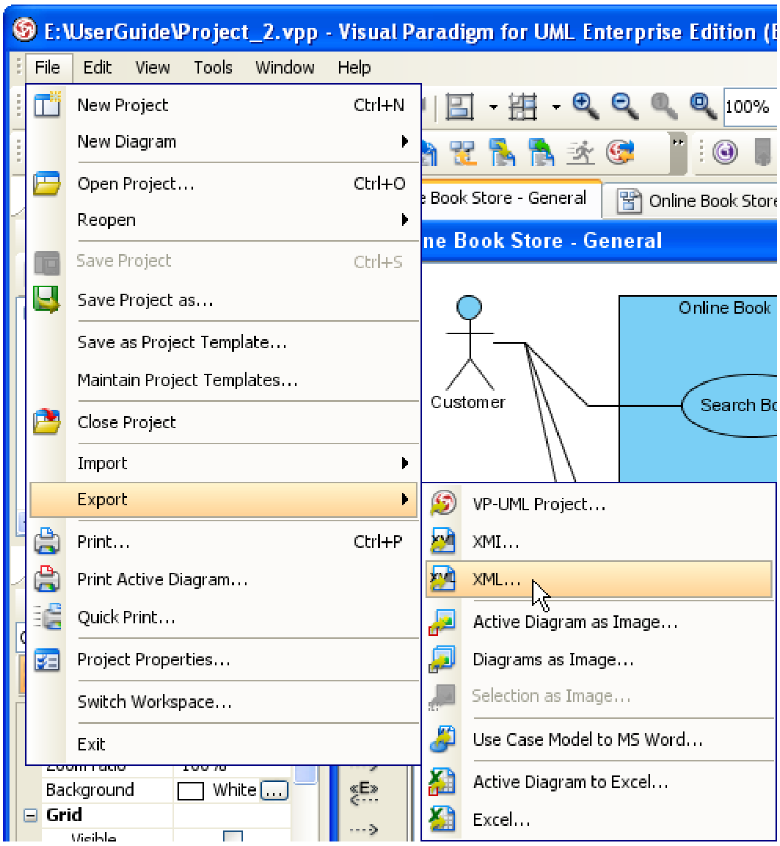
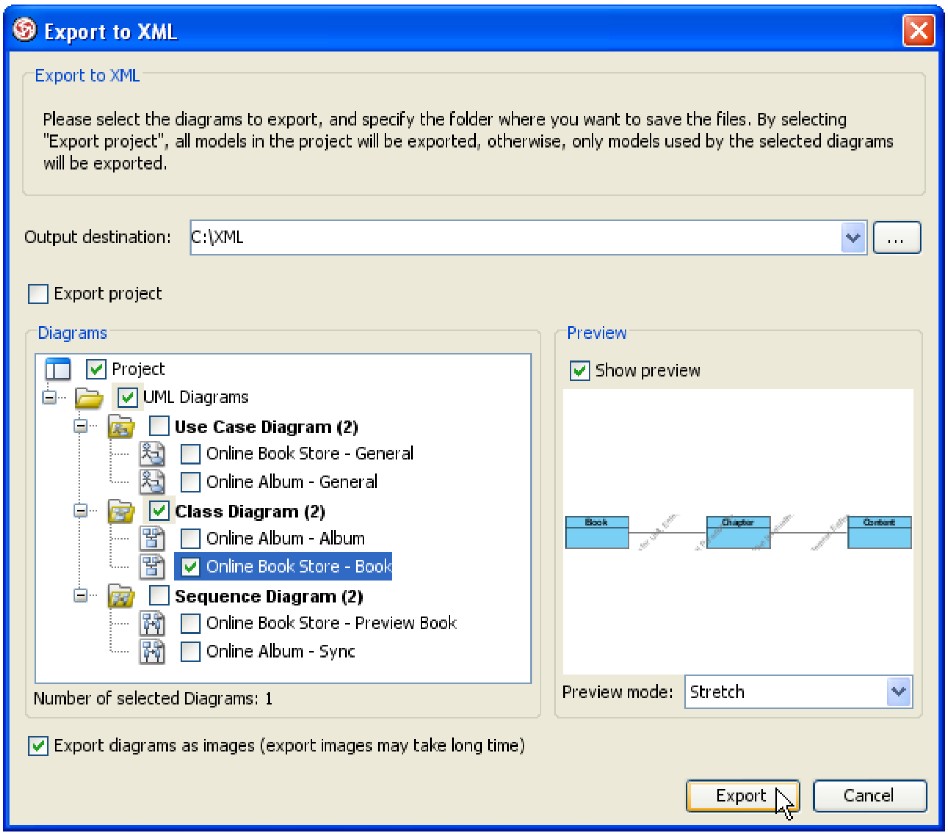


Figure 1-45 Exporting XML from main menu

2. Only select class diagram *Online Book Store - Book* and press Export button.



Export to XML

Export to XML

Please select the diagrams to export, and specify the folder where you want to save the files. By selecting "Export project", all models in the project will be exported, otherwise, only models used by the selected diagrams will be exported.

Output destination: C:\XML ...

☐ Export project

Diagrams

- ☑ Project
  - ☑ UML Diagrams
    - ☐ Use Case Diagram (2)
      - ☐ Online Book Store - General
      - ☐ Online Album - General
    - ☑ Class Diagram (2)
      - ☐ Online Album - Album
      - ☑ Online Book Store - Book
    - ☐ Sequence Diagram (2)
      - ☐ Online Book Store - Preview Book
      - ☐ Online Album - Sync

Number of selected Diagrams: 1

Preview

☑ Show preview

Preview mode: Stretch

☑ Export diagrams as images (export images may take long time)

Export Cancel

Figure 1-46 Export class diagram to XML

3. Rename use case *Search Book* to *Locate Book*.

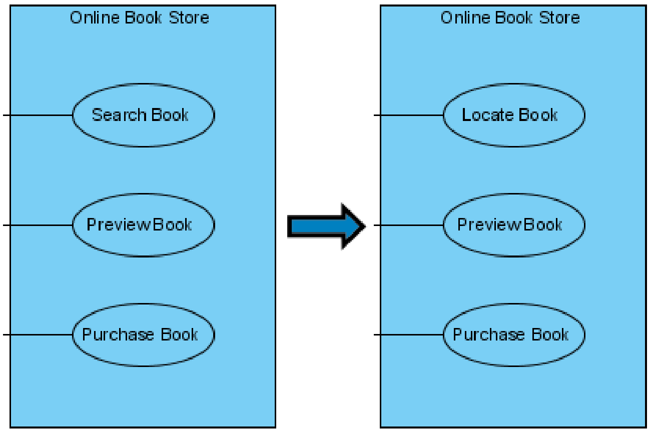


Figure 1-47 *Renaming use case*

4. Rename class *Content* to *Book Content*.

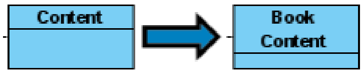


Figure 1-48 *Renaming class*

5. Import XML from main menu.

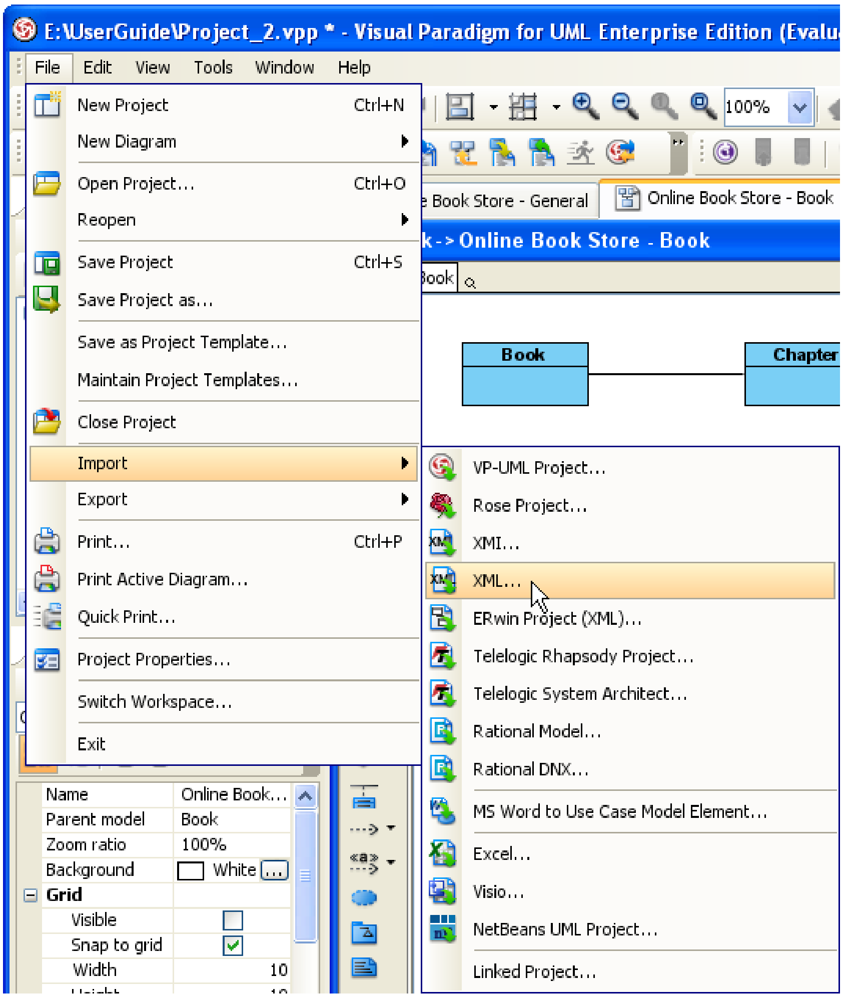
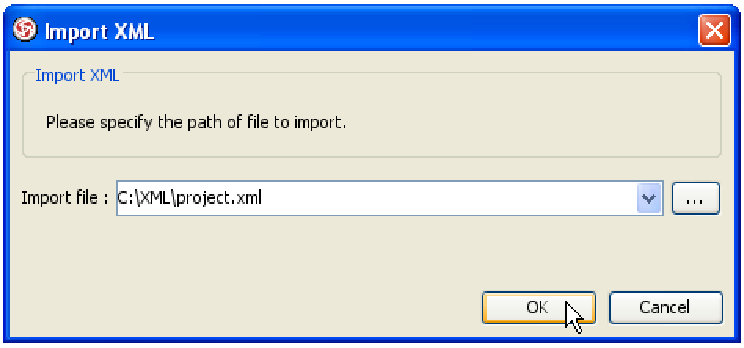


Figure 1-49 *Importing XML from main menu*

6. Press OK button.



Import XML

Import XML

Please specify the path of file to import.

Import file : C:\XML\project.xml

Figure 1-50 *Import XML*

7. Class *BookContent* is renamed back to *Content*.

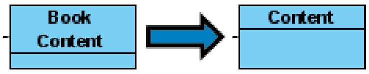


Figure 1-51 *Different of class after import XML*

8. Use case *Locate Book* is remain unchanged.

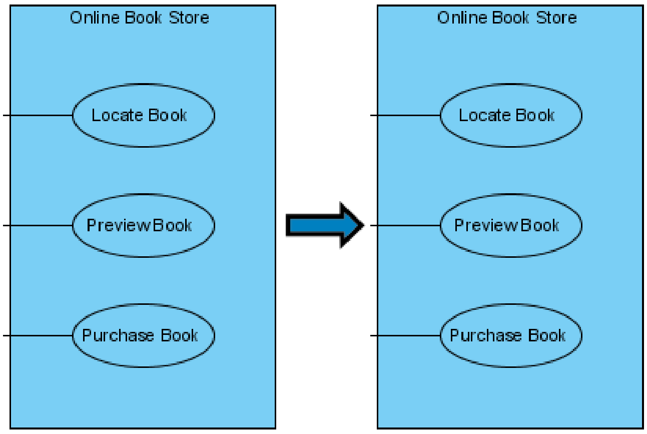


Figure 1-52 *Different of use case after import XML*

Exporting VP Project

Exporting selected diagrams to VP Project

1. Mouse over main menu.

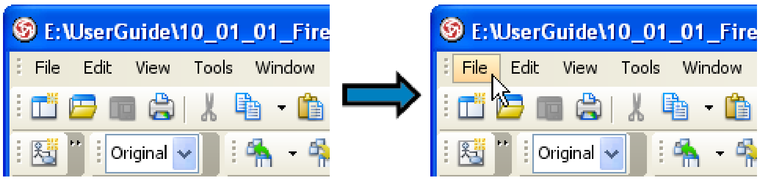


Figure 2-1 *Mouse over main menu*

2. Click on File of main menu.

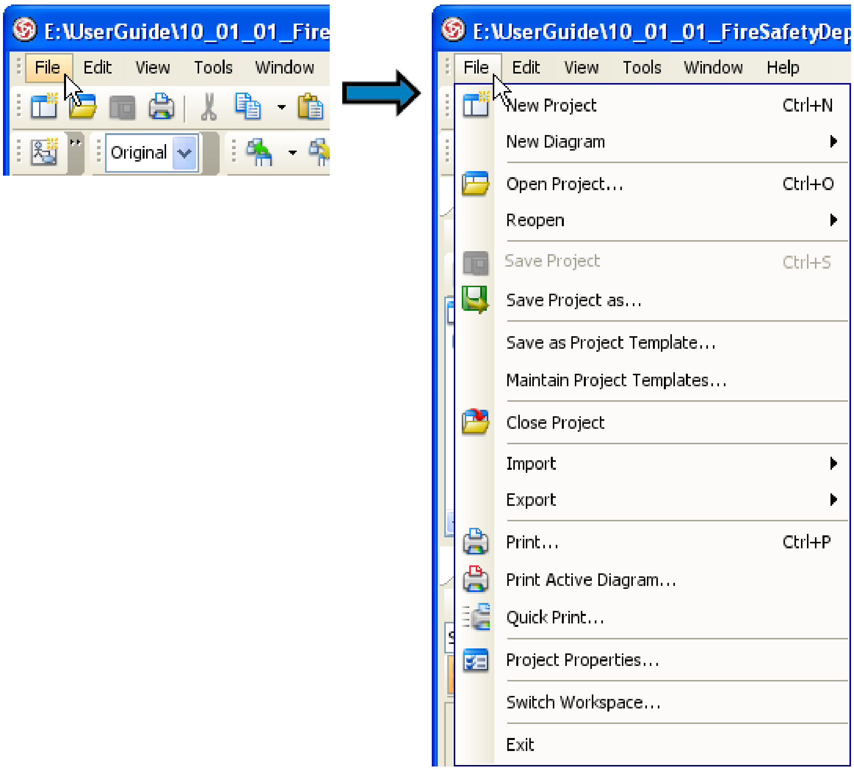


Figure 2-2 *Showing popup of main menu*

3. Click on Export > VP-UML Project... menu item.

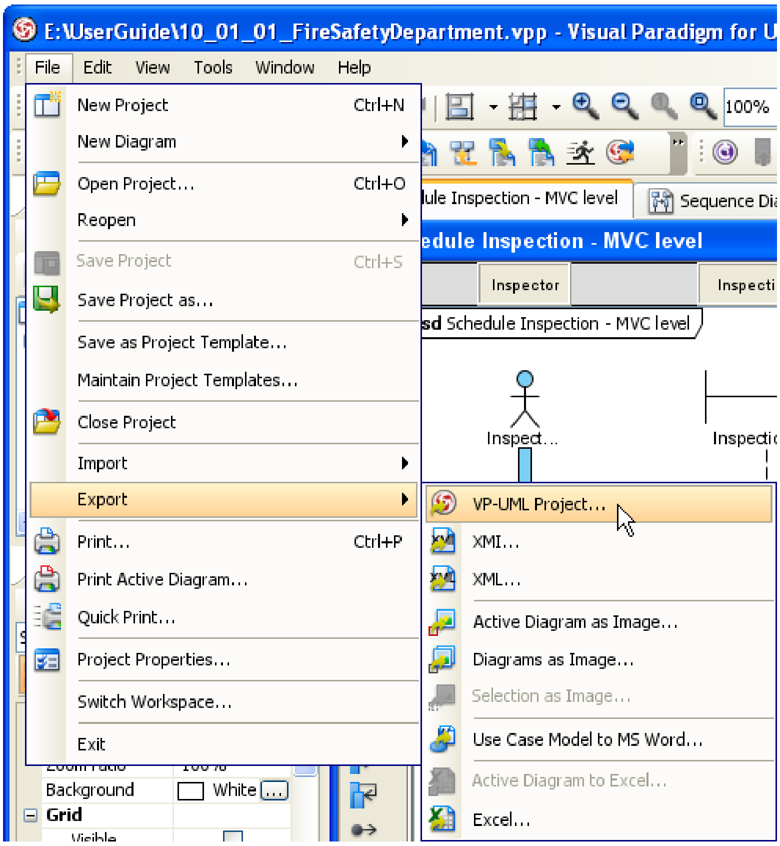
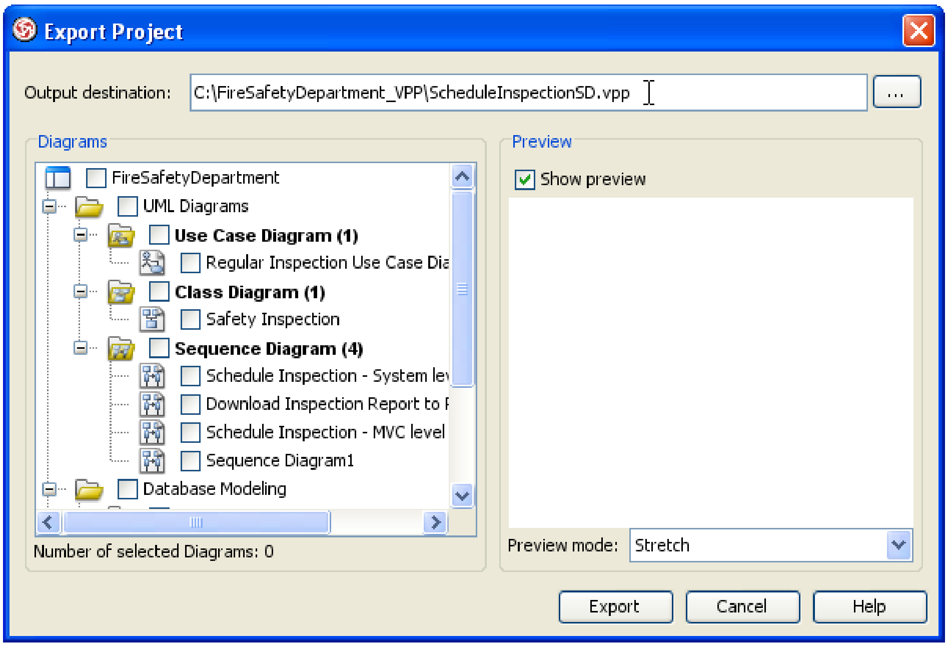


Figure 2-3 *Export VP project*

4. Input Output destination.



Output destination: C:\FireSafetyDepartment_VPP\ScheduleInspectionSD.vpp

- □ FireSafetyDepartment
  - □ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Regular Inspection Use Case Dia
    - □ **Class Diagram (1)**
      - □ Safety Inspection
    - □ **Sequence Diagram (4)**
      - □ Schedule Inspection - System lev
      - □ Download Inspection Report to F
      - □ Schedule Inspection - MVC level
      - □ Sequence Diagram1
  - □ Database Modeling

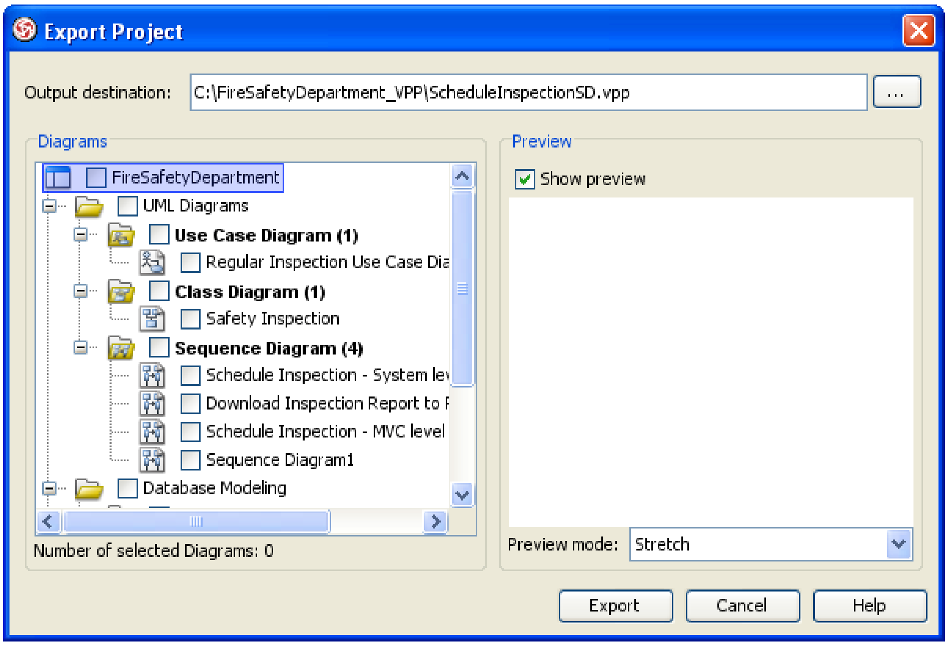
Number of selected Diagrams: 0

☑ Show preview

Preview mode: Stretch

Figure 2-4 *Inputting output destination*

5. It is possible to check "FireSafetyDepartment" checkbox to export whole project to another VPP.


Export Project

Output destination: C:\FireSafetyDepartment_VPP\ScheduleInspectionSD.vpp

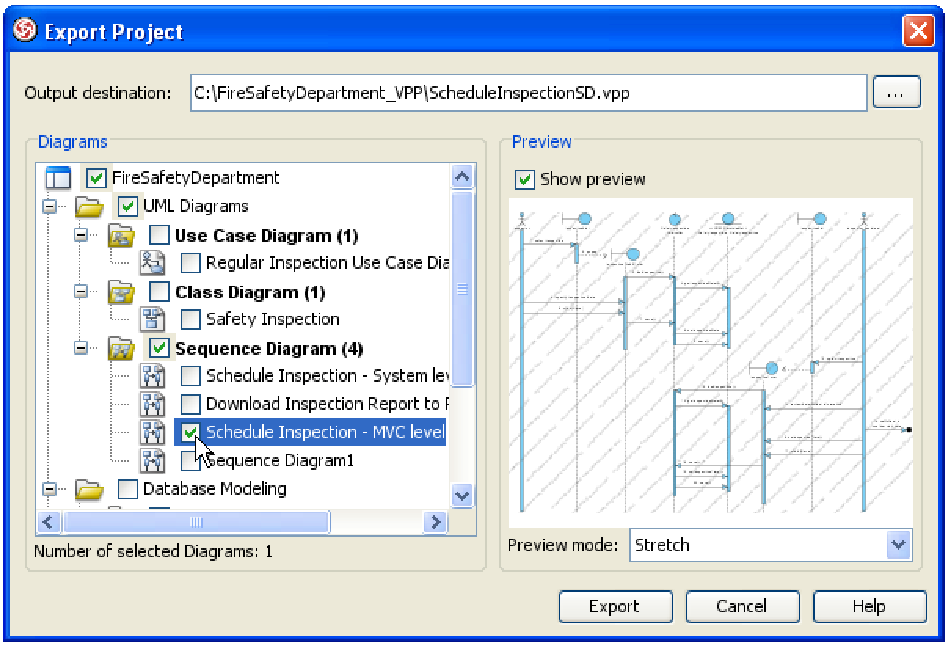
Number of selected Diagrams: 0

☑ Show preview

Preview mode: Stretch

Figure 2-5 Highlight export whole VP project checkbox

6. Select Sequence Diagram *Schedule Inspection - MVC level* checkbox.


Export Project

Output destination: C:\FireSafetyDepartment_VPP\ScheduleInspectionSD.vpp

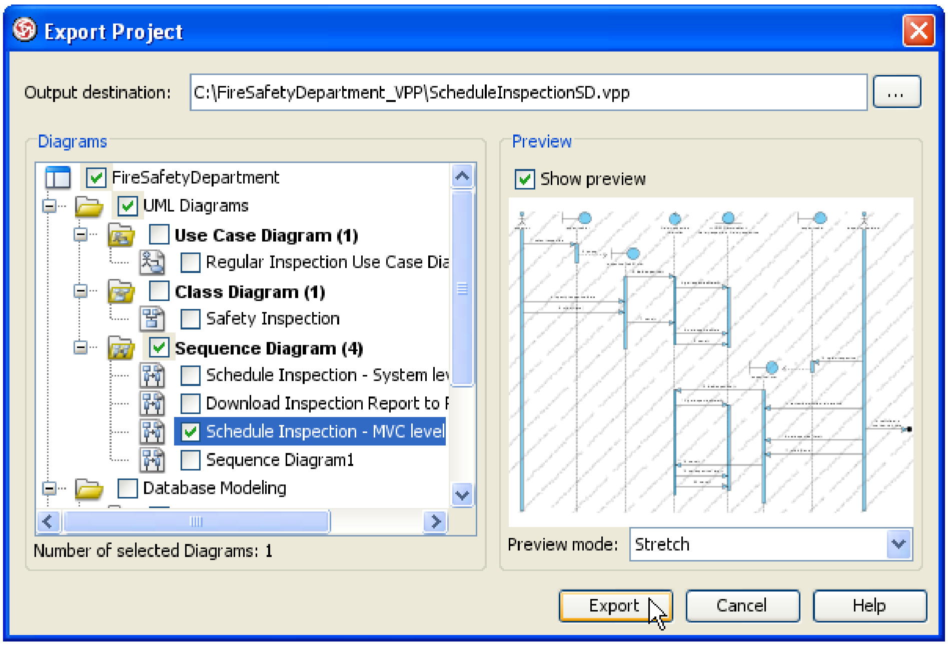
Number of selected Diagrams: 1

☑ Show preview

Preview mode: Stretch

Figure 2-6 Selecting diagrams to export VP project

7. Press Export button.



Export Project

Output destination: C:\FireSafetyDepartment_VPP\ScheduleInspectionSD.vpp

Diagrams

- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Regular Inspection Use Case Dia
    - □ **Class Diagram (1)**
      - □ Safety Inspection
    - ☑ **Sequence Diagram (4)**
      - □ Schedule Inspection - System le
      - □ Download Inspection Report to F
      - ☑ Schedule Inspection - MVC level
      - □ Sequence Diagram1
  - □ Database Modeling

Number of selected Diagrams: 1

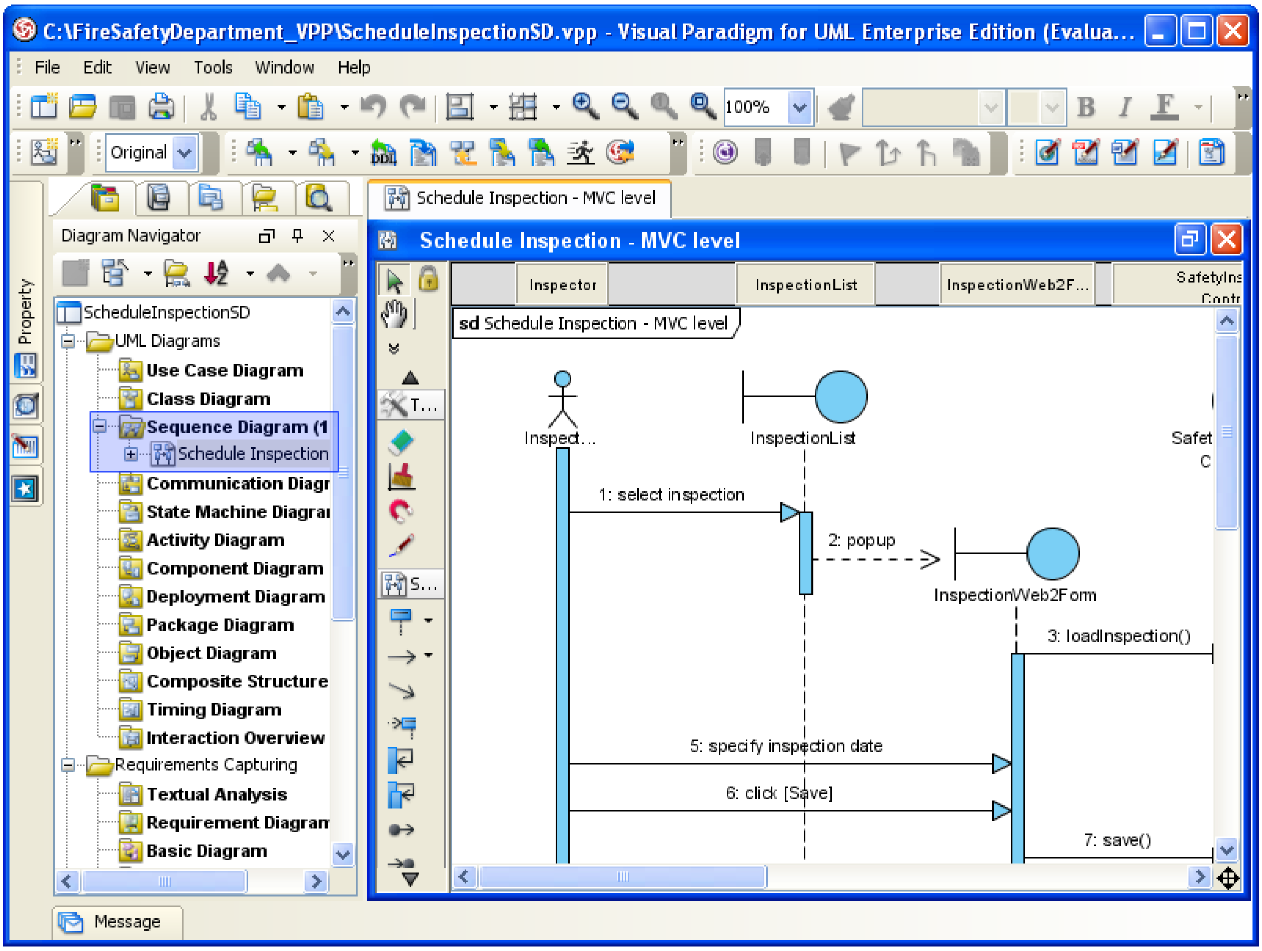
Preview

☑ Show preview

Preview mode: Stretch

Figure 2-7 Export to VP project

8. There is only one Sequence Diagram in Diagram Navigator and Sequence Diagram is exported successfully.



C:\FireSafetyDepartment_VPP\ScheduleInspectionSD.vpp - Visual Paradigm for UML Enterprise Edition (Evalua...

- ScheduleInspectionSD
  - UML Diagrams
    - **Use Case Diagram**
    - **Class Diagram**
    - **Sequence Diagram (1**
      - Schedule Inspection
    - **Communication Diagr**
    - **State Machine Diagra**
    - **Activity Diagram**
    - **Component Diagram**
    - **Deployment Diagram**
    - **Package Diagram**
    - **Object Diagram**
    - **Composite Structure**
    - **Timing Diagram**
    - **Interaction Overview**
  - Requirements Capturing
    - **Textual Analysis**
    - **Requirement Diagram**
    - **Basic Diagram**

Schedule Inspection - MVC level

Figure 2-8 Review exported VP project

Importing VP Project

Importing VP Project to current project

1. Open exported project and rename LifeLine *InspectionList* to *Web base InspectionList*.

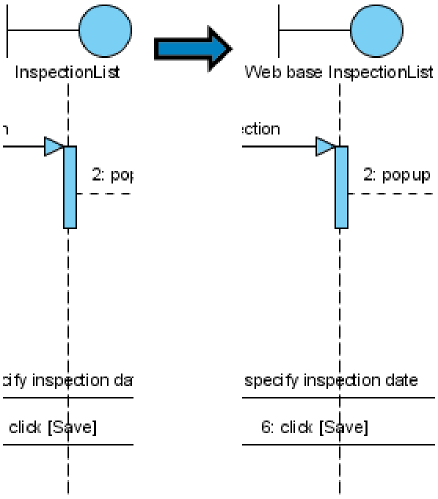


Figure 2-9 *Renaming LifeLine*

2. Press Save button in toolbar.

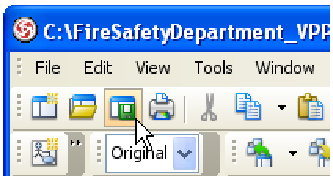


Figure 2-10 *Save project*

3. Open original project and mouse over main menu.

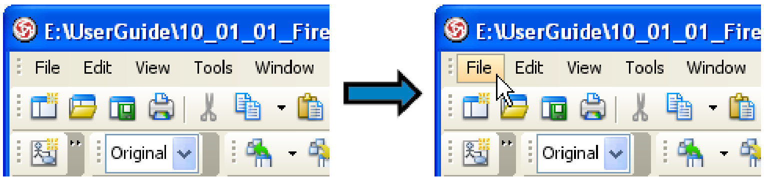


Figure 2-11 *Mouse over main menu*

4. Click on File of main menu.

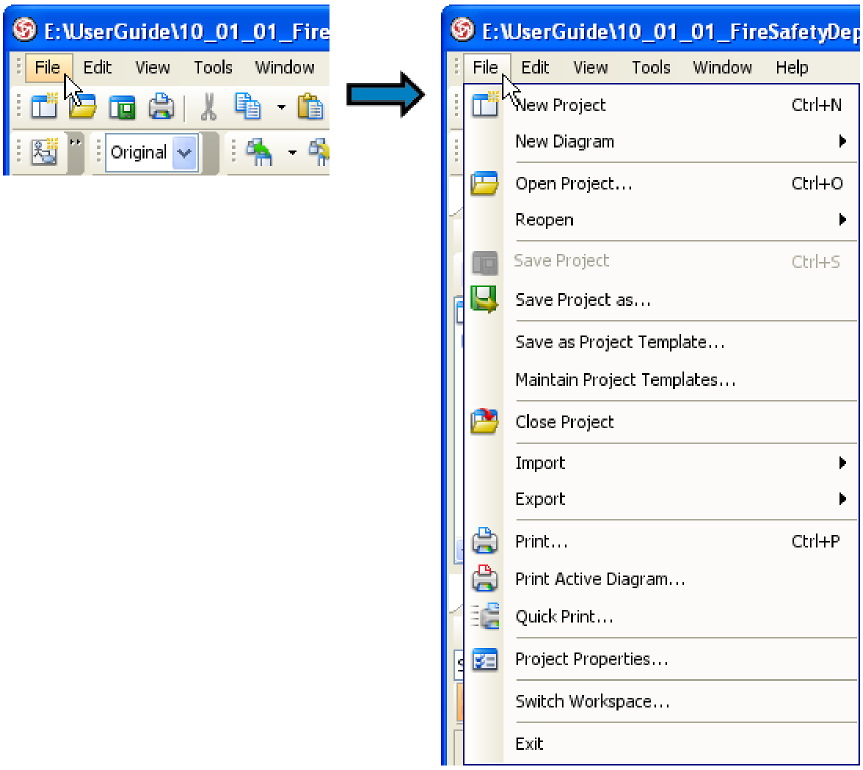
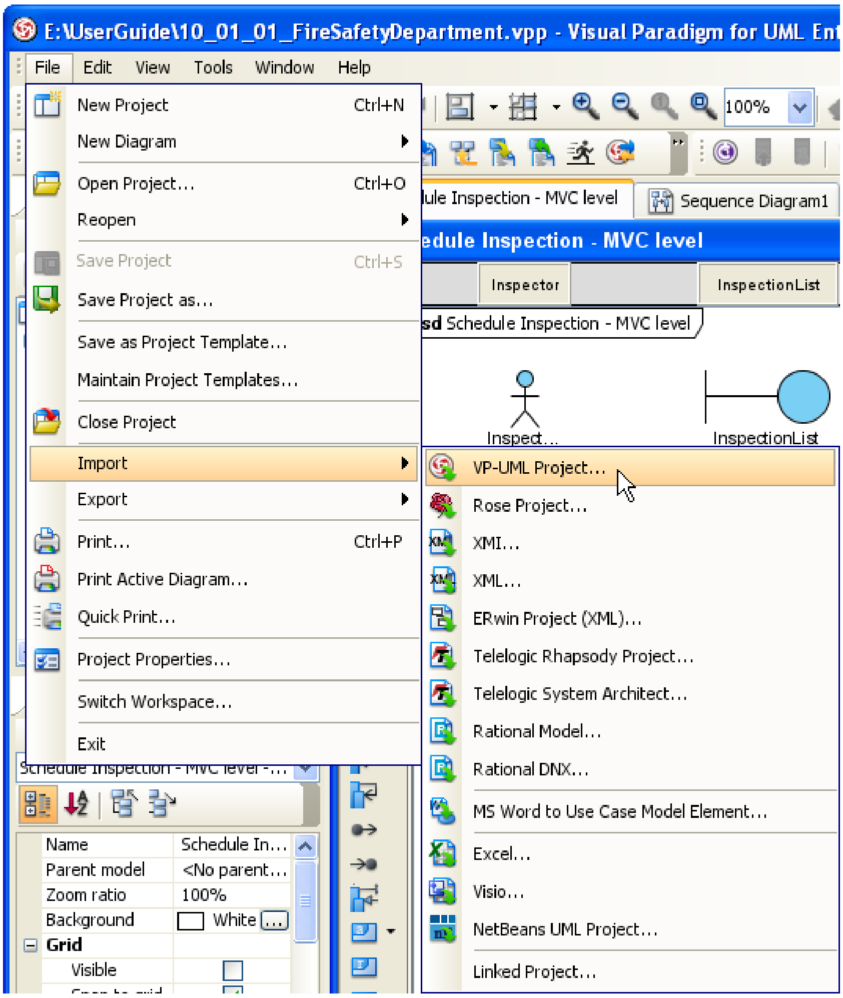


Figure 2-12 *Showing popup of main menu*

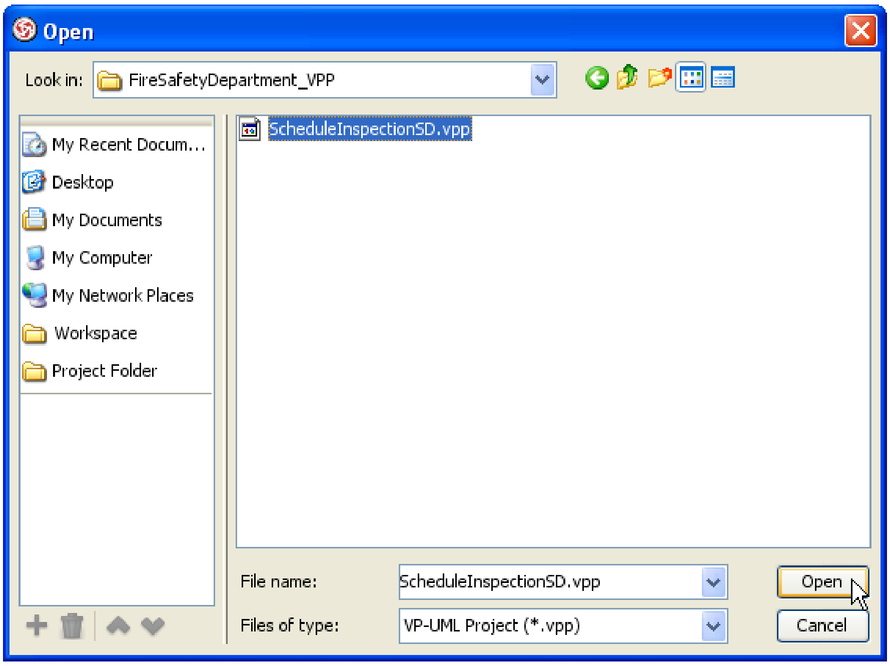
5. Click on Export > VP-UML Project... menu item.



Name: Schedule In...
Parent model: <No parent...
Zoom ratio: 100%
Background: White

Figure 2-13 Import VP project

6. Select exported project.



Look in: FireSafetyDepartment_VPP

File name: ScheduleInspectionSD.vpp

Files of type: VP-UML Project (*.vpp)

Figure 2-14 Selecting VP project to be import

7. LifeLine *InspectionList* is renamed to *Web base InspectionList*.

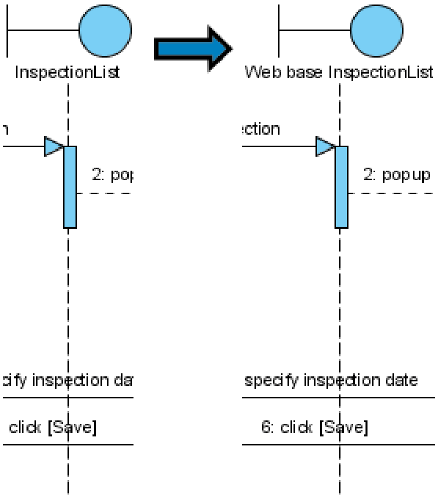


Figure 2-15 *Different of lifeline after import VP project*

Import overwrite mechanism

Importing VPP will only overwrite the modified Model Element

1. Below image showing original project.

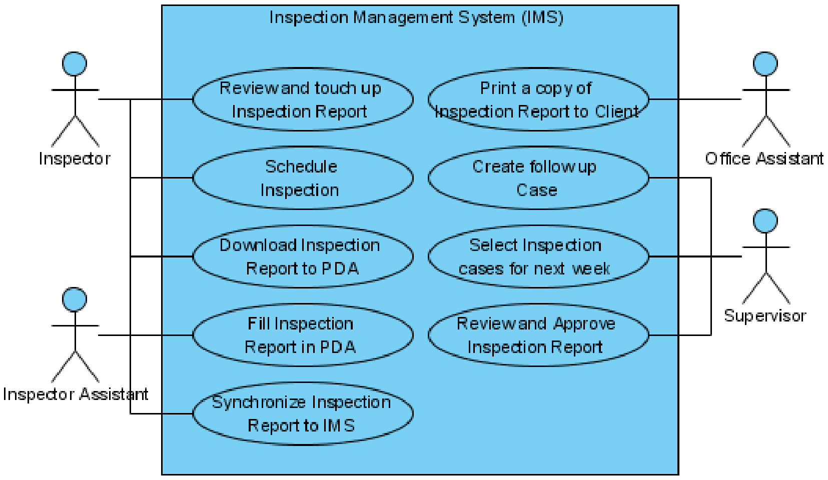

Inspection Management System (IMS)
Inspector
Inspector Assistant
Office Assistant
Supervisor
Review and touch up Inspection Report
Schedule Inspection
Download Inspection Report to PDA
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS
Print a copy of Inspection Report to Client
Create follow up Case
Select Inspection cases for next week
Review and Approve Inspection Report

Figure 2-16 *Original project*

2. Click Export > VP-UML Project... menu item to export original project as VP-UML Project.

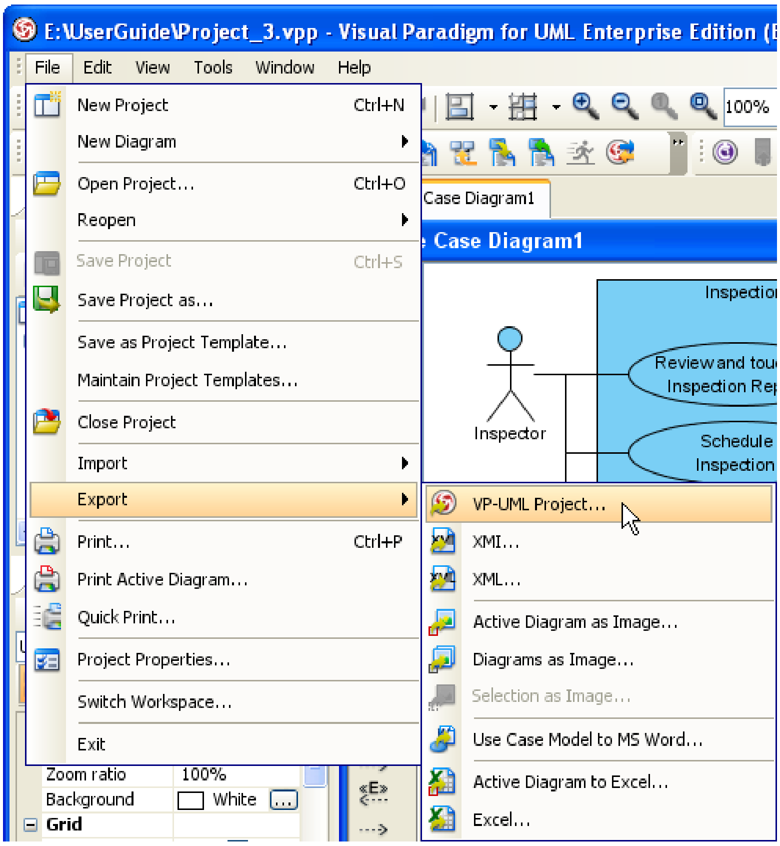
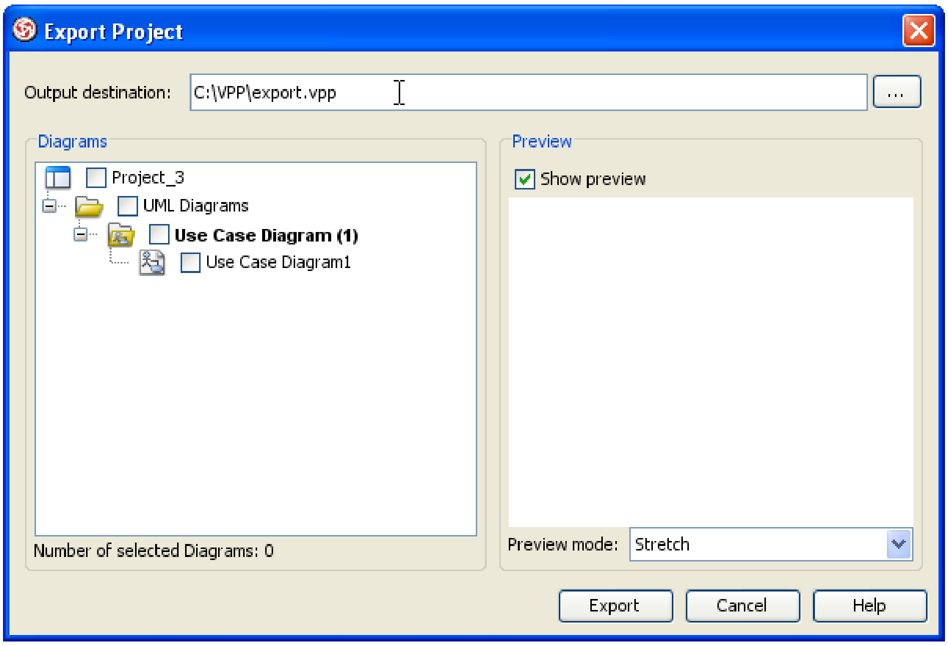


Figure 2-17 *Export VP-UML Project*

3. Input *C:\VPP\export.vpp* in Output destination.



Export Project

Output destination: C:\VPP\export.vpp

Diagrams

- □ Project_3
  - □ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Use Case Diagram1

Number of selected Diagrams: 0

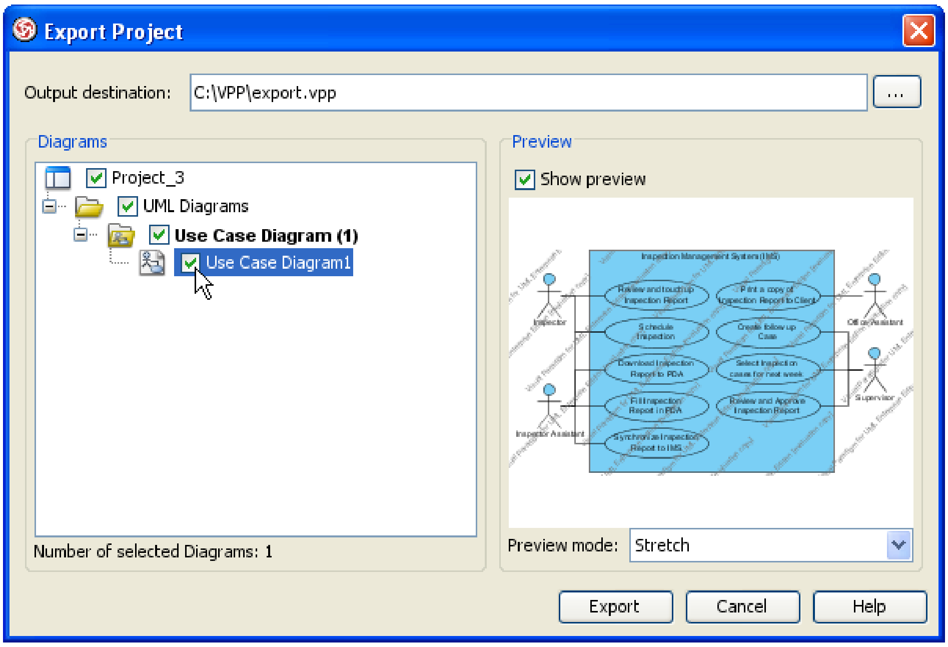
Preview

☑ Show preview

Preview mode: Stretch

Figure 2-18 *Inputting output destination*

4. Select *Use Case Diagram1* checkbox.



Export Project

Output destination: C:\VPP\export.vpp

Diagrams

- ☑ Project_3
  - ☑ UML Diagrams
    - ☑ **Use Case Diagram (1)**
      - ☑ Use Case Diagram1

Number of selected Diagrams: 1

Preview

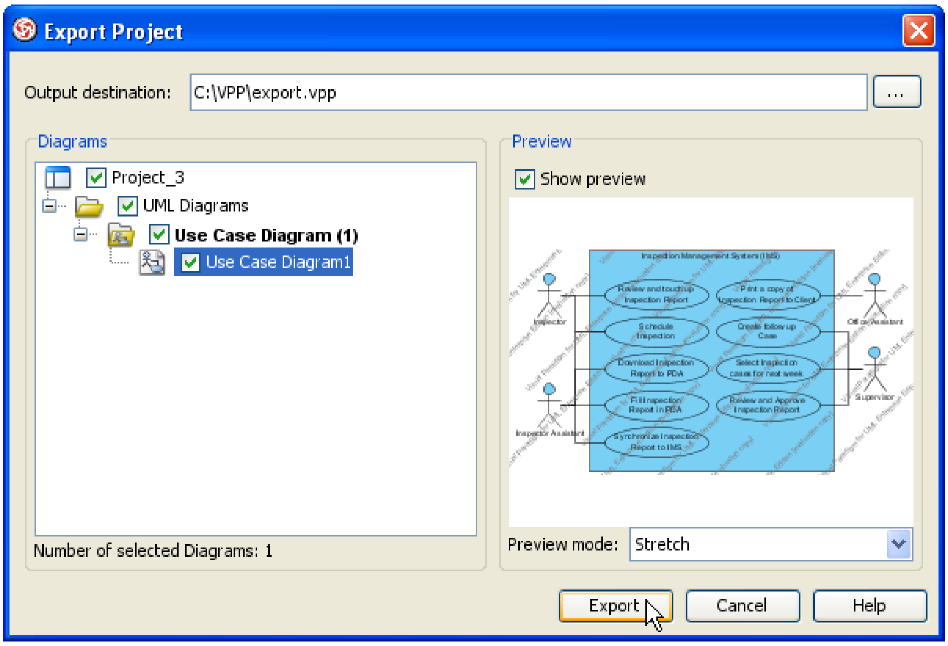
☑ Show preview

Preview mode: Stretch

Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Fill Inspection Report in PDA
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Inspector
Inspector Assistant
Office Assistant
Supervisor
...
Export
Cancel
Help

Figure 2-19 *Selecting diagram to be export*

5. Press Export button to perform the export.



Export Project

Output destination: C:\VPP\export.vpp

Diagrams

- ☑ Project_3
  - ☑ UML Diagrams
    - ☑ **Use Case Diagram (1)**
      - ☑ Use Case Diagram1

Number of selected Diagrams: 1

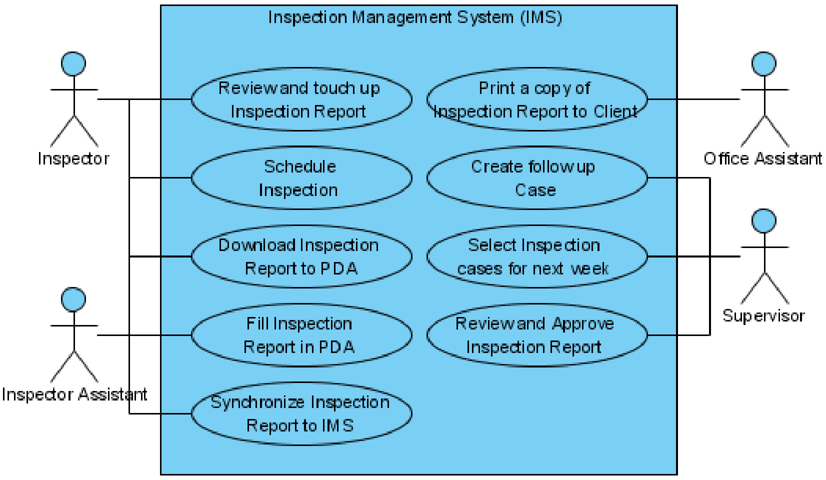
Preview

☑ Show preview

Preview mode: Stretch

Figure 2-20 *Export project*

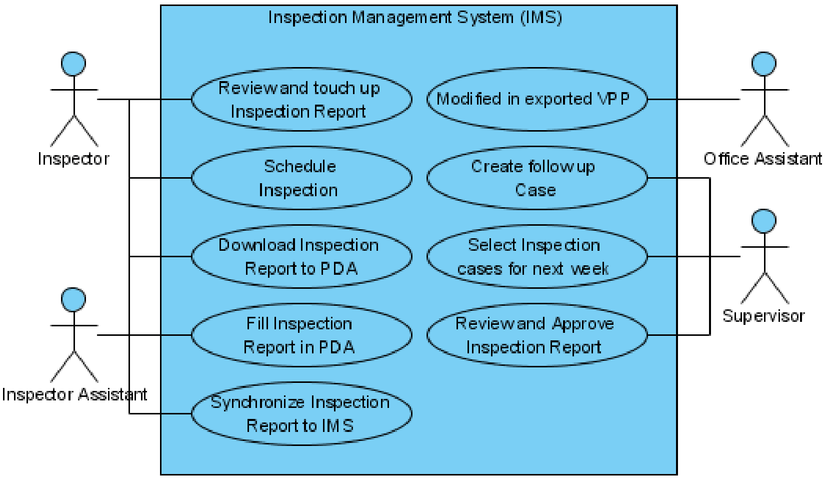
6. Switch to exported project by opening *C:\VPP\export.vpp*.



Inspection Management System (IMS)
Review and touch up Inspection Report
Print a copy of Inspection Report to Client
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Fill Inspection Report in PDA
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Inspector
Inspector Assistant
Office Assistant
Supervisor

Figure 2-21 *Exported project*

7. Rename Use Case *Print a copy of Inspection Report to Client* to *Modified in exported VPP*.



Inspection Management System (IMS)
Review and touch up Inspection Report
Modified in exported VPP
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Fill Inspection Report in PDA
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Inspector
Inspector Assistant
Office Assistant
Supervisor

Figure 2-22 *Renamed use case in exported project*

8. Save exported project by pressing Save button in tool bar.

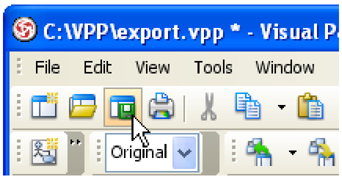


Figure 2-23 *Save exported project*

9. Switch back to original project.

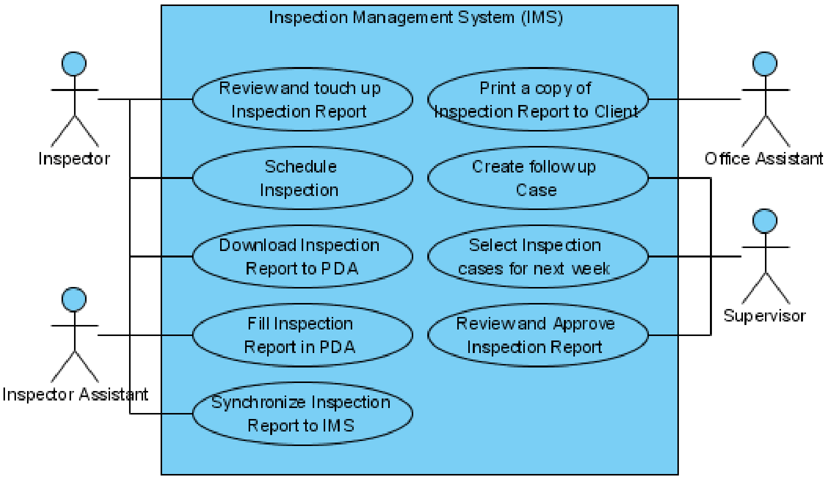

Inspection Management System (IMS)
Inspector
Inspector Assistant
Office Assistant
Supervisor
Review and touch up Inspection Report
Schedule Inspection
Download Inspection Report to PDA
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS
Print a copy of Inspection Report to Client
Create follow up Case
Select Inspection cases for next week
Review and Approve Inspection Report

Figure 2-24 *Original project*

10. Rename Use Case *Review and touch up Inspection Report* to *Modified in original VPP*.

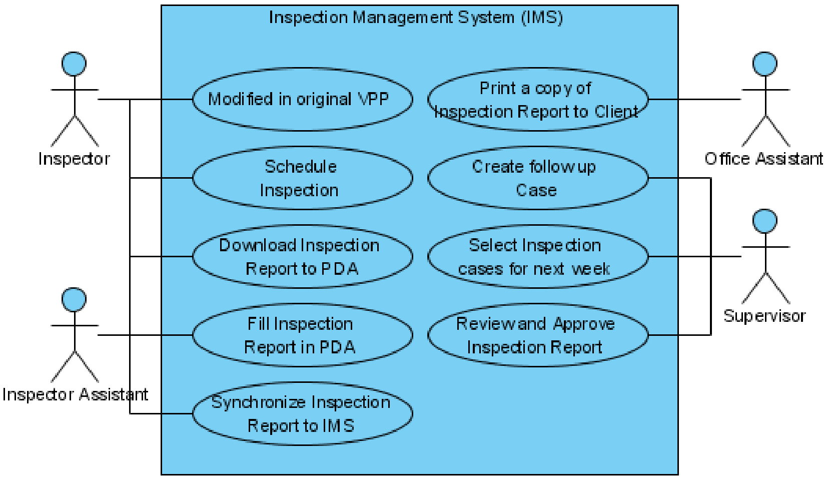
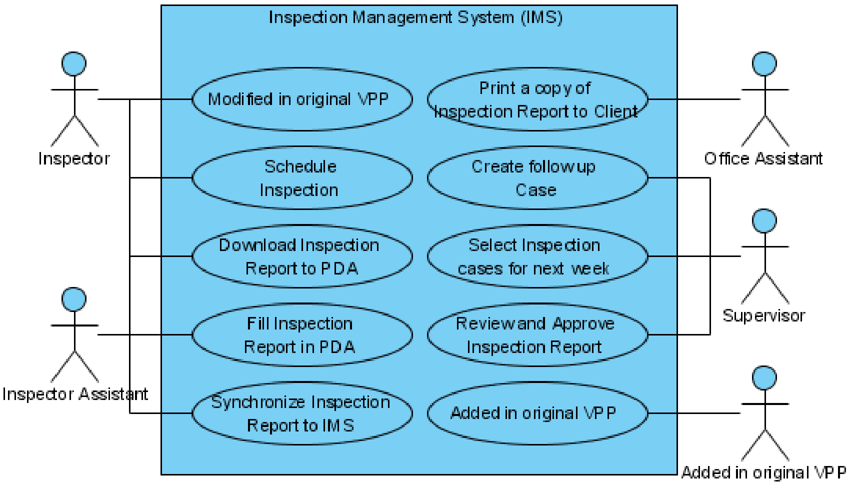

Inspection Management System (IMS)
Inspector
Inspector Assistant
Office Assistant
Supervisor
Modified in original VPP
Schedule Inspection
Download Inspection Report to PDA
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS
Print a copy of Inspection Report to Client
Create follow up Case
Select Inspection cases for next week
Review and Approve Inspection Report

Figure 2-25 *Renamed use case in original project*

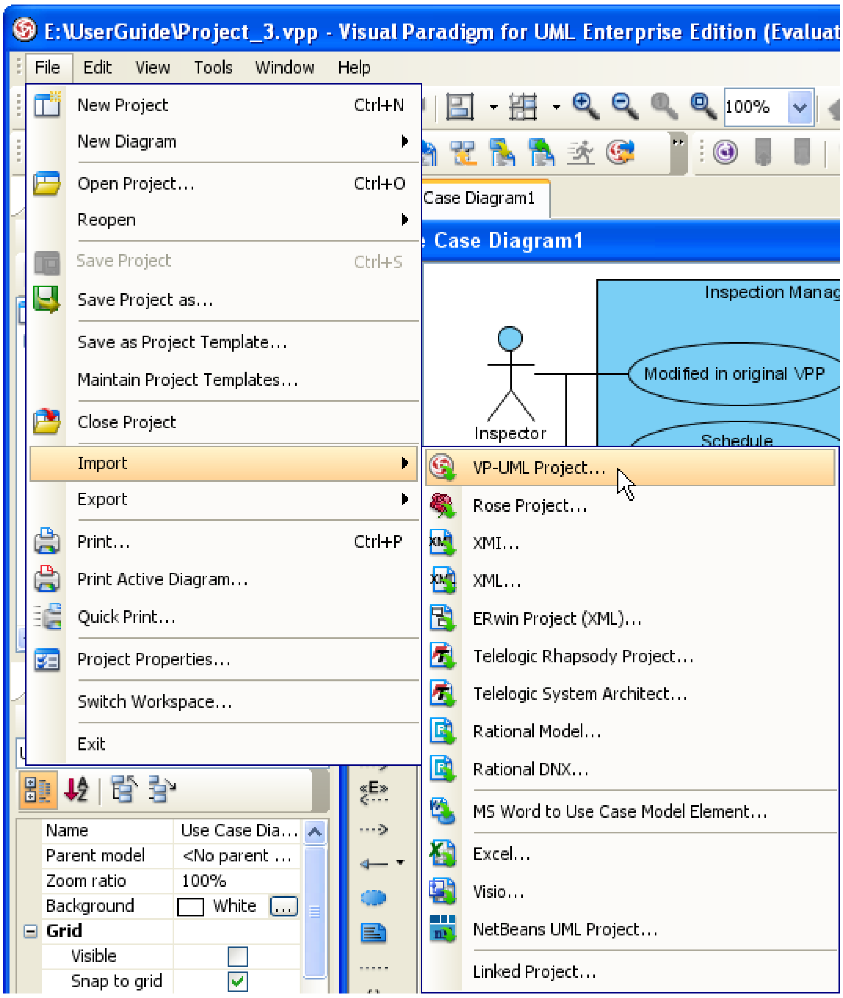
11. Add Use Case, Association and Actor.



Inspection Management System (IMS)
Inspector
Inspector Assistant
Modified in original VPP
Schedule Inspection
Download Inspection Report to PDA
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS
Print a copy of Inspection Report to Client
Create follow up Case
Select Inspection cases for next week
Review and Approve Inspection Report
Added in original VPP
Office Assistant
Supervisor
Added in original VPP

Figure 2-26 *Added shapes in original project*

12. Click Import > VP-UML Project... menu item to import another VP-UML Project.



Name	Use Case Dia...
Parent model	<No parent ...
Zoom ratio	100%
Background	White
Grid	
Visible	□
Snap to grid	☑

Figure 2-27 *Import VP-UML Project*

13. Select *C:\VPP\export.vpp* and press Open button to perform the import.

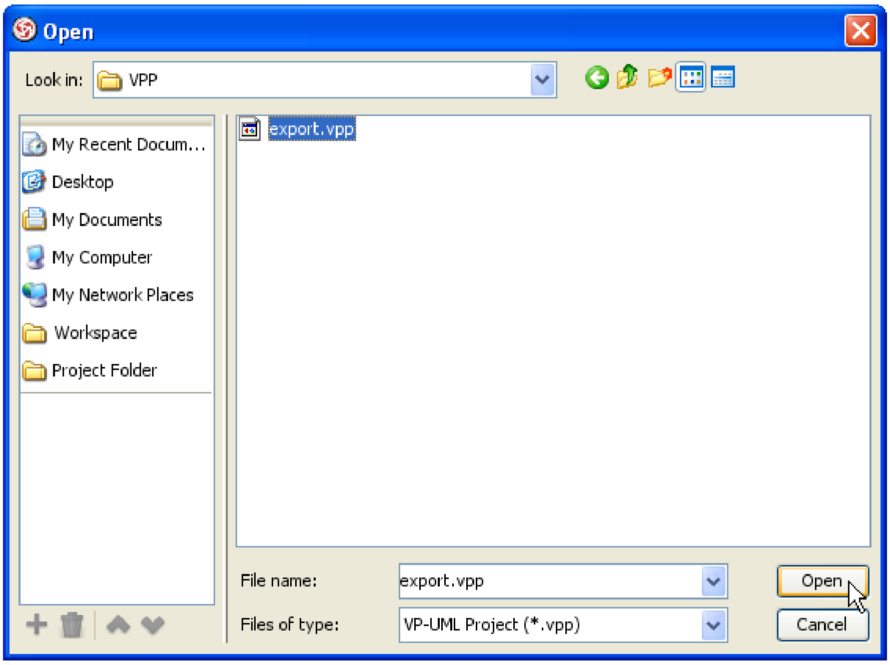


Figure 2-28 *Import project*

14. Below image showing result after the import. You can found only Use Case *Print a copy of Inspection Report to Client* is renamed to *Modified in exported VPP* and other model elements are remain unchange. In other words only modified Model Element will be overwrite.

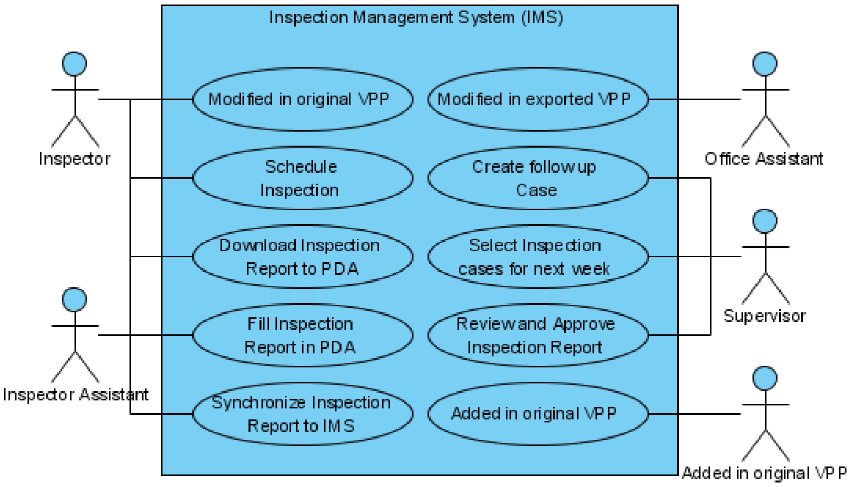

Inspection Management System (IMS)
Modified in original VPP
Modified in exported VPP
Schedule Inspection
Create follow up Case
Download Inspection Report to PDA
Select Inspection cases for next week
Fill Inspection Report in PDA
Review and Approve Inspection Report
Synchronize Inspection Report to IMS
Added in original VPP
Inspector
Inspector Assistant
Office Assistant
Supervisor
Added in original VPP

Figure 2-29 *Result of import project*

Exporting Diagrams to Microsoft Excel Format

1. Mouse over main menu.

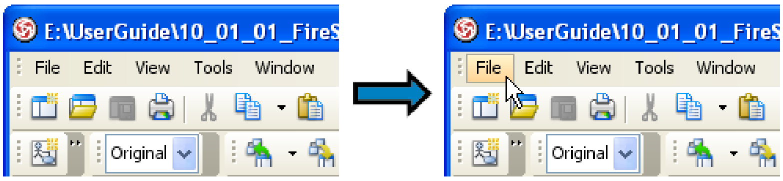


Figure 3-1 *Mouse over main menu*

2. Click on File of main menu.

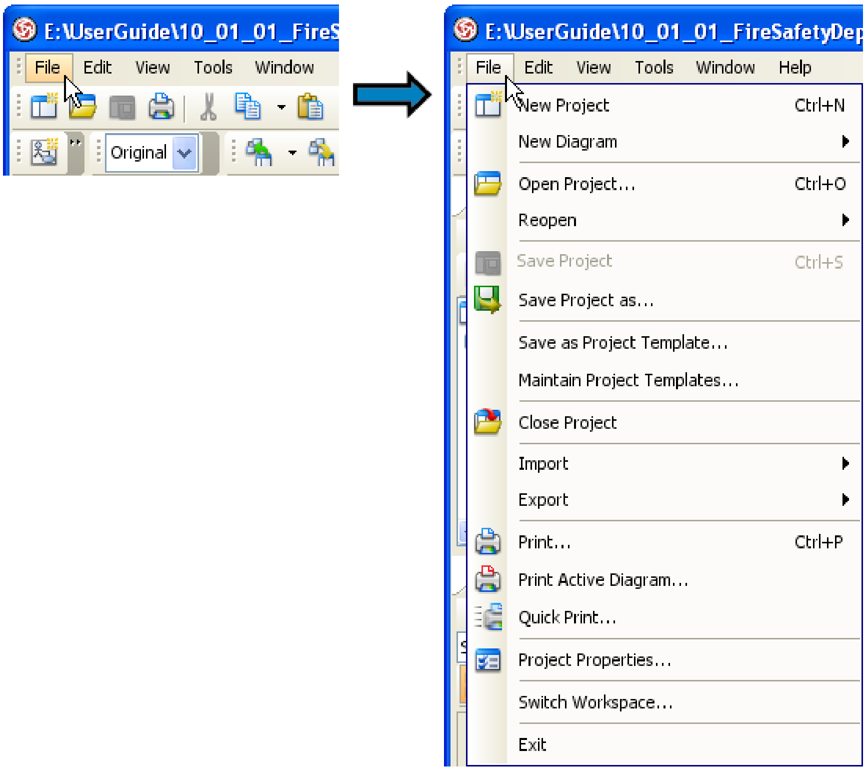


Figure 3-2 *Showing popup of main menu*

3. Click on Export > Excel... menu item.

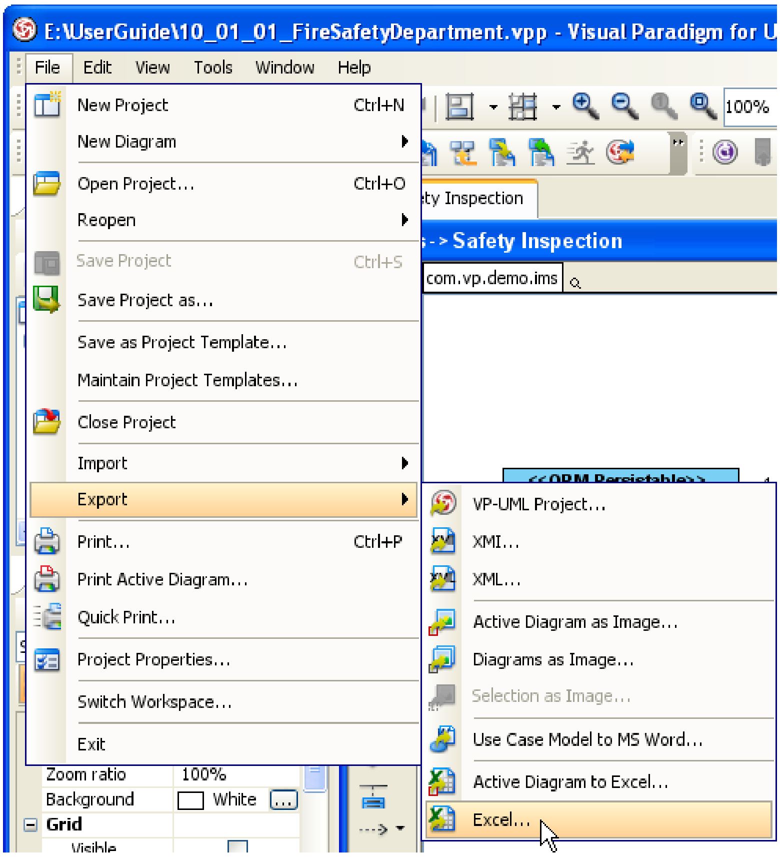
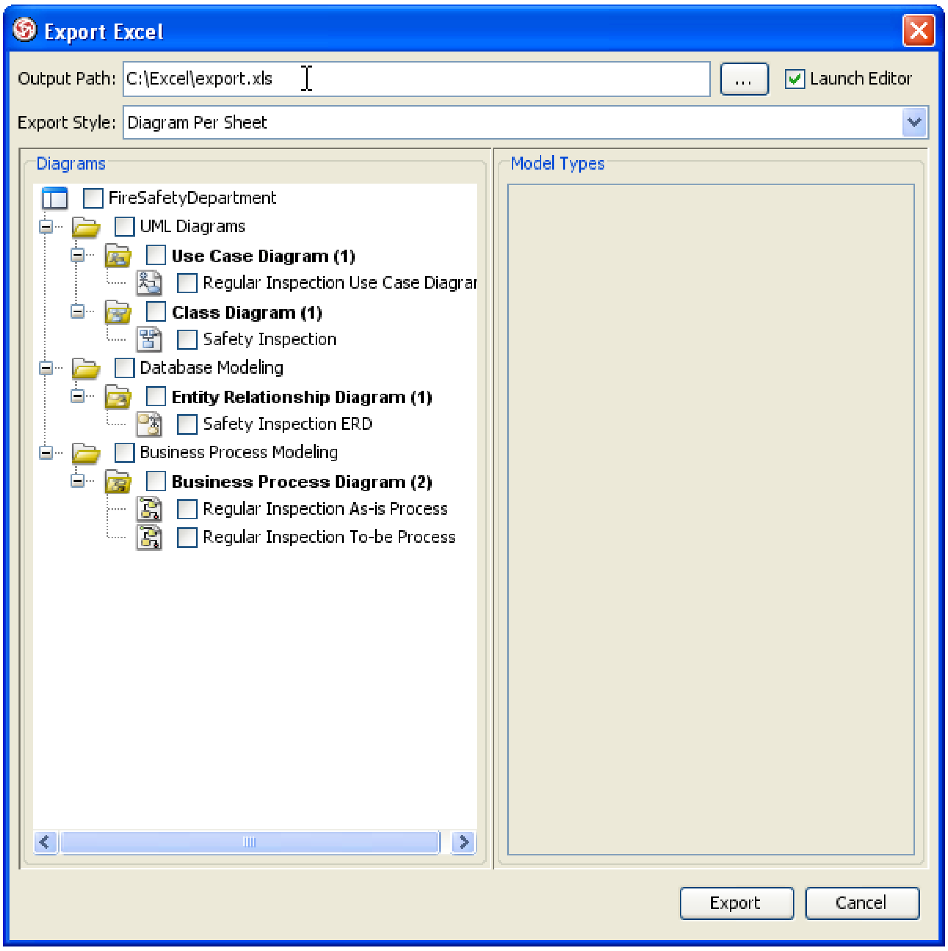


Figure 3-3 Export Excel

4. Input *C:\Excel\export.xls* in Output Path.



Export Excel

Output Path: C:\Excel\export.xls

☑ Launch Editor

Export Style: Diagram Per Sheet

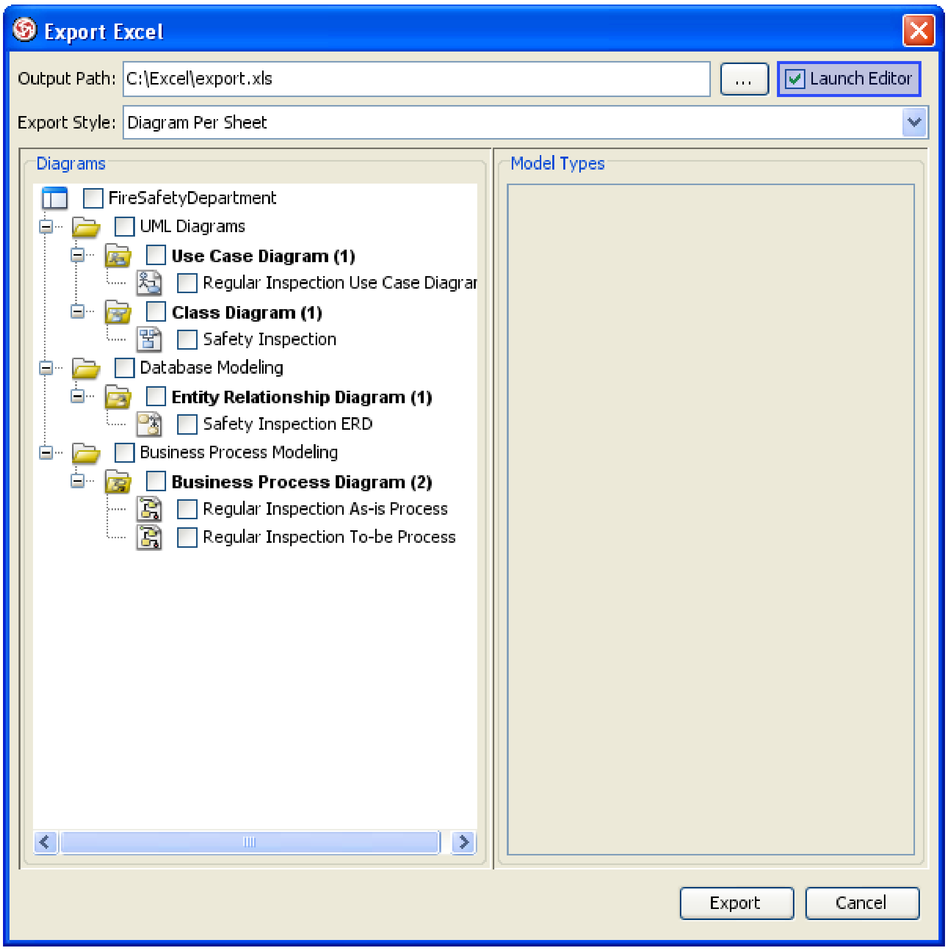
Diagrams

- □ FireSafetyDepartment
  - □ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Regular Inspection Use Case Diagrar
    - □ **Class Diagram (1)**
      - □ Safety Inspection
  - □ Database Modeling
    - □ **Entity Relationship Diagram (1)**
      - □ Safety Inspection ERD
  - □ Business Process Modeling
    - □ **Business Process Diagram (2)**
      - □ Regular Inspection As-is Process
      - □ Regular Inspection To-be Process

Model Types

Figure 3-4 *Inputting output path*

5. When Launch Editor is checked, VP-UML will launch pre-configured editor for exported result.



Export Excel

Output Path: C:\Excel\export.xls

☑ Launch Editor

Export Style: Diagram Per Sheet

...
Diagrams
□ FireSafetyDepartment
□ UML Diagrams
□ Use Case Diagram (1)
□ Regular Inspection Use Case Diagrar
□ Class Diagram (1)
□ Safety Inspection
□ Database Modeling
□ Entity Relationship Diagram (1)
□ Safety Inspection ERD
□ Business Process Modeling
□ Business Process Diagram (2)
□ Regular Inspection As-is Process
□ Regular Inspection To-be Process
Model Types
Export
Cancel

Figure 3-5 *Describe launch editor*

6. When Export Style is Diagram Per Sheet, VP-UML will export each selected diagrams in separate sheet.

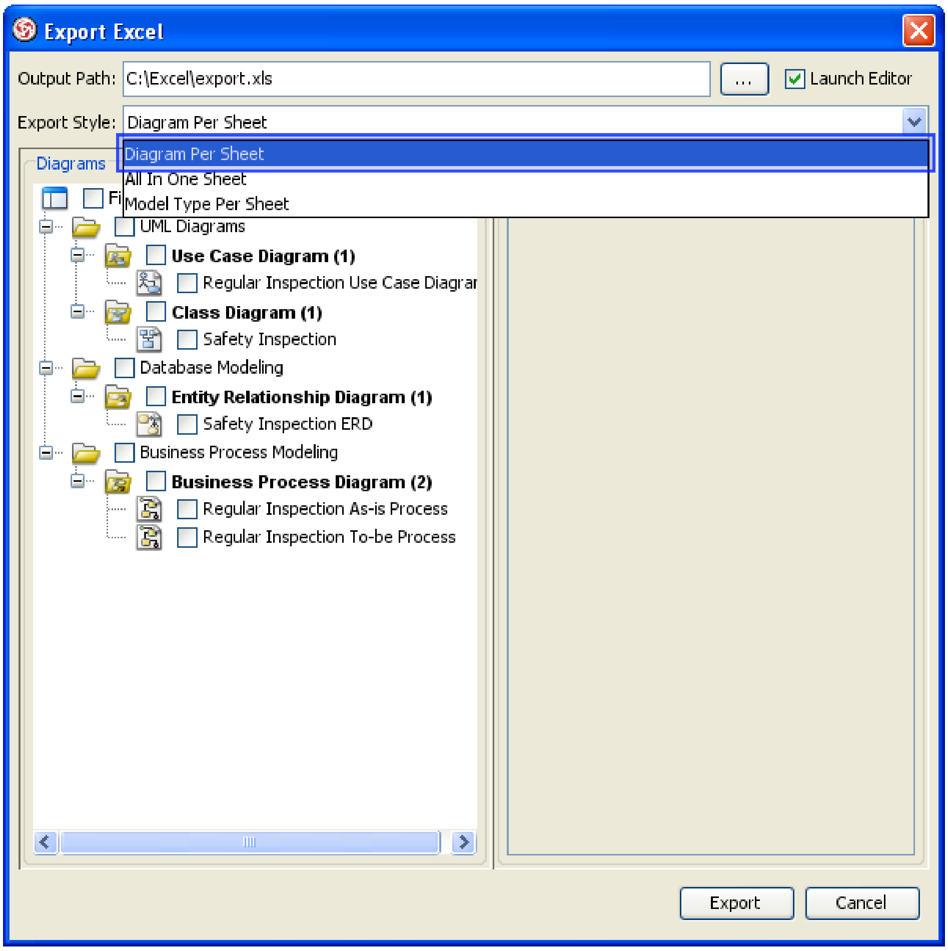
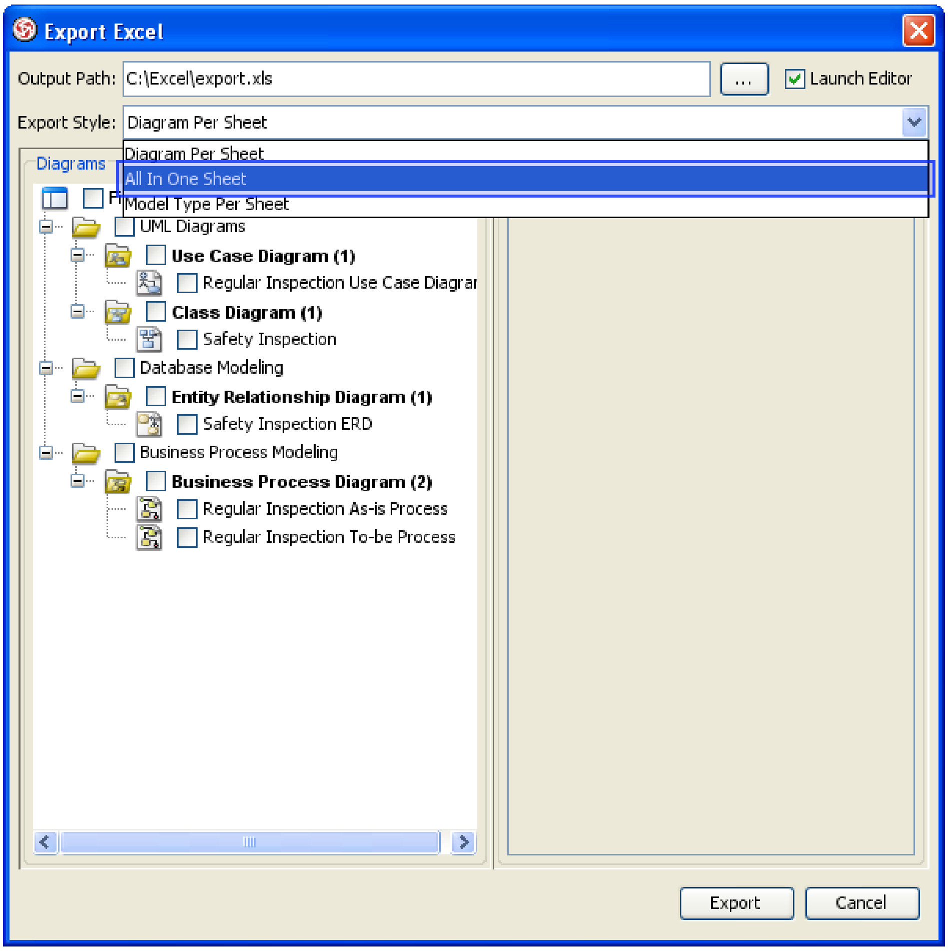

Export Excel
Output Path: C:\Excel\export.xls
...
☑ Launch Editor
Export Style: Diagram Per Sheet
Diagram Per Sheet
All In One Sheet
Model Type Per Sheet
Diagrams
□ Fi
□ UML Diagrams
□ Use Case Diagram (1)
□ Regular Inspection Use Case Diagrar
□ Class Diagram (1)
□ Safety Inspection
□ Database Modeling
□ Entity Relationship Diagram (1)
□ Safety Inspection ERD
□ Business Process Modeling
□ Business Process Diagram (2)
□ Regular Inspection As-is Process
□ Regular Inspection To-be Process
Export
Cancel

Figure 3-6 *Describe diagram per sheet export style*

7. When Export Style is Diagram Per Sheet, VP-UML will export all selected diagrams in one sheet.



Export Excel

Output Path: C:\Excel\export.xls

☑ Launch Editor

Export Style: Diagram Per Sheet

...
Diagram Per Sheet
All In One Sheet
Model Type Per Sheet
Diagrams
□ F
□ UML Diagrams
□ Use Case Diagram (1)
□ Regular Inspection Use Case Diagrar
□ Class Diagram (1)
□ Safety Inspection
□ Database Modeling
□ Entity Relationship Diagram (1)
□ Safety Inspection ERD
□ Business Process Modeling
□ Business Process Diagram (2)
□ Regular Inspection As-is Process
□ Regular Inspection To-be Process
Export
Cancel

Figure 3-7 *Describe all in one sheet export style*

8. When Export Style is Diagram Per Sheet, VP-UML will export each selected model types in separate sheet.

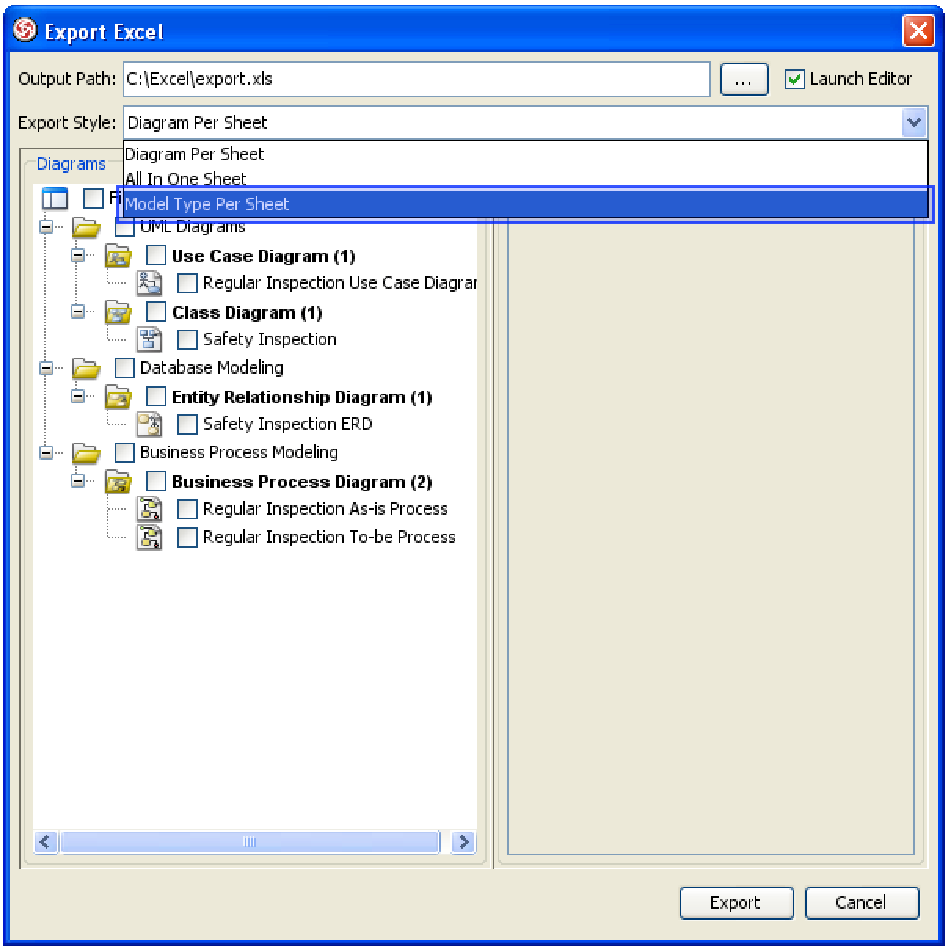
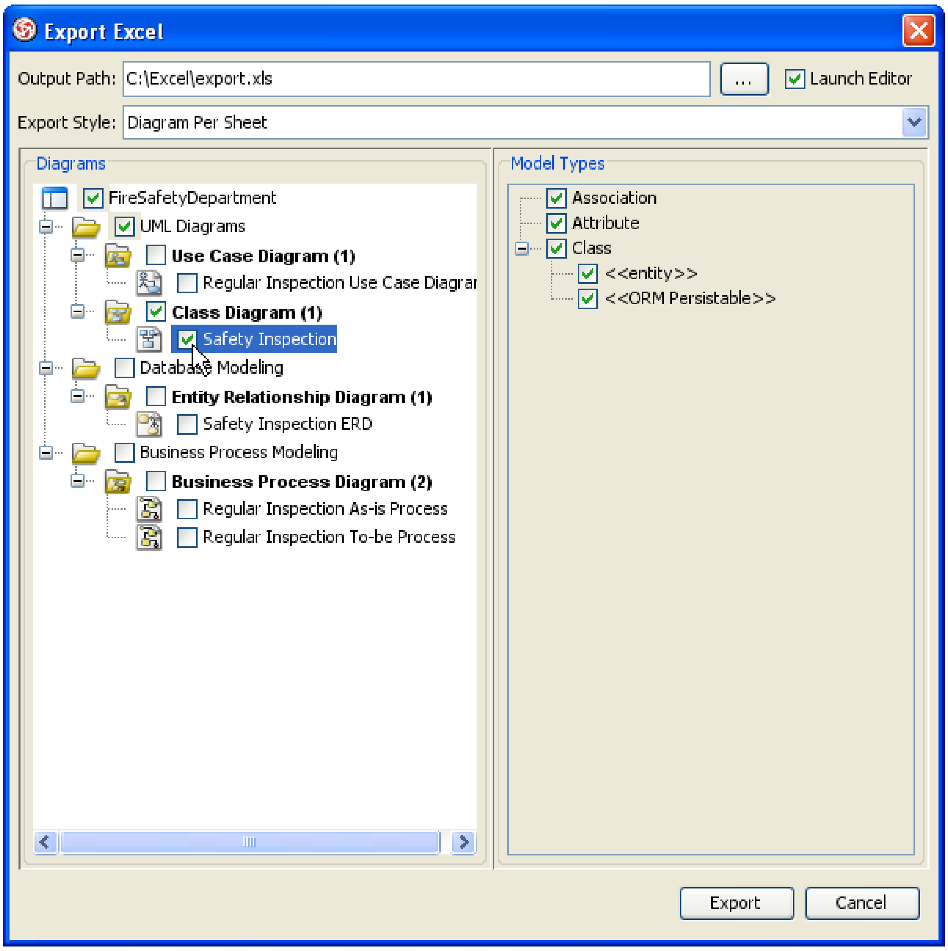

Export Excel
Output Path: C:\Excel\export.xls
...
☑ Launch Editor
Export Style: Diagram Per Sheet
Diagram Per Sheet
All In One Sheet
Model Type Per Sheet
Diagrams
UML Diagrams
Use Case Diagram (1)
Regular Inspection Use Case Diagrar
Class Diagram (1)
Safety Inspection
Database Modeling
Entity Relationship Diagram (1)
Safety Inspection ERD
Business Process Modeling
Business Process Diagram (2)
Regular Inspection As-is Process
Regular Inspection To-be Process
Export
Cancel

Figure 3-8 *Describe model type per sheet export style*

9. Select *Safety Inspection* checkbox to be export.



Output Path: C:\Excel\export.xls ☑ Launch Editor

Export Style: Diagram Per Sheet

Diagrams

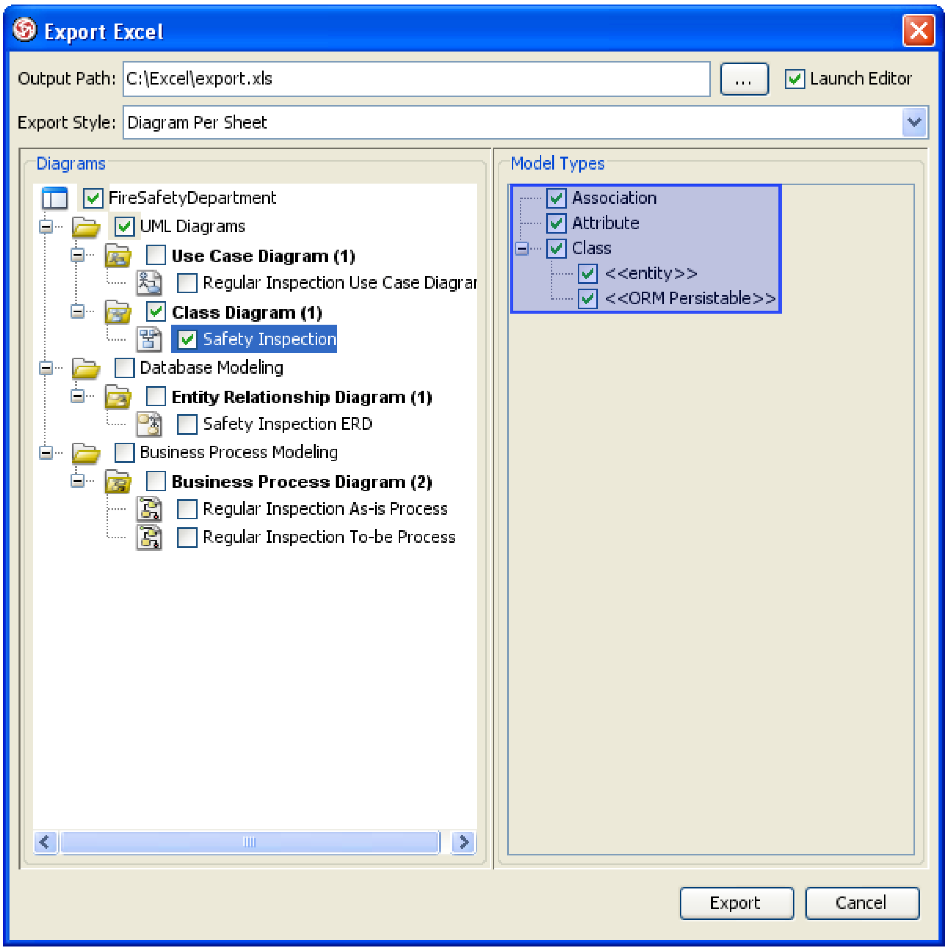
- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Regular Inspection Use Case Diagrar
    - ☑ **Class Diagram (1)**
      - ☑ Safety Inspection
  - □ Database Modeling
    - □ **Entity Relationship Diagram (1)**
      - □ Safety Inspection ERD
  - □ Business Process Modeling
    - □ **Business Process Diagram (2)**
      - □ Regular Inspection As-is Process
      - □ Regular Inspection To-be Process

Model Types

- ☑ Association
- ☑ Attribute
- ☑ Class
  - ☑ <<entity>>
  - ☑ <<ORM Persistable>>

Figure 3-9 *Selecting diagrams to export Excel*

10. Only selected model type will include in the export.



Output Path: C:\Excel\export.xls

☑ Launch Editor

Export Style: Diagram Per Sheet

Diagrams

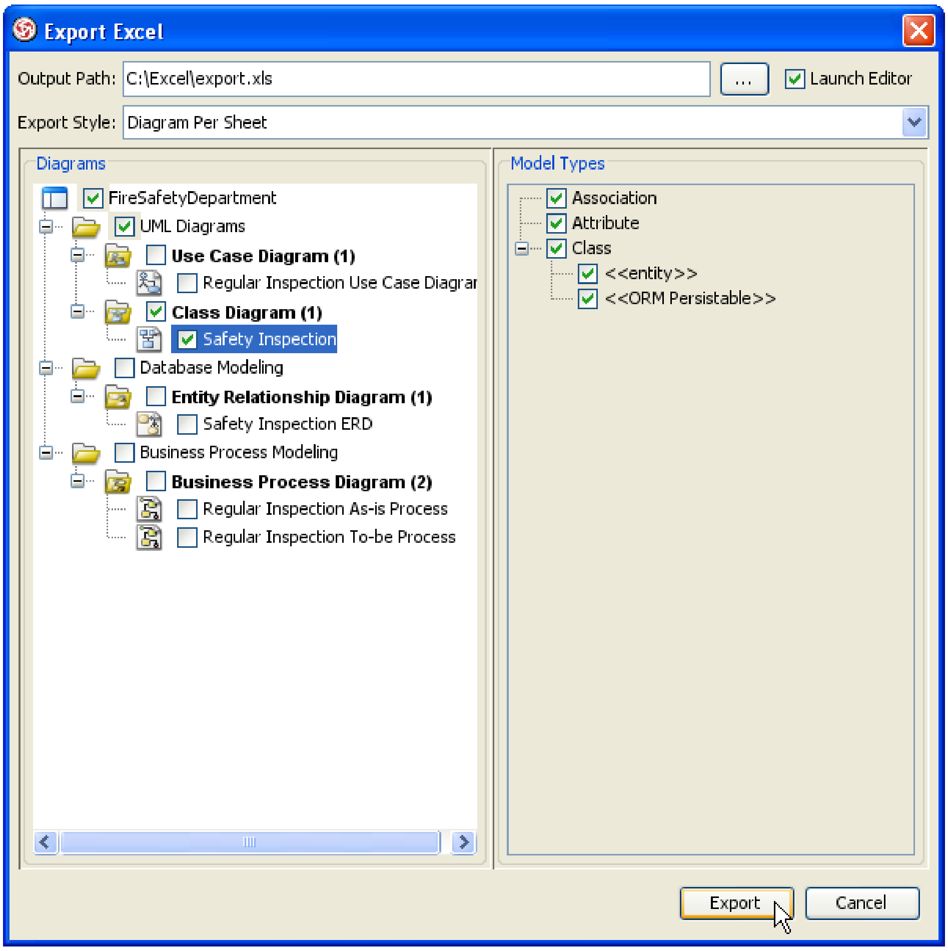
- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Regular Inspection Use Case Diagrar
    - ☑ **Class Diagram (1)**
      - ☑ Safety Inspection
  - □ Database Modeling
    - □ **Entity Relationship Diagram (1)**
      - □ Safety Inspection ERD
  - □ Business Process Modeling
    - □ **Business Process Diagram (2)**
      - □ Regular Inspection As-is Process
      - □ Regular Inspection To-be Process

Model Types

- ☑ Association
- ☑ Attribute
- ☑ Class
  - ☑ <<entity>>
  - ☑ <<ORM Persistable>>

Figure 3-10 *Describe model types*

11. Press Export button to perform the export.



Output Path: C:\Excel\export.xls ☑ Launch Editor

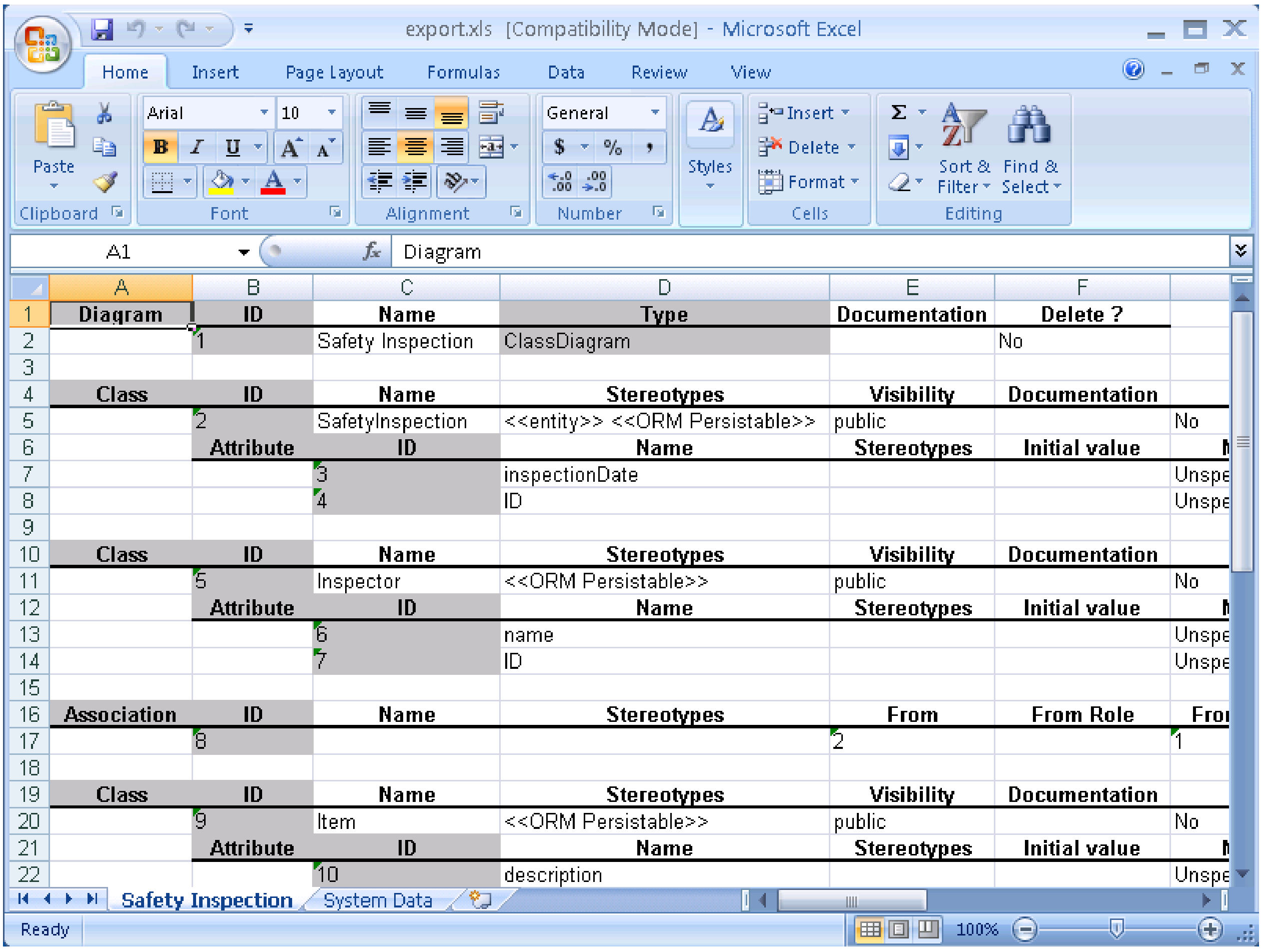
Export Style: Diagram Per Sheet

- ☑ FireSafetyDepartment
  - ☑ UML Diagrams
    - □ **Use Case Diagram (1)**
      - □ Regular Inspection Use Case Diagrar
    - ☑ **Class Diagram (1)**
      - ☑ Safety Inspection
  - □ Database Modeling
    - □ **Entity Relationship Diagram (1)**
      - □ Safety Inspection ERD
  - □ Business Process Modeling
    - □ **Business Process Diagram (2)**
      - □ Regular Inspection As-is Process
      - □ Regular Inspection To-be Process

- ☑ Association
- ☑ Attribute
- ☑ Class
  - ☑ <<entity>>
  - ☑ <<ORM Persistable>>

Figure 3-11 Export to Excel

12. Below image showing the exported Excel.



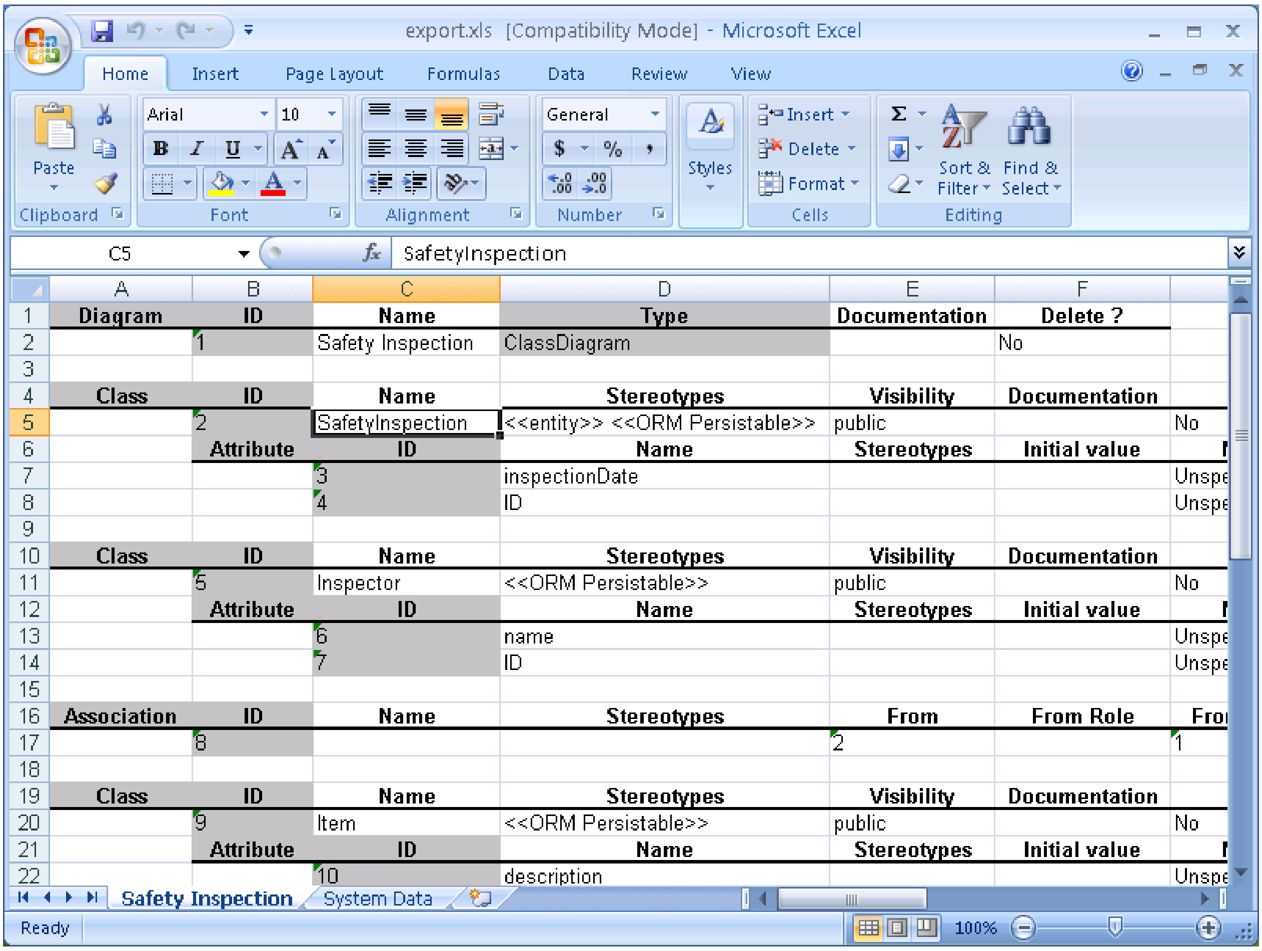
A1 | *fx* | Diagram

	A	B	C	D	E	F	
1	**Diagram**	**ID**	**Name**	**Type**	**Documentation**	**Delete ?**	
2		1	Safety Inspection	ClassDiagram		No	
3							
4	**Class**	**ID**	**Name**	**Stereotypes**	**Visibility**	**Documentation**	
5		2	SafetyInspection	<<entity>> <<ORM Persistable>>	public		No
6		**Attribute**	**ID**	**Name**	**Stereotypes**	**Initial value**	**M**
7			3	inspectionDate			Unspe
8			4	ID			Unspe
9							
10	**Class**	**ID**	**Name**	**Stereotypes**	**Visibility**	**Documentation**	
11		5	Inspector	<<ORM Persistable>>	public		No
12		**Attribute**	**ID**	**Name**	**Stereotypes**	**Initial value**	**M**
13			6	name			Unspe
14			7	ID			Unspe
15							
16	**Association**	**ID**	**Name**	**Stereotypes**	**From**	**From Role**	**Fro**
17		8			2		1
18							
19	**Class**	**ID**	**Name**	**Stereotypes**	**Visibility**	**Documentation**	
20		9	Item	<<ORM Persistable>>	public		No
21		**Attribute**	**ID**	**Name**	**Stereotypes**	**Initial value**	**M**
22			10	description			Unspe

Figure 3-12 *Review exported Excel*

Importing Microsoft Excel file

1. Open exported Excel file.

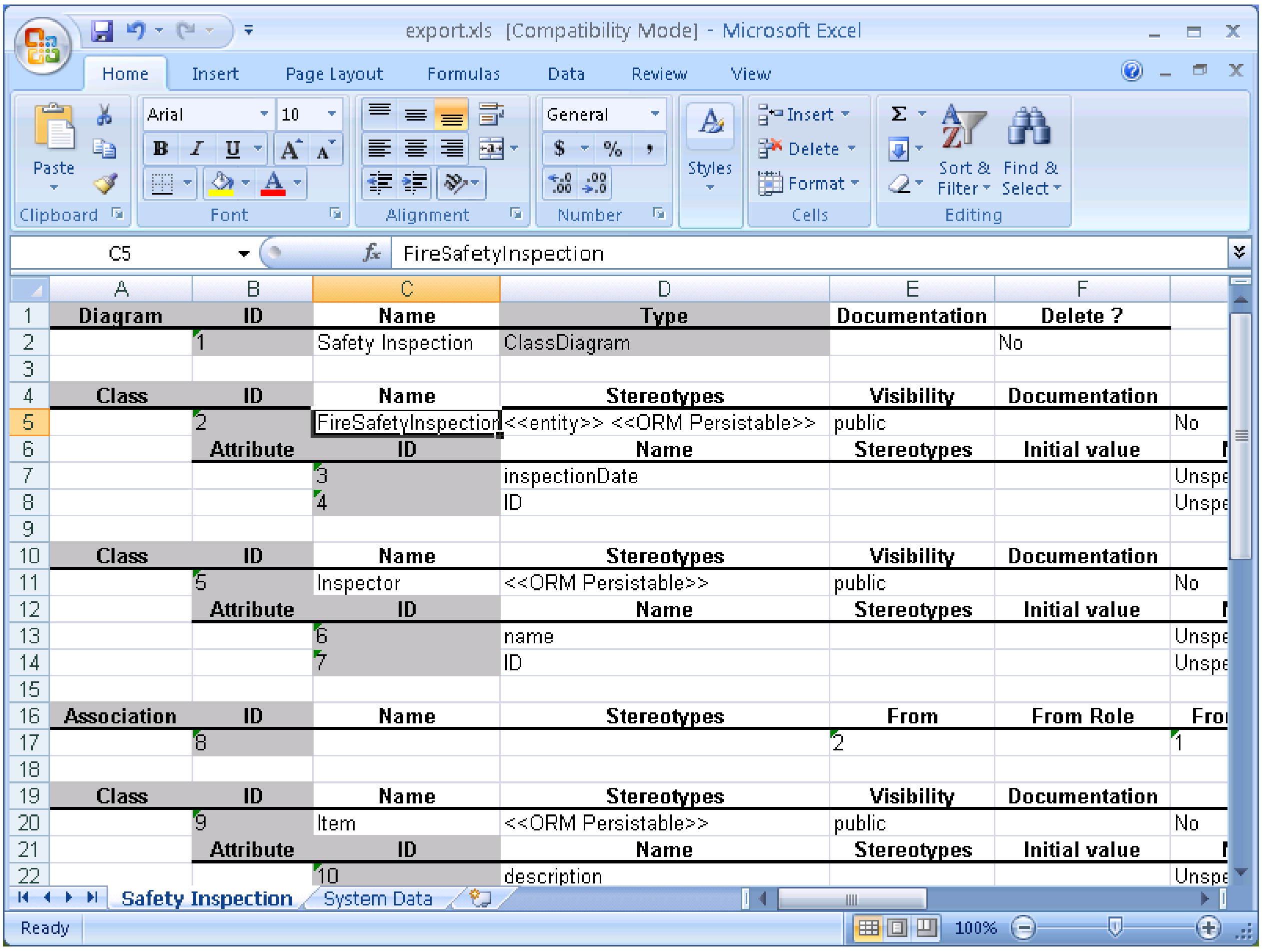


C5 *fx* SafetyInspection

	A	B	C	D	E	F	
1	Diagram	ID	Name	Type	Documentation	Delete ?	
2		1	Safety Inspection	ClassDiagram		No	
3							
4	Class	ID	Name	Stereotypes	Visibility	Documentation	
5		2	SafetyInspection	<<entity>> <<ORM Persistable>>	public		No
6		Attribute	ID	Name	Stereotypes	Initial value	I
7			3	inspectionDate			Unspe
8			4	ID			Unspe
9							
10	Class	ID	Name	Stereotypes	Visibility	Documentation	
11		5	Inspector	<<ORM Persistable>>	public		No
12		Attribute	ID	Name	Stereotypes	Initial value	I
13			6	name			Unspe
14			7	ID			Unspe
15							
16	Association	ID	Name	Stereotypes	From	From Role	Fro
17		8			2		1
18							
19	Class	ID	Name	Stereotypes	Visibility	Documentation	
20		9	Item	<<ORM Persistable>>	public		No
21		Attribute	ID	Name	Stereotypes	Initial value	I
22			10	description			Unspe

Figure 3-13 *Exported Excel*

2. Rename Class *SafetyInspection* to *FireSafetyInspection*.

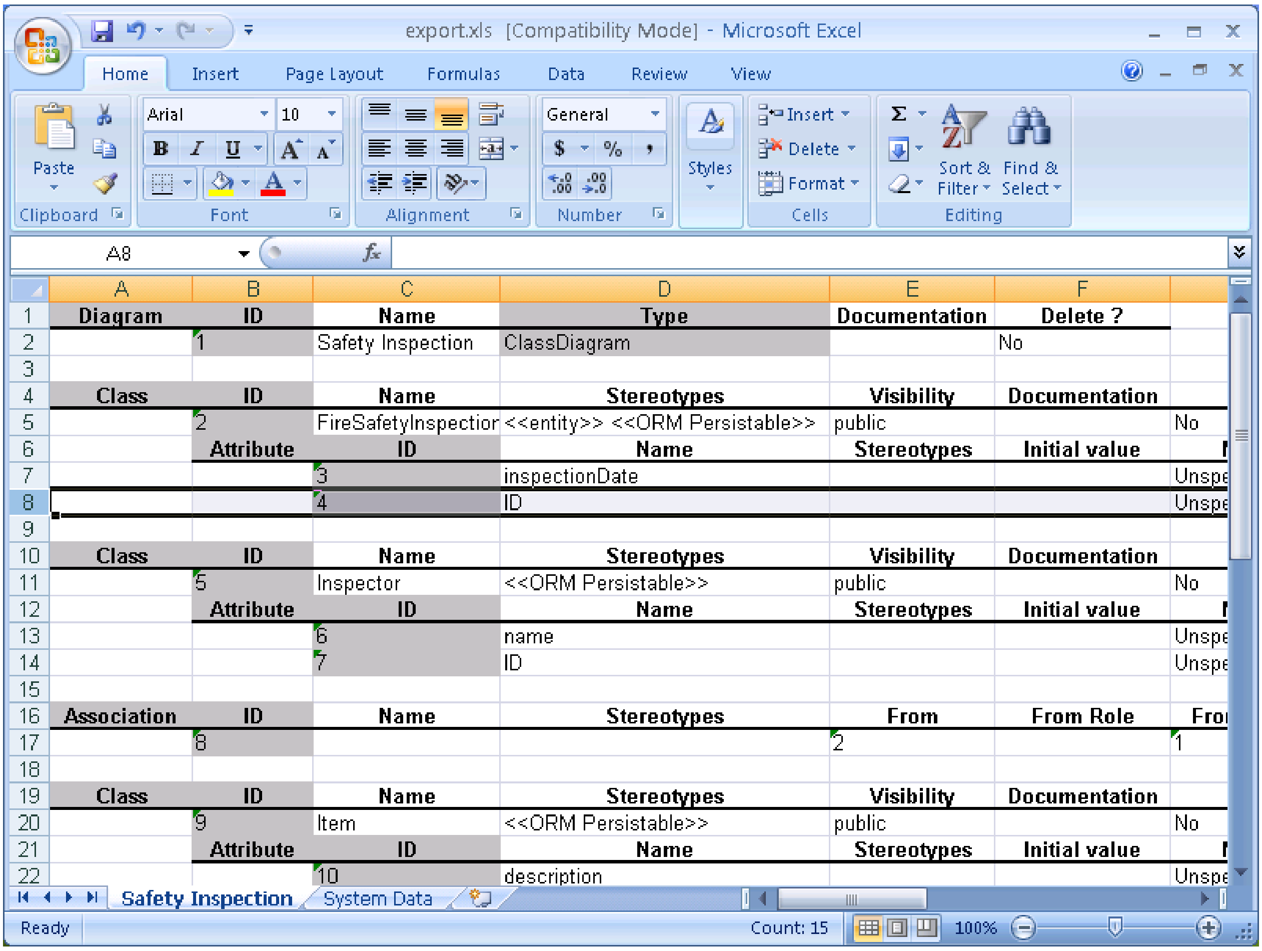


C5 FireSafetyInspection

	A	B	C	D	E	F	
1	Diagram	ID	Name	Type	Documentation	Delete ?	
2		1	Safety Inspection	ClassDiagram		No	
3							
4	Class	ID	Name	Stereotypes	Visibility	Documentation	
5		2	FireSafetyInspection	<<entity>> <<ORM Persistable>>	public		No
6		Attribute	ID	Name	Stereotypes	Initial value	I
7			3	inspectionDate			Unspe
8			4	ID			Unspe
9							
10	Class	ID	Name	Stereotypes	Visibility	Documentation	
11		5	Inspector	<<ORM Persistable>>	public		No
12		Attribute	ID	Name	Stereotypes	Initial value	I
13			6	name			Unspe
14			7	ID			Unspe
15							
16	Association	ID	Name	Stereotypes	From	From Role	Fro
17		8			2		1
18							
19	Class	ID	Name	Stereotypes	Visibility	Documentation	
20		9	Item	<<ORM Persistable>>	public		No
21		Attribute	ID	Name	Stereotypes	Initial value	I
22			10	description			Unspe

Figure 3-14 *Renaming class in Excel*

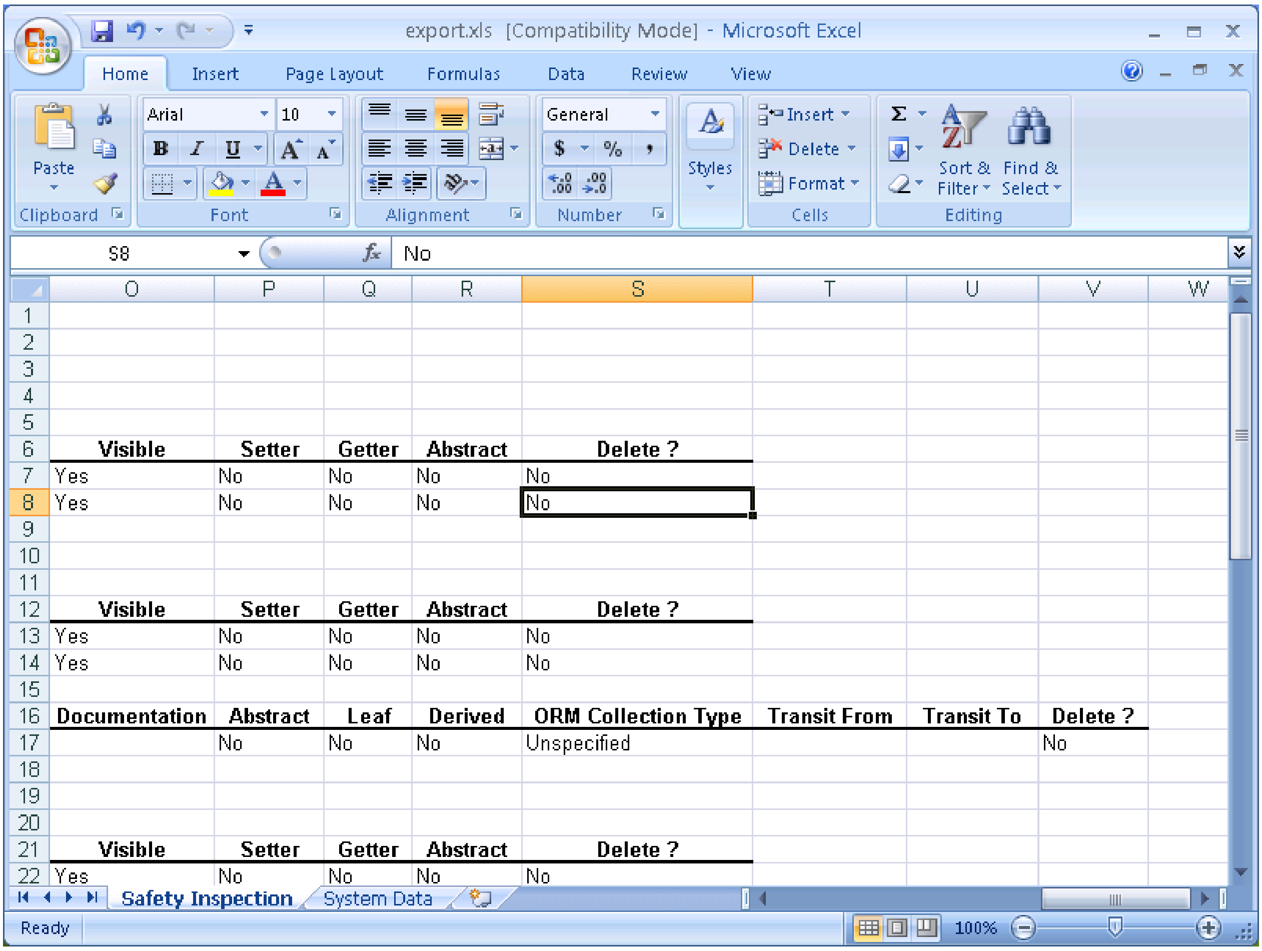
3. It is possible to delete Attribute from Class such as selected Attribute.



	A	B	C	D	E	F	
1	Diagram	ID	Name	Type	Documentation	Delete ?	
2		1	Safety Inspection	ClassDiagram		No	
3							
4	Class	ID	Name	Stereotypes	Visibility	Documentation	
5		2	FireSafetyInspectior	<<entity>> <<ORM Persistable>>	public		No
6		Attribute	ID	Name	Stereotypes	Initial value	I
7			3	inspectionDate			Unspe
8			4	ID			Unspe
9							
10	Class	ID	Name	Stereotypes	Visibility	Documentation	
11		5	Inspector	<<ORM Persistable>>	public		No
12		Attribute	ID	Name	Stereotypes	Initial value	I
13			6	name			Unspe
14			7	ID			Unspe
15							
16	Association	ID	Name	Stereotypes	From	From Role	Fro
17		8			2		1
18							
19	Class	ID	Name	Stereotypes	Visibility	Documentation	
20		9	Item	<<ORM Persistable>>	public		No
21		Attribute	ID	Name	Stereotypes	Initial value	I
22			10	description			Unspe

Figure 3-15 *Highlight attribute in Excel*

4. Delete cell can be used to identify delete or not.



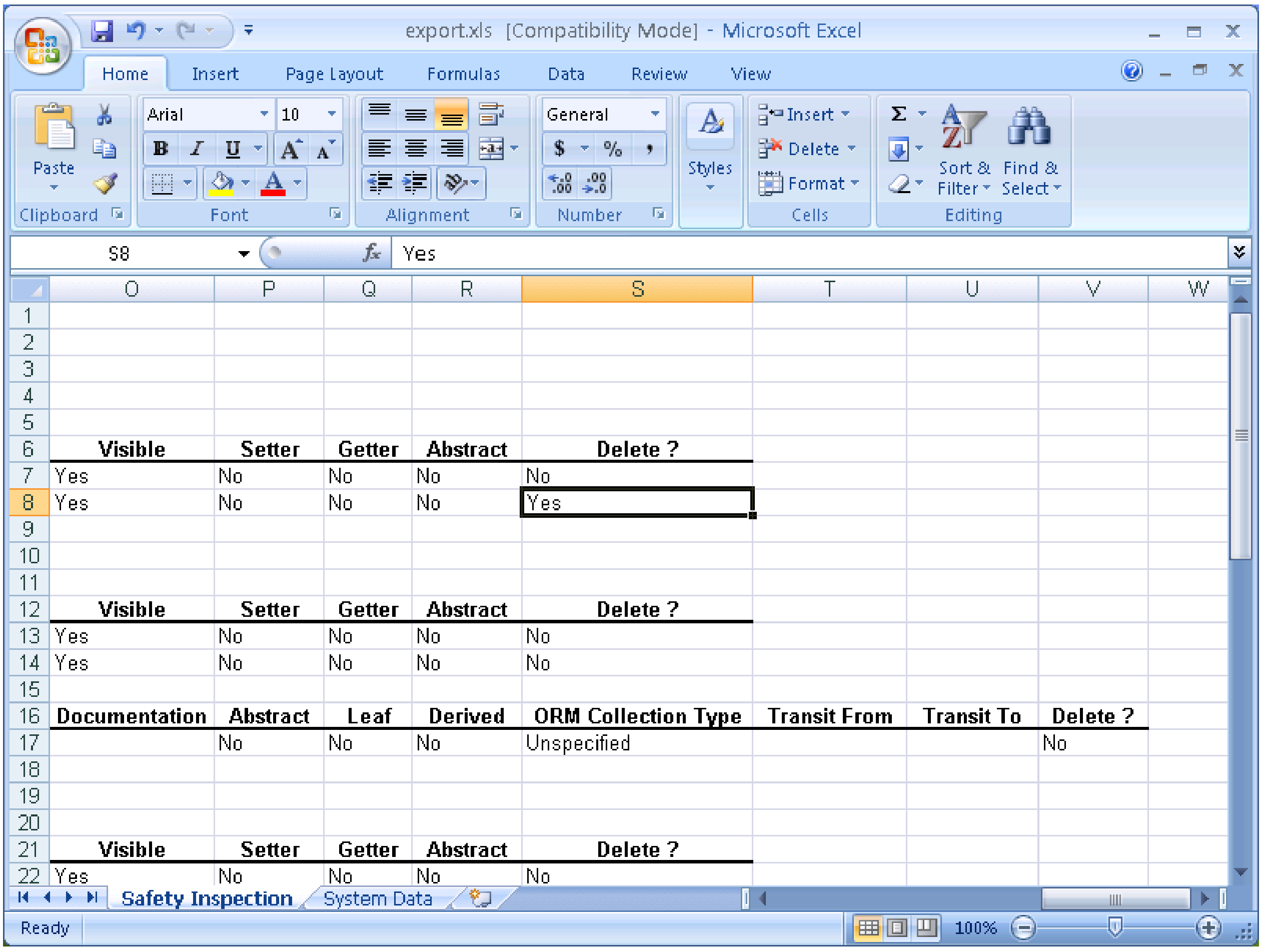
S8 | *fx* | No

	O	P	Q	R	S	T	U	V	W
1									
2									
3									
4									
5									
6	**Visible**	**Setter**	**Getter**	**Abstract**	**Delete ?**				
7	Yes	No	No	No	No				
8	Yes	No	No	No	No				
9									
10									
11									
12	**Visible**	**Setter**	**Getter**	**Abstract**	**Delete ?**				
13	Yes	No	No	No	No				
14	Yes	No	No	No	No				
15									
16	**Documentation**	**Abstract**	**Leaf**	**Derived**	**ORM Collection Type**	**Transit From**	**Transit To**	**Delete ?**	
17		No	No	No	Unspecified			No	
18									
19									
20									
21	**Visible**	**Setter**	**Getter**	**Abstract**	**Delete ?**				
22	Yes	No	No	No	No				

Safety Inspection | System Data

Figure 3-16 Highlight delete cell in Excel

5. Change value of Delete cell from *No* to *Yes* to mark Attribute to be delete.

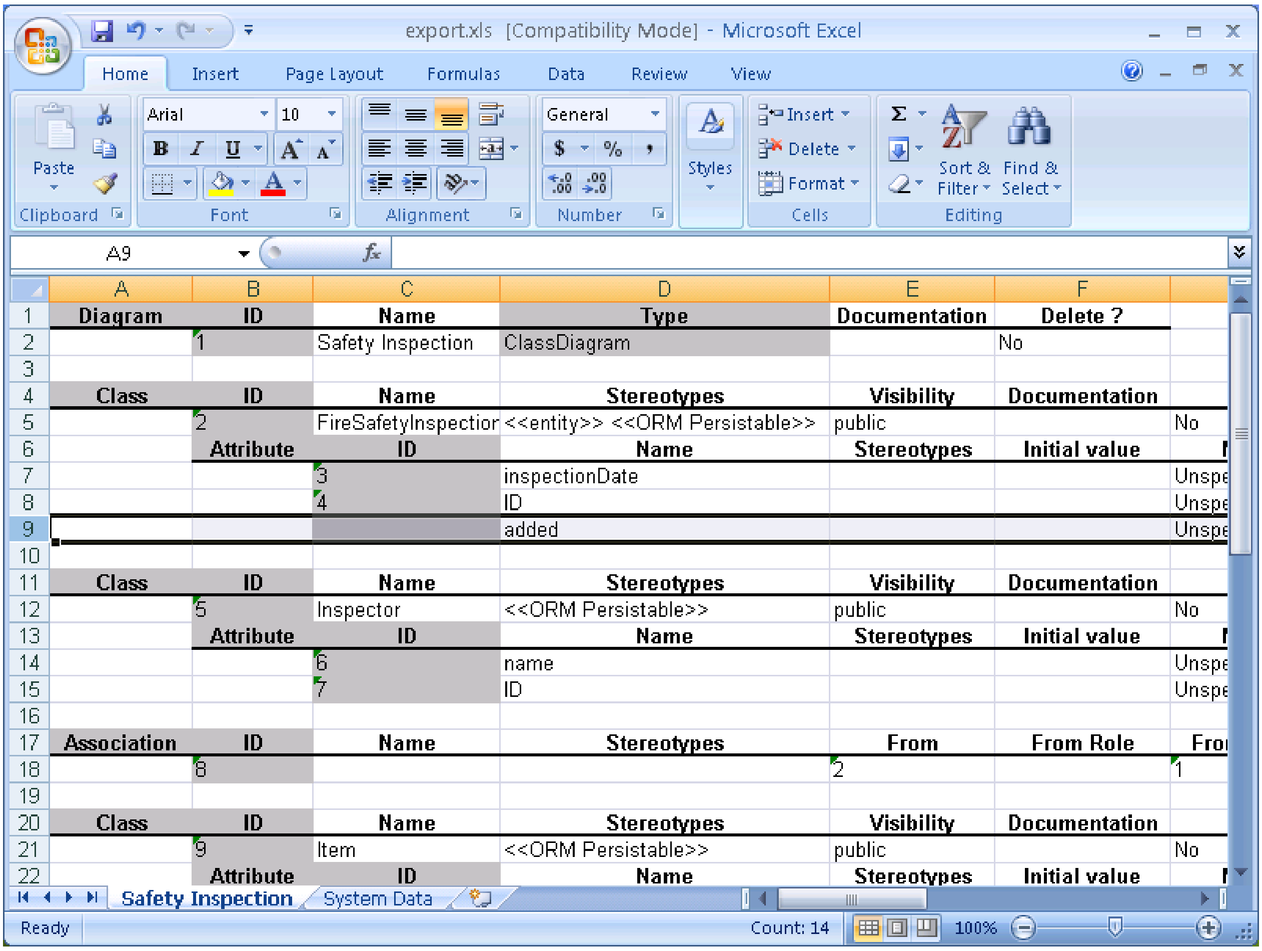


S8 *fx* Yes

	O	P	Q	R	S	T	U	V	W
1									
2									
3									
4									
5									
6	**Visible**	**Setter**	**Getter**	**Abstract**	**Delete ?**				
7	Yes	No	No	No	No				
8	Yes	No	No	No	Yes				
9									
10									
11									
12	**Visible**	**Setter**	**Getter**	**Abstract**	**Delete ?**				
13	Yes	No	No	No	No				
14	Yes	No	No	No	No				
15									
16	**Documentation**	**Abstract**	**Leaf**	**Derived**	**ORM Collection Type**	**Transit From**	**Transit To**	**Delete ?**	
17		No	No	No	Unspecified			No	
18									
19									
20									
21	**Visible**	**Setter**	**Getter**	**Abstract**	**Delete ?**				
22	Yes	No	No	No	No				

Figure 3-17 *Deleting attribute in Excel*

6. It is possible to add new Attribute similar to existing Attribute except gray cell. Notice, do not modify values in gray cell or System Data Sheet.



	A	B	C	D	E	F	
1	Diagram	ID	Name	Type	Documentation	Delete ?	
2		1	Safety Inspection	ClassDiagram		No	
3							
4	Class	ID	Name	Stereotypes	Visibility	Documentation	
5		2	FireSafetyInspectior	<<entity>> <<ORM Persistable>>	public		No
6		Attribute	ID	Name	Stereotypes	Initial value	I
7			3	inspectionDate			Unspe
8			4	ID			Unspe
9				added			Unspe
10							
11	Class	ID	Name	Stereotypes	Visibility	Documentation	
12		5	Inspector	<<ORM Persistable>>	public		No
13		Attribute	ID	Name	Stereotypes	Initial value	I
14			6	name			Unspe
15			7	ID			Unspe
16							
17	Association	ID	Name	Stereotypes	From	From Role	Fro
18		8			2		1
19							
20	Class	ID	Name	Stereotypes	Visibility	Documentation	
21		9	Item	<<ORM Persistable>>	public		No
22		Attribute	ID	Name	Stereotypes	Initial value	I

Figure 3-18 *Adding attribute in Excel*

7. Below image showing project will to import Excel.

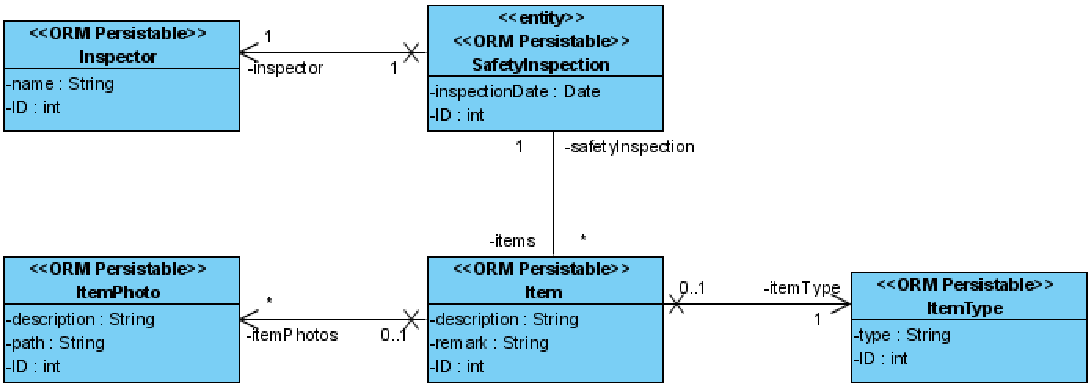


Figure 3-19 *Project before import*

8. Mouse over main menu.

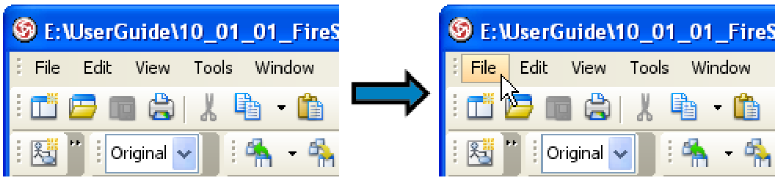


Figure 3-20 *Mouse over main menu*

9. Click on File of main menu.

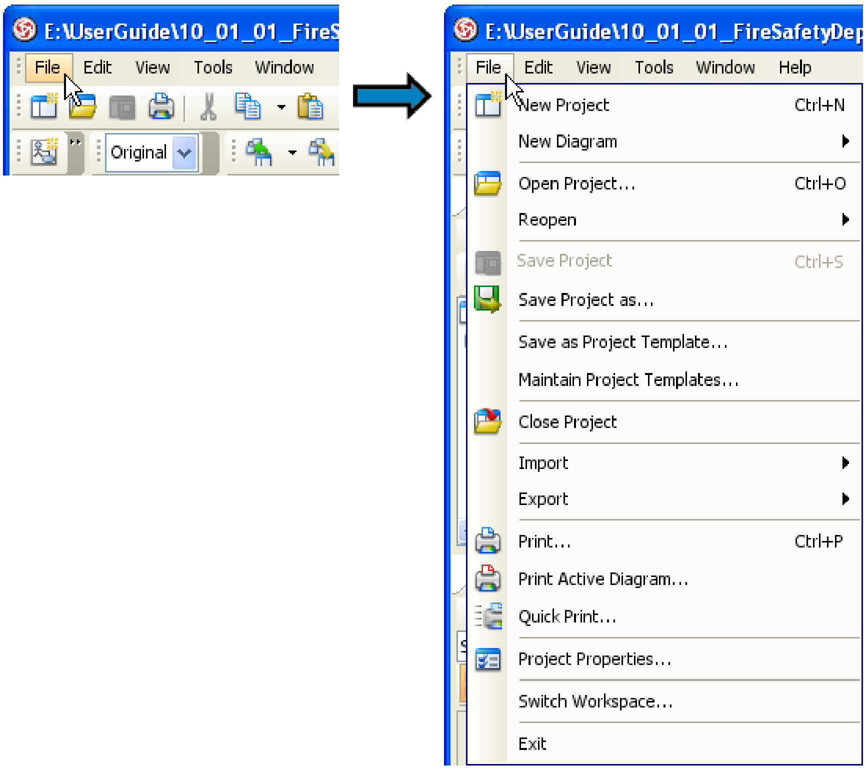
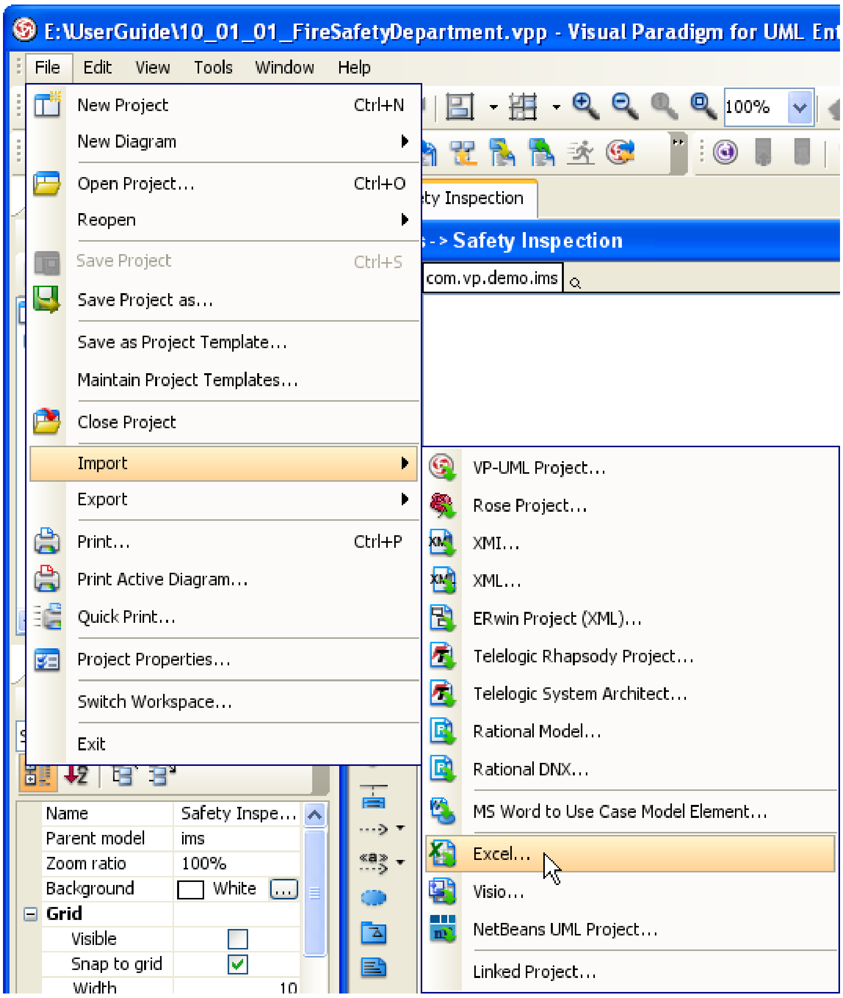


Figure 3-21 *Showing popup of main menu*

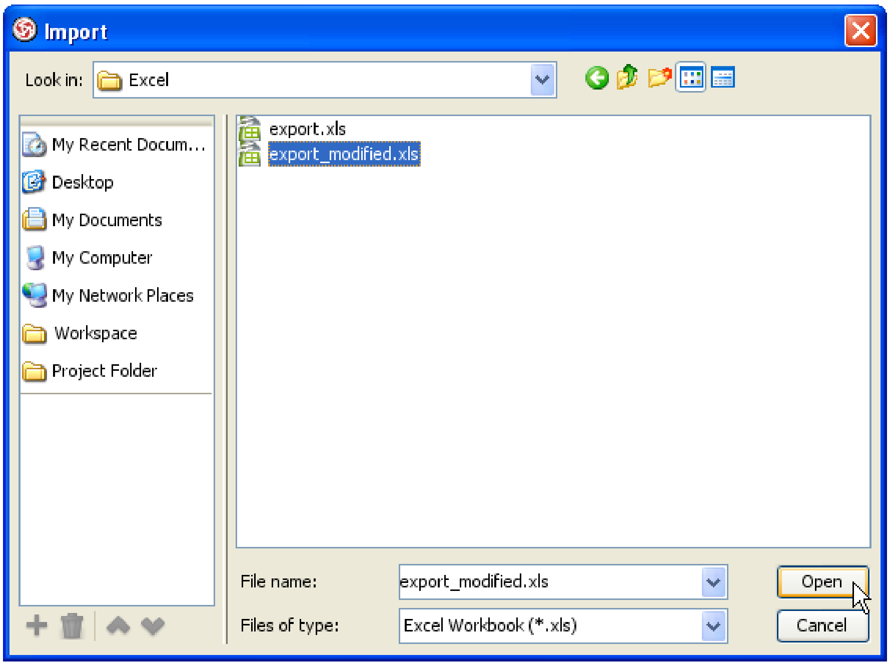
10. Click on Import > Excel... menu item.



Name	Safety Inspe...
Parent model	ims
Zoom ratio	100%
Background	□ White
Grid	
Visible	□
Snap to grid	☑
Width	10

Figure 3-22 Import Excel

11. Select modified Excel then press Open button.



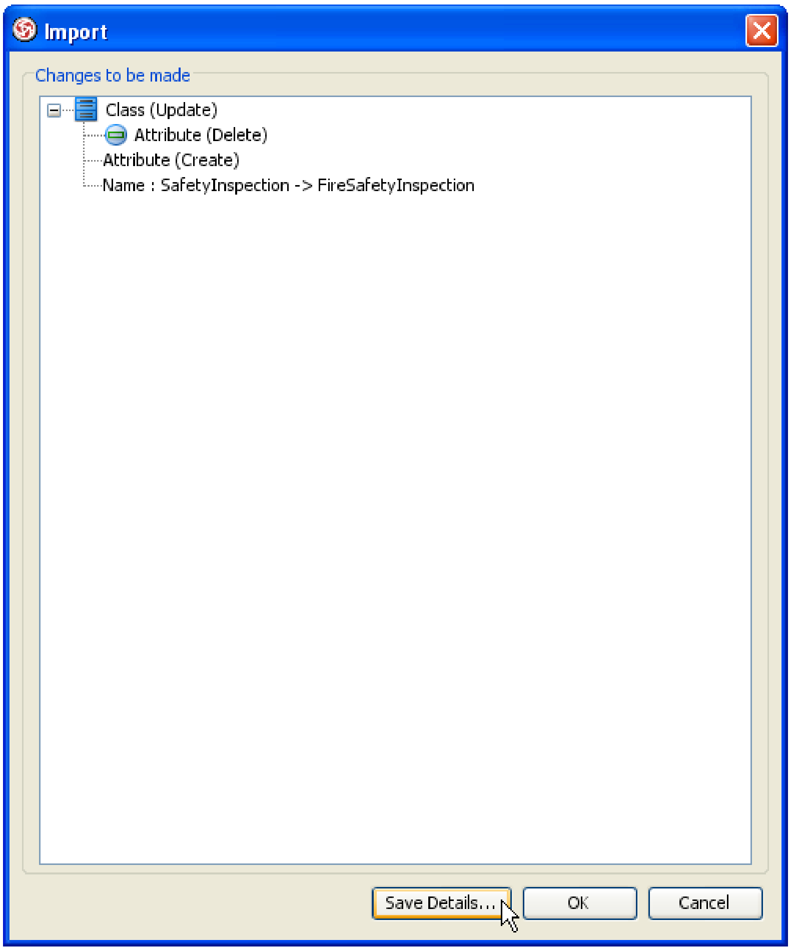
Look in: Excel

File name: export_modified.xls

Files of type: Excel Workbook (*.xls)

Figure 3-23 Selecting Excel to be import

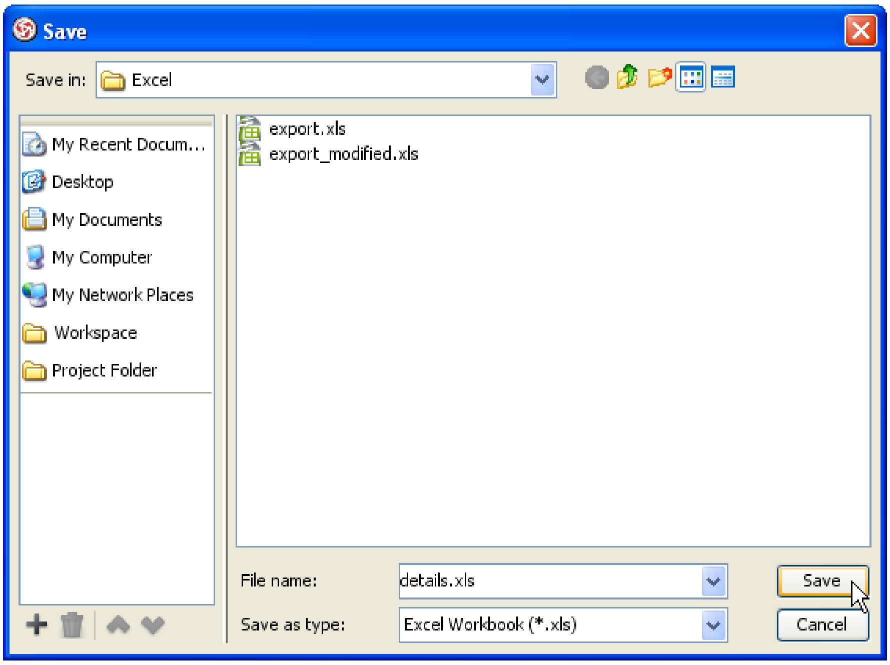
12. Press Save Details... button to save changes to Excel.



Class (Update)
Attribute (Delete)
Attribute (Create)
Name : SafetyInspection -> FireSafetyInspection

Figure 3-24 Save details

13. Press Save button to perform the save.



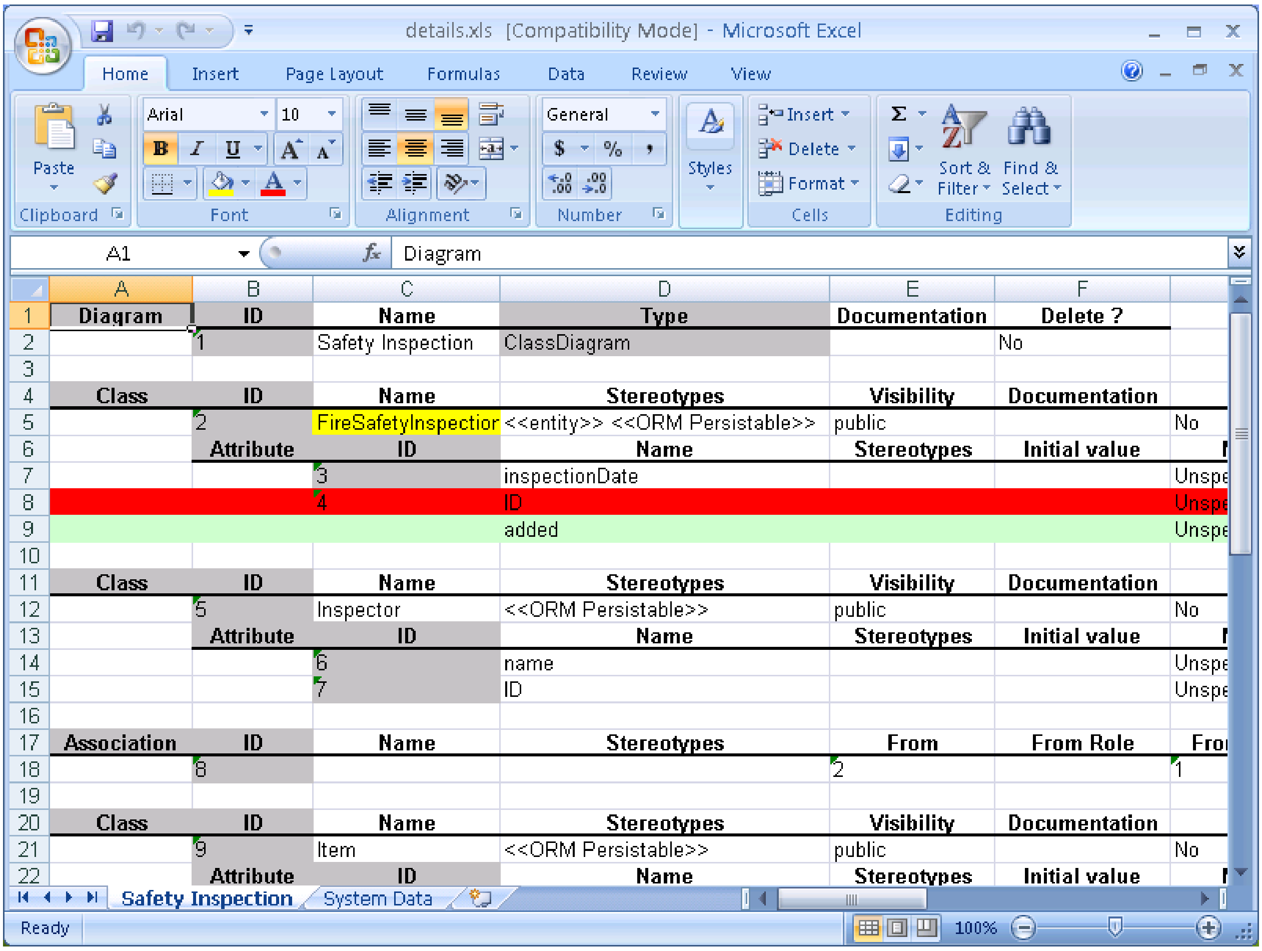
Save in: Excel

File name: details.xls

Save as type: Excel Workbook (*.xls)

Figure 3-25 Inputting Excel to be saved

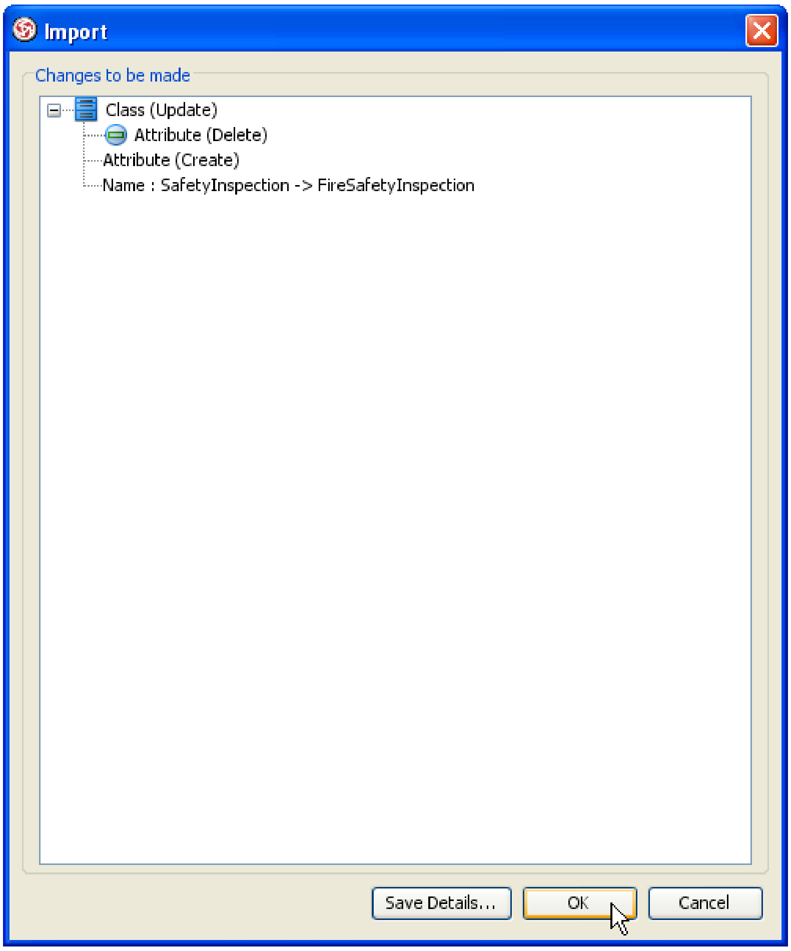
14. Below image showing saved details.



	A	B	C	D	E	F	
1	Diagram	ID	Name	Type	Documentation	Delete ?	
2		1	Safety Inspection	ClassDiagram		No	
3							
4	Class	ID	Name	Stereotypes	Visibility	Documentation	
5		2	FireSafetyInspectior	<<entity>> <<ORM Persistable>>	public		No
6		Attribute	ID	Name	Stereotypes	Initial value	I
7			3	inspectionDate			Unspe
8			4	ID			Unspe
9				added			Unspe
10							
11	Class	ID	Name	Stereotypes	Visibility	Documentation	
12		5	Inspector	<<ORM Persistable>>	public		No
13		Attribute	ID	Name	Stereotypes	Initial value	I
14			6	name			Unspe
15			7	ID			Unspe
16							
17	Association	ID	Name	Stereotypes	From	From Role	Fro
18		8			2		1
19							
20	Class	ID	Name	Stereotypes	Visibility	Documentation	
21		9	Item	<<ORM Persistable>>	public		No
22		Attribute	ID	Name	Stereotypes	Initial value	I

Figure 3-26 *Saved details*

15. Press OK button to perform the import.



Changes to be made

- Class (Update)
  - Attribute (Delete)
  - Attribute (Create)
  - Name : SafetyInspection -> FireSafetyInspection

Figure 3-27 *Import Excel*

16. After the import, Class *SafetyInspection* renamed to *FireSafetyInspection and* Attribute *ID* is deleted also Attribute *added* is added.

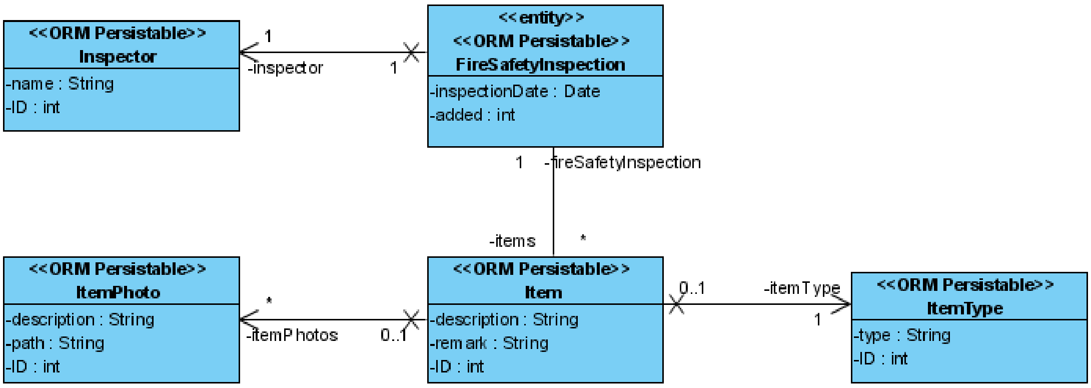


Figure 3-28 *Result of import Excel*

Exporting XMI

Exporting project to XMI

1. Mouse over main menu.

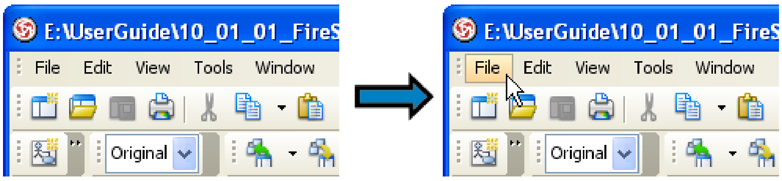


Figure 4-1 *Mouse over main menu*

2. Click on File of main menu.

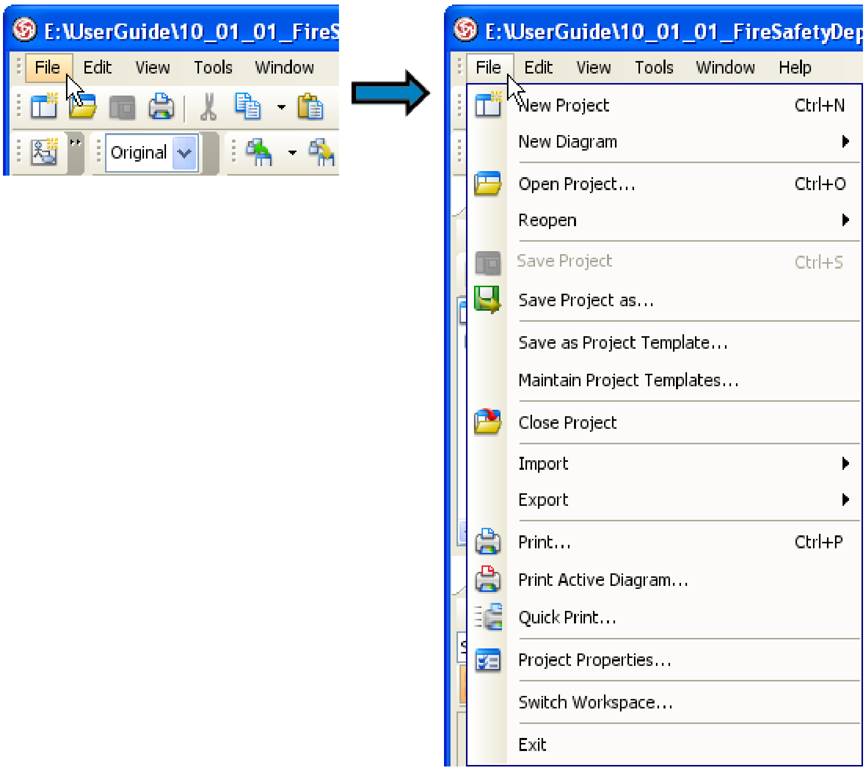


Figure 4-2 *Showing popup of main menu*

3. Click on Export > XMI... menu item.

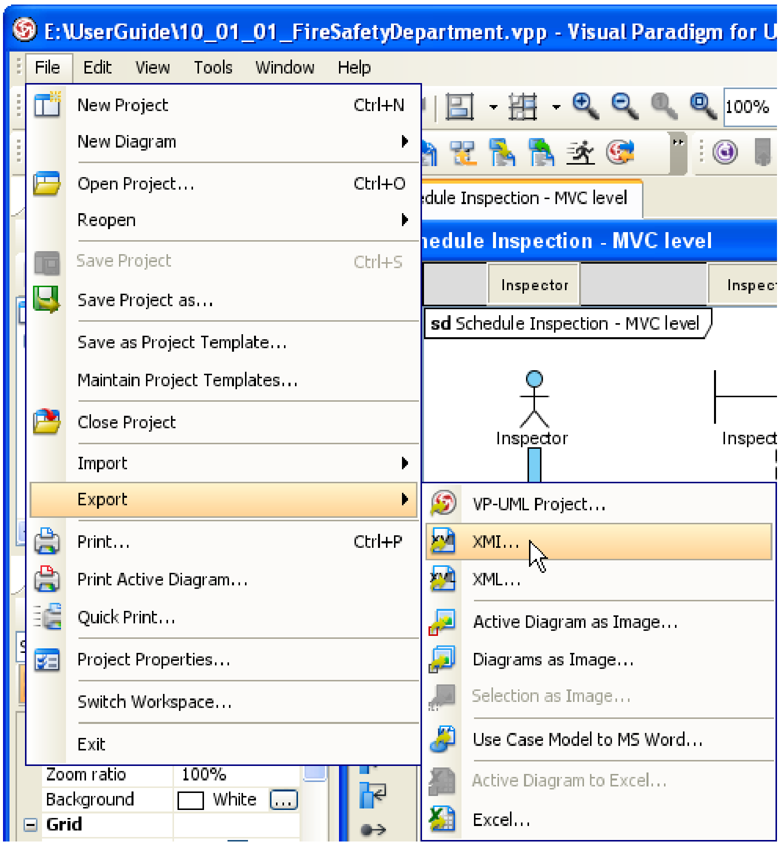
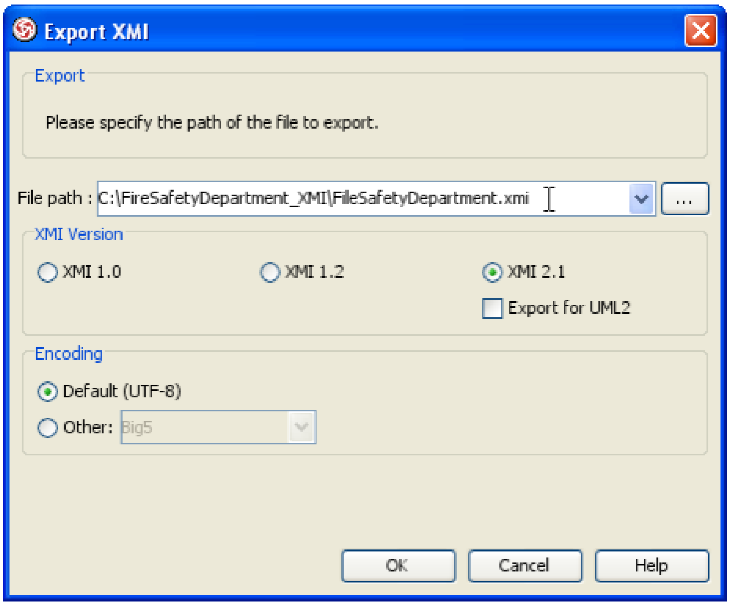


Figure 4-3 *Export XMI*

4. Input *C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi* in File path.


Export XMI

Export

Please specify the path of the file to export.

File path : C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi

XMI Version

○ XMI 1.0 ○ XMI 1.2 ● XMI 2.1

□ Export for UML2

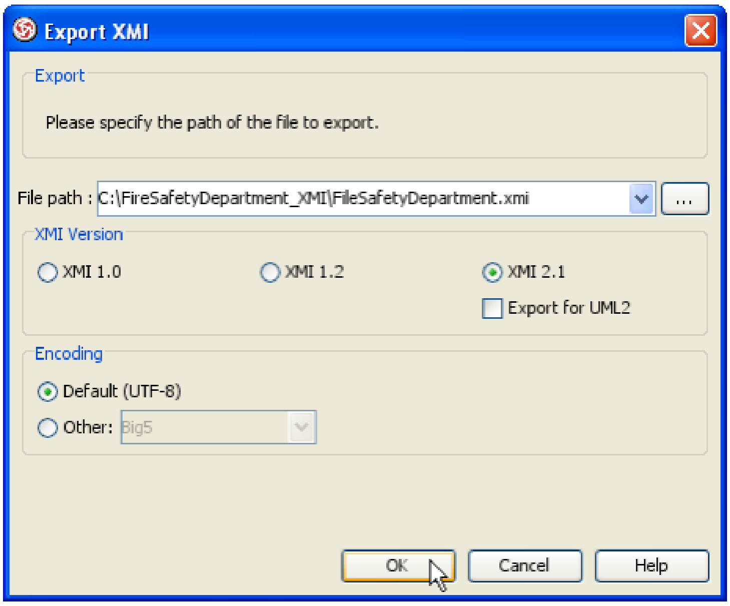
Encoding

● Default (UTF-8)

○ Other: Big5

Figure 4-4 *Inputting file path*

5. Check XMI 2.1 is checkbox and press OK button.



Export XMI

Export

Please specify the path of the file to export.

File path : C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi

XMI Version

○ XMI 1.0 ○ XMI 1.2 ● XMI 2.1

□ Export for UML2

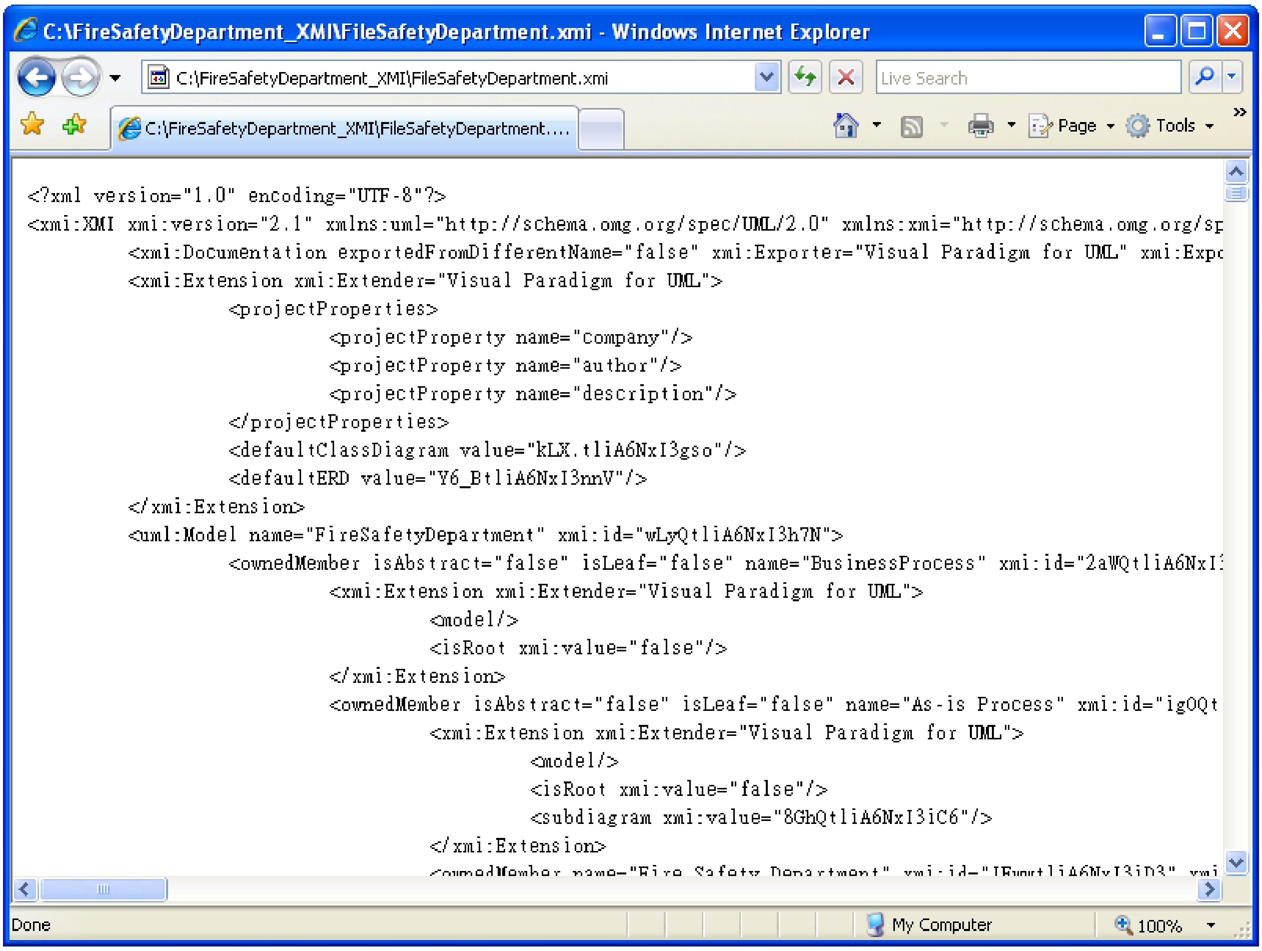
Encoding

● Default (UTF-8)

○ Other: Big5

Figure 4-5 *Export to XMI*

6. Below image showing the exported XMI.



C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi - Windows Internet Explorer

```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.1" xmlns:uml="http://schema.omg.org/spec/UML/2.0" xmlns:xmi="http://schema.omg.org/sp
        <xmi:Documentation exportedFromDifferentName="false" xmi:Exporter="Visual Paradigm for UML" xmi:Expo
        <xmi:Extension xmi:Extender="Visual Paradigm for UML">
                <projectProperties>
                        <projectProperty name="company"/>
                        <projectProperty name="author"/>
                        <projectProperty name="description"/>
                </projectProperties>
                <defaultClassDiagram value="kLX.tliA6NxI3gso"/>
                <defaultERD value="Y6_BtliA6NxI3nnV"/>
        </xmi:Extension>
        <uml:Model name="FireSafetyDepartment" xmi:id="wLyQtliA6NxI3h7N">
                <ownedMember isAbstract="false" isLeaf="false" name="BusinessProcess" xmi:id="2aWQtliA6NxI
                        <xmi:Extension xmi:Extender="Visual Paradigm for UML">
                                <model/>
                                <isRoot xmi:value="false"/>
                        </xmi:Extension>
                        <ownedMember isAbstract="false" isLeaf="false" name="As-is Process" xmi:id="igOQt
                                <xmi:Extension xmi:Extender="Visual Paradigm for UML">
                                        <model/>
                                        <isRoot xmi:value="false"/>
                                        <subdiagram xmi:value="8GhQtliA6NxI3iC6"/>
                                </xmi:Extension>
                                <ownedMember name="Fire Safety Department" xmi:id="IFwwtliA6NxI3iD3" xmi
```

Figure 4-6 *Review exported XMI*

Exporting diagrams to XML with command line interface

1. Mouse over Start button.

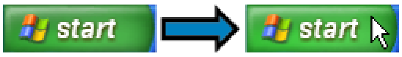


Figure 4-7 *Mouse over start button*

2. Press Start button.

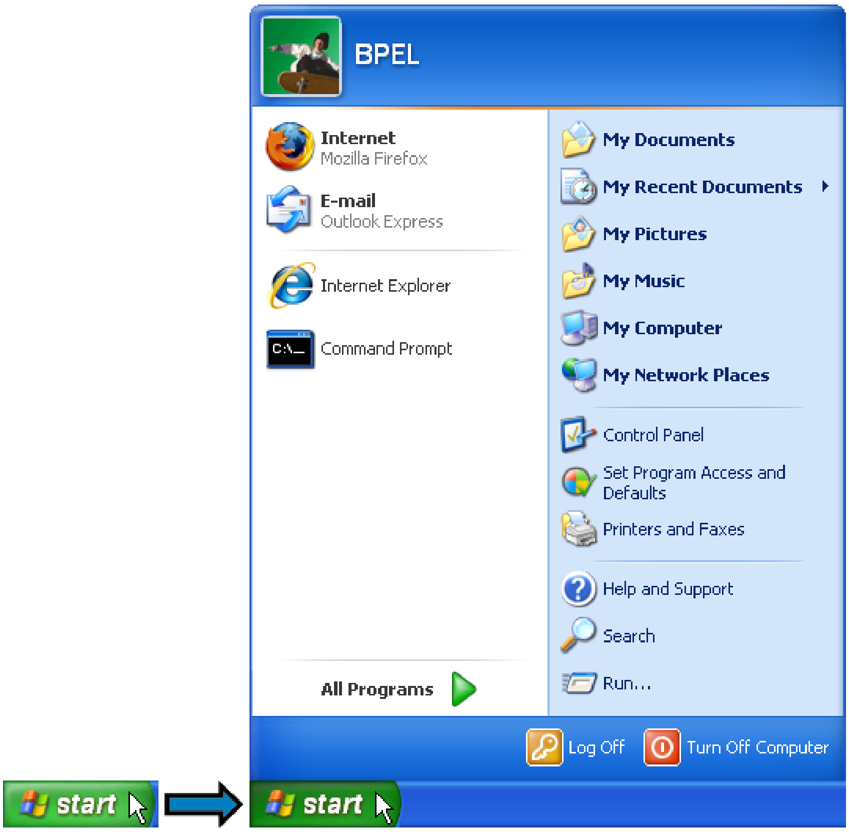


Figure 4-8 *Showing start menu*

3. Click on Run... button.

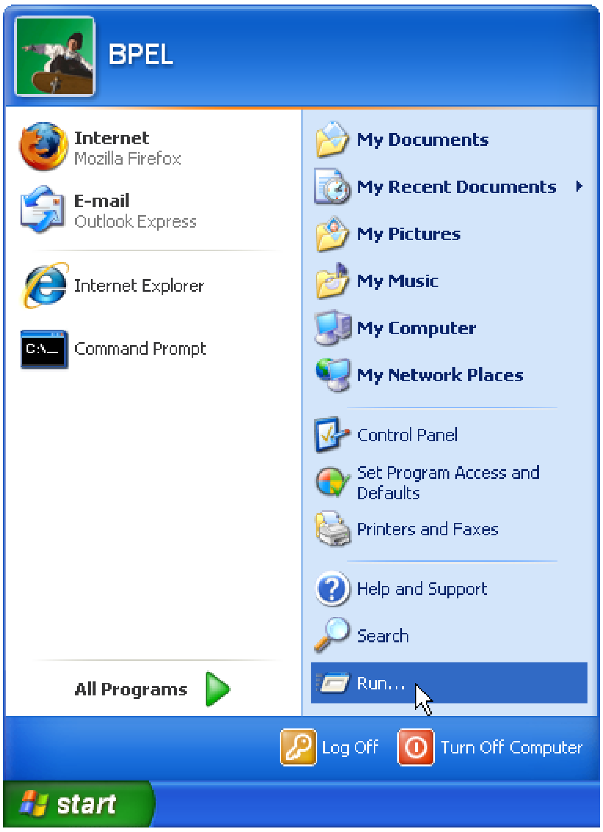
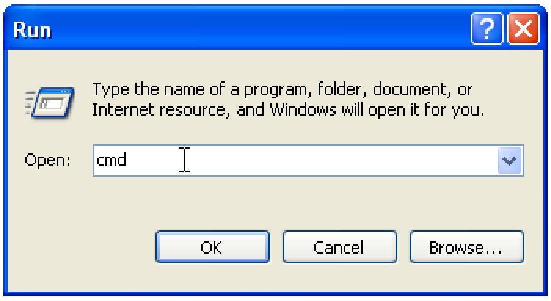


Figure 4-9 *Run from start menu*

4. Input cmd as command.

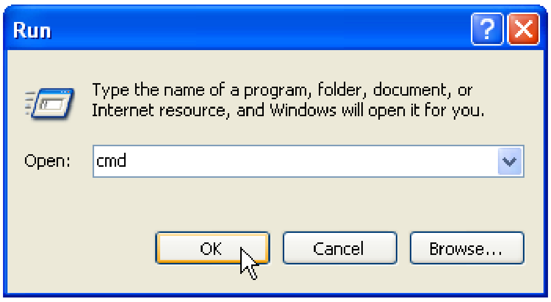


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Figure 4-10 *Inputting command to run*

5. Press OK button.

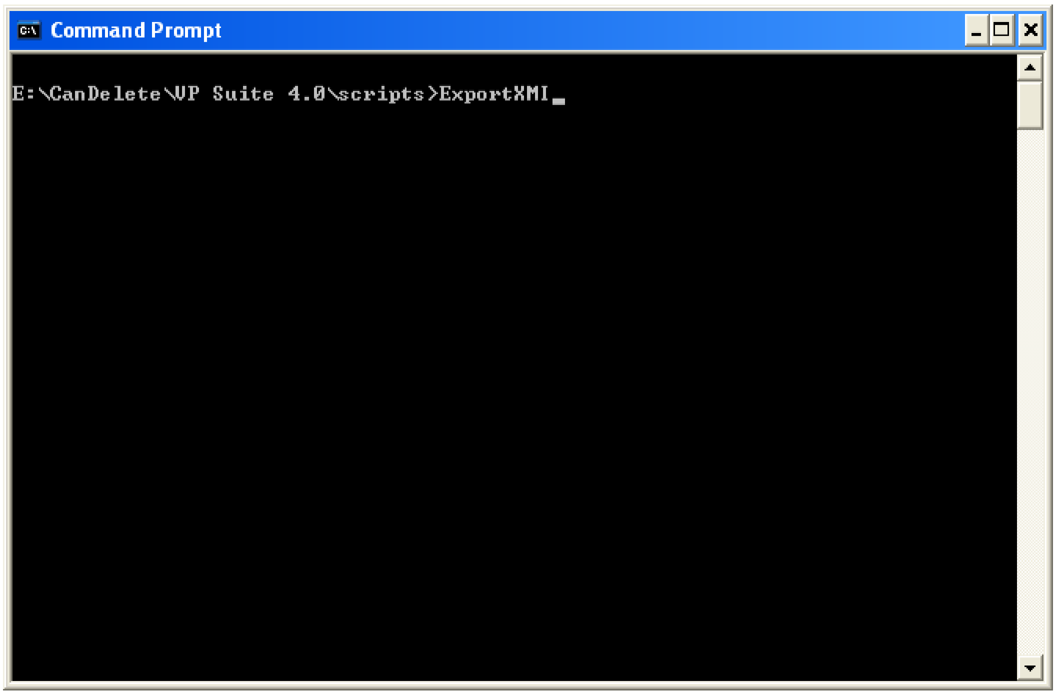


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Figure 4-11 *Run command*

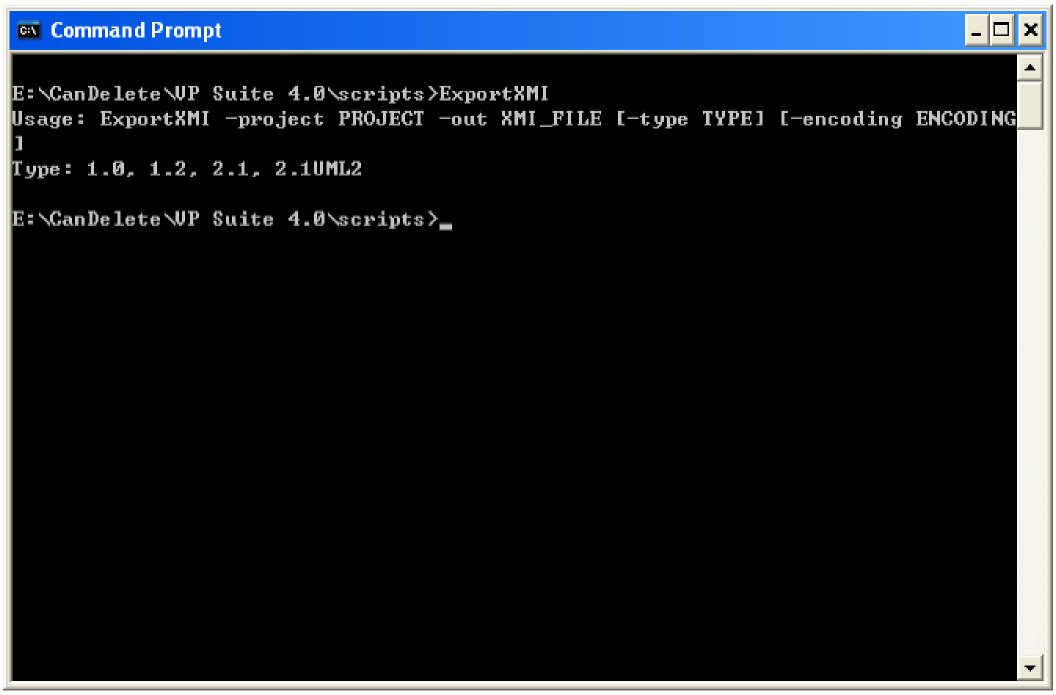
6. Change directory to scripts folder and execute ExportXMI.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXMI_
```

Figure 4-12 *Inputting command*

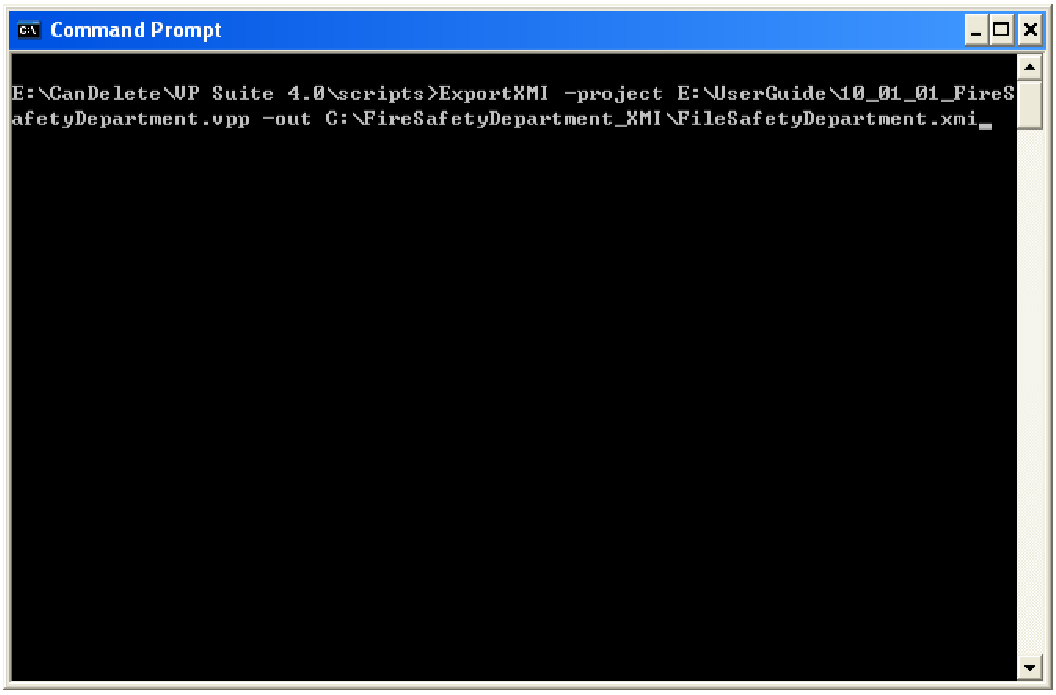
7. Usage of ExportXMI will be shown.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXMI
Usage: ExportXMI -project PROJECT -out XMI_FILE [-type TYPE] [-encoding ENCODING
]
Type: 1.0, 1.2, 2.1, 2.1UML2

E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 4-13 *Execute result of command*

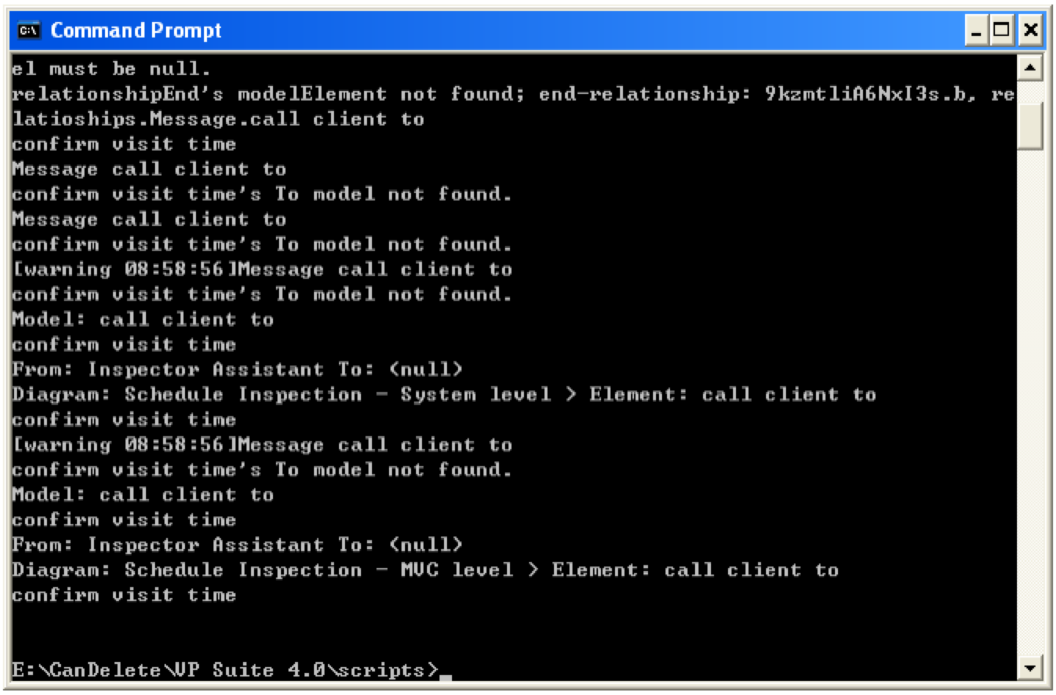
8. Example of exporting XMI.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXMI -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi_
```

Figure 4-14 Inputting command

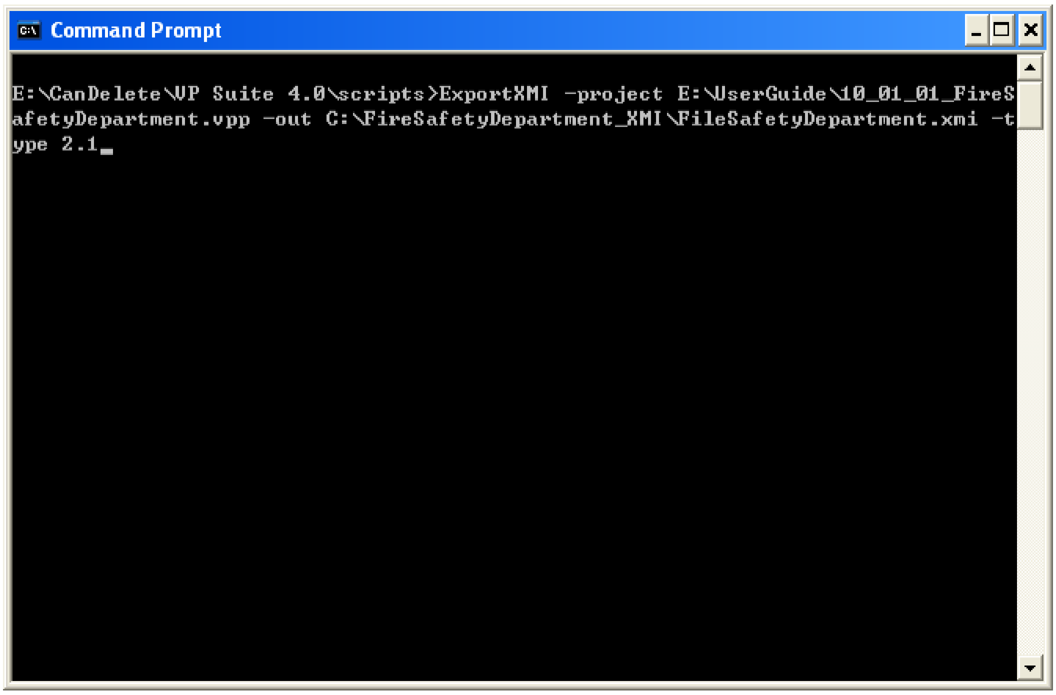
9. Exeute result of exporting XMI.



```
el must be null.
relationshipEnd's modelElement not found; end-relationship: 9kzmtliA6NxI3s.b, re
latioships.Message.call client to
confirm visit time
Message call client to
confirm visit time's To model not found.
Message call client to
confirm visit time's To model not found.
[warning 08:58:56]Message call client to
confirm visit time's To model not found.
Model: call client to
confirm visit time
From: Inspector Assistant To: <null>
Diagram: Schedule Inspection - System level > Element: call client to
confirm visit time
[warning 08:58:56]Message call client to
confirm visit time's To model not found.
Model: call client to
confirm visit time
From: Inspector Assistant To: <null>
Diagram: Schedule Inspection - MVC level > Element: call client to
confirm visit time


E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 4-15 Execute result

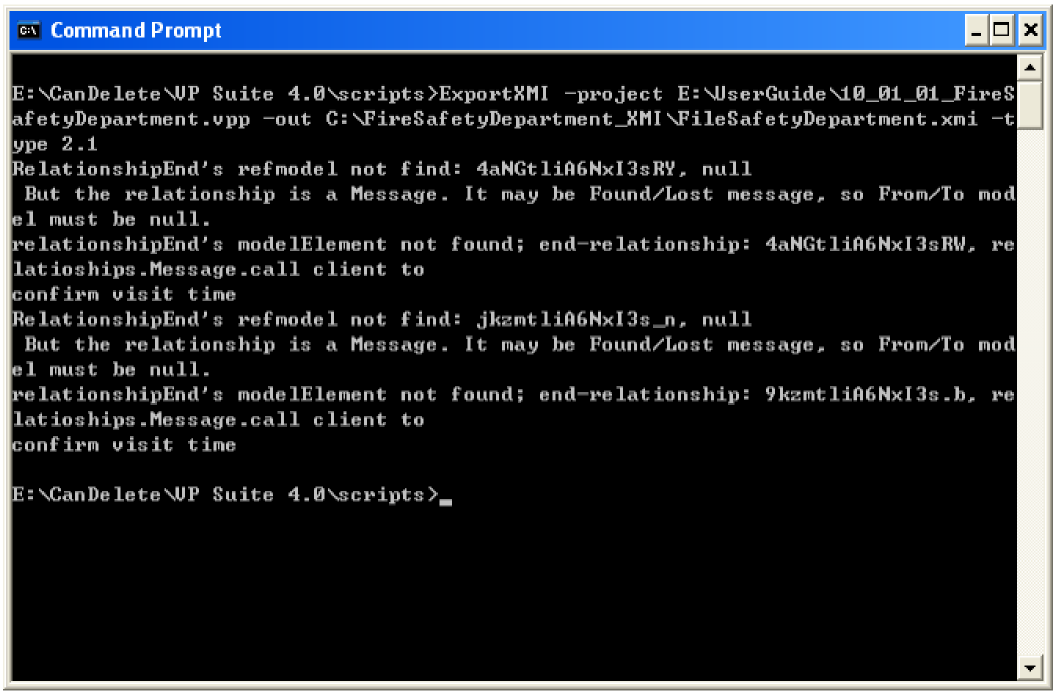
10. Example of exporting XMI with specified type.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXMI -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi -t
ype 2.1_
```

__Figure 4-16__ Inputting command

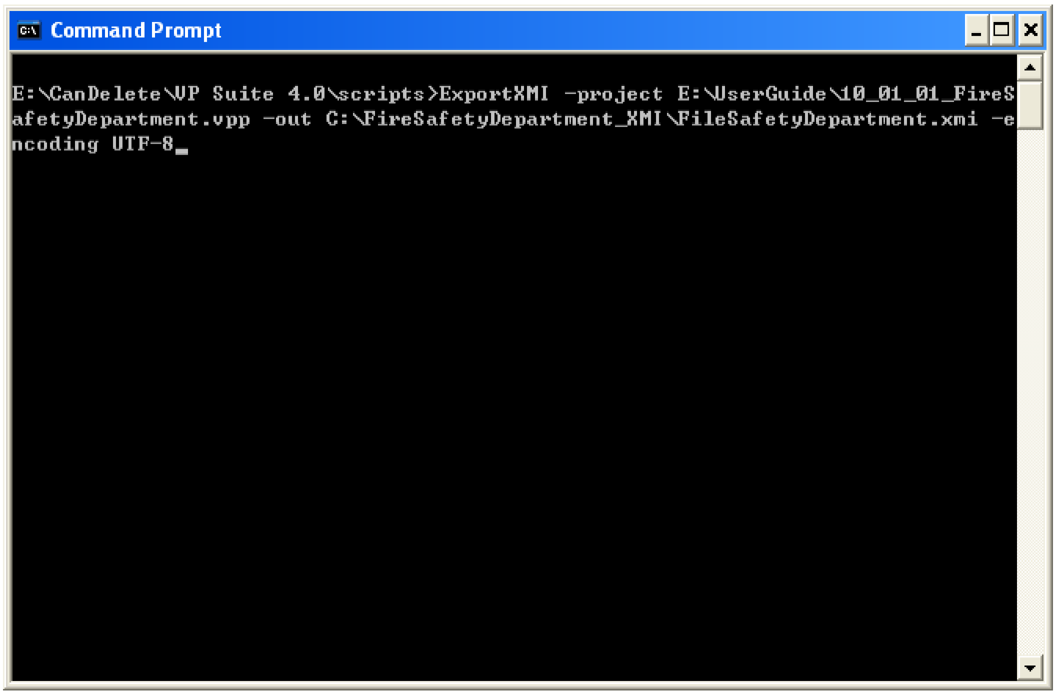
11. Execute result of exporting XMI.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXMI -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi -t
ype 2.1
RelationshipEnd's refmodel not find: 4aNGtliA6NxI3sRY, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 4aNGtliA6NxI3sRW, re
latioships.Message.call client to
confirm visit time
RelationshipEnd's refmodel not find: jkzmtliA6NxI3s_n, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 9kzmtliA6NxI3s.b, re
latioships.Message.call client to
confirm visit time

E:\CanDelete\VP Suite 4.0\scripts>_
```

__Figure 4-17__ Execute result

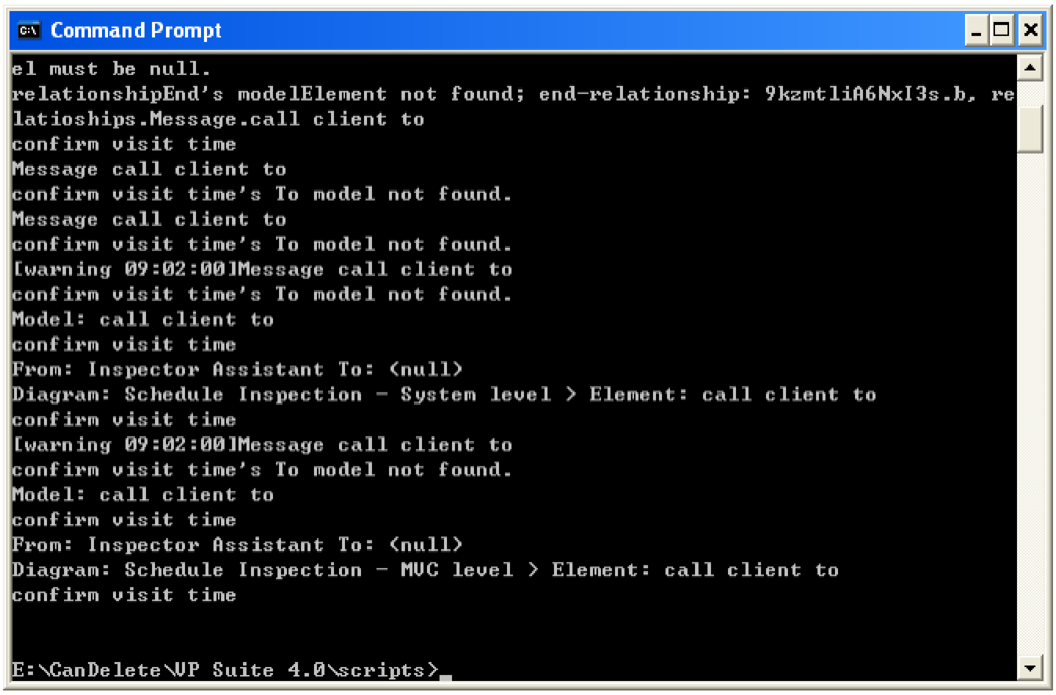
12. Example of exporting XMI with specified encoding.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXMI -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi -e
ncoding UTF-8_
```

Figure 4-18 Inputting command

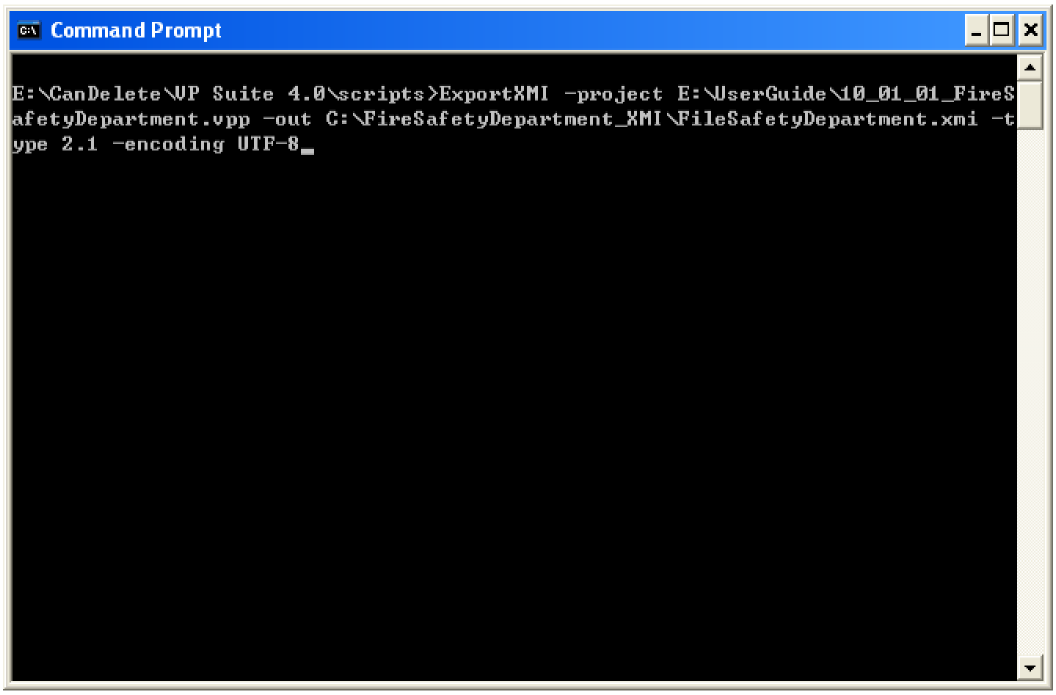
13. Execute result of exporting XMI.



```
el must be null.
relationshipEnd's modelElement not found; end-relationship: 9kzmtliA6NxI3s.b, re
latioships.Message.call client to
confirm visit time
Message call client to
confirm visit time's To model not found.
Message call client to
confirm visit time's To model not found.
[warning 09:02:00]Message call client to
confirm visit time's To model not found.
Model: call client to
confirm visit time
From: Inspector Assistant To: <null>
Diagram: Schedule Inspection - System level > Element: call client to
confirm visit time
[warning 09:02:00]Message call client to
confirm visit time's To model not found.
Model: call client to
confirm visit time
From: Inspector Assistant To: <null>
Diagram: Schedule Inspection - MVC level > Element: call client to
confirm visit time


E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 4-19 Execute result

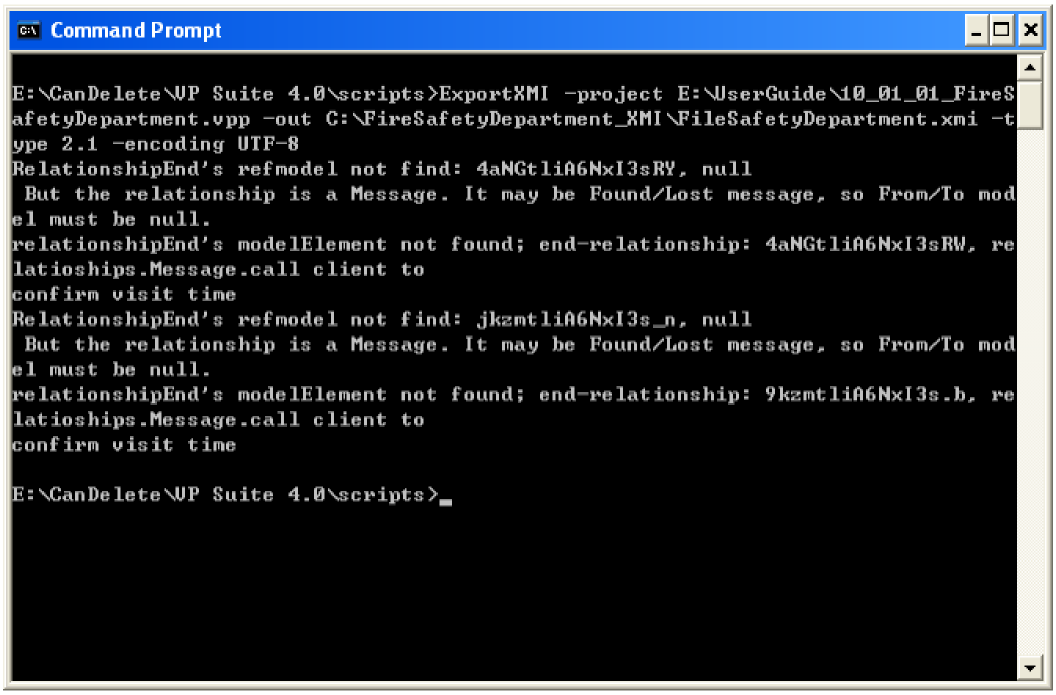
14. Example of exporting XMI with specified type and specified encoding.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXMI -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi -t
ype 2.1 -encoding UTF-8_
```

Figure 4-20 *Inputting command*

15. Execute result of exporting XMI.



```
E:\CanDelete\VP Suite 4.0\scripts>ExportXMI -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -out C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi -t
ype 2.1 -encoding UTF-8
RelationshipEnd's refmodel not find: 4aNGtliA6NxI3sRY, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 4aNGtliA6NxI3sRW, re
latioships.Message.call client to
confirm visit time
RelationshipEnd's refmodel not find: jkzmtliA6NxI3s_n, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 9kzmtliA6NxI3s.b, re
latioships.Message.call client to
confirm visit time

E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 4-21 *Execute result*

Importing XMI

Importing XMI

1. Press New Project button.

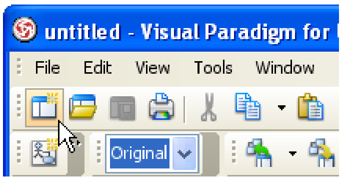
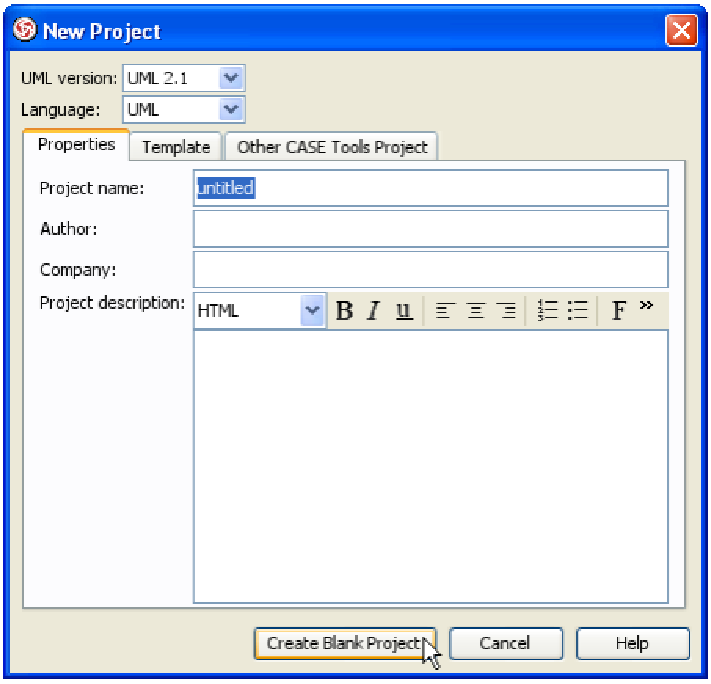


Figure 4-22 *New project from tool bar*

2. Press Create Blank Project button.



UML version: UML 2.1

Language: UML

Project name: untitled

Figure 4-23 *Create new project*

3. Mouse over main menu.

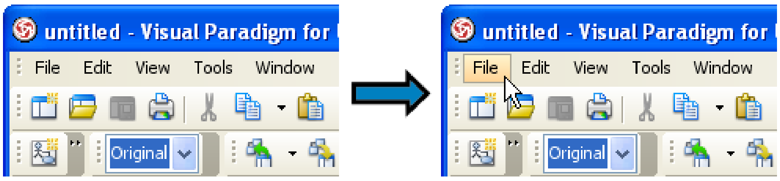


Figure 4-24 *Mouse over main menu*

4. Click on File of main menu.

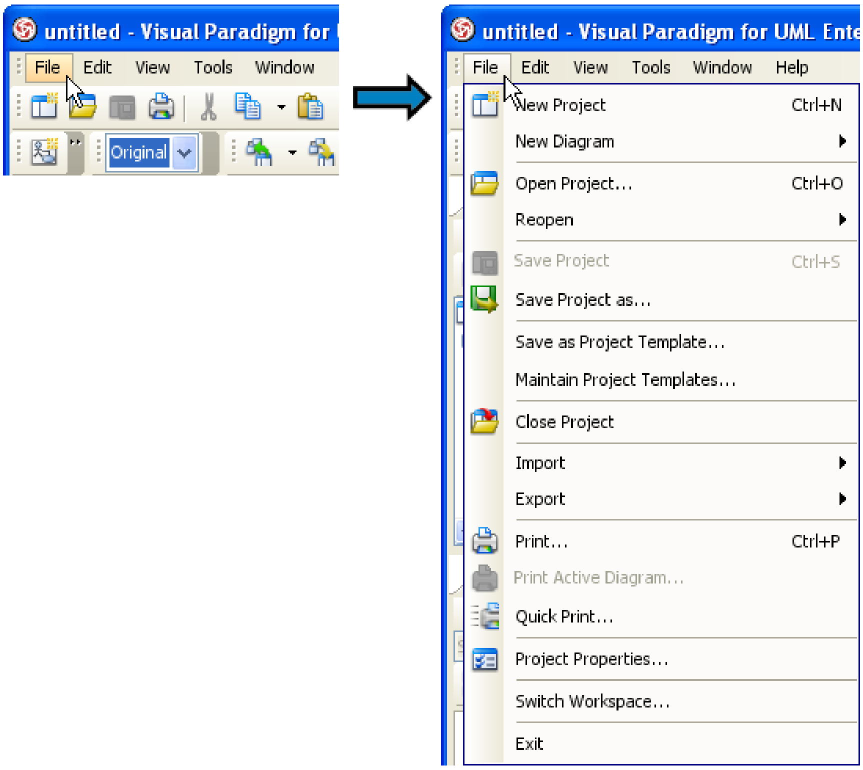


Figure 4-25 *Showing popup of main menu*

5. Click on Import > XMI... menu item.

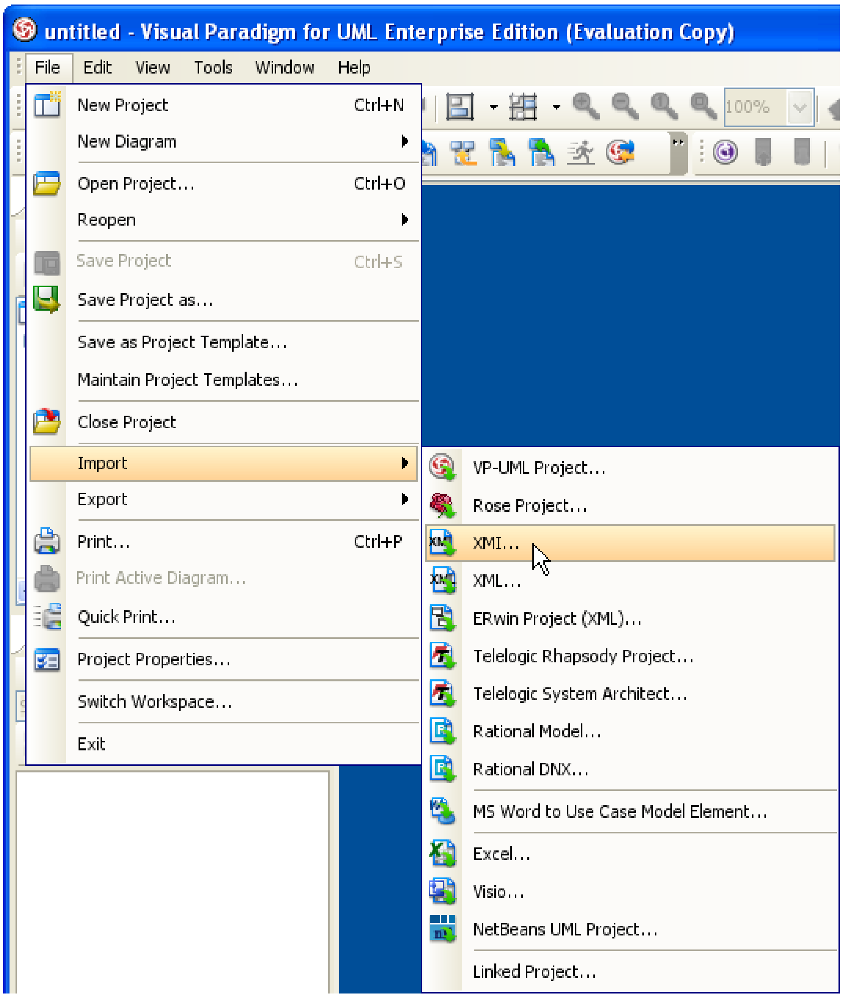
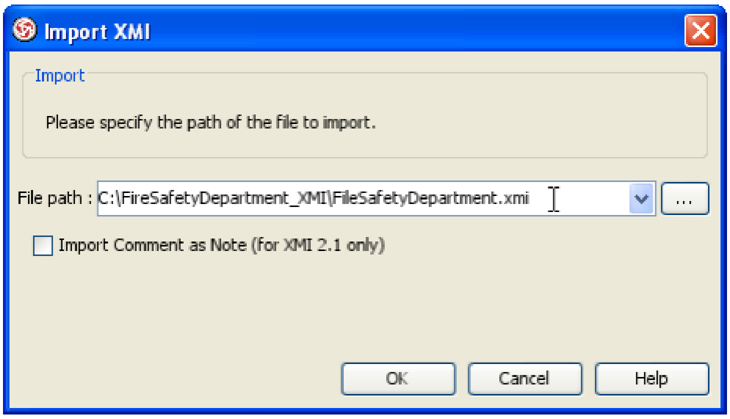


Figure 4-26 *Export XMI*

6. Input File path.


Import XMI

Import

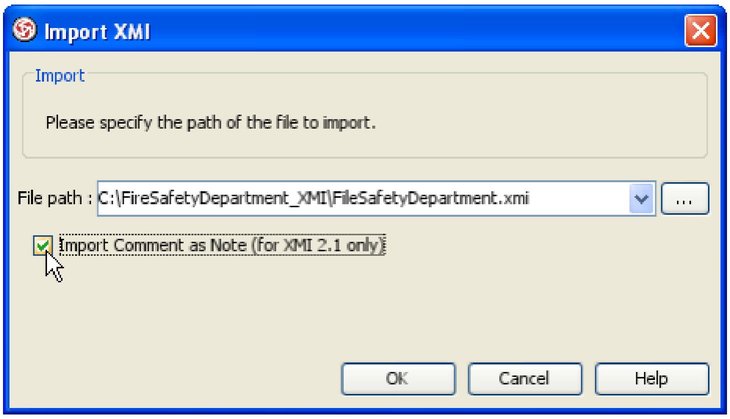
Please specify the path of the file to import.

File path : C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi

□ Import Comment as Note (for XMI 2.1 only)

Figure 4-27 *Inputting file path*

7. Check Import Comment as Note (for XMI 2.1 only) checkbox.



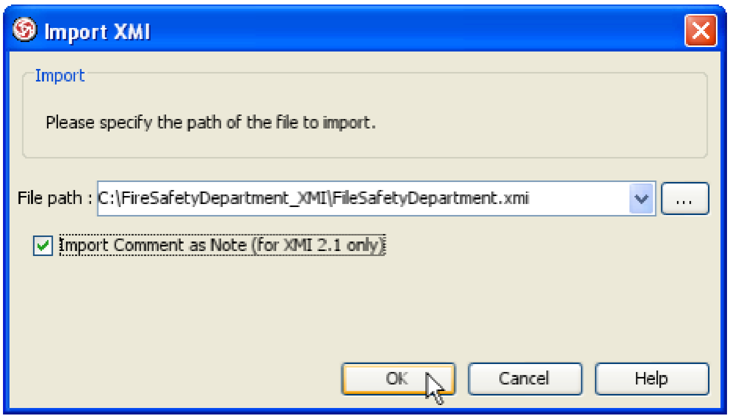
Please specify the path of the file to import.

File path : C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi

☑ Import Comment as Note (for XMI 2.1 only)

Figure 4-28 *Check Import Comment as Note (for XMI 2.1 only)*

8. Press OK button.



Please specify the path of the file to import.

File path : C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi

☑ Import Comment as Note (for XMI 2.1 only)

Figure 4-29 *Import from XMI*

Importing XML to project with command line interface

1. Mouse over Start button.

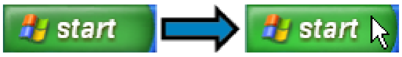


Figure 4-30 *Mouse over start button*

2. Press Start button.

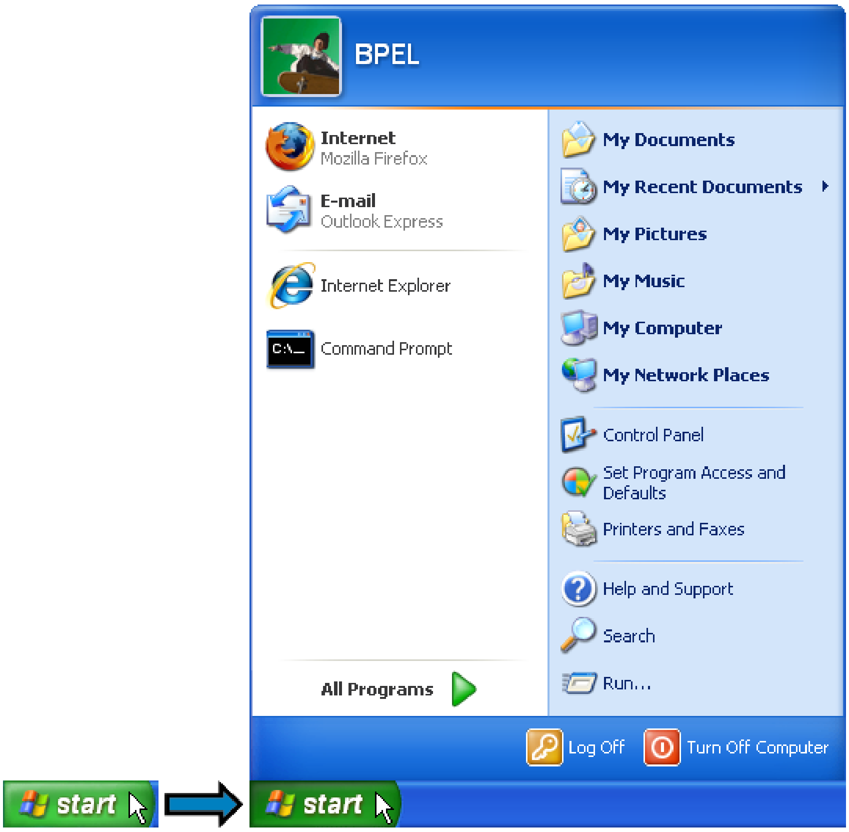


Figure 4-31 *Showing start menu*

3. Press Run... button.

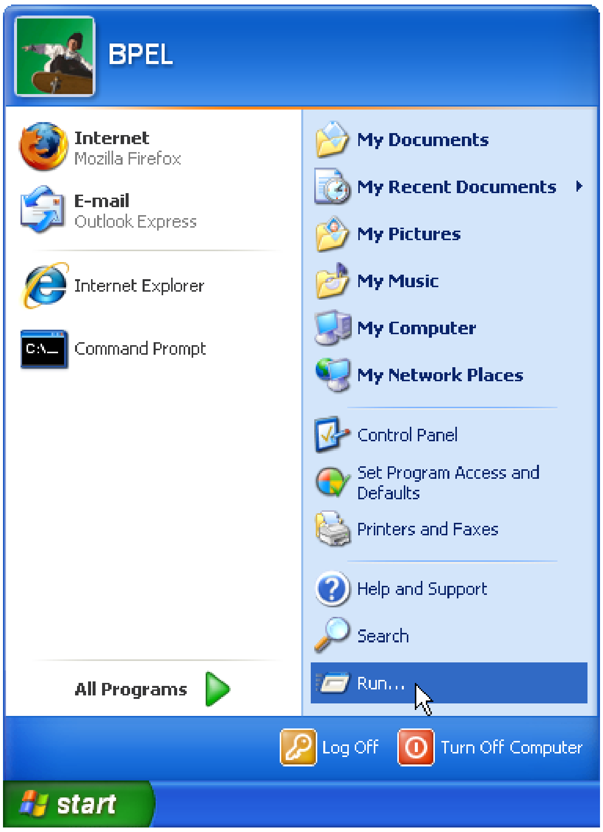
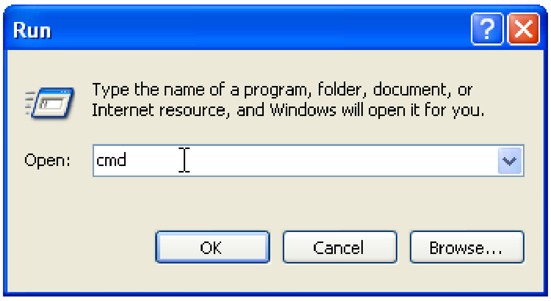


Figure 4-32 *Run from start menu*

4. Input cmd as command.

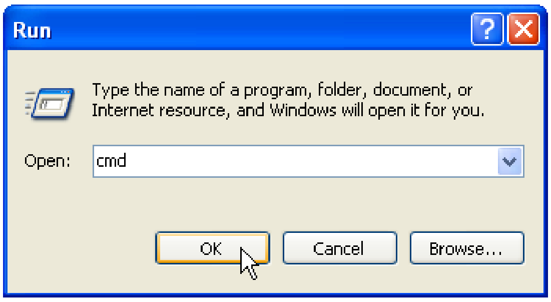

Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Figure 4-33 Inputting command to run

5. Press OK button.

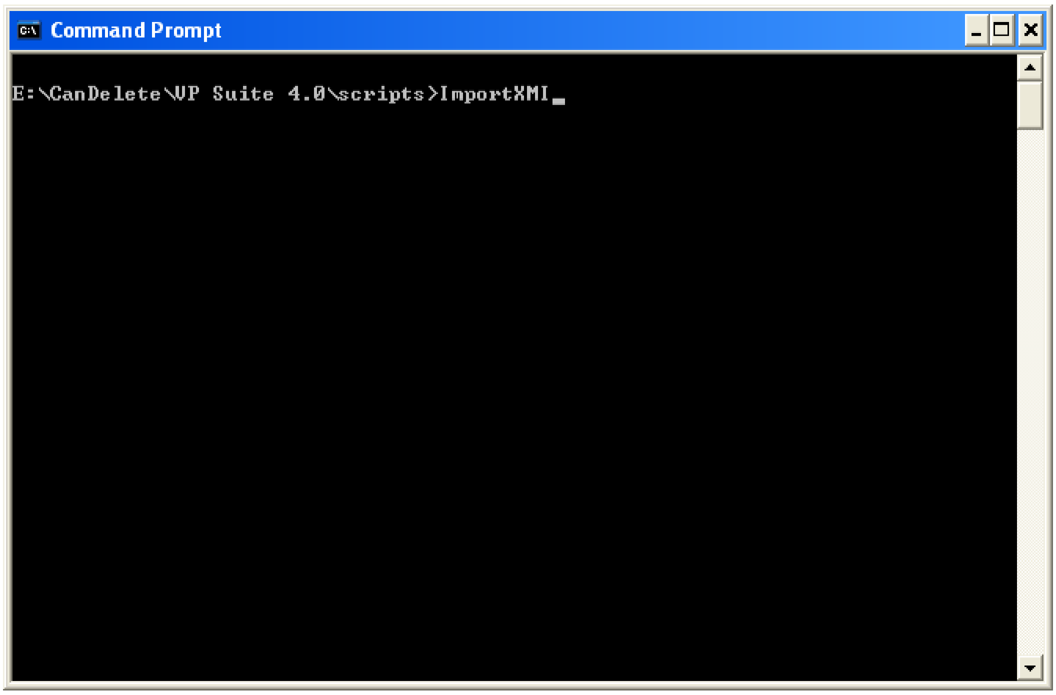

Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Figure 4-34 Run command

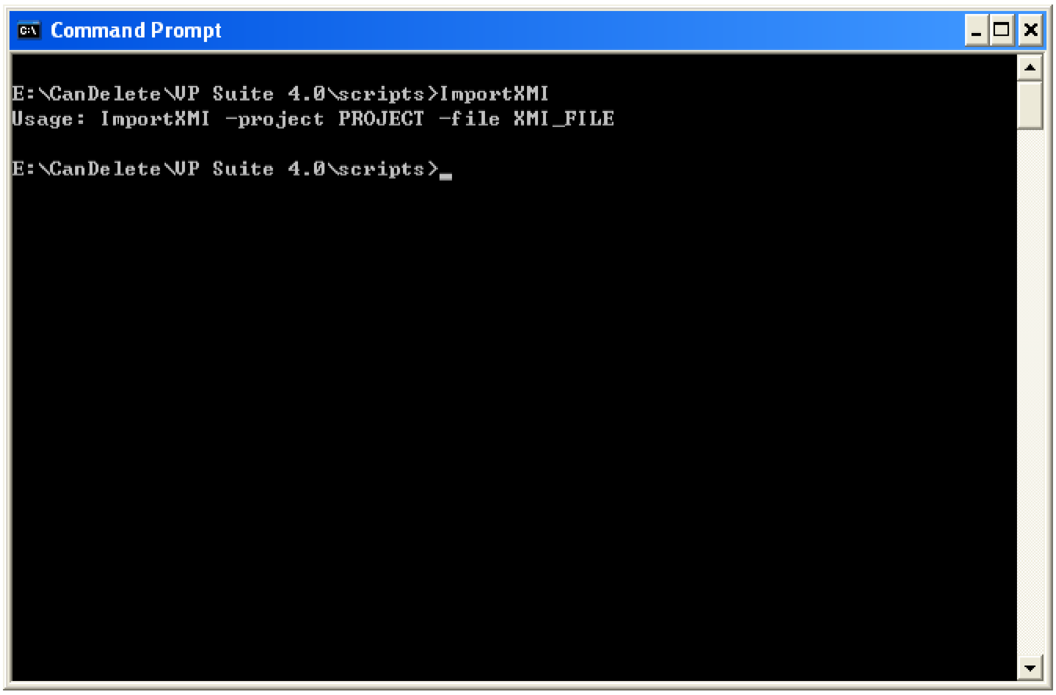
6. Change directory to scripts folder and execute ImportXMI.


Command Prompt

```
E:\CanDelete\VP Suite 4.0\scripts>ImportXMI_
```

Figure 4-35 Inputting command

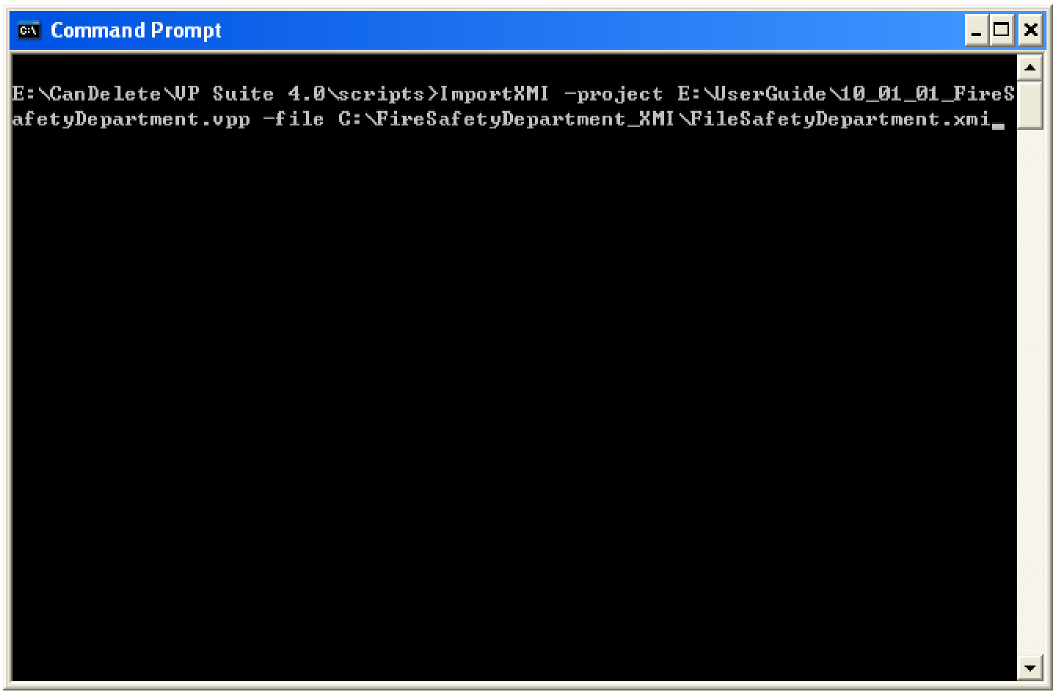
7. Usage of ImportXMI will be shown.



```
E:\CanDelete\VP Suite 4.0\scripts>ImportXMI
Usage: ImportXMI -project PROJECT -file XMI_FILE

E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 4-36 *Execute result of command*

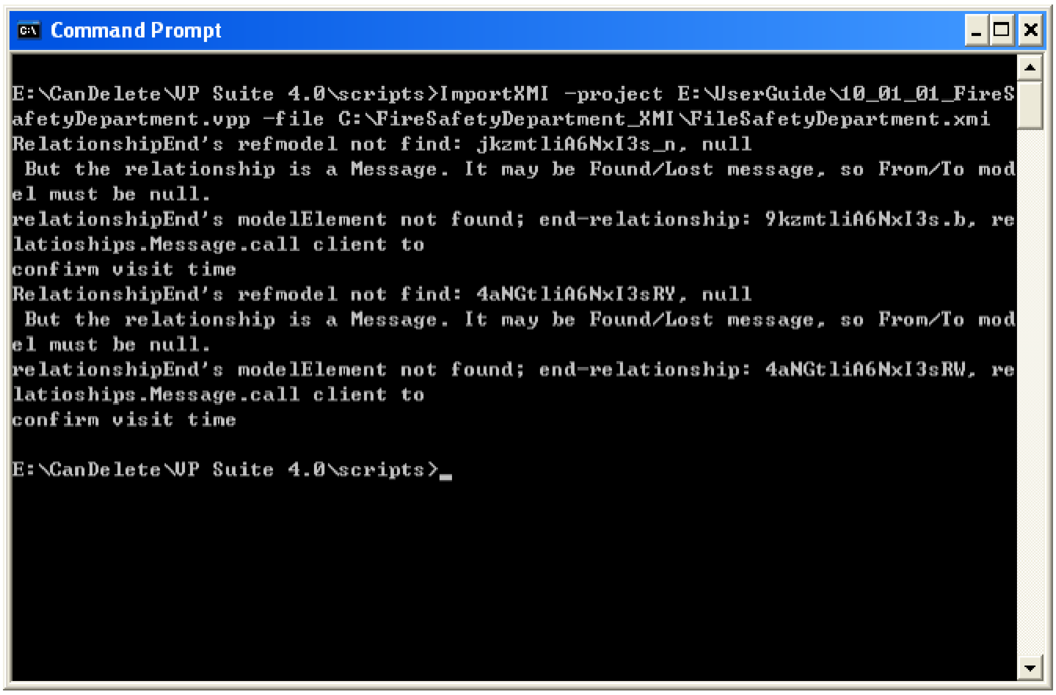
8. Example of importing XMI.



```
E:\CanDelete\VP Suite 4.0\scripts>ImportXMI -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -file C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi_
```

Figure 4-37 *Inputting command*

9. Execute result of importing XMI.



```
E:\CanDelete\VP Suite 4.0\scripts>ImportXMI -project E:\UserGuide\10_01_01_FireS
afetyDepartment.vpp -file C:\FireSafetyDepartment_XMI\FileSafetyDepartment.xmi
RelationshipEnd's refmodel not find: jkzmtliA6NxI3s_n, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 9kzmtliA6NxI3s.b, re
latioships.Message.call client to
confirm visit time
RelationshipEnd's refmodel not find: 4aNGtliA6NxI3sRY, null
 But the relationship is a Message. It may be Found/Lost message, so From/To mod
el must be null.
relationshipEnd's modelElement not found; end-relationship: 4aNGtliA6NxI3sRW, re
latioships.Message.call client to
confirm visit time

E:\CanDelete\VP Suite 4.0\scripts>_
```

Figure 4-38 *Execute result*

Importing Visio Drawing into VP-UML

1. Here is a Visio drawing:

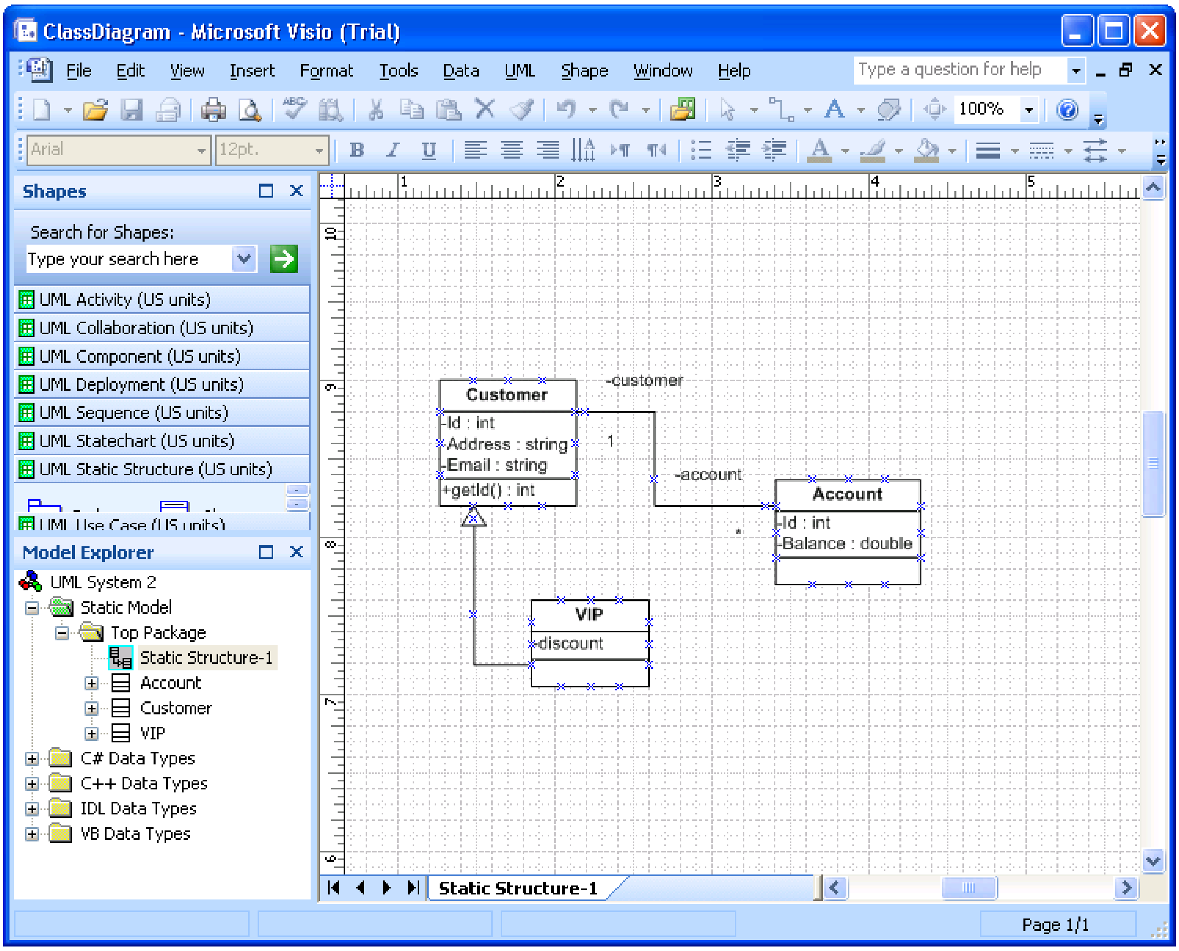


Figure 5-1 A Visio drawing

To import a Visio drawing into VP-UML, select **File > Import > Visio...** in the main menu of VP-UML.

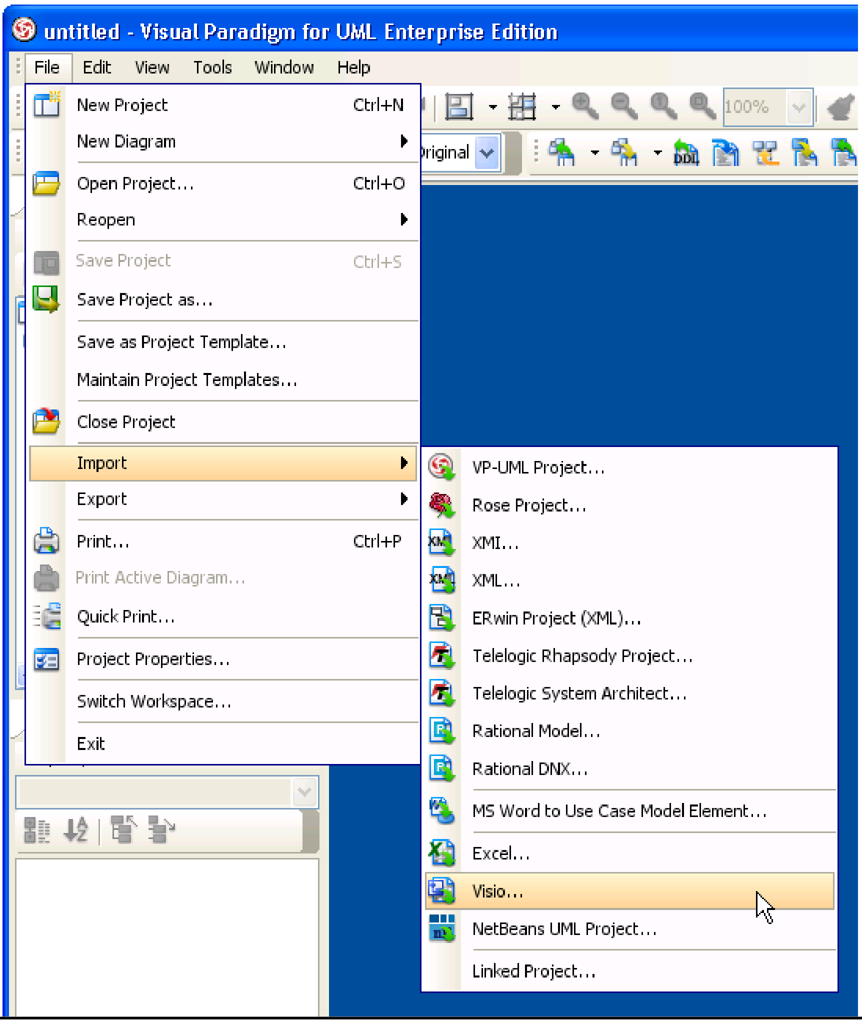
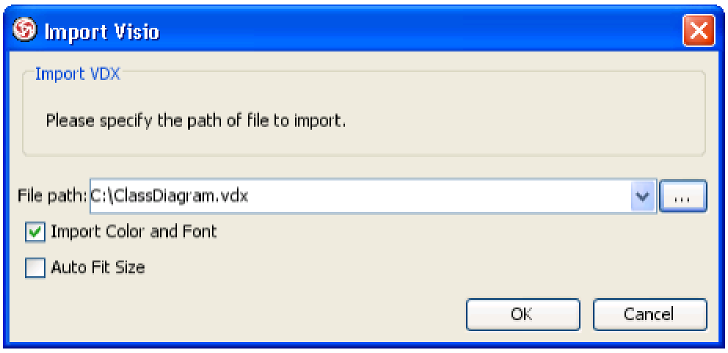


Figure 5-2 The import Visio drawing menu

2. Specify the file path of the Visio drawing.



Please specify the path of file to import.

File path: C:\ClassDiagram.vdx

☑ Import Color and Font

□ Auto Fit Size

Figure 5-3 *Specifying Visio drawing path*

Here are the configurable options:

Option	Description
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.
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.

Table 5-1 *Description of available import options*

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.

3. 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 VP-UML.

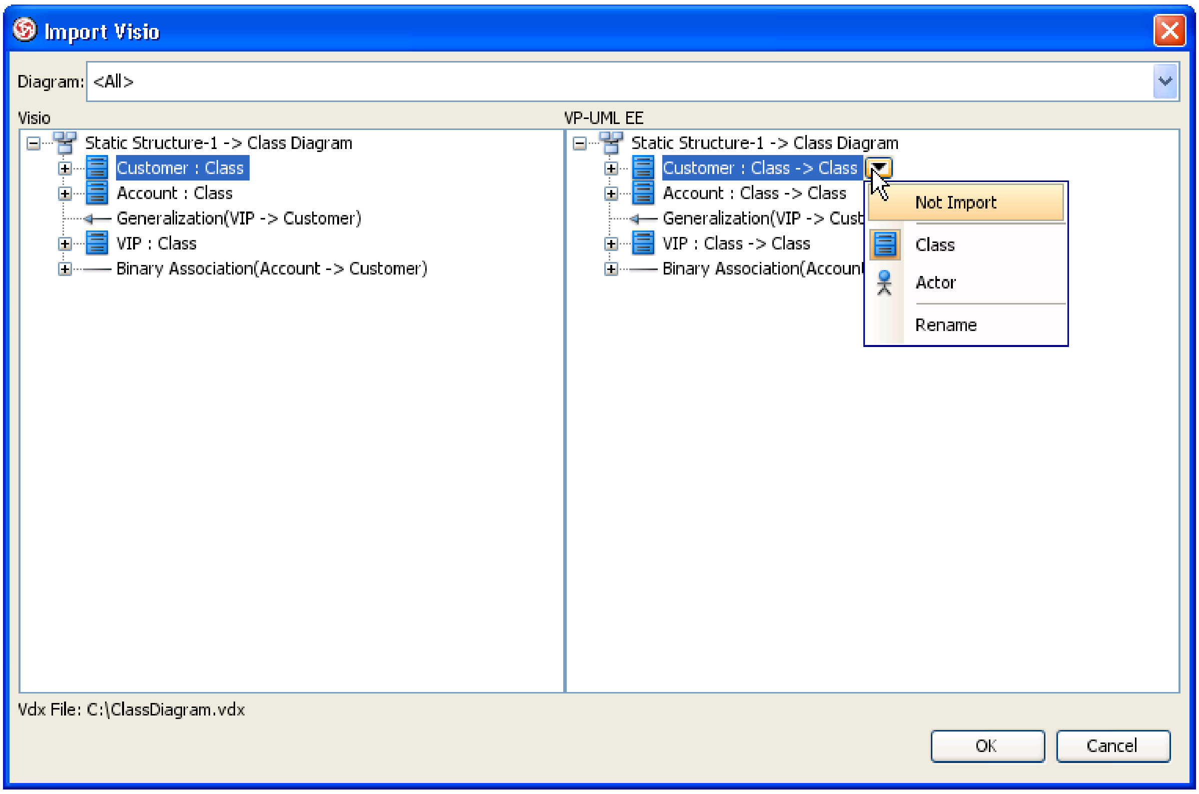


Diagram: <All>

Vdx File: C:\ClassDiagram.vdx

Import Visio
Visio
Static Structure-1 -> Class Diagram
Customer : Class
Account : Class
Generalization(VIP -> Customer)
VIP : Class
Binary Association(Account -> Customer)
VP-UML EE
Static Structure-1 -> Class Diagram
Customer : Class -> Class
Account : Class -> Class
Generalization(VIP -> Cust
VIP : Class -> Class
Binary Association(Accoun
Not Import
Class
Actor
Rename
OK
Cancel

__Figure 5-4__ 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 VP-UML 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.

__Table 5-2__ Description of available import options on individual shape

4. Click **OK** when the import is configured. The drawings can then be accessible in the **Diagram Navigator**.

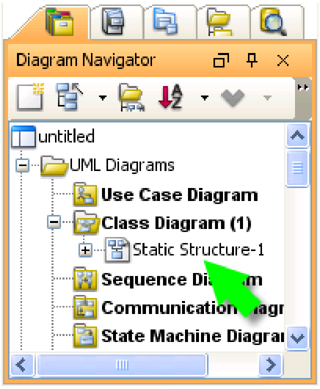
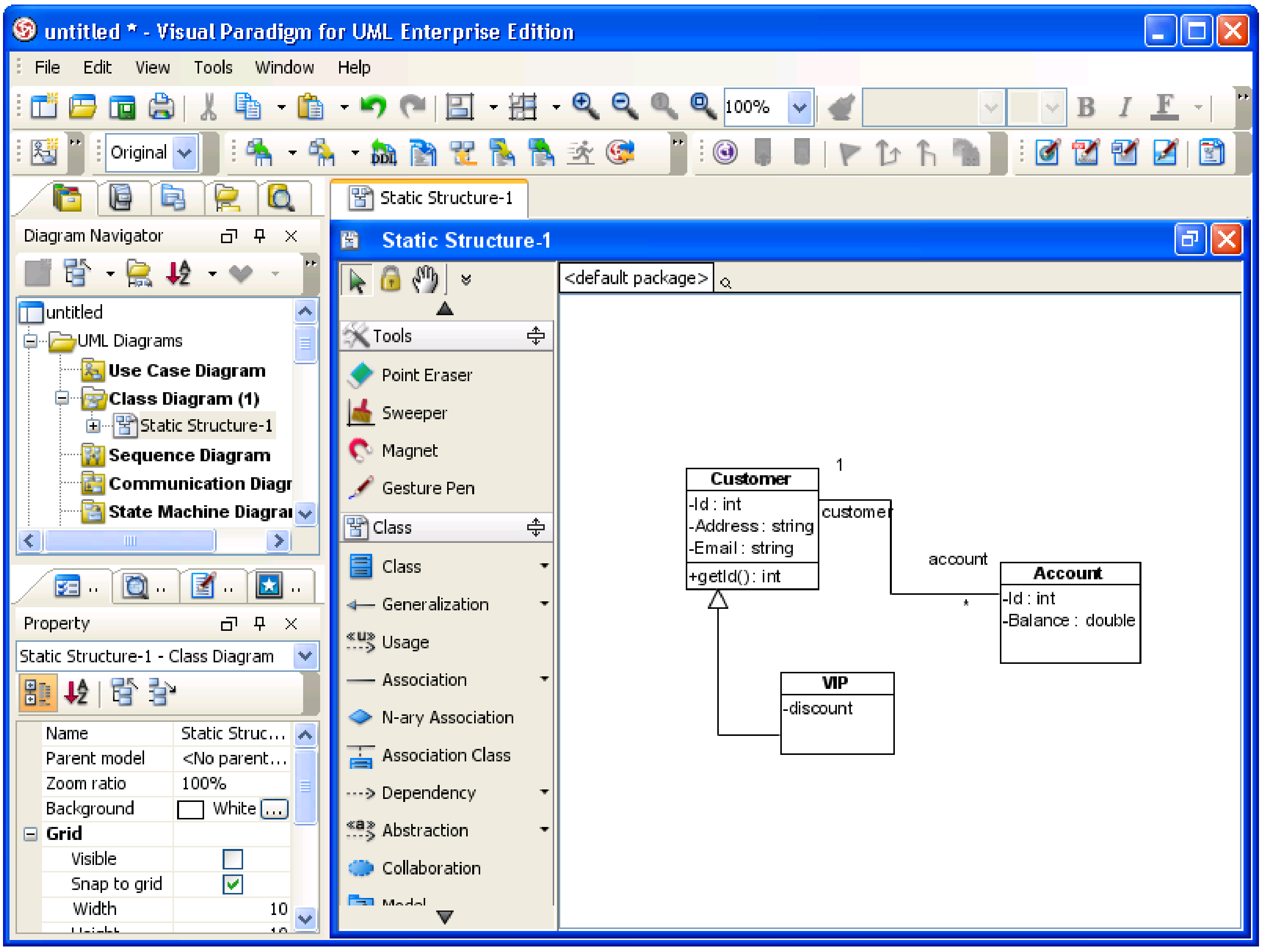


Figure 5-5 *Diagram Navigator listing the imported diagram(s)*

You can then double click on the diagram node to open the diagram.



Name: Static Struc...
Parent model: <No parent...
Zoom ratio: 100%
Background: White
Grid
Visible: □
Snap to grid: ☑
Width: 10

Figure 5-6 *A Class Diagram imported from Visio drawing*

Importing Rational Rose Model into VP-UML

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 VP-UML, with all the model data as well as formatting retained.

1. Here is a Rose Model:

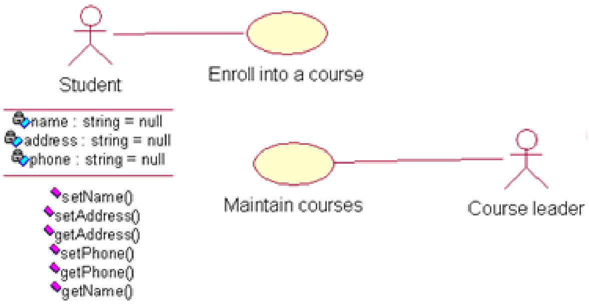


***Figure 6-1** A Rose drawing*

To import a Rose model into VP-UML, select **File > Import > Rose Project...** in the main menu of VP-UML.

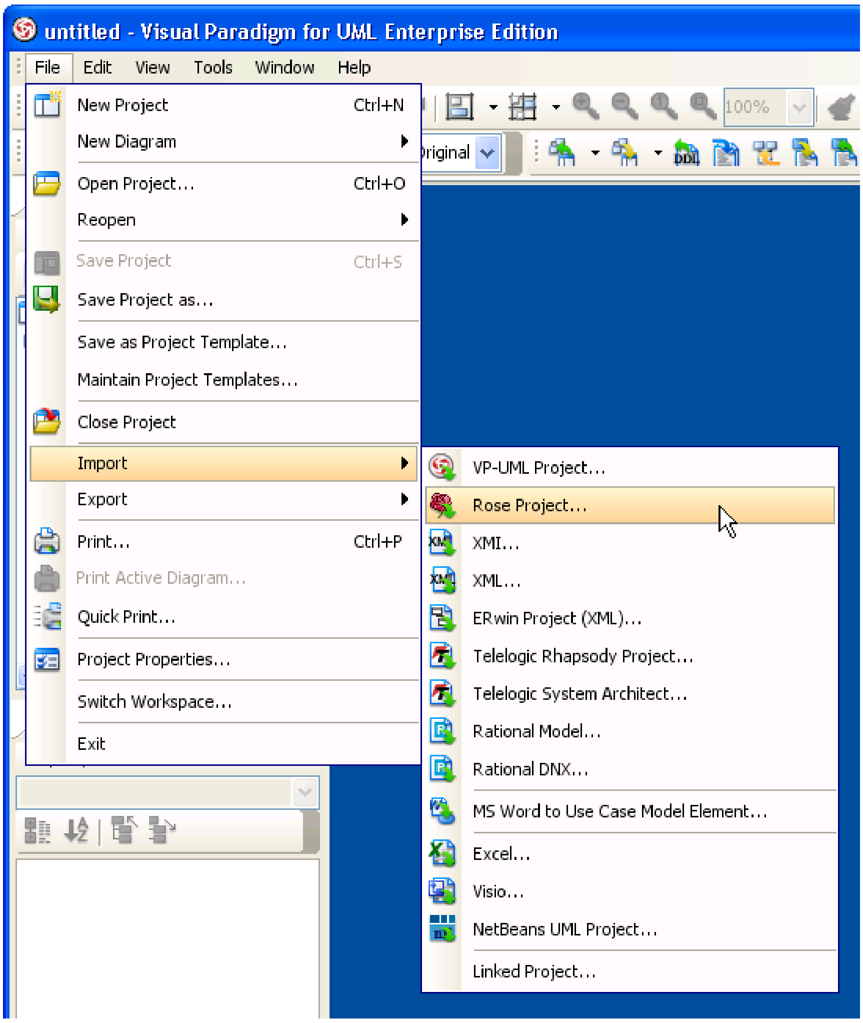
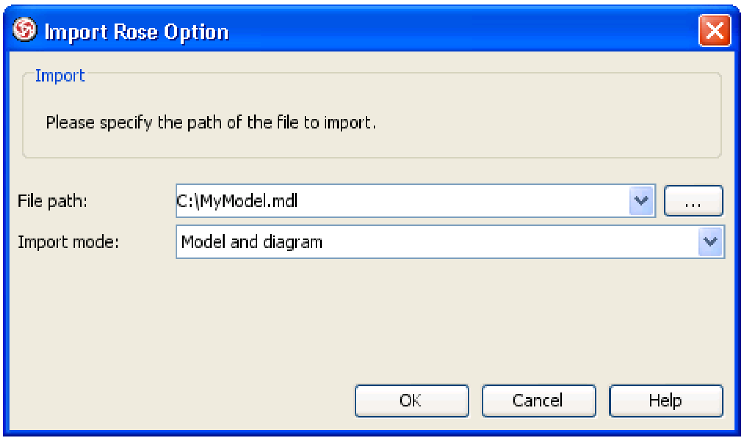


***Figure 6-2** The import Rose Project menu*

2. Specify the file path of the Rose model.



Import Rose Option

Import

Please specify the path of the file to import.

File path: C:\MyModel.mdl

Import mode: Model and diagram

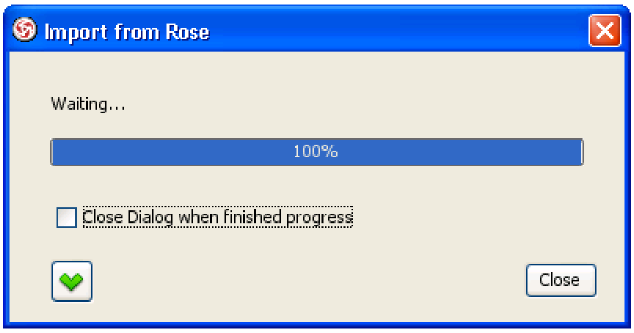
Figure 6-3 *Specifying Rose model path*

Furthermore, you can select the import mode. Below is a description of the modes.

Option	Description
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.

Table 6-1 *Description of available import modes*

3. Click **OK** to start importing. This popup a dialog box indicating the progress of importing.



Import from Rose

Waiting...

100%

□ Close Dialog when finished progress

Figure 6-4 *The dialog box indicating the progress of importing*

4. 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**.

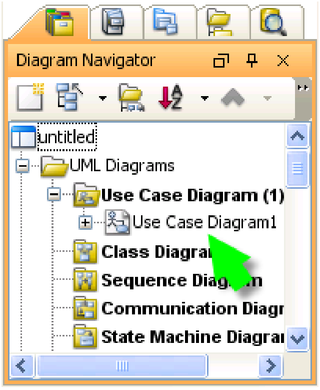


Figure 6-5 *Diagram Navigator listing the imported diagram(s)*

Model element can be accessed in the **Model Explorer**. User can form diagrams with them by dragging and dropping them onto diagrams.

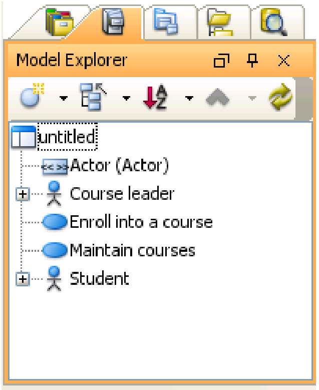


Figure 6-6 *Model Explorer listing the imported model element(s)*

You can then double click on the diagram node to open the diagram.

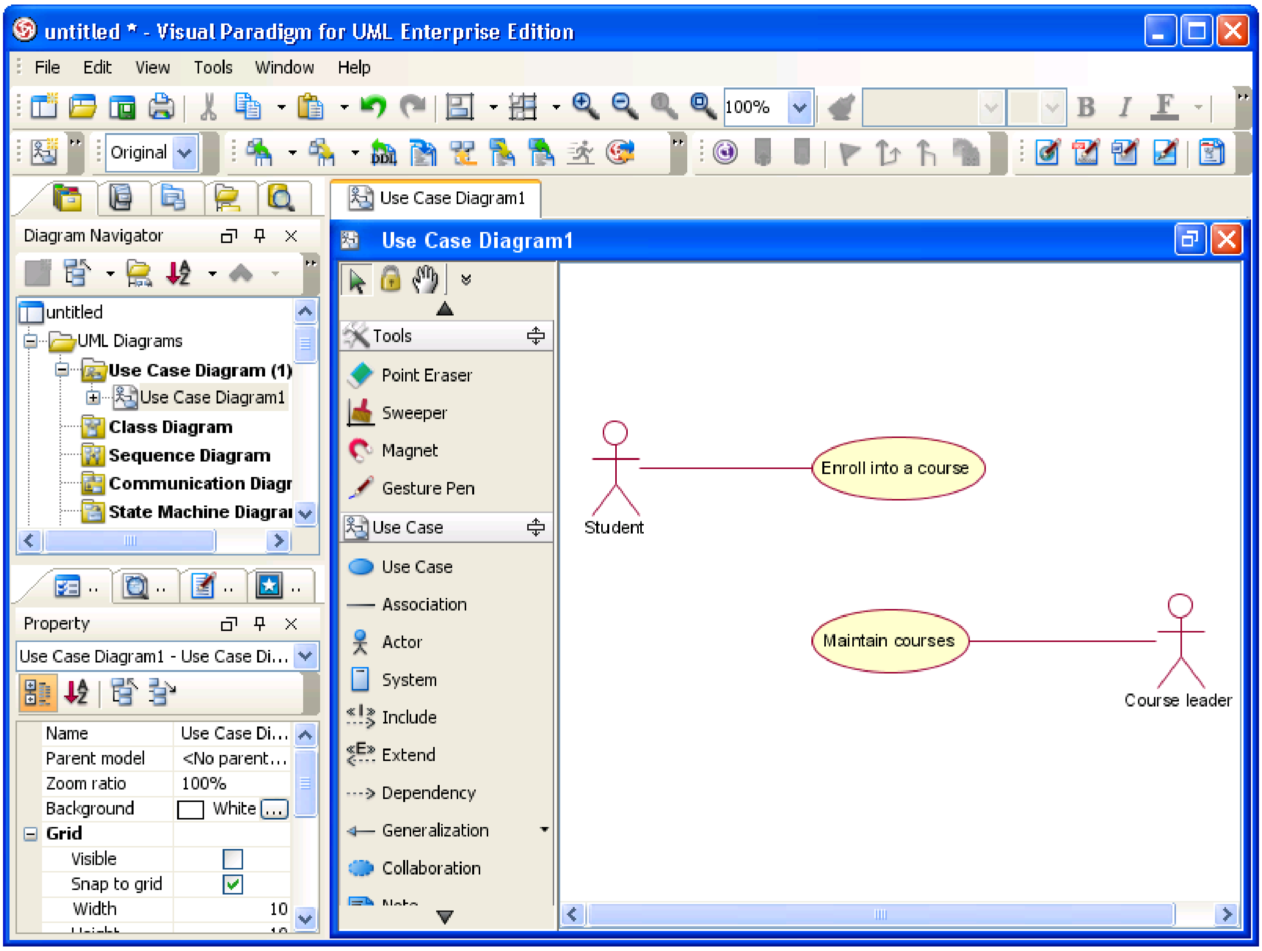


Figure 6-7 *A Use Case Diagram imported from Rose model*

Importing Rational Software Architect EMX into VP-UML

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 VP-UML 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. Here is a Rational model:

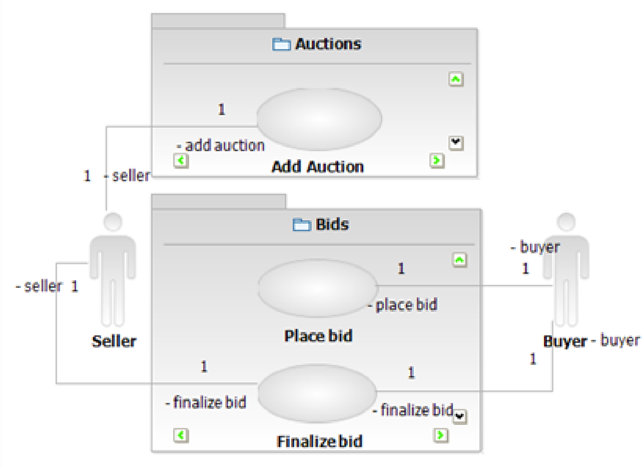


Figure 7-1 *A Rational drawing*

To import a Rational model into VP-UML, select **File > Import > Rational Model...** in the main menu of VP-UML.

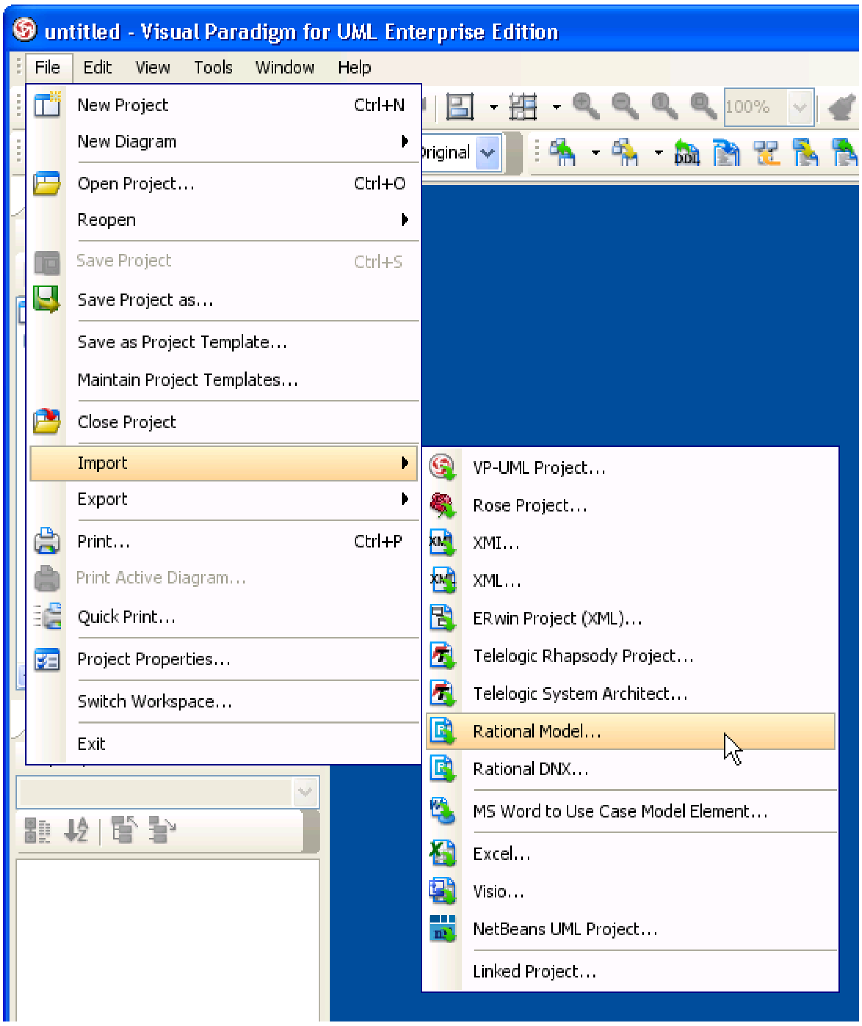
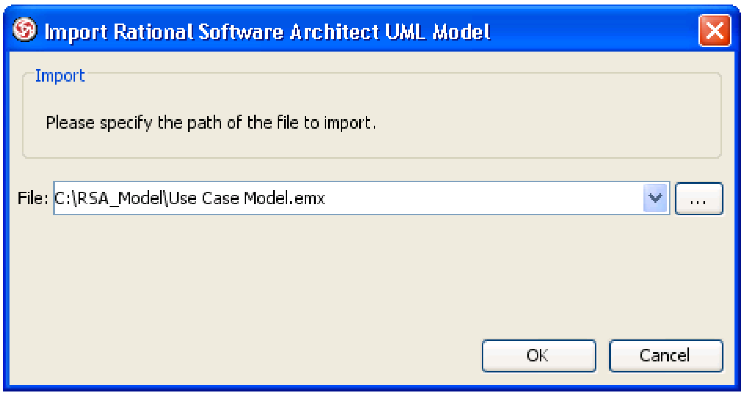


Figure 7-2 *The import Rational model menu*

2. Specify the file path of the Rational model.



Import Rational Software Architect UML Model

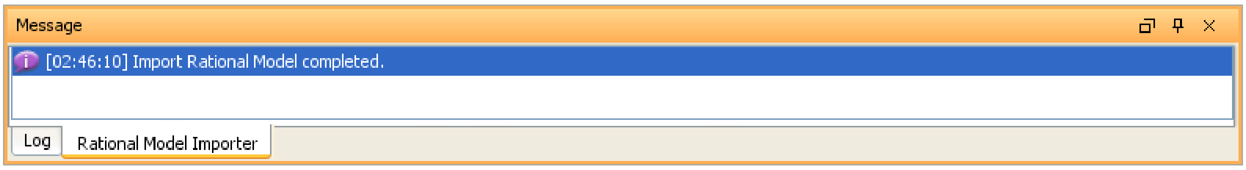
Import

Please specify the path of the file to import.

File: C:\RSA_Model\Use Case Model.emx

Figure 7-3 *Specifying Rational model path*

3. Click **OK** to start importing. When import is completed, the message pane will popup, with a notification appear in it.



[02:46:10] Import Rational Model completed.

Figure 7-4 *The dialog box indicating the progress of importing*

The drawings can then be accessible in the **Diagram Navigator**.

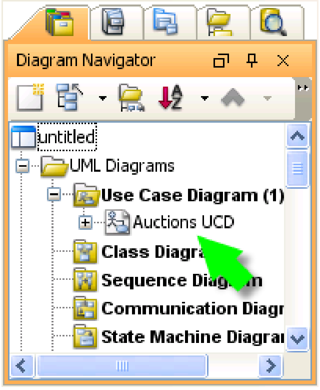
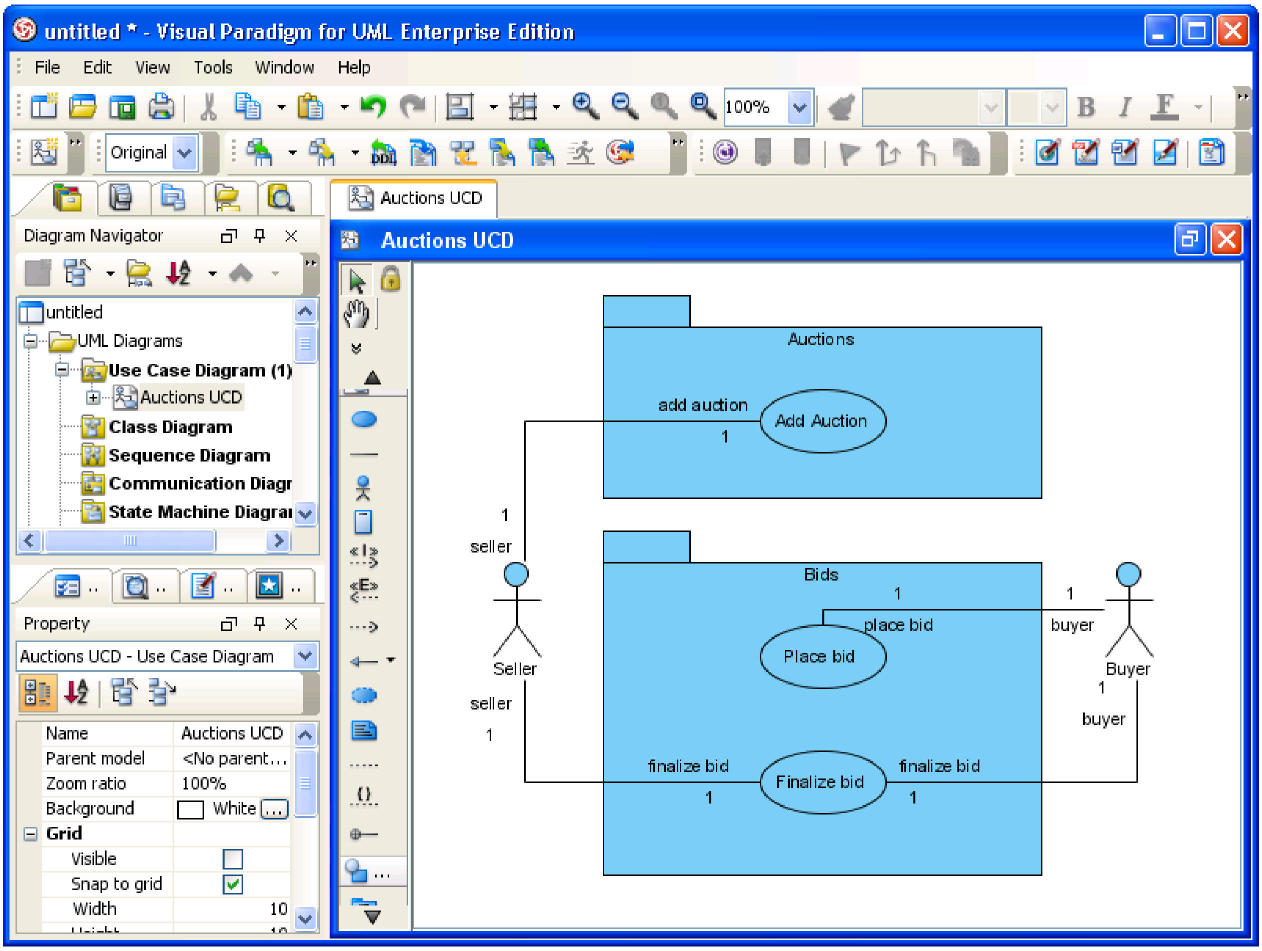


Figure 7-5 *Diagram Navigator listing the imported diagram(s)*

You can then double click on the diagram node to open the diagram.



Auctions UCD - Use Case Diagram

Name	Auctions UCD
Parent model	<No parent…
Zoom ratio	100%
Background	White
Grid	
Visible	□
Snap to grid	☑
Width	10

Figure 7-6 *A Use Case Diagram imported from Rational model*

Importing Rational Software Architect DNX into VP-UML

VP-UML supports importing drawing drew in Rational Software Architect with a .dnx extension. By importing a drawing, all diagrams, shapes and model information will be imported.

1. Here is a drawing in Rational Software Architect:

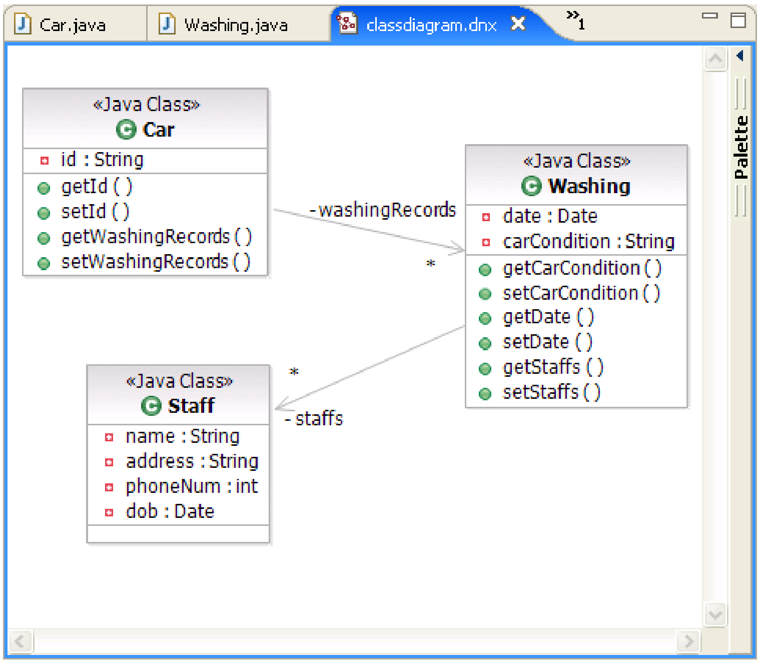


Figure 7-7 *A Rational drawing*

To import the drawing into VP-UML, select **File > Import > Rational DNX...** in the main menu of VP-UML.

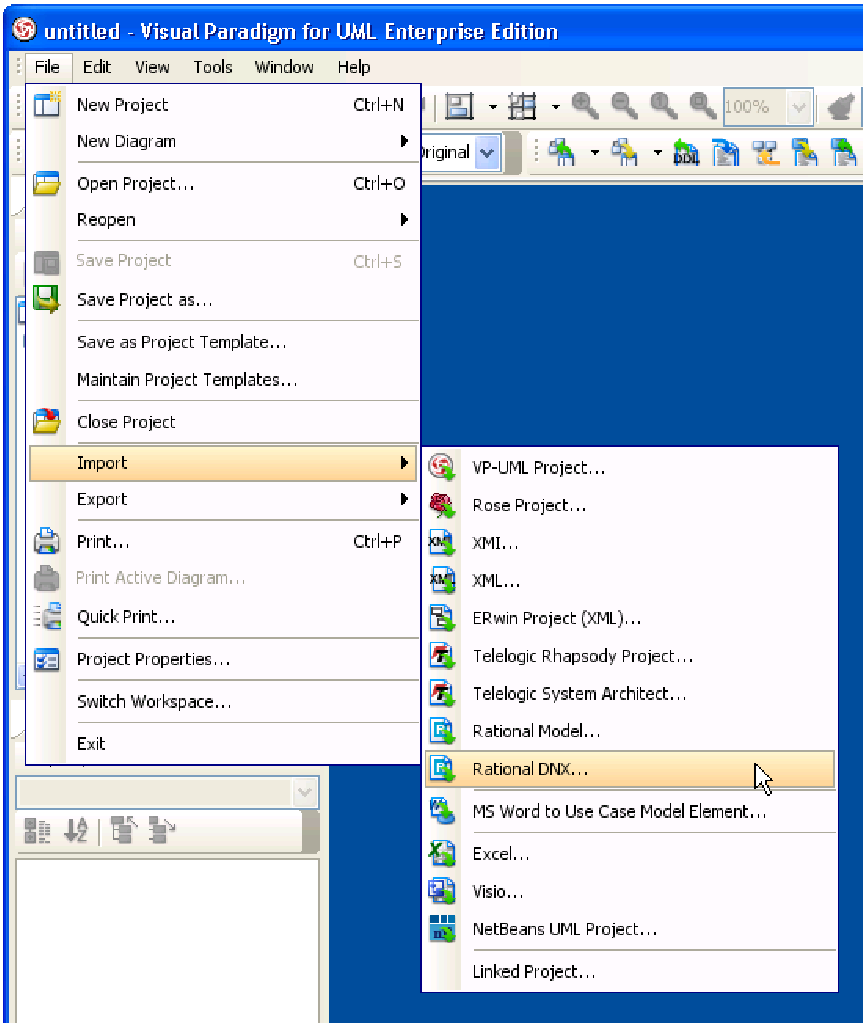
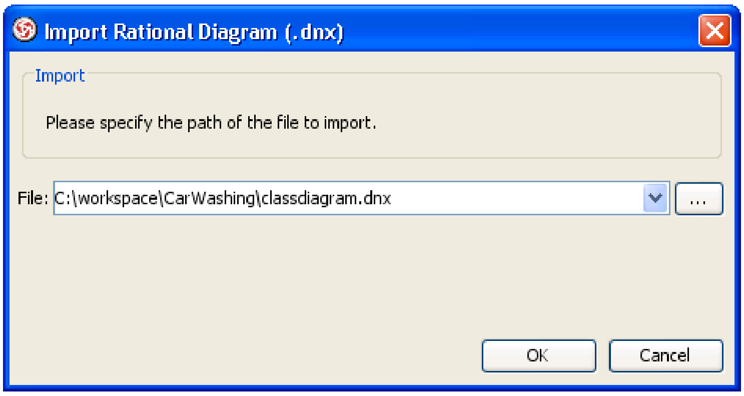


Figure 7-8 *The import Rational DNX menu*

2. Specify the file path of the .dnx file.



Import Rational Diagram (.dnx)

Import

Please specify the path of the file to import.

File: C:\workspace\CarWashing\classdiagram.dnx

Figure 7-9 *Specifying .dnx file path*

3. Click **OK** to start importing. When import is completed, the drawings can then be accessible in the **Diagram Navigator**.

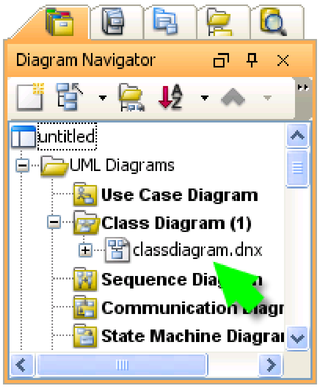


Figure 7-10 *Diagram Navigator listing the imported diagram(s)*

You can then double click on the diagram node to open the diagram.

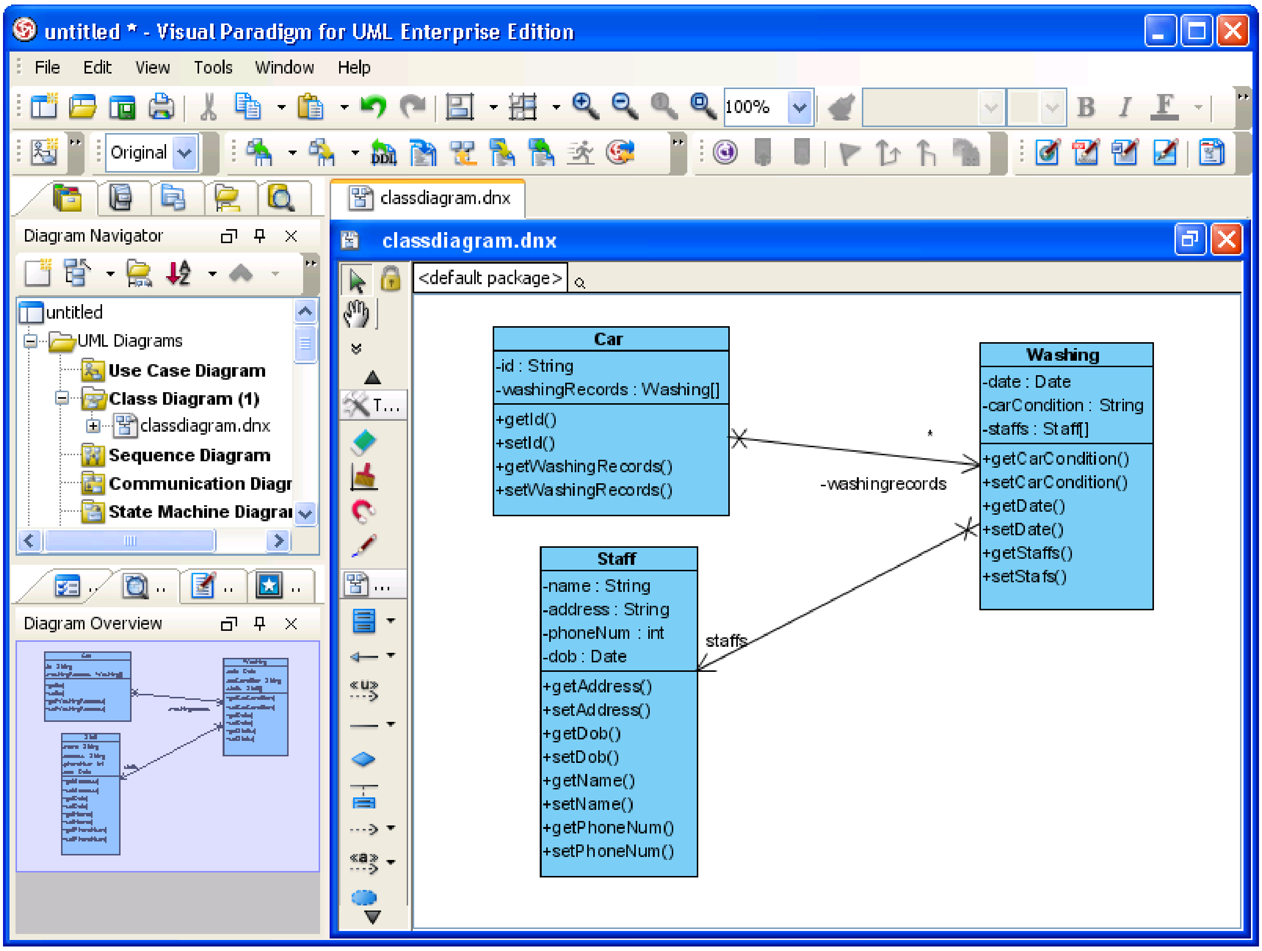


Figure 7-11 *A Class Diagram imported from Rational .dnx file*

Importing ERwin Data Modeler Project into VP-UML

AllFusion ERwin Data Modeler is a popular tool for data modeling. You can import ERwin diagrams and entity models into VP-UML with all properties preserved.

1. Here is a ERwin Data Modeler Project. In order to let VP-UML import it, you need to save it as an XML file. Select **File > Save As...** from the menu.

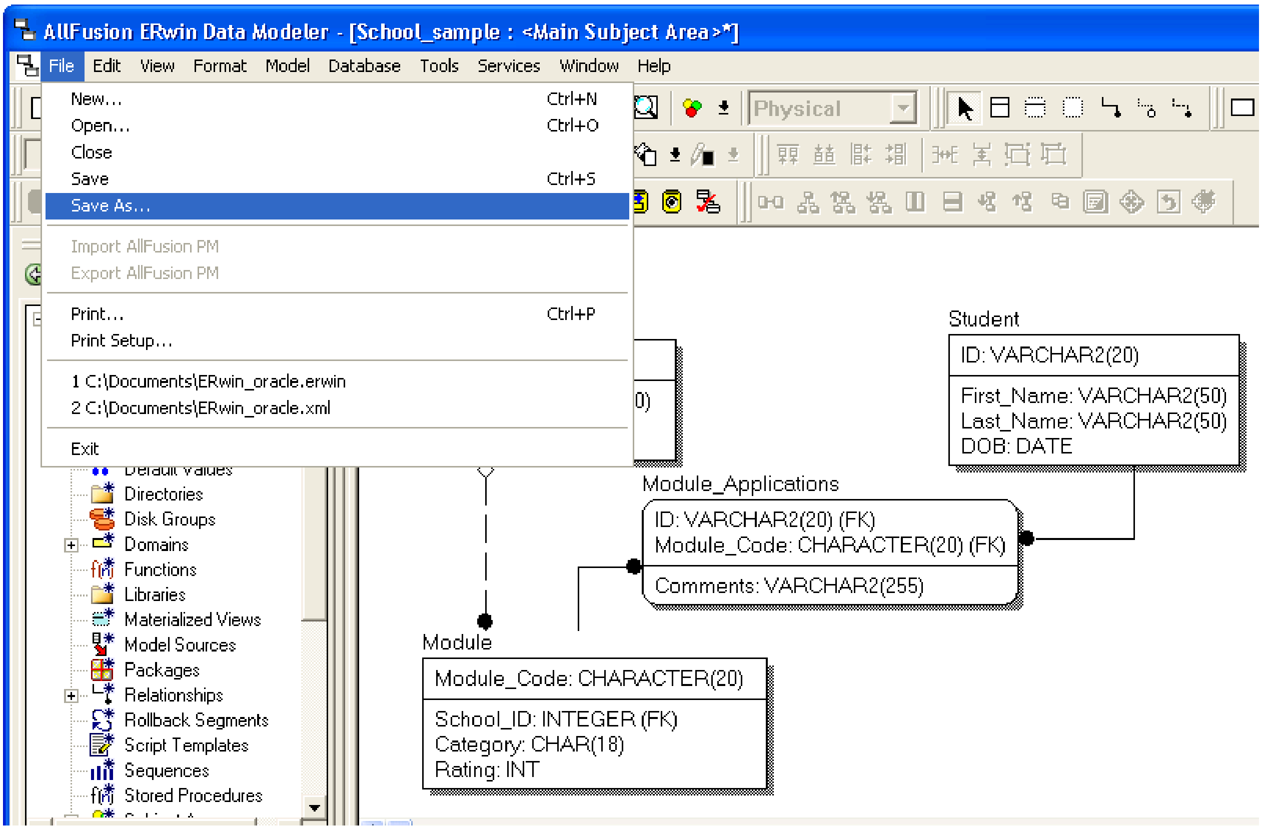
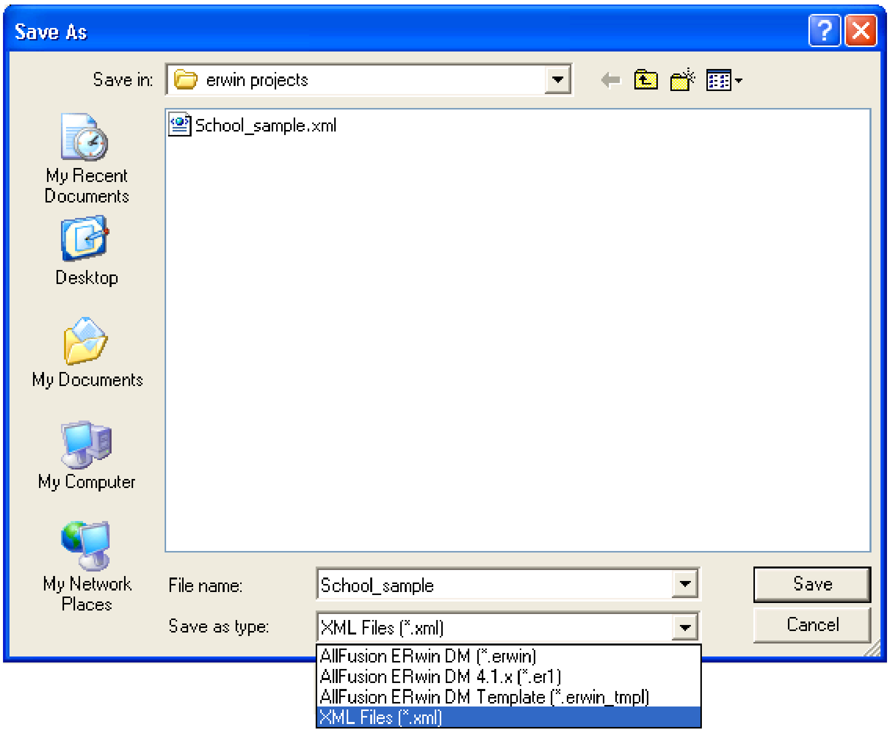


Figure 8-1 To save an ERwin Data Modeler project as XML

2. Select **XML Files (*.xml)** in **Save as type** and enter the file name in the **Save As** dialog box.



Save in: erwin projects

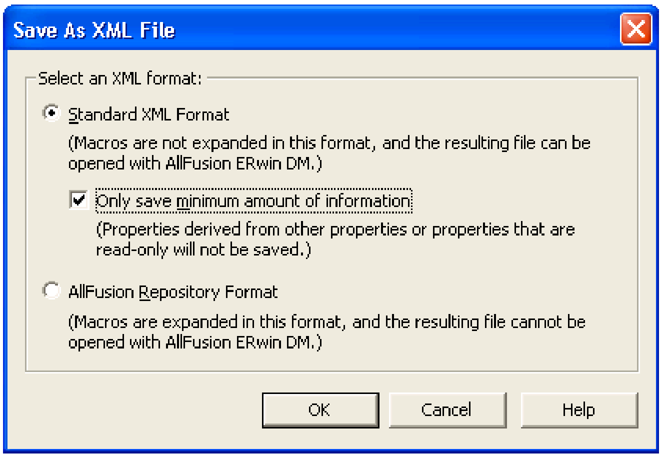
File name: School_sample

Save as type: XML Files (*.xml)

Figure 8-2 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**.



Save As XML File

Select an XML format:

● Standard XML Format

(Macros are not expanded in this format, and the resulting file can be opened with AllFusion ERwin DM.)

☑ Only save minimum amount of information

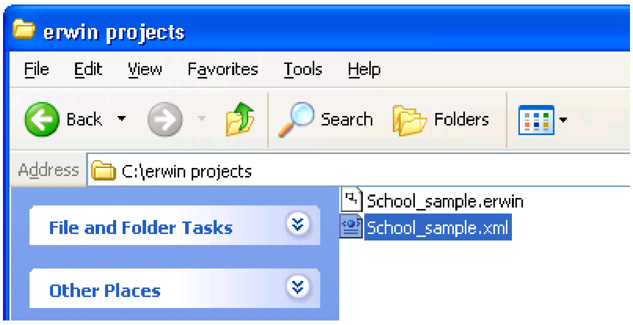
(Properties derived from other properties or properties that are read-only will not be saved.)

○ AllFusion Repository Format

(Macros are expanded in this format, and the resulting file cannot be opened with AllFusion ERwin DM.)

Figure 8-3 *Exporting the XML in standard XML format*

5. Click **OK** to confirm. This saves an XML file that can be used for importing into VP-UML.



C:\erwin projects

School_sample.erwin

School_sample.xml

Figure 8-4 *XML file saved from ERwin Data Modeler*

6. To import an ERwin Data Modeler project into VP-UML, select **File > Import > ERwin Project(XML)...** in the main menu of VP-UML.

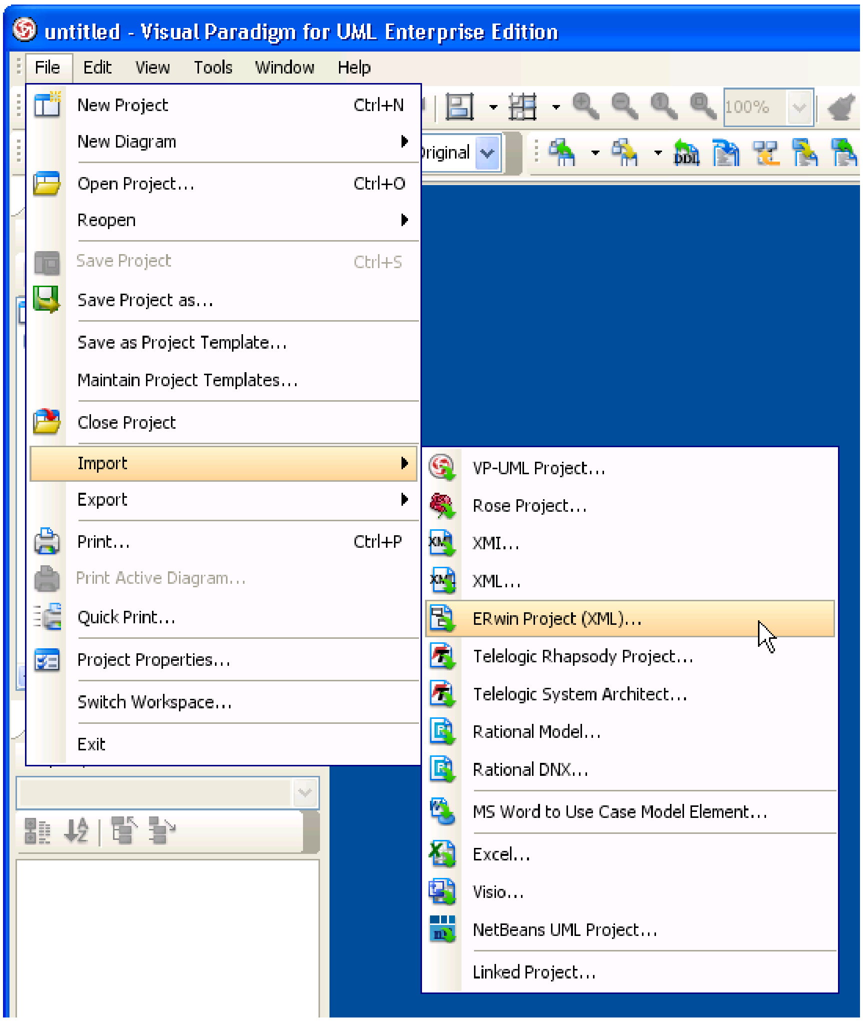
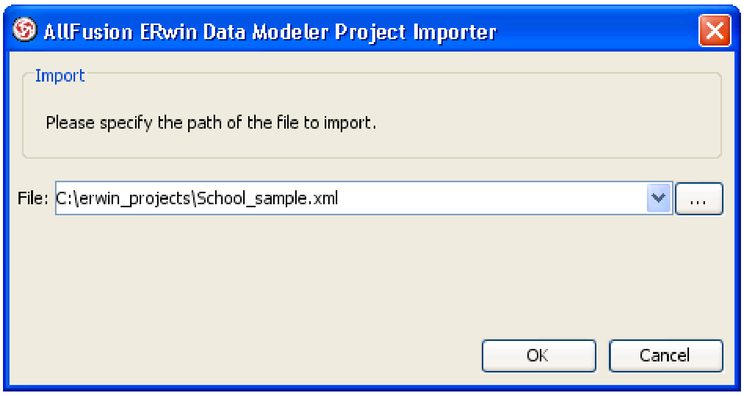


Figure 8-5 The import ERwin Data Modeler Project menu

7. Specify the file path of the XML file.

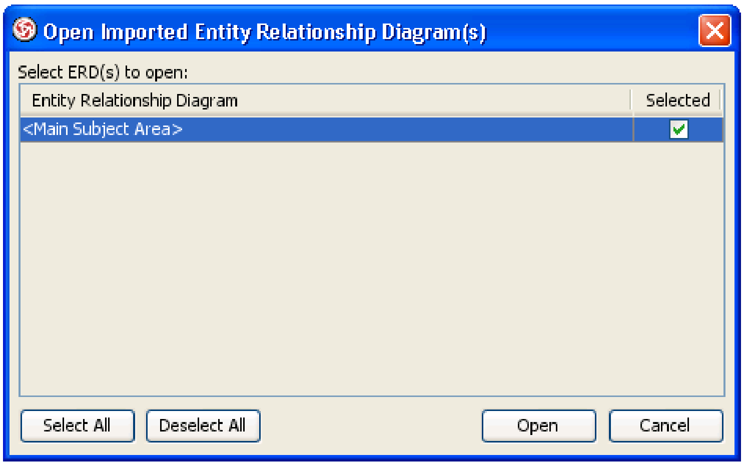


Please specify the path of the file to import.

File: C:\erwin_projects\School_sample.xml

Figure 8-6 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.



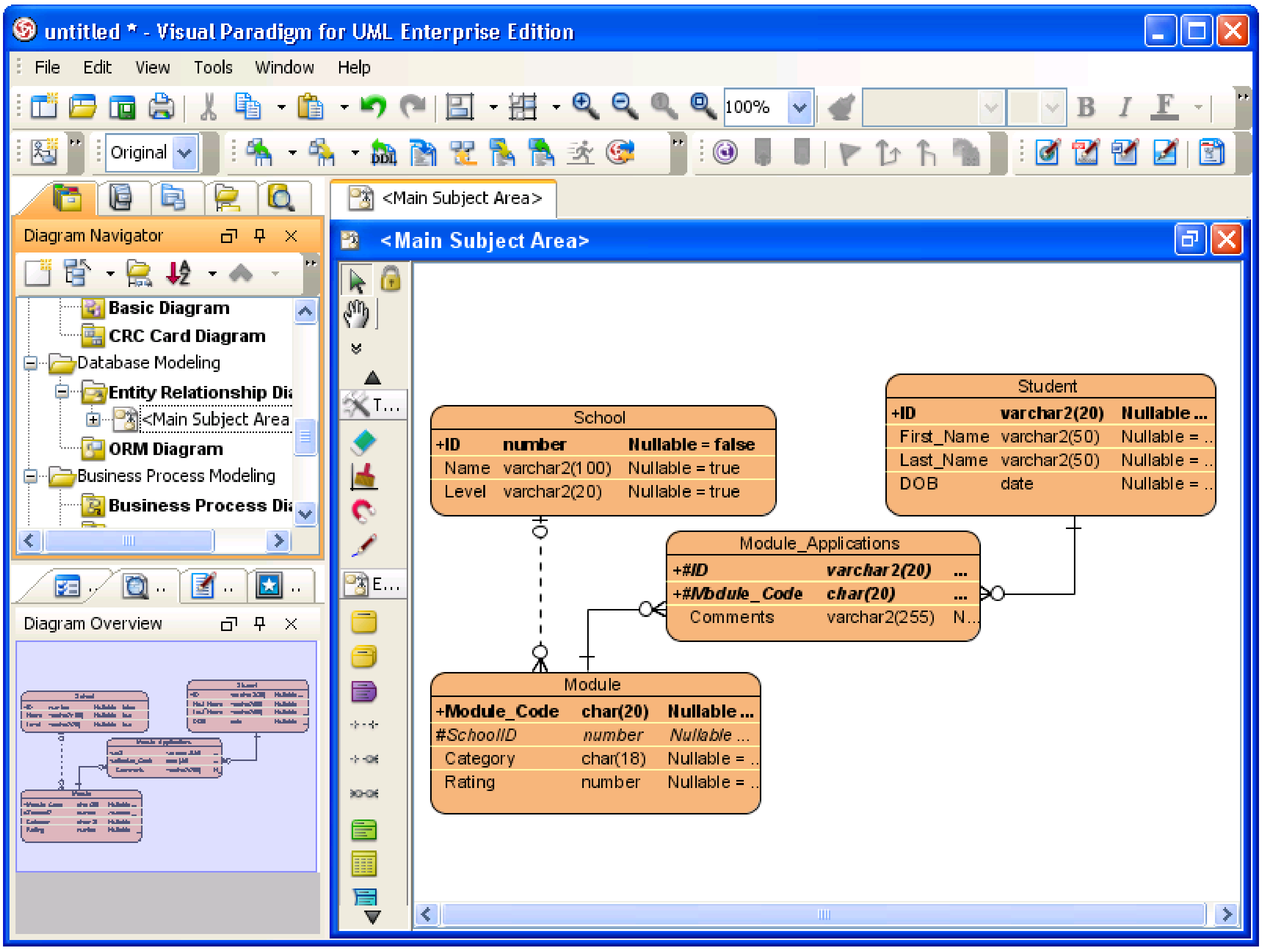
Open Imported Entity Relationship Diagram(s)

Select ERD(s) to open:

Entity Relationship Diagram	Selected
<Main Subject Area>	☑

Figure 8-7 *Select the diagram(s) to open after importing*

9. Select the diagram(s) to open and click **Open** to open them. The drawings will then be opened.



untitled * - Visual Paradigm for UML Enterprise Edition

School

+ID	number	Nullable = false
Name	varchar2(100)	Nullable = true
Level	varchar2(20)	Nullable = true

Student

+ID	varchar2(20)	Nullable ...
First_Name	varchar2(50)	Nullable = ..
Last_Name	varchar2(50)	Nullable = ..
DOB	date	Nullable = ..

Module_Applications

+#ID	varchar2(20)	...
+#Module_Code	char(20)	...
Comments	varchar2(255)	N...

Module

+Module_Code	char(20)	Nullable ...
#SchoolID	number	Nullable ...
Category	char(18)	Nullable = ..
Rating	number	Nullable = ..

Figure 8-8 *An Entity Relationship Diagram imported from ERwin Data Modeler*

Importing Telelogic Rhapsody Project into VP-UML

1. Here is a Telelogic Rhapsody Project:

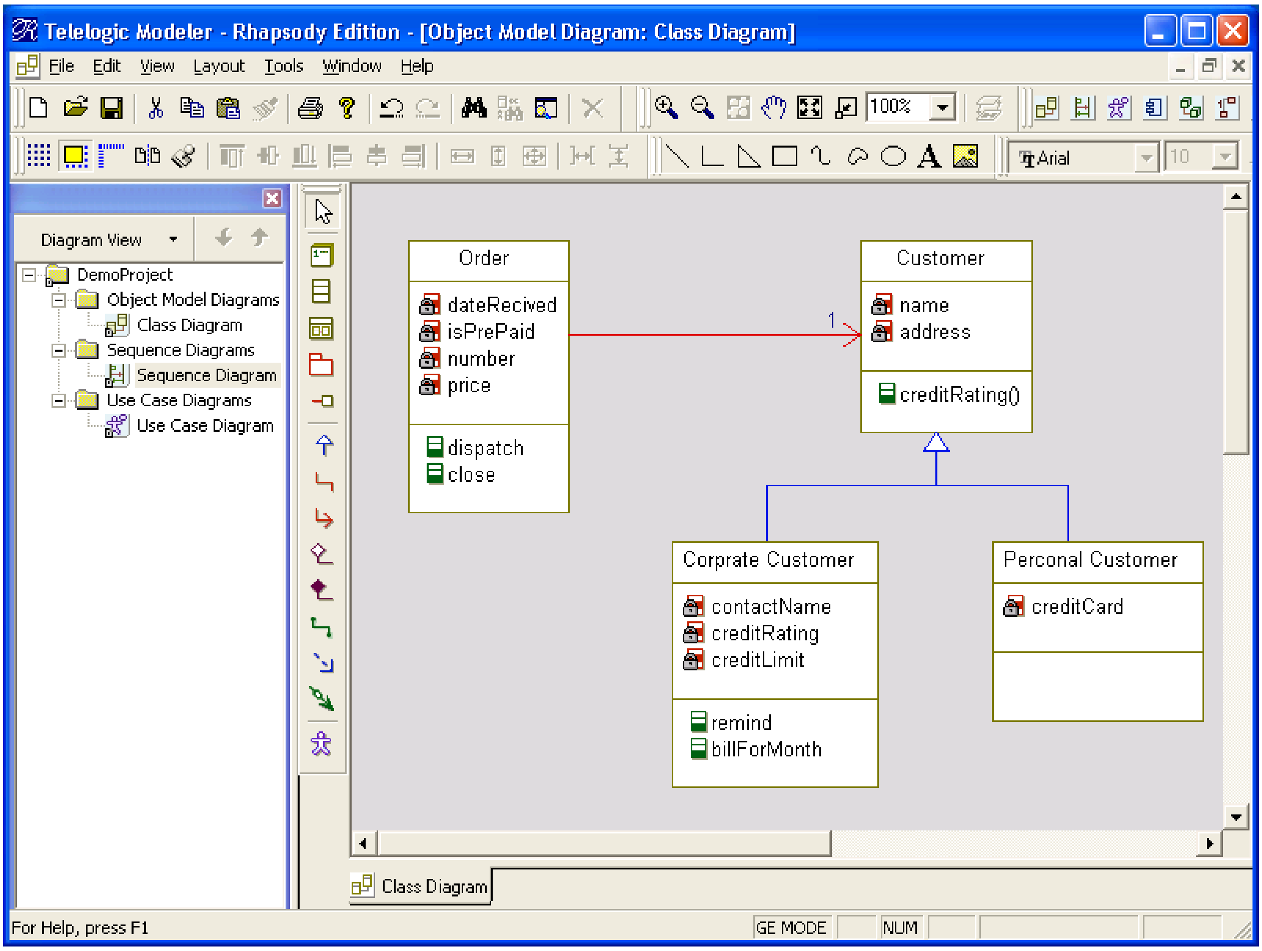


Figure 9-1 *A drawing in Telelogic Modeler*

To import a Telelogic Phapsody project into VP-UML, select **File > Import > Telelogic Rhapsody Project ...** in the main menu of VP-UML.

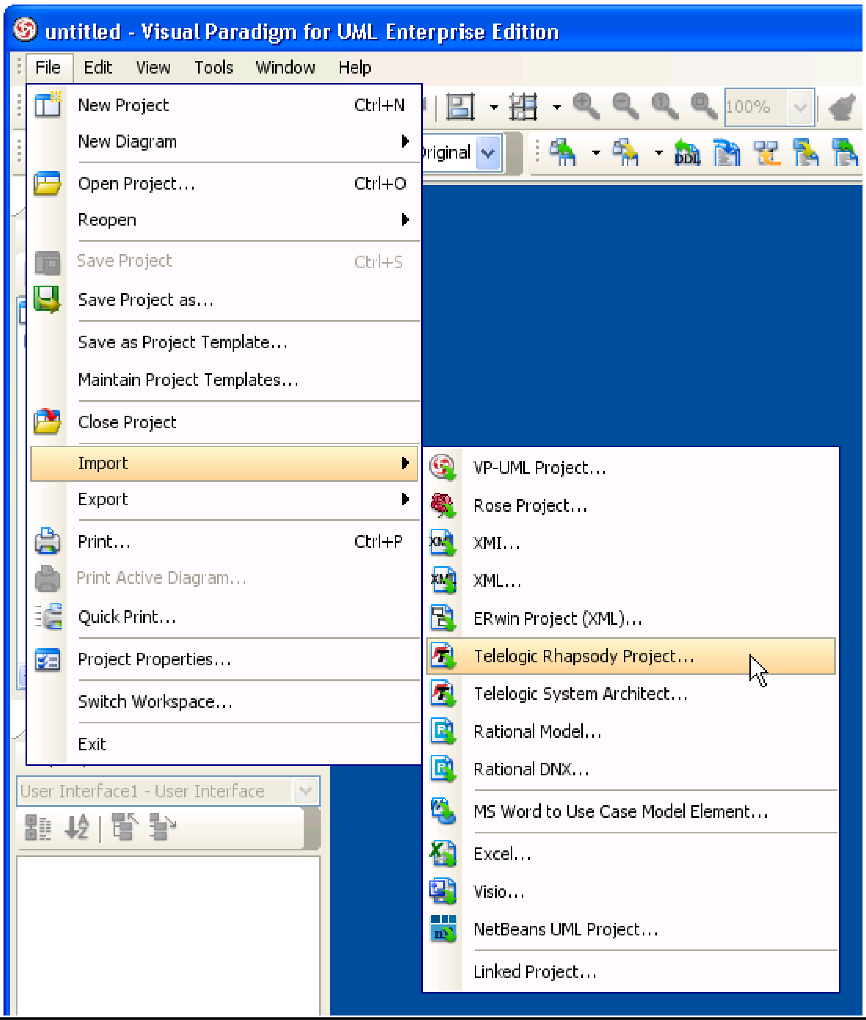
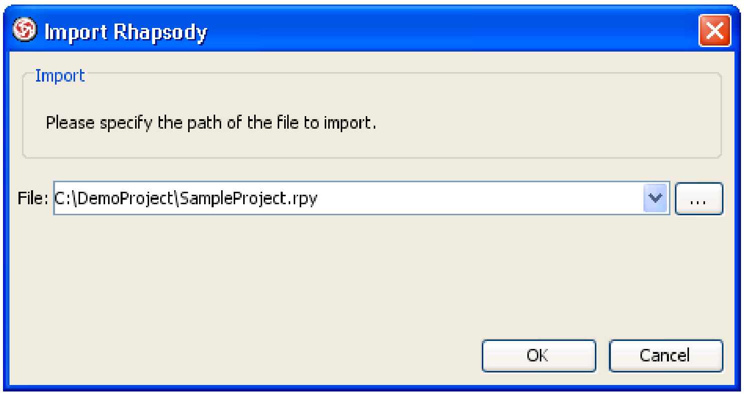


Figure 9-2 *The import Telelogic Rhapsody Project menu*

2. Specify the file path of the Telelogic Rhapsody Project.



Please specify the path of the file to import.

File: C:\DemoProject\SampleProject.rpy

Figure 9-3 *Specifying Telelogic Rhapsody Project path*

3. Click **OK** to start importing. When import is completed, the drawings can then be accessible in the **Diagram Navigator**.

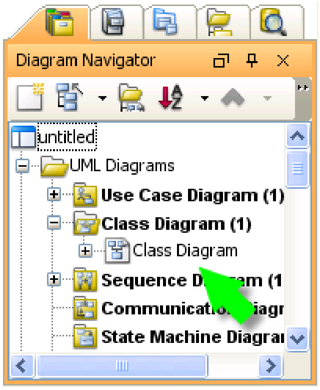


Figure 9-4 *Diagram Navigator listing the imported diagram(s)*

You can then double click on the diagram node to open the diagram.

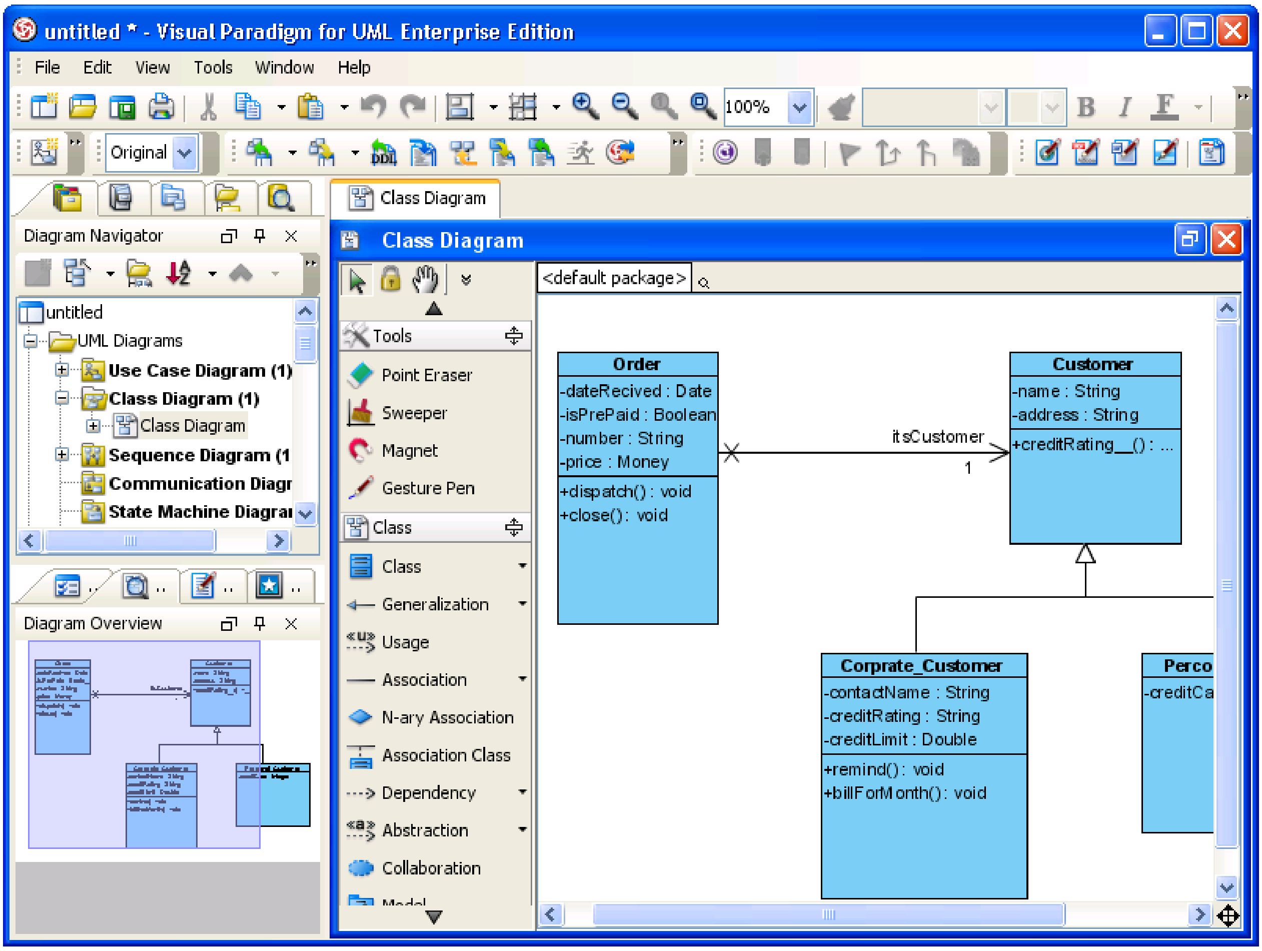


Figure 9-5 *A Class Diagram imported from Telelogic Rhapsody Project*

Importing Telelogic System Architect into VP-UML

1. Here is a Telelogic System Architect drawing:

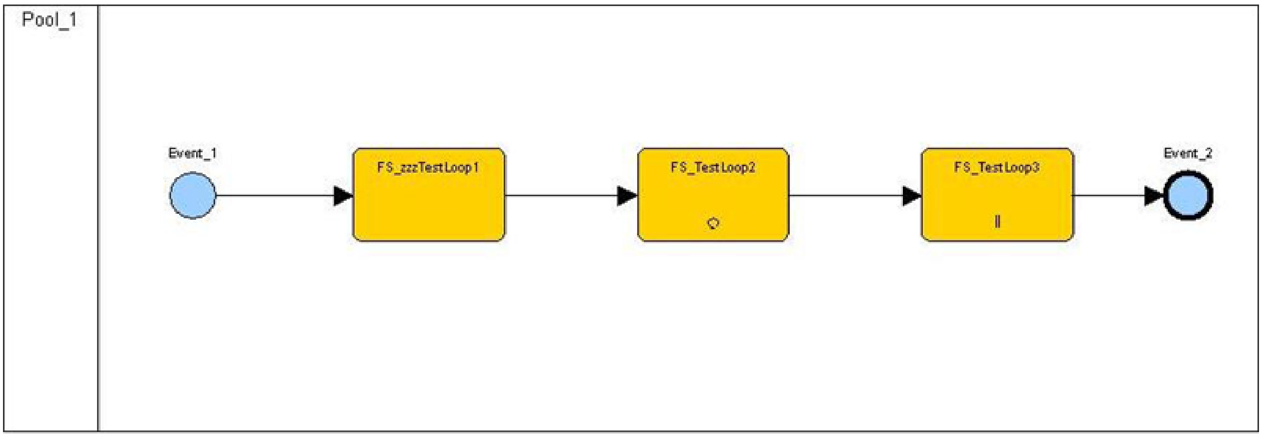


Figure 9-6 *A Telelogic System Architect drawing*

To import a Telelogic System Architect project into VP-UML, select **File > Import > Telelogic System Architect ...** in the main menu of VP-UML.

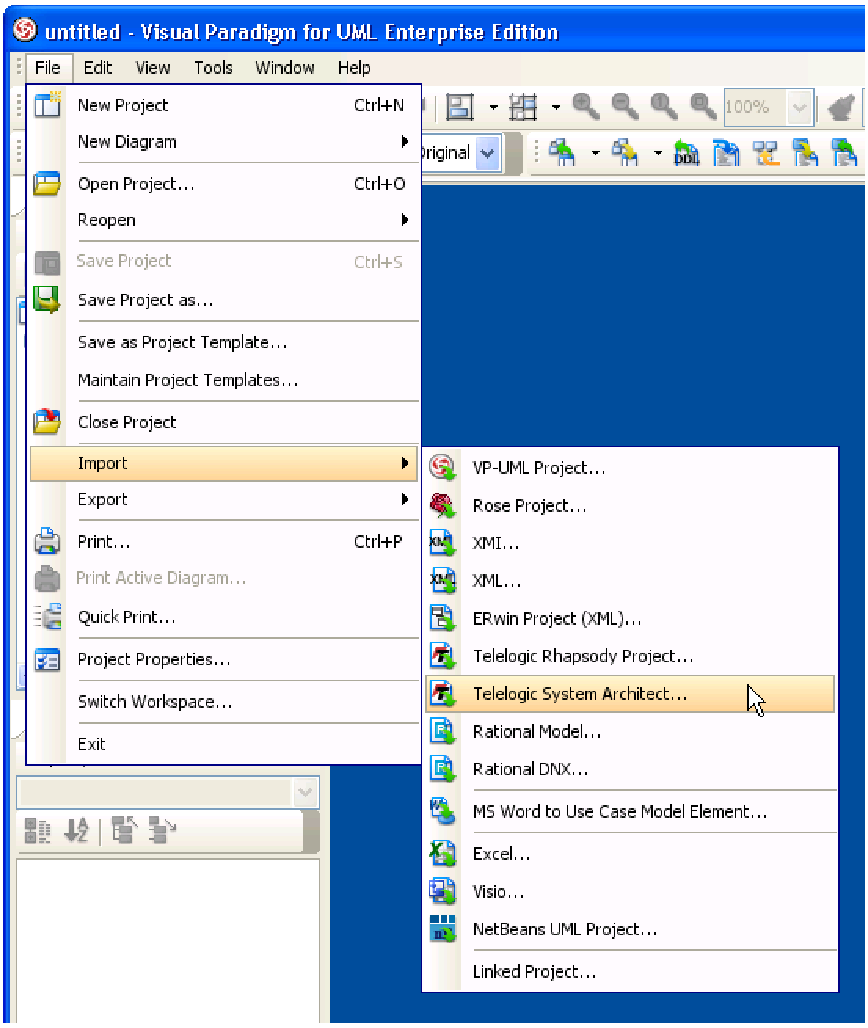
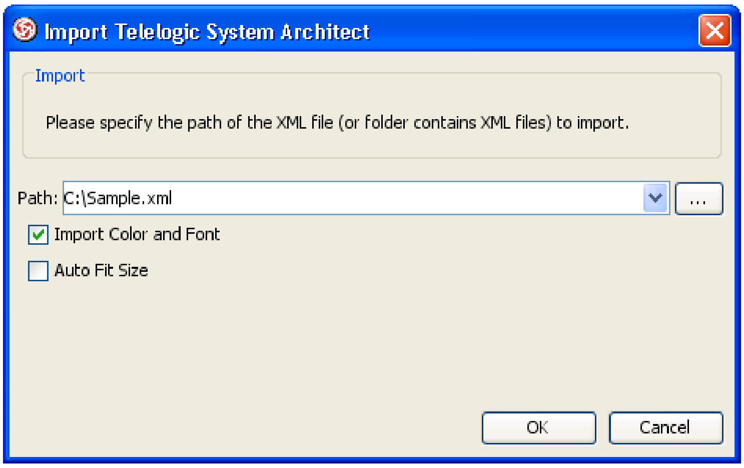


Figure 9-7 *The import Telelogic System Architect menu*

2. Specify the file path of the Telelogic System Architect project.



Import Telelogic System Architect

Import

Please specify the path of the XML file (or folder contains XML files) to import.

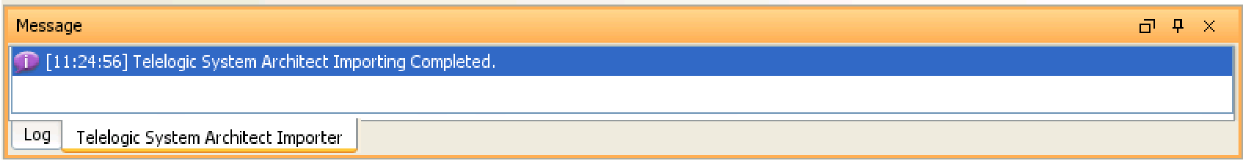
Path: C:\Sample.xml

☑ Import Color and Font

□ Auto Fit Size

Figure 9-8 *Specifying Telelogic System Architect path*

3. Click **OK** to start importing. When import is completed, the message pane will popup, with a notification appear in it.



[11:24:56] Telelogic System Architect Importing Completed.

Figure 9-9 *The dialog box indicating the progress of importing*

The drawings can then be accessible in the **Diagram Navigator**.

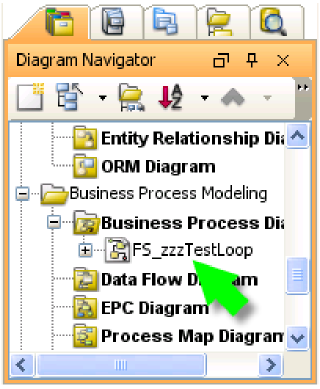


Figure 9-10 *Diagram Navigator listing the imported diagram(s)*

You can then double click on the diagram node to open the diagram.

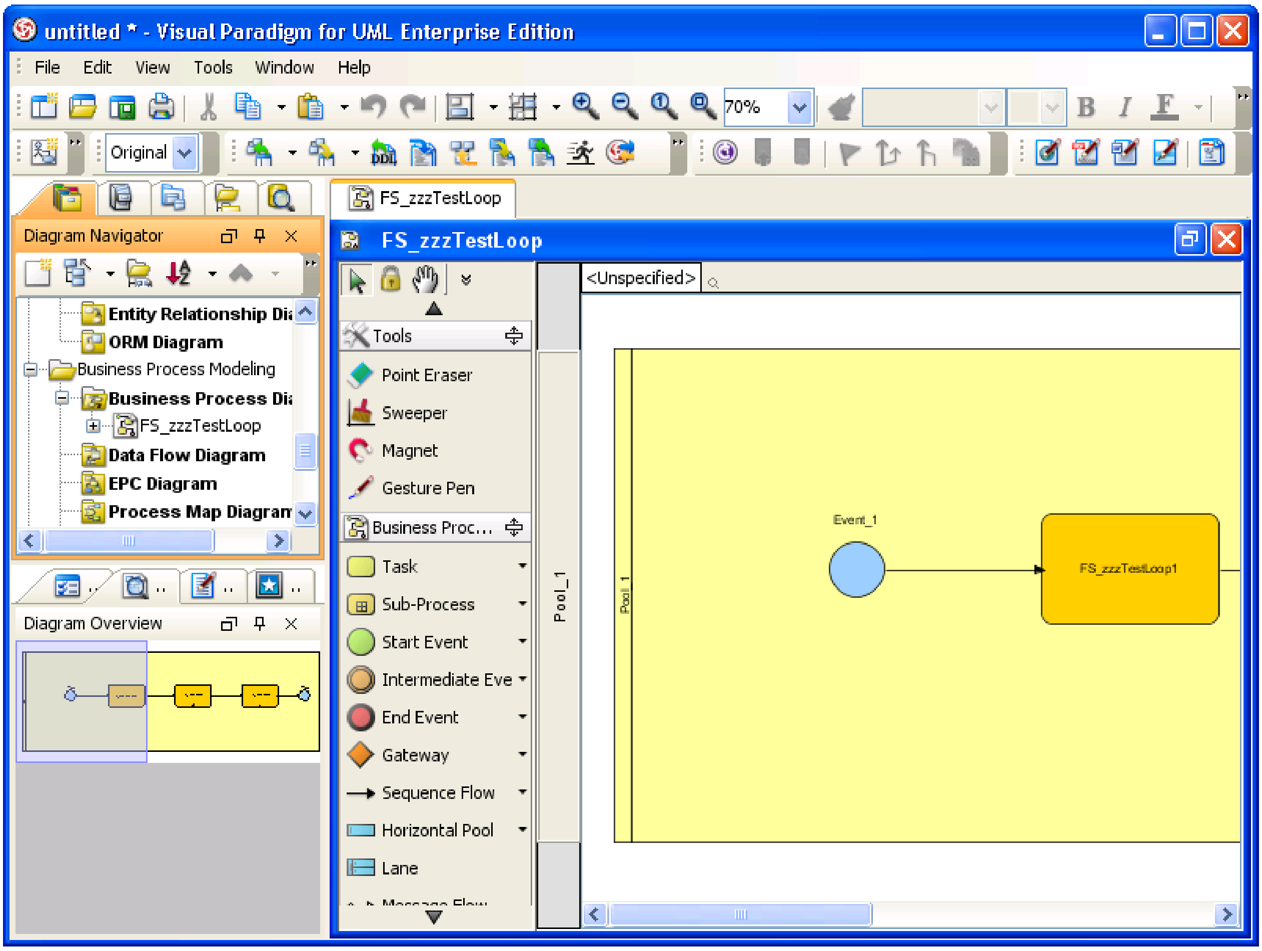


Figure 9-11 *A Business Process Diagram imported from Telelogic System Architect*

1. Here is a NetBeans UML Diagram:

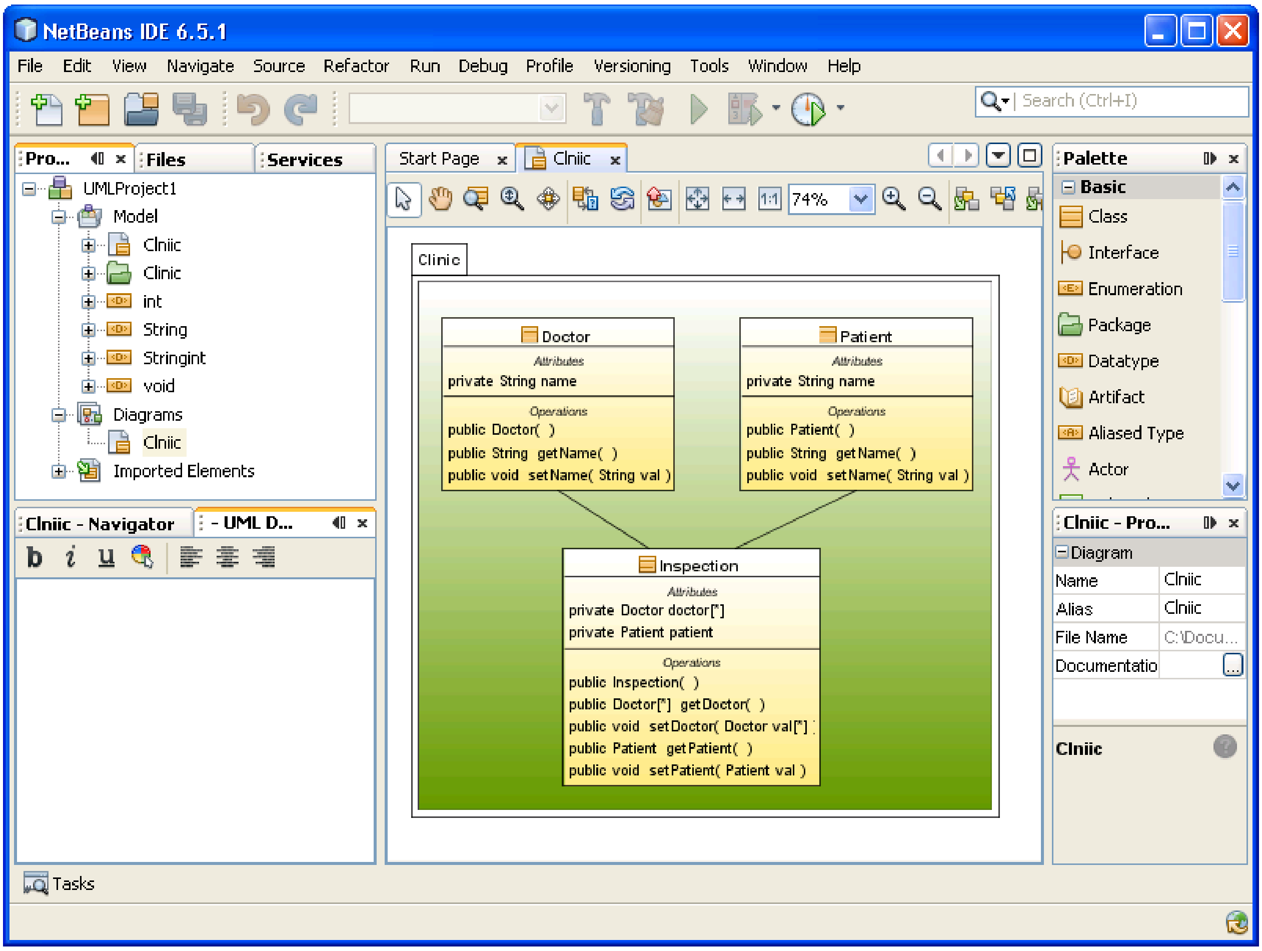


Figure 10-1 A Class Diagram drew in NetBeans

To import a NetBeans UML project into VP-UML, select **File > Import > NetBeans UML Project...** in the main menu of VP-UML.

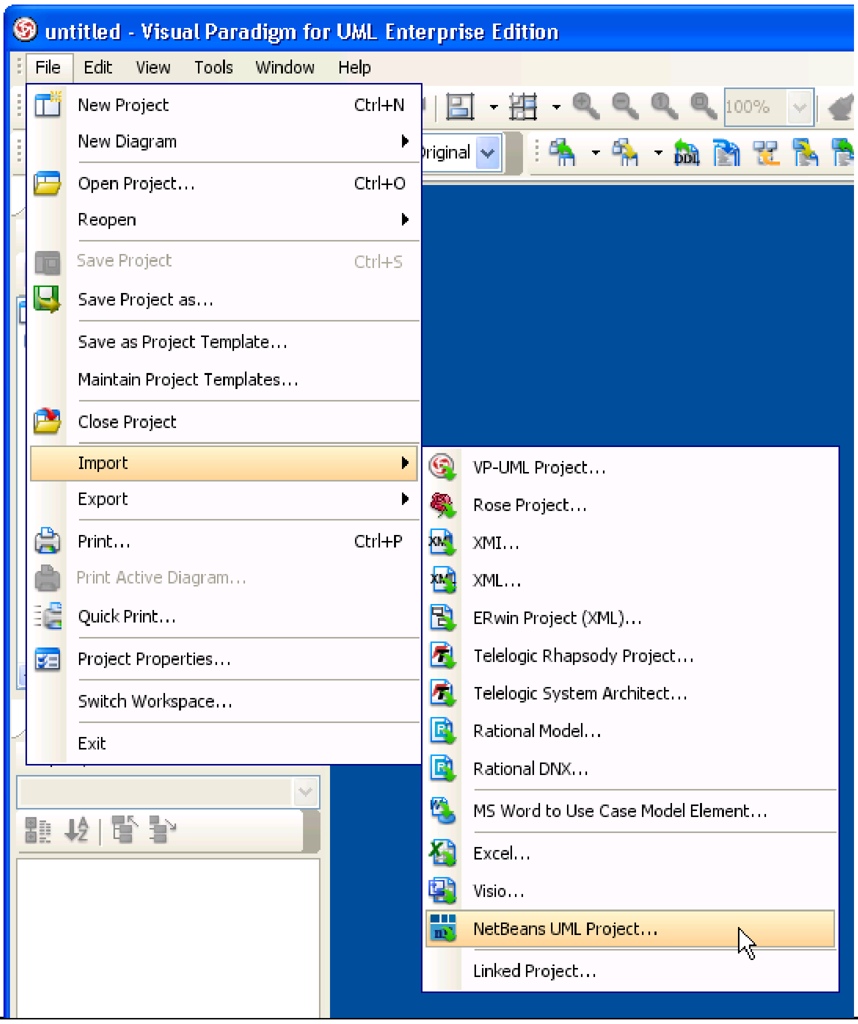
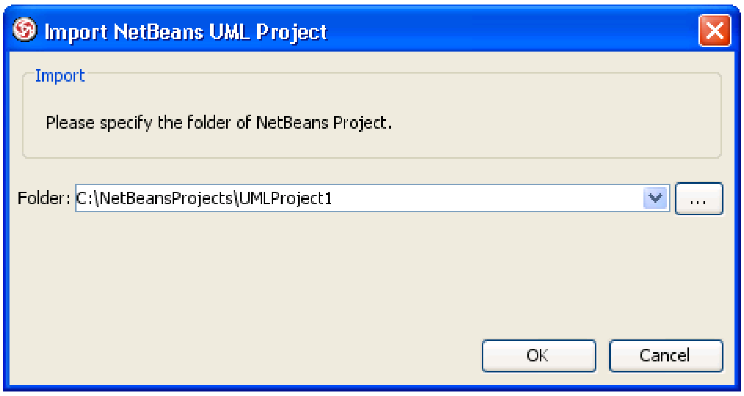


Figure 10 -2 The import NetBeans UML Project menu

2. Specify the file path of the NetBeans project.



Import NetBeans UML Project

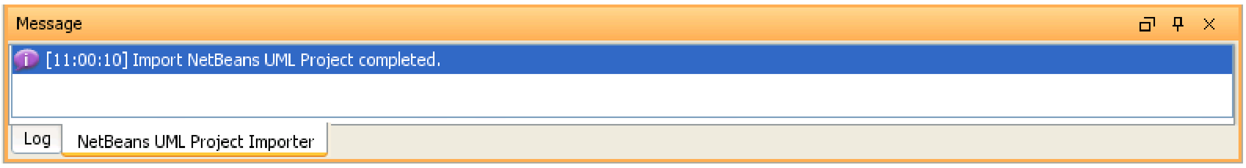
Import

Please specify the folder of NetBeans Project.

Folder: C:\NetBeansProjects\UMLProject1

Figure 10 -3 *Specifying NetBeans UML Project path*

3. Click **OK** to start importing. When import is completed, the message pane will popup, with a notification appear in it.



[11:00:10] Import NetBeans UML Project completed.

Figure 10 -4 *The dialog box indicating the progress of importing*

The drawings can then be accessible in the **Diagram Navigator**.

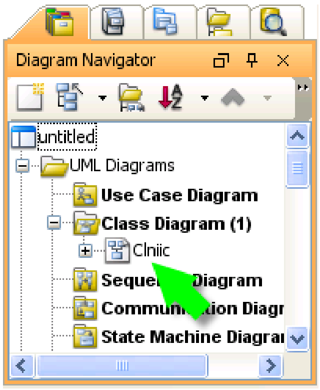
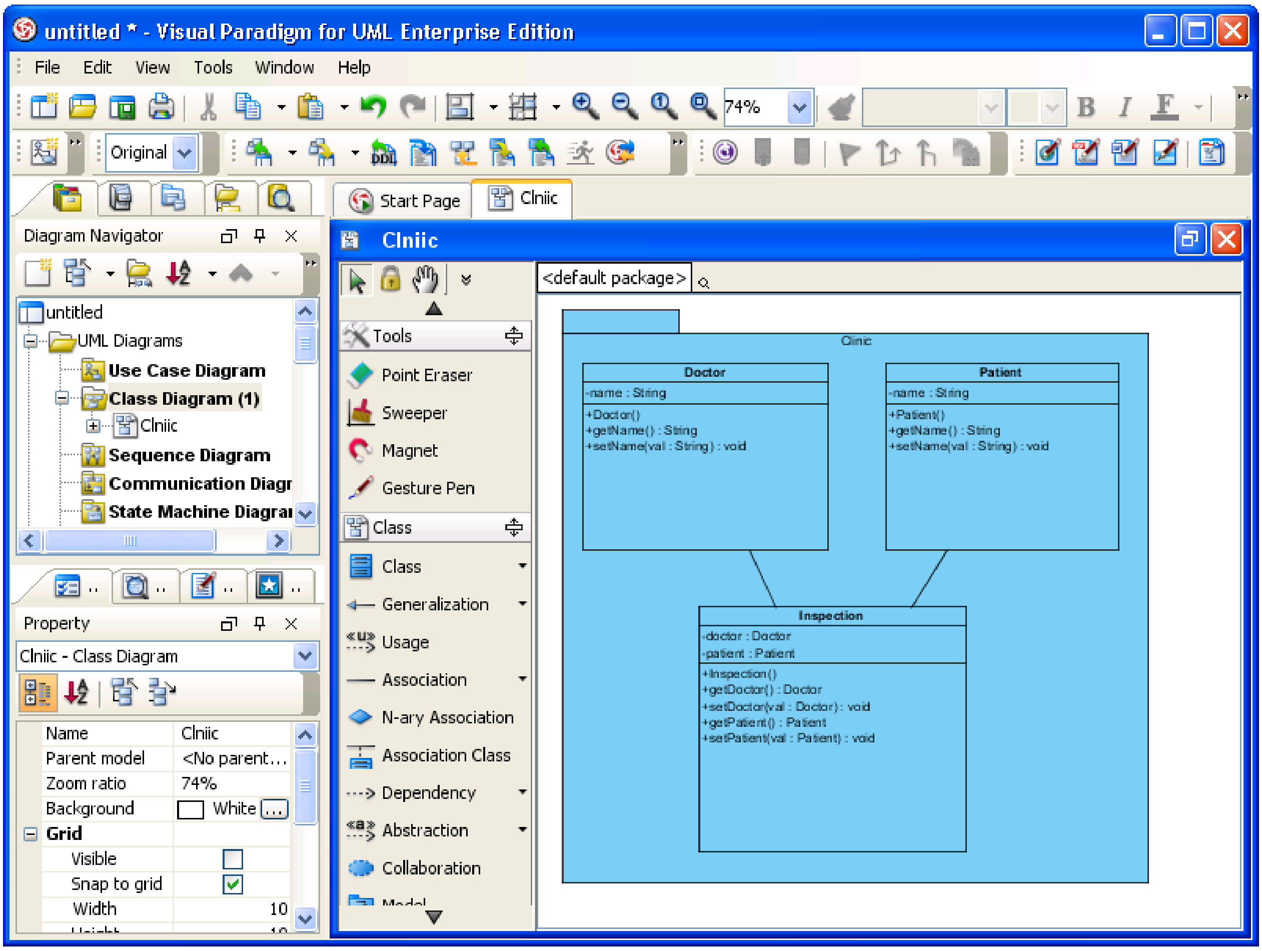


Figure 10 -5 *Diagram Navigator listing the imported diagram(s)*

You can then double click on the diagram node to open the diagram.



Name	Clniic
Parent model	<No parent…
Zoom ratio	74%
Background	White
Grid	
Visible	☐
Snap to grid	☑
Width	10

Figure 10 -6 *A Class Diagram imported from NetBeans UML Project*

NOTE: Due to different ways in presenting diagrams in VP-UML and NetBeans, the imported shapes may contain unnecessary spaces in them. 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 on a diagram, right click on the diagram background and select **Diagram Content > Auto Fit Shapes Size** in the popup menu.

Exporting Active Diagram as Image

This feature exports the active (i.e. opening) diagram as an image file. To export the active diagram as an image file:

1. Perform one of the following actions:
   - Select **File > Export > Active Diagram as Image...** from main menu.

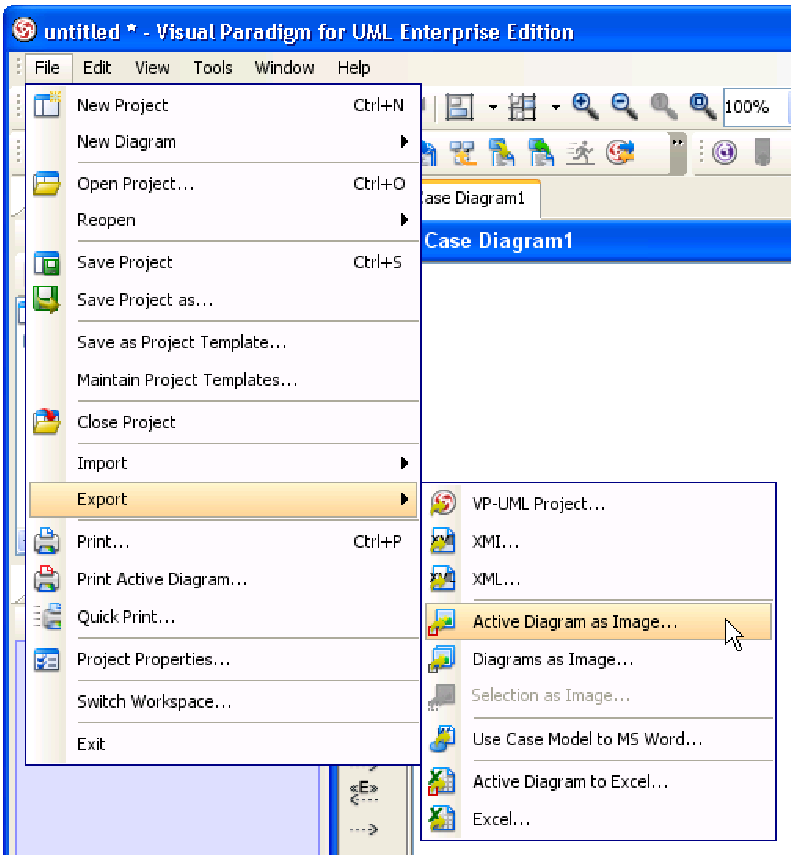


Figure 11-1 *Export active diagram as image*

   - Click on the **Export Active Diagram as Image** button on the **Import and Export** toolbar. The toolbar is hidden by default. You need to right click on the toolbar and select **Import and Export** to enable.

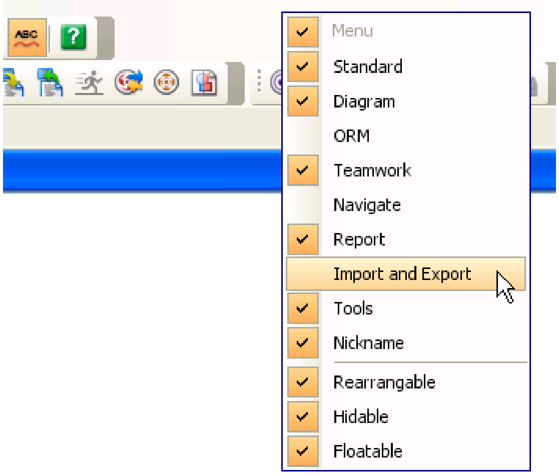
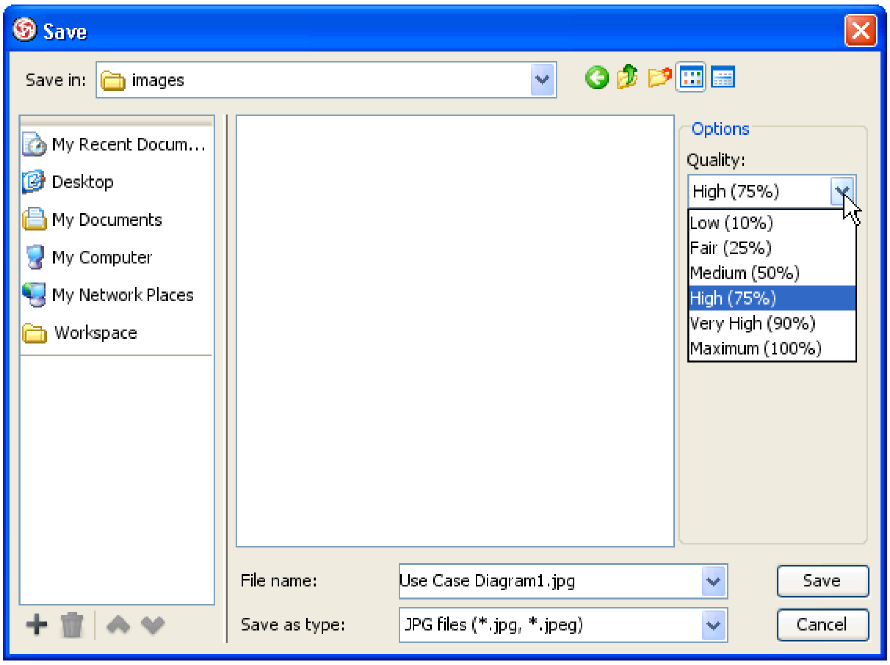


Figure 11-2 *Enable Import and Export toolbar*

2. In the **Save** dialog box, set the image quality. The higher the quality, the clearer the image, the larger the image size.



Save in: images

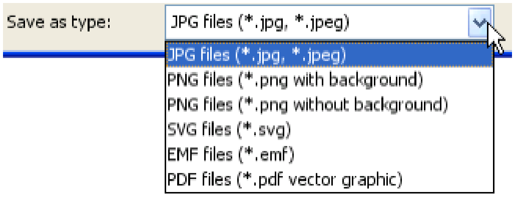
Quality: High (75%)

File name: Use Case Diagram1.jpg

Save as type: JPG files (*.jpg, *.jpeg)

Figure 11-3 *Set image quality*

3. Select the image format.



Save as type: JPG files (*.jpg, *.jpeg)

JPG files (*.jpg, *.jpeg)
PNG files (*.png with background)
PNG files (*.png without background)
SVG files (*.svg)
EMF files (*.emf)
PDF files (*.pdf vector graphic)

Figure 11-4 *Select image format*

NOTE: 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.

NOTE: You can export VP-UML 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.

4. Specify the filename.
5. Click **Save** to export.

Exporting Multiple Diagrams as Images

To Export Diagrams

This feature exports one or more diagrams as images. To export diagrams as image files:

1. Perform one of the following actions:
   - Select **File > Export > Diagrams as Image...** from main menu.

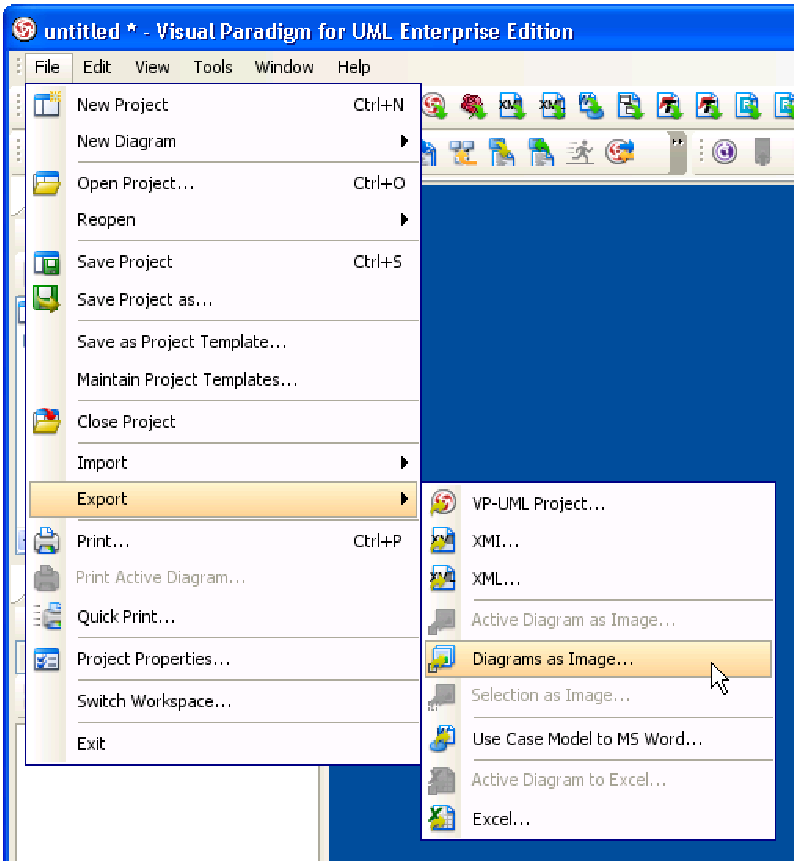


Figure 11-5 *To export diagrams as images*

   - Click on the **Export Diagram as Image** button on the toolbar. The toolbar is hidden by default. You need to right click on the toolbar and select **Import and Export** to enable.

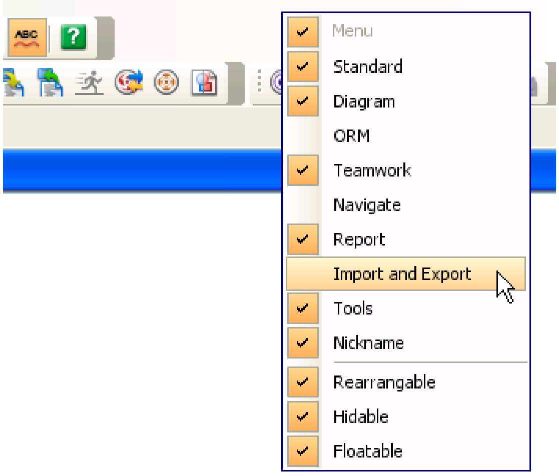


Figure 11-6 *Enable Import and Export toolbar*

2. In the **Diagram Exporter** dialog box, specify the diagrams to export and preview the exported image.

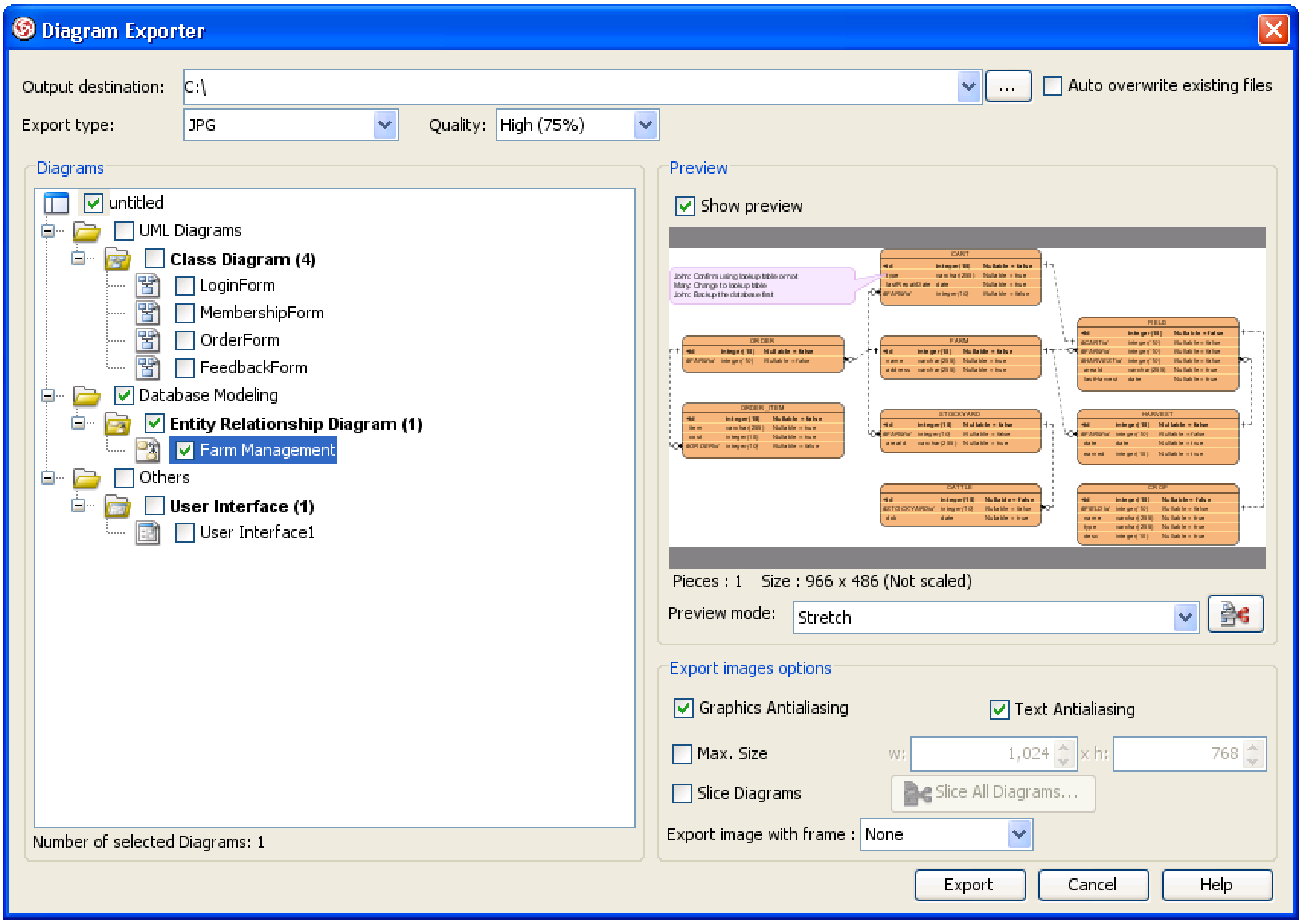


Diagram Exporter

Output destination: C:\ … □ Auto overwrite existing files

Export type: JPG Quality: High (75%)

Diagrams

- ☑ untitled
  - □ UML Diagrams
    - □ **Class Diagram (4)**
      - □ LoginForm
      - □ MembershipForm
      - □ OrderForm
      - □ FeedbackForm
  - ☑ Database Modeling
    - ☑ **Entity Relationship Diagram (1)**
      - ☑ Farm Management
  - □ Others
    - □ **User Interface (1)**
      - □ User Interface1

Number of selected Diagrams: 1

Preview

☑ Show preview

Pieces : 1 Size : 966 x 486 (Not scaled)

Preview mode: Stretch

Export images options

☑ Graphics Antialiasing ☑ Text Antialiasing

□ Max. Size w: 1,024 x h: 768

□ Slice Diagrams Slice All Diagrams…

Export image with frame : None

Figure 11-7 *The Diagram Exporter*

3. Configure the export settings. Below is a description of the available settings.

Field	Description
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.
Auto overwrite existing files	You can check the **Auto overwrite existing files** checkbox to allow overwriting of files in the export process.
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 VP-UML 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.
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.
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.
Preview	The **Preview** pane shows the preview of the exported image of the selected diagram in the Diagrams pane. You can check/uncheck the **Show preview** checkbox to enable/disable the preview.
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.
✂	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.
Graphics/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 graphics, check the **Graphics Anti-aliasing** checkbox in the **Anti-aliasing pane**. Likewise, you can check the **Text checkbox** in the **Anti-aliasing pane** to apply anti-aliasing to text.
Max. Size	Maximum size of exported images. If the diagram size is larger than the max. size, it will be resized.
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.
Export image with frame	A frame is a border that print 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.

Table 11-1 *Description of Diagram Exporter dialog box*

4. Click **Export** to export the diagrams.

Slice a Diagram into Pieces with Diagram Slicer

User can slice large diagrams into pieces(number of files), as well as restrict the size of the exported diagram.

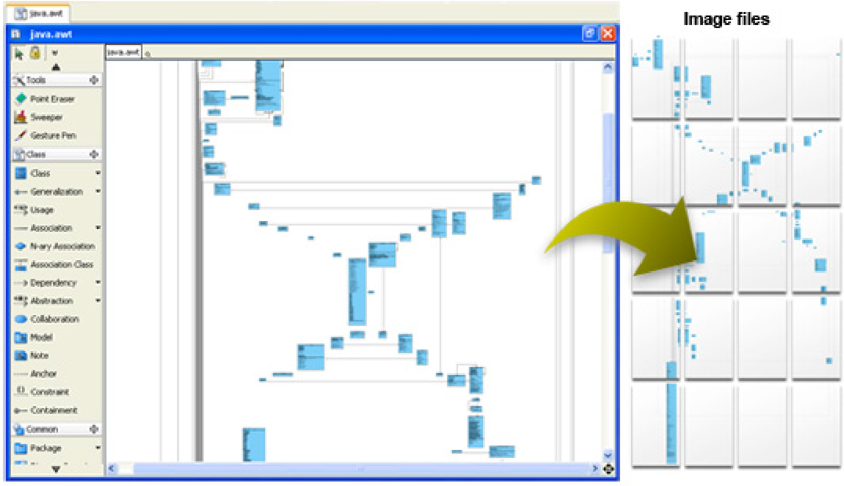
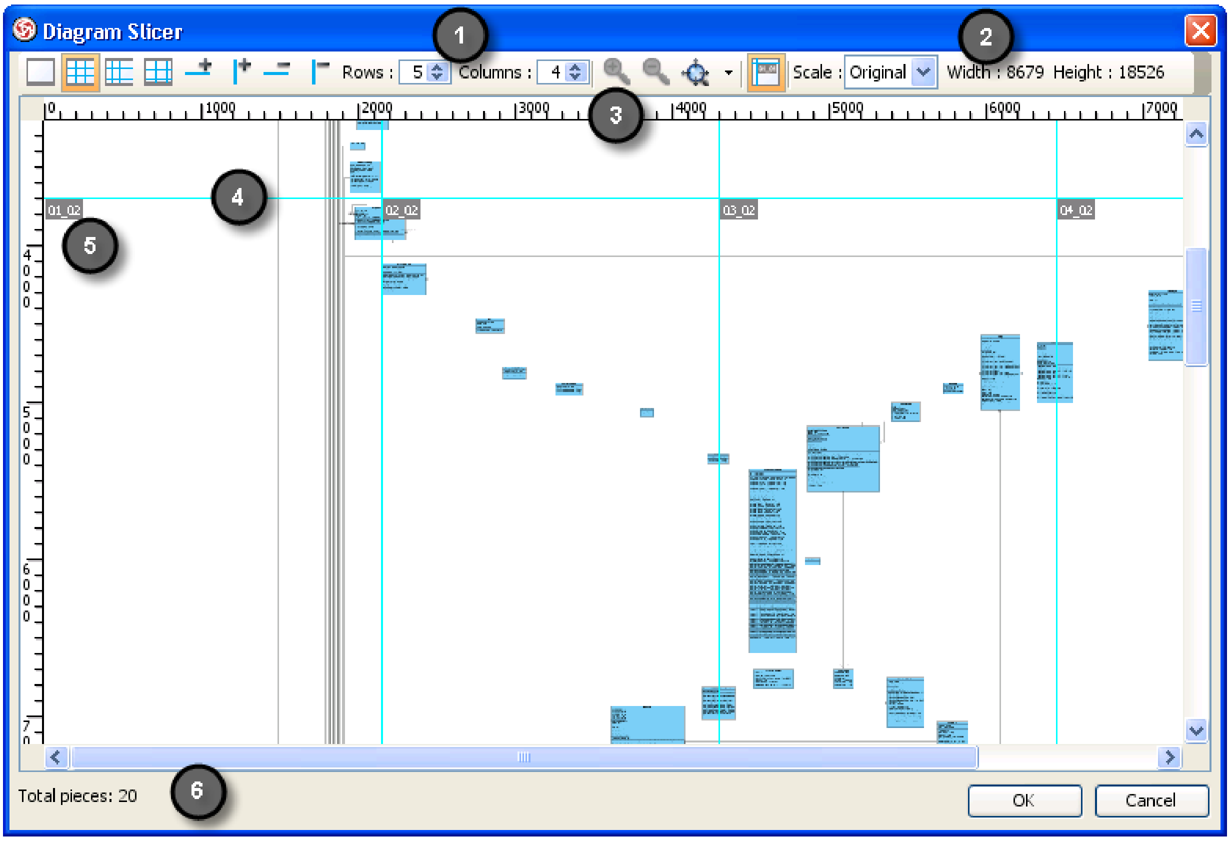


Figure 11-8 Slicing diagram with Diagram Slicer

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.



Rows : 5 Columns : 4 Scale : Original Width : 8679 Height : 18526

Total pieces: 20

Figure 11-9 The Diagram Slicer dialog box

Below is the description of different parts of the dialog box.

Region	Description
1	Control the number of rows and columns.
2	Control the scale of the exported diagram. There are three types of scale - Original, Size and Ratio. When configuring slicing for all diagrams, this part will not be displayed.
3	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.
4	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.
5	Index label - Shows the index of the pieces. This index will be printed on the exported file as well.
6	Show the number of total pieces to export.
☐	Set to **No Slicing**.
▦	Set to **Fixed Size**.
▤	Set to **Free Slicing**.
⊞	Set to **Free Ratio**.
+̲	Add a horizontal slice.
|+	Add a vertical slice.
−̲	Remove a row. When **Free Slice** is selected, the selected horizontal line will be removed.
|−	Remove a column. When **Free Slice** is selected, the selected vertical line will be removed.
🔍	Magnify the preview of diagram.
🔍	Diminish the preview of diagram.
⌖	Re-size the preview to fit the visible area.
🔍	Show actual size of diagram. It will be disabled if the diagram is too large.
▣	Control the visibility of index label

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.

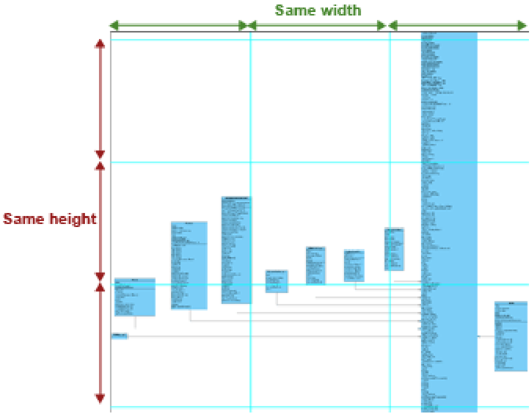


Figure 11-10 *Fixed Size*

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.

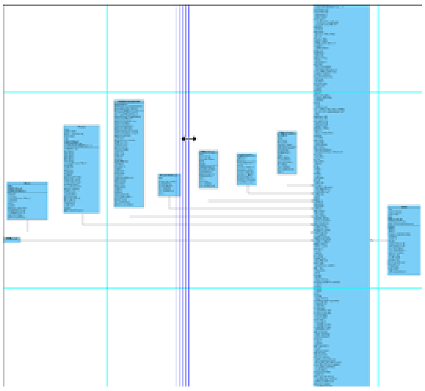


Figure 11-11 *Free Slicing*

Fixed Ratio

Fixed Ratio strategy gain 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.

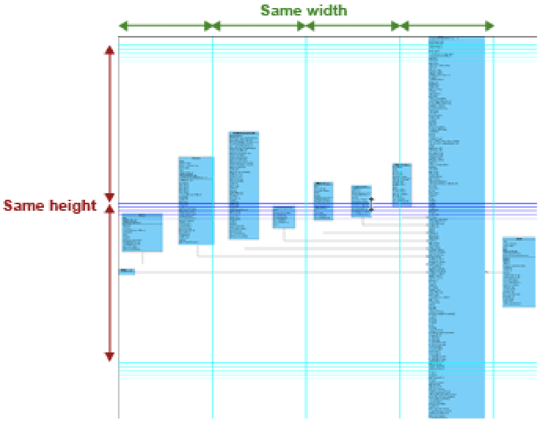


Figure 11-12 *Fixed Ratio*

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.

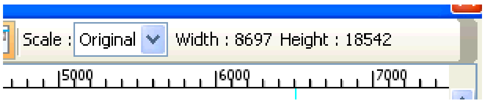

Scale : Original Width : 8697 Height : 18542

Figure 11-13 *Size of image can be controlled by Diagram Slicer*

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.

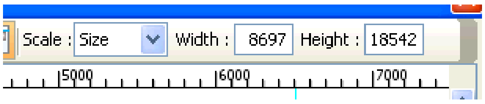

Scale : Size Width : 8697 Height : 18542

Figure 11-14 *Control image size by setting exact size - before set*

Let&rsquo;s type 1000 to Width, then the Height and the ruler will be updated automatically. The ruler shows the size of the diagram.

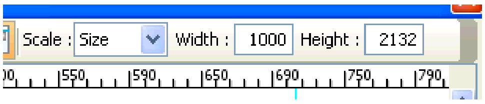

Scale : Size Width : 1000 Height : 2132

Figure 11-15 *Control image size by setting exact size - after set*

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.

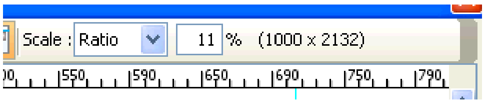

Scale : Ratio 11 % (1000 x 2132)

Figure 11-16 *Control image size by ratio - before set*

Let&rsquo;s input *50* to the percentage field. The total size and ruler will be updated automatically.

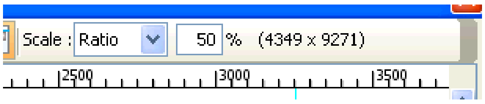

Scale : Ratio 50 % (4349 x 9271)

Figure 11-17 *Control image size by ratio - after set*

Exporting Portion of Diagram as Image

This feature exports a portion of diagram as an image file by selecting the shapes need to export. To export the selected shapes as image file:

1. Select the shapes to be exported. Multiple shape selection can be made in the ways below:
   - Click on a shape to select it, then press the **Control** or **Shift** key then select other shapes
   - Press on diagram and move the mouse cursor to select a range of shapes
   - Select a shape, select **Edit > Select All of Same Type** to select all shapes in same type

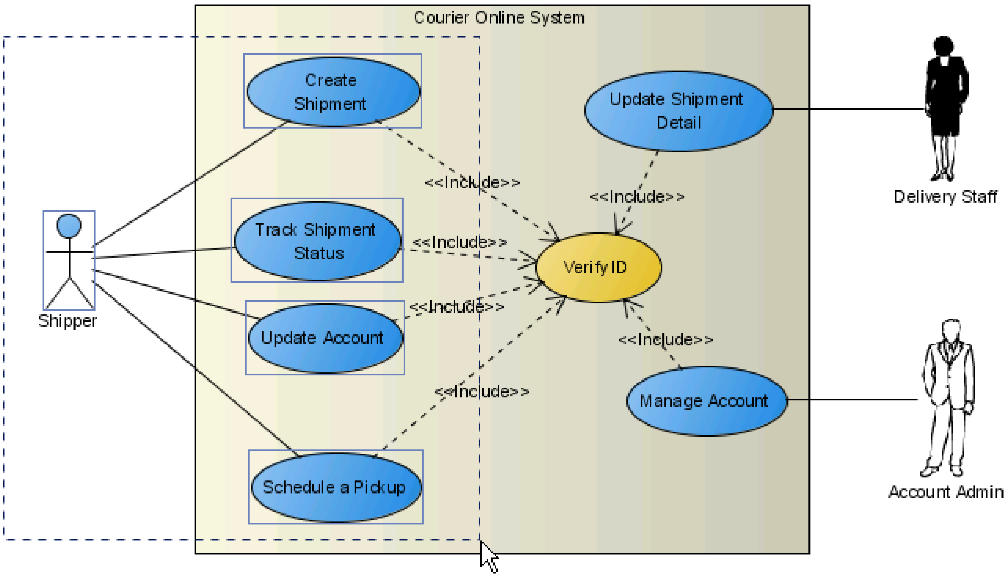


Figure 11-18 *Selecting shapes to export as image*

2. Perform one of the following actions:

- Select **File > Export > Selection as Image...** from main menu.

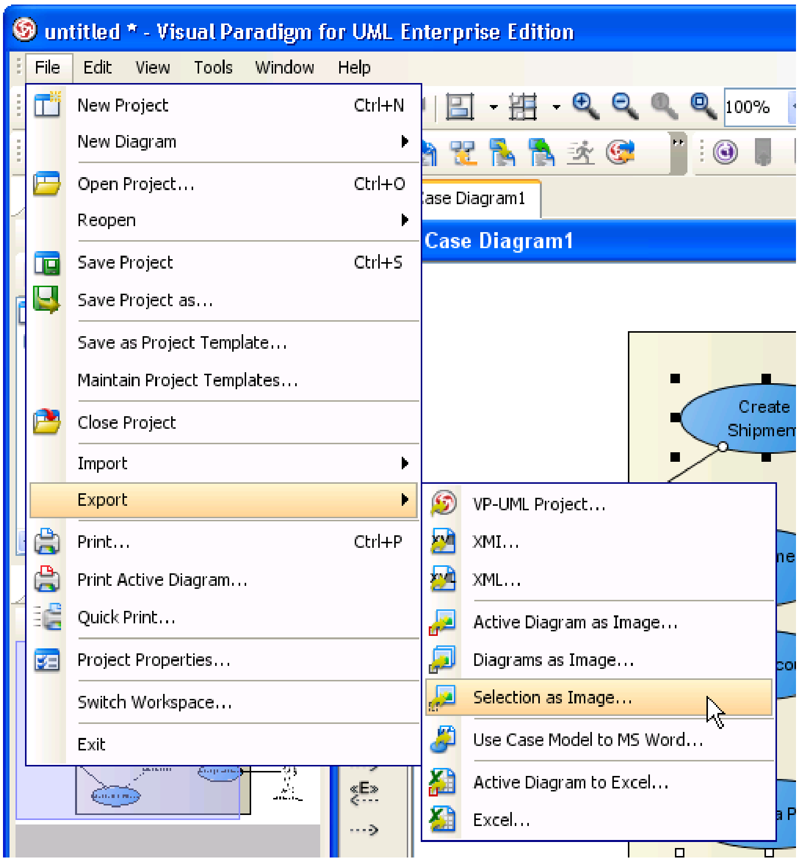


Figure 11-19 Export selected shapes as image

- Click on the **Export Selection as Image** button on the **Import and Export** toolbar. The toolbar is hidden by default. You need to right click on the toolbar and select **Import and Export** to enable.

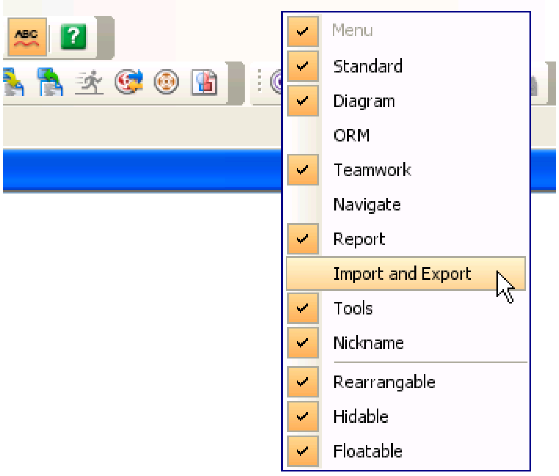
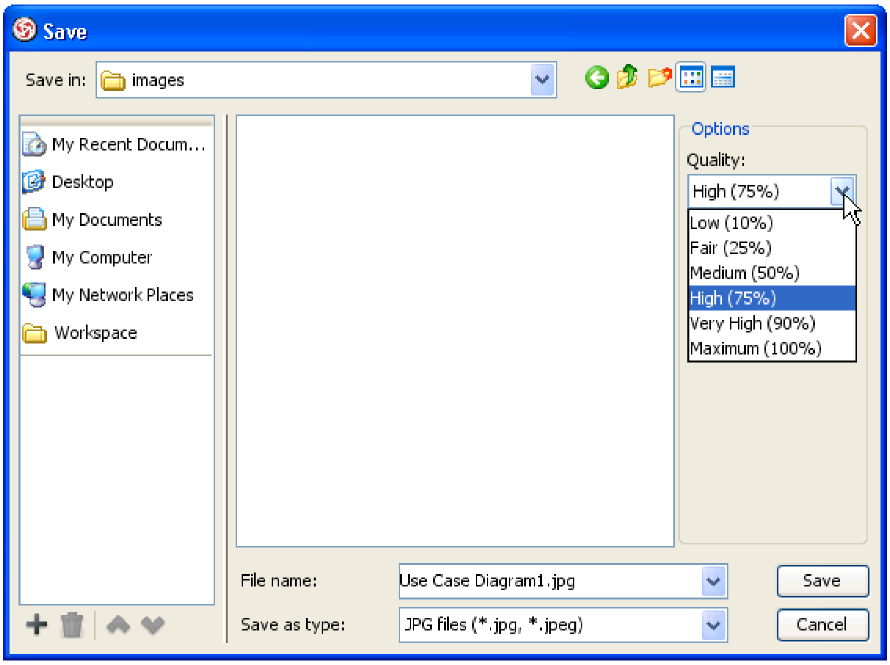


Figure 11-20 Enable Import and Export toolbar

3. In the **Save** dialog box, set the image quality. The higher the quality, the clearer the image, the larger the image size.



Save in: images

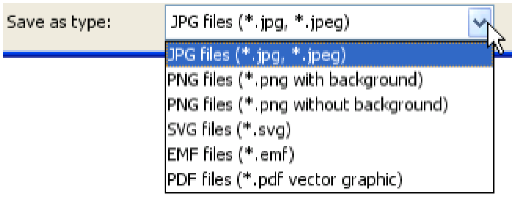
Quality: High (75%)

File name: Use Case Diagram1.jpg

Save as type: JPG files (*.jpg, *.jpeg)

Figure 11-21 *Set image quality*

4. Select the image format.



JPG files (*.jpg, *.jpeg)
PNG files (*.png with background)
PNG files (*.png without background)
SVG files (*.svg)
EMF files (*.emf)
PDF files (*.pdf vector graphic)

Save as type: JPG files (*.jpg, *.jpeg)

Figure 11-22 *Select image format*

NOTE: 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.

NOTE: You can export VP-UML 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.

5. Specify the filename.
6. Click **Save** to export.

Creating Shape in Shape Editor

Although UML and BPMN are well-established notations, sometimes they 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 VP-UML. To use shape Editor:

1. To launch Shape Editor, select **Tools > Shape Editor...** from the main menu.

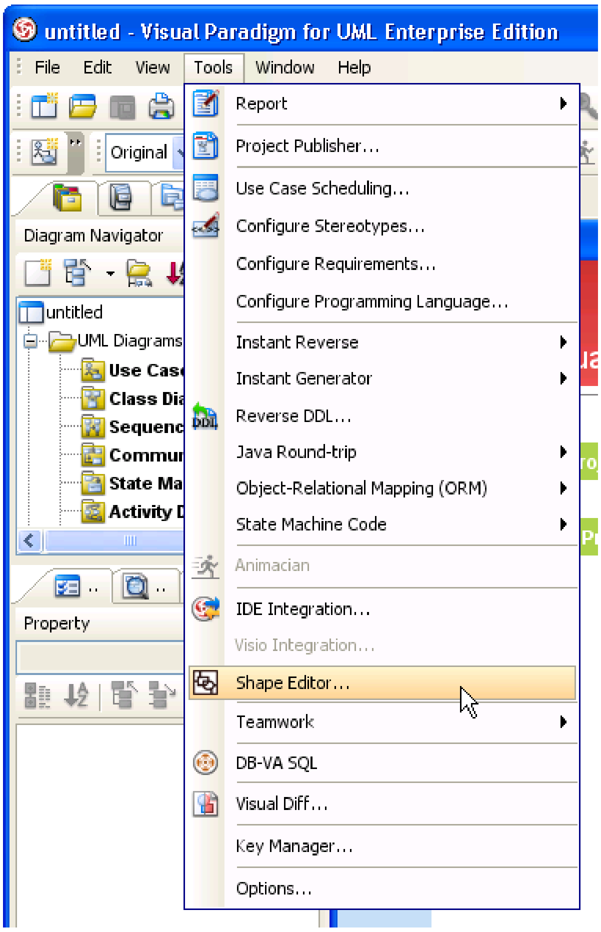
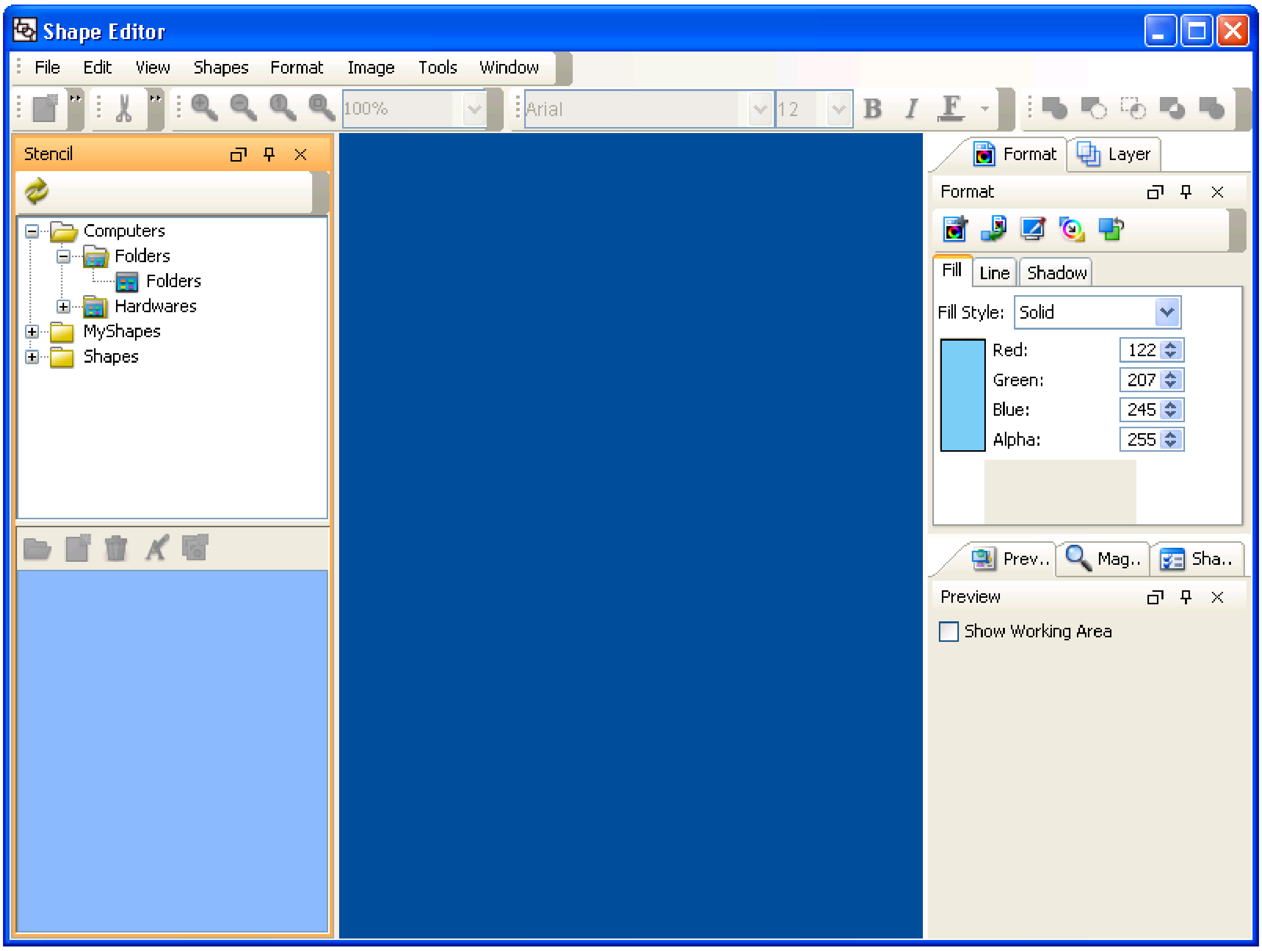


Figure 11-1 Opening Shape Editor through the main menu

This starts Shape Editor.



Fill Style: Solid

Red: 122

Green: 207

Blue: 245

Alpha: 255

□ Show Working Area

Figure 11-2 Shape Editor

2. 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 and select **Add Gallery** in the popup menu.

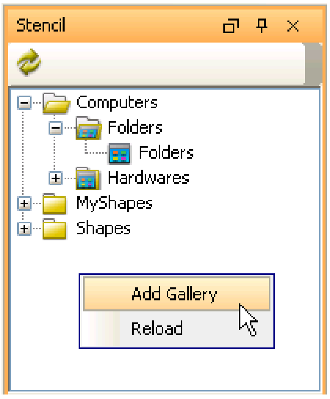


Figure 11-3 *To add a gallery*

3. Enter the gallery name and press the **Enter** key to confirm editing.

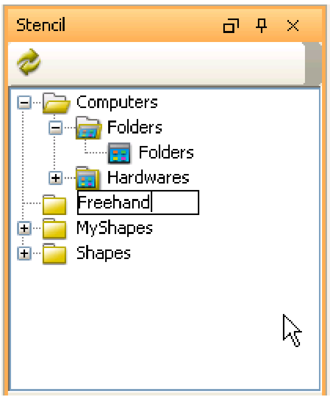


Figure 11-4 *Naming a gallery*

4. To create a category, right click on a gallery and select **Add Category** in the popup menu.

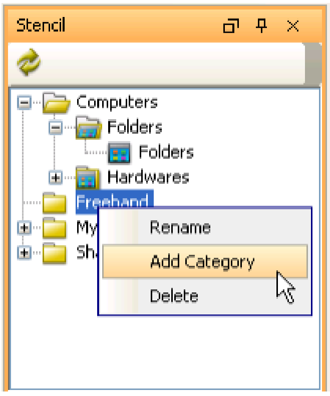


Figure 11-5 *To add a category*

5. Enter the category name in the Input dialog box and click OK to confirm.

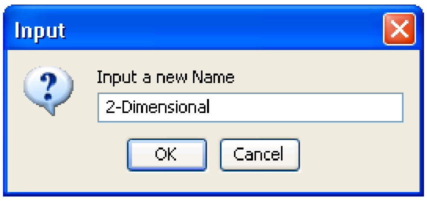


Figure 11-6 *Naming a category*

6. To create a stencil, right-click on a category and select **Add Stencil** in the popup menu.

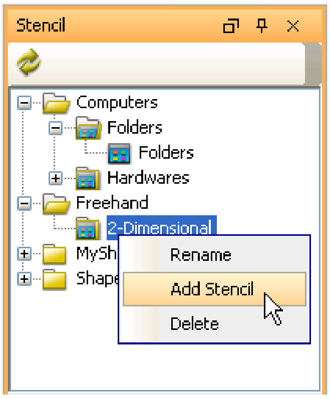


Figure 11-7 *To add a stencil*

7. Enter the stencil name and press the **Enter** key to confirm editing.

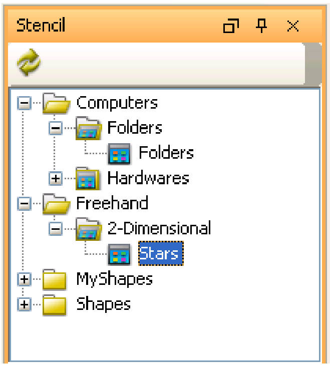


Figure 11-8 *Naming a stencil*

8. To create a shape, click on the (**New Shape**) button in the bottom part of the **Stencil** pane to create a blank drawing for drawing the shape.

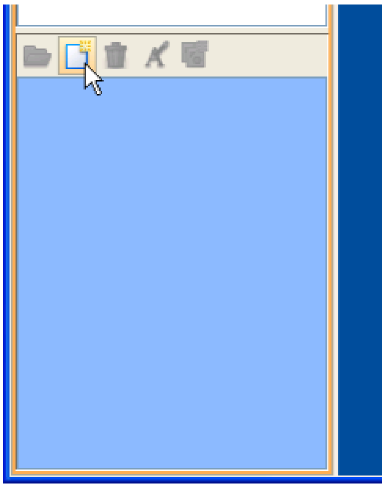
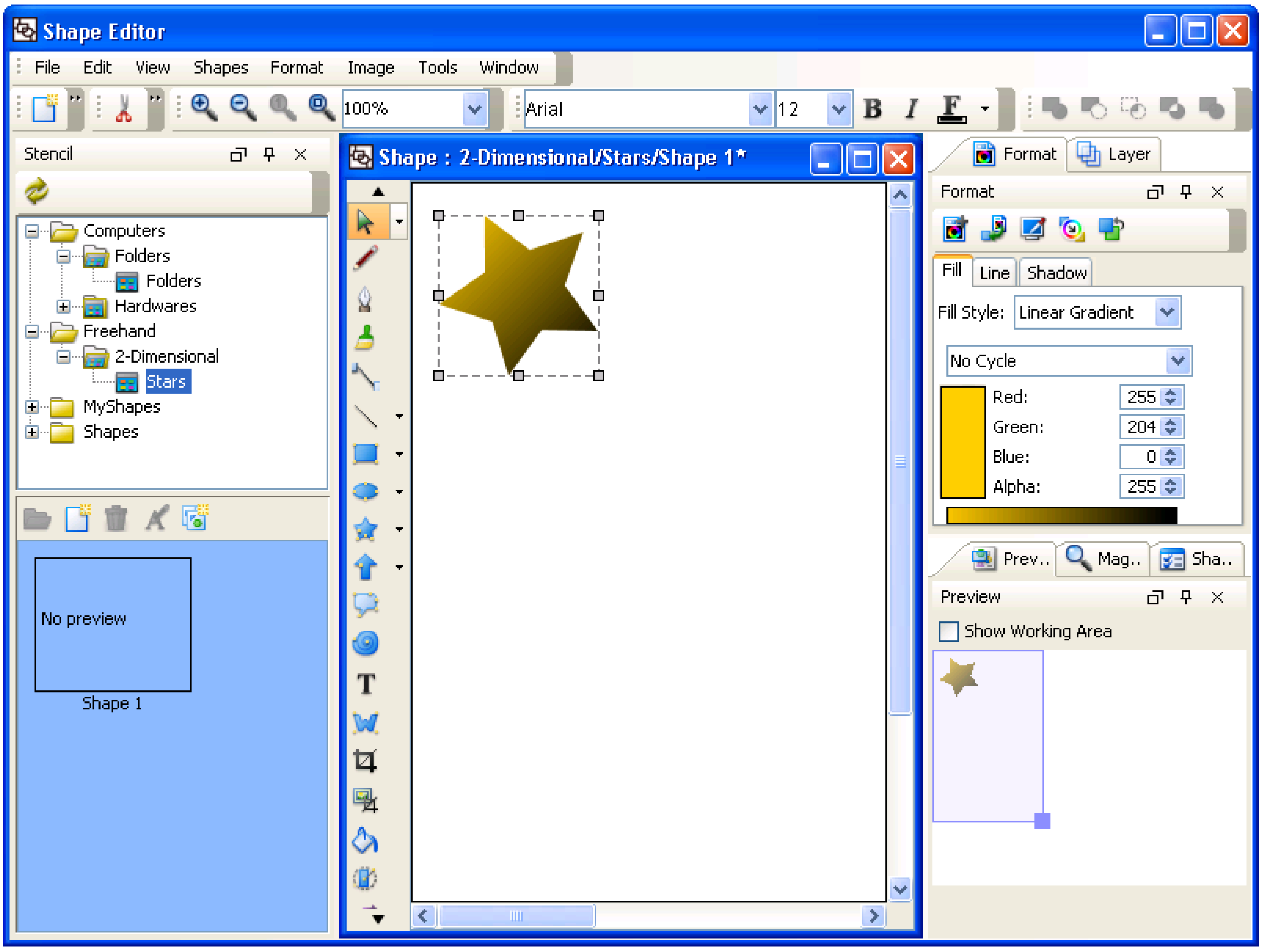


Figure 11-9 *To add a shape*

9. Draw the shape in the drawing pane.



Fill Style: Linear Gradient

No Cycle

Red: 255

Green: 204

Blue: 0

Alpha: 255

□ Show Working Area

Figure 11-10 *Drawing a shape in drawing pane*

10. Save the drawing by selecting **File > Save** in the main menu.

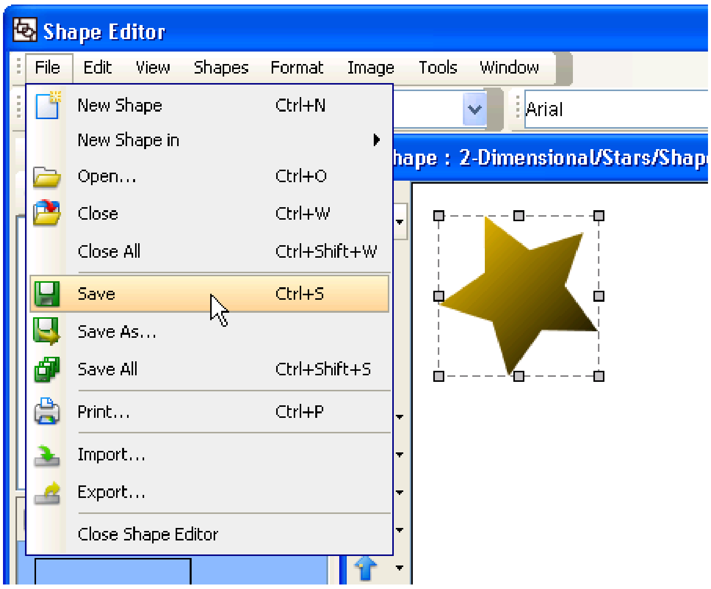


Figure 11-11 Saving a drawing

Shapes created in Shape Editor can be used in VP-UML. For details, please refer to the next section.

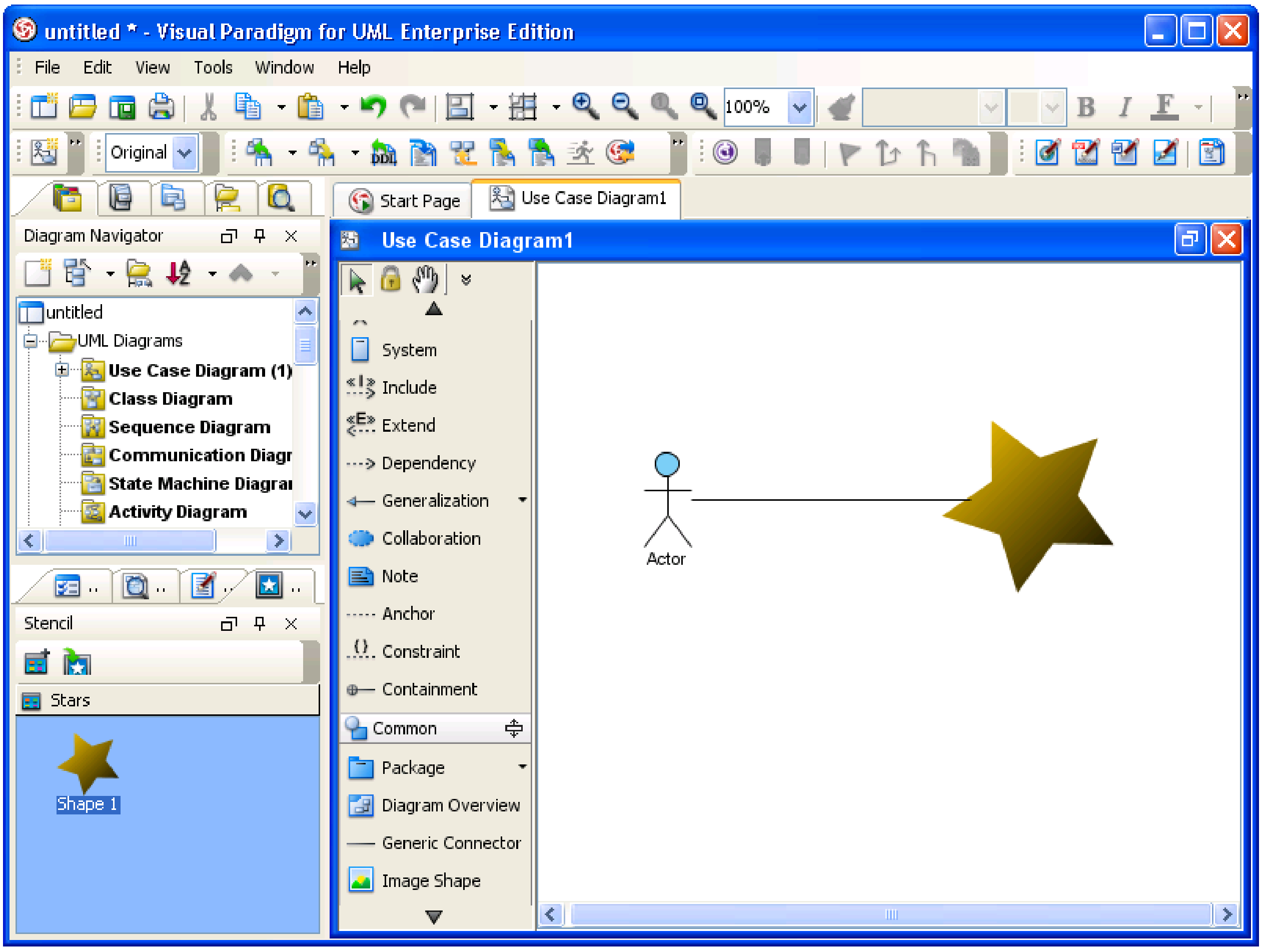


Figure 11-12 Apply a stencil shape in VP-UML

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. Open the **Stencil** pane. It is by default resided at the panes at the bottom left of VP-UML. If it does not appear, select **View > Panes > Stencil** from the main menu.
2. Click on the (Add Stencil) button in the top of **Stencil** pane. This popup a list of gallery. Select the stencil to add.

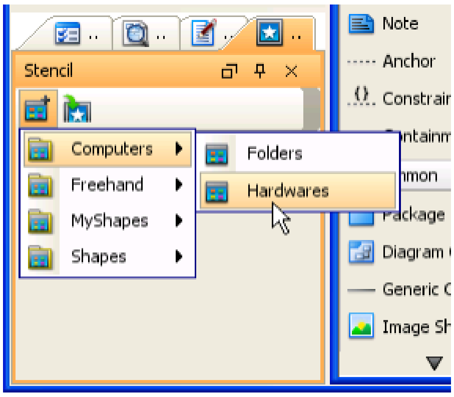


Figure 11-13 *To show a stencil*

The stencil is added to the **Stencil** pane.

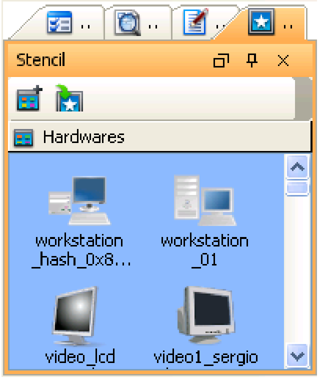


Figure 11-14 *Stencil with shapes appearing in Stencil pane*

NOTE: You can add multiple stencil by repeating this step.

3. Press on a shape in the **Stencil** pane and drag it out of the **Stencil** pane.

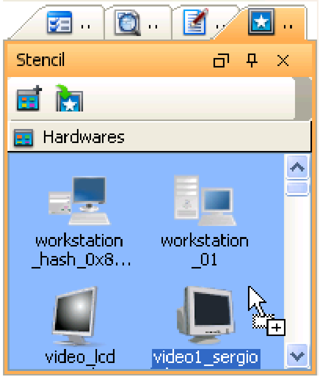


Figure 11-15 *Dragging shape out of Stencil pane*

4. Drop it onto the diagram to create the shape.

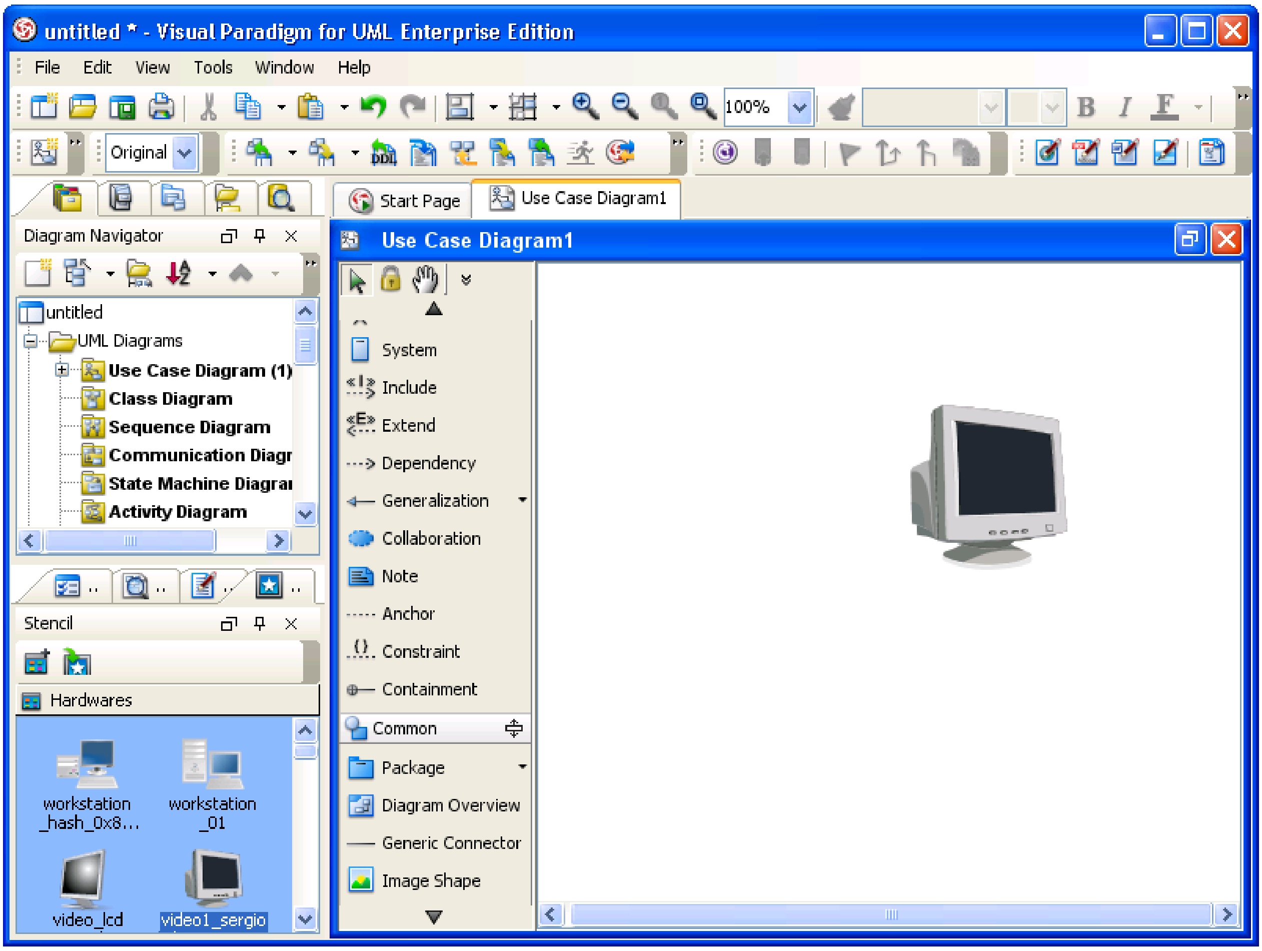


Figure 11-16 Stencil shape is created on diagram

5. You can also use generic connector to connect built-in notations shapes and stencil shapes. To do so, select **Generic Connector** in the diagram toolbar, under the **Common** category.

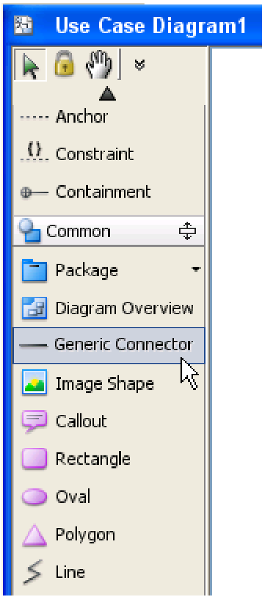


Figure 11-17 Selecting Generic Connector from the diagram toolbar

6. Press on the source shape, hold the mouse button, move the mouse cursor to the target shape and release the mouse button.

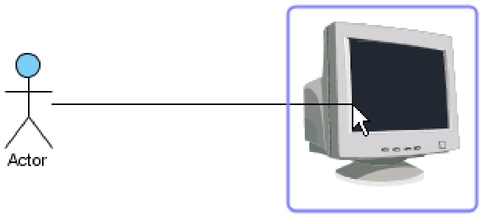


Figure 11-18 *Connecting an Actor with a stencil shape*

Connector is created.

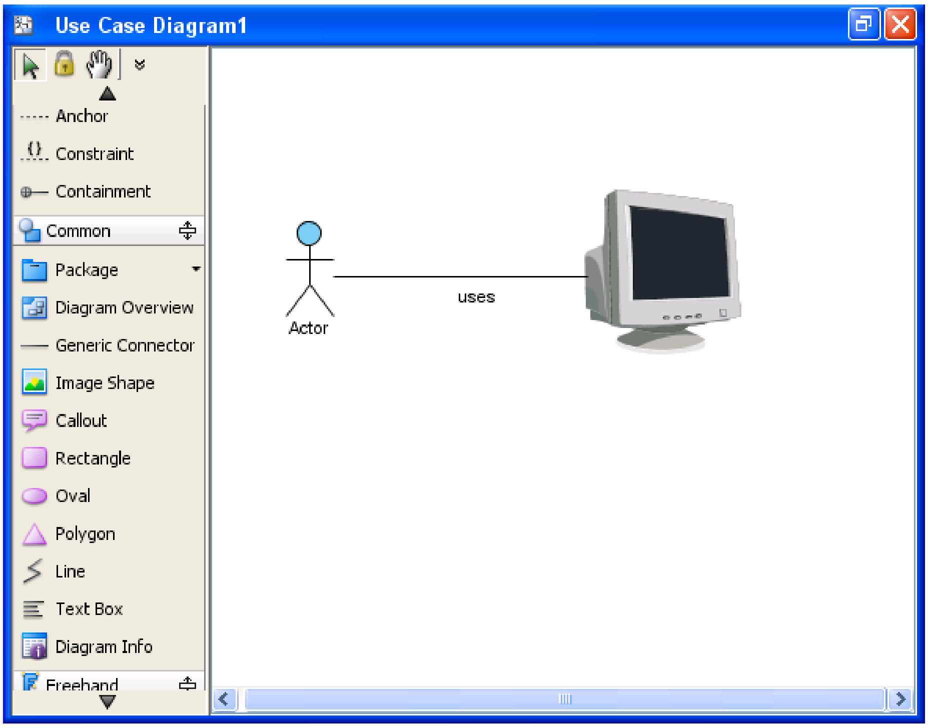
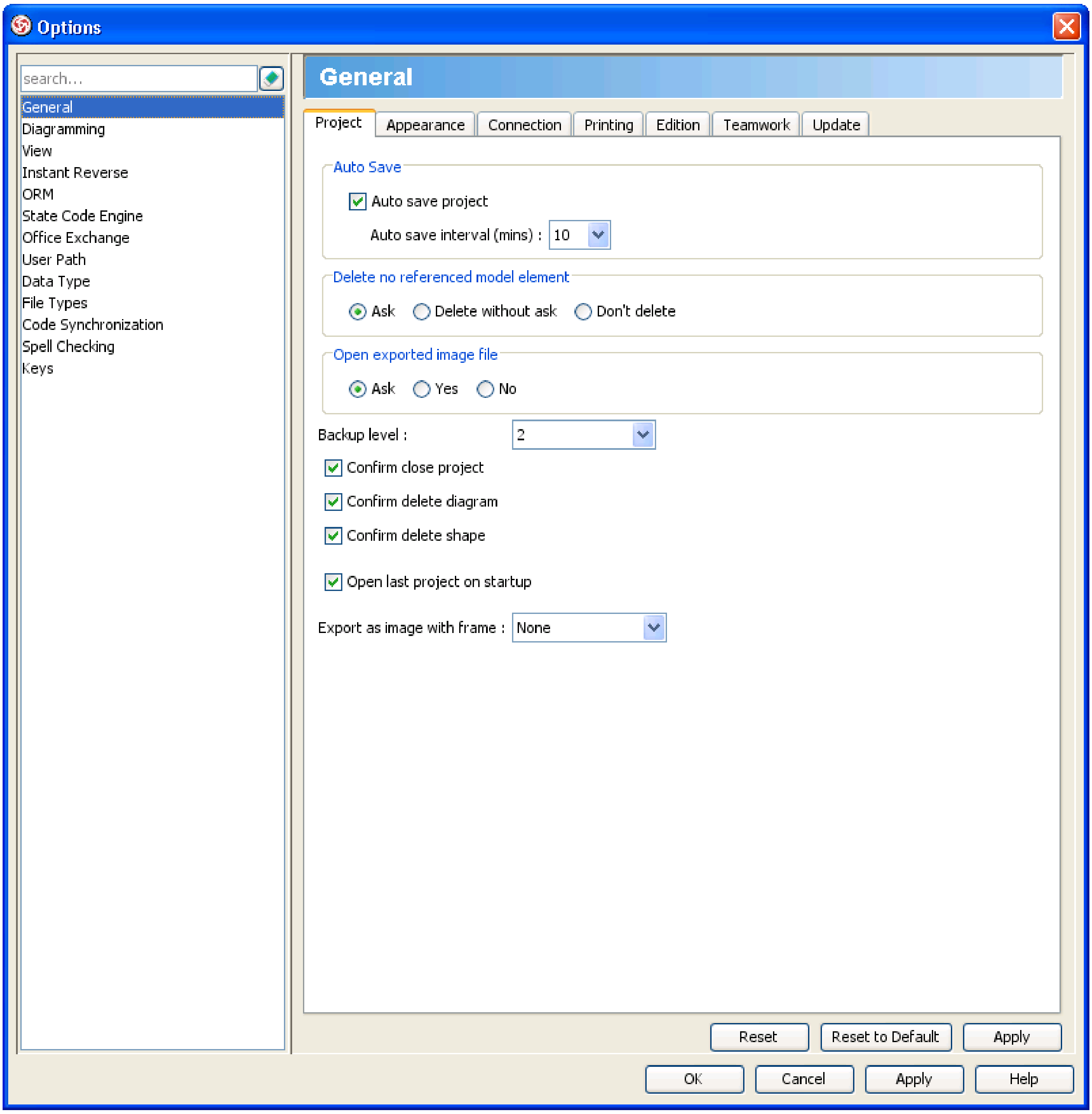


Figure 11-19 *Connector is created*

General Options



Auto Save

☑ Auto save project

Auto save interval (mins) : 10

Delete no referenced model element

● Ask ○ Delete without ask ○ Don't delete

Open exported image file

● Ask ○ Yes ○ No

Backup level : 2

☑ Confirm close project

☑ Confirm delete diagram

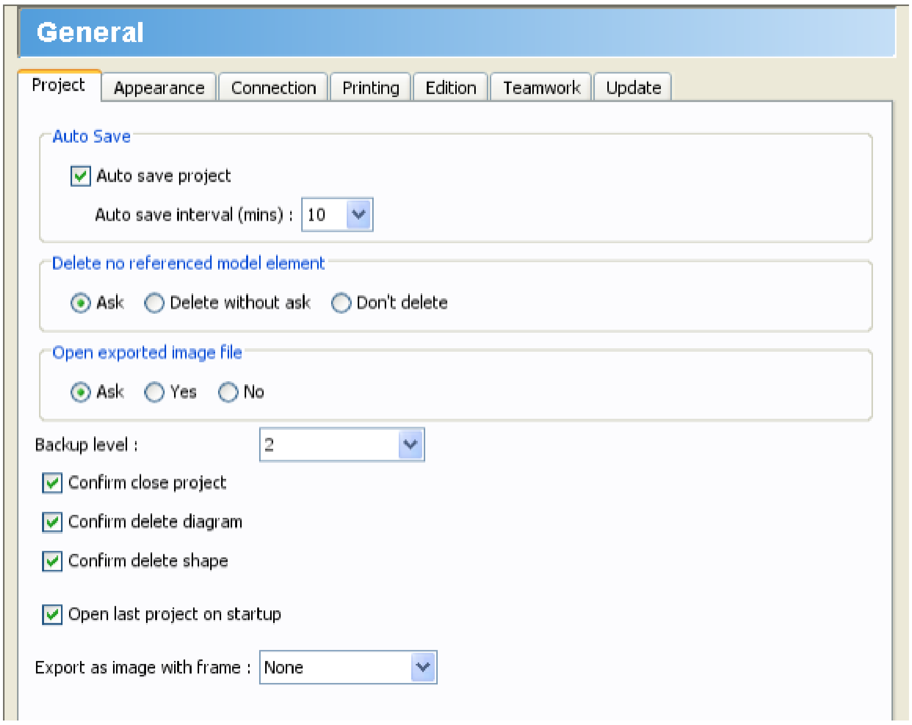
☑ Confirm delete shape

☑ Open last project on startup

Export as image with frame : None

Figure 2-1 *General Options in Option List*

Project



Auto Save

☑ Auto save project

Auto save interval (mins) : 10

Delete no referenced model element

● Ask ○ Delete without ask ○ Don't delete

Open exported image file

● Ask ○ Yes ○ No

Backup level : 2

☑ Confirm close project

☑ Confirm delete diagram

☑ Confirm delete shape

☑ Open last project on startup

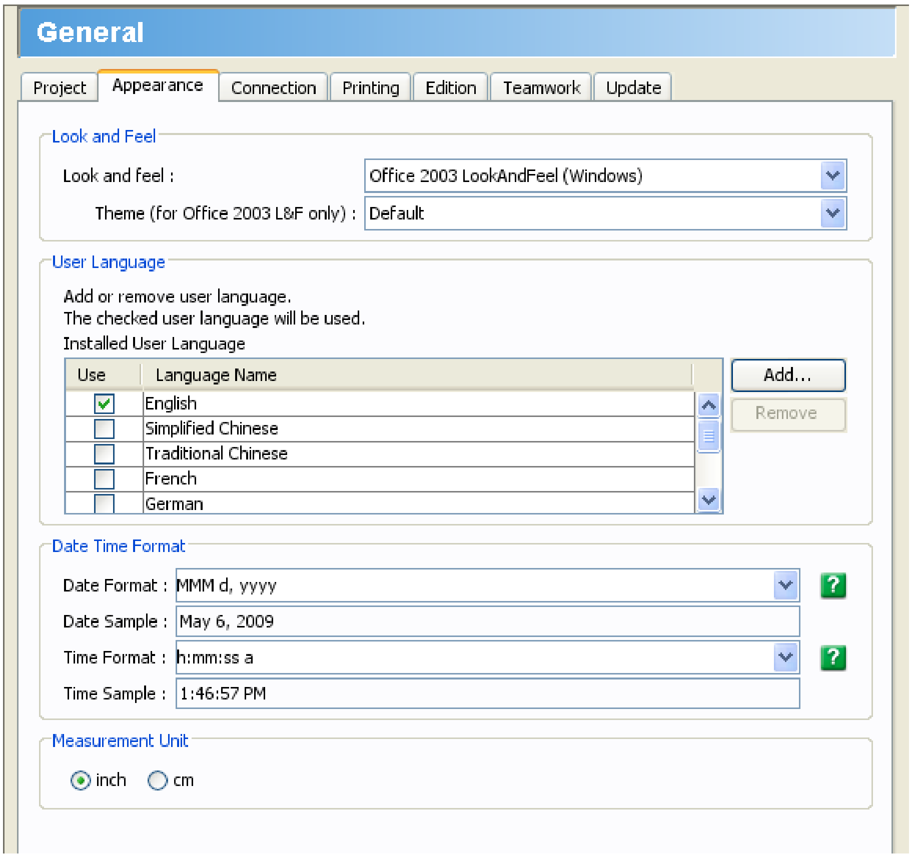
Export as image with frame : None

Figure 2-2 *Project Options in General*

Option Name	Description
Auto save project	Save the opening project as backup periodically
Auto save interval (mins)	Time needed to wait until the next auto saving
Delete no referenced model element	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 delete diagram	(default true) Prompt if you really want to close diagram
Confirm delete shape	(default true) Prompt if you really want to delete a shape on diagram
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

Table 2-1 *Project Options details*

Appearance



Look and Feel

Look and feel : Office 2003 LookAndFeel (Windows)

Theme (for Office 2003 L&F only) : Default

User Language

Add or remove user language.
The checked user language will be used.
Installed User Language

Use	Language Name
☑	English
□	Simplified Chinese
□	Traditional Chinese
□	French
□	German

Date Time Format

Date Format : MMM d, yyyy

Date Sample : May 6, 2009

Time Format : h:mm:ss a

Time Sample : 1:46:57 PM

Measurement Unit

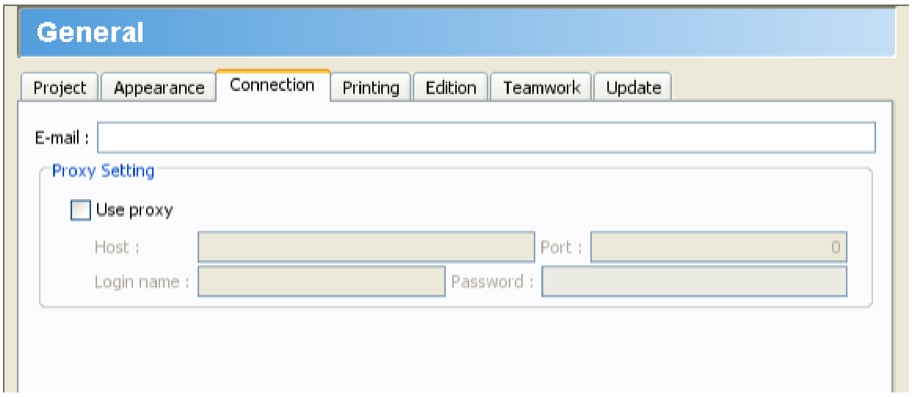
● inch ○ cm

Figure 2-3 *Appearence Options in General*

Option Name	Description
Look and feel	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.
Date Format	Format of date values that appear in the application
Time Format	Format of time values that appear in the application
Measurement Unit	• inch - (default) Set the measurement unit to be inch • cm - Set the measurement unit to be centimeters

Table 2-2 *Appearence Options details*

Connection



E-mail :

Proxy Setting

□ Use proxy

Host :

Port : 0

Login name :

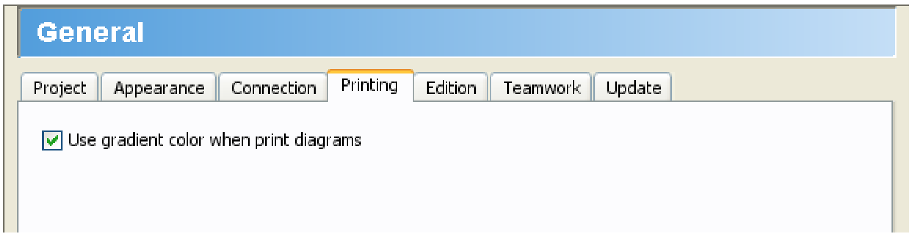
Password :

Figure 2-4 *Connection Options in General*

Option Name	Description
E-mail	Enter the Email field to specify your email address
Use proxy	(default false) To enable/disable the need of using 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).

Table 2-3 *Connection Options details*

Printing



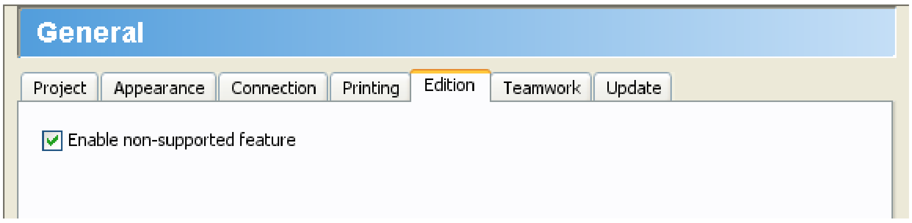
☑ Use gradient color when print diagrams

Figure 2-5 *Connection Options in General*

Option Name	Description
Use gradient color when print diagrams	Control whether to use gradient or solid color for shapes and diagrams when printing.

Table 2-4 *Printing Options details*

Edition



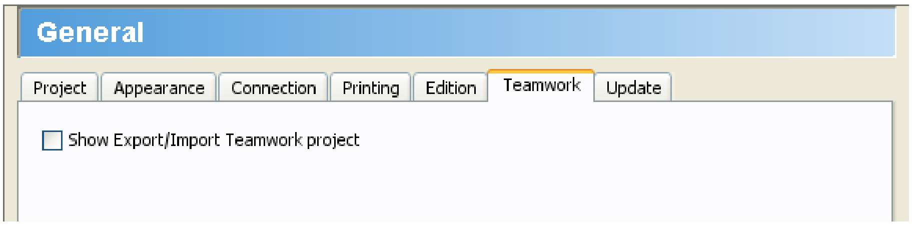
☑ Enable non-supported feature

Figure 2-6 *Edition Options in General*

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.

Table 2-5 *Printing Options details*

Teamwork



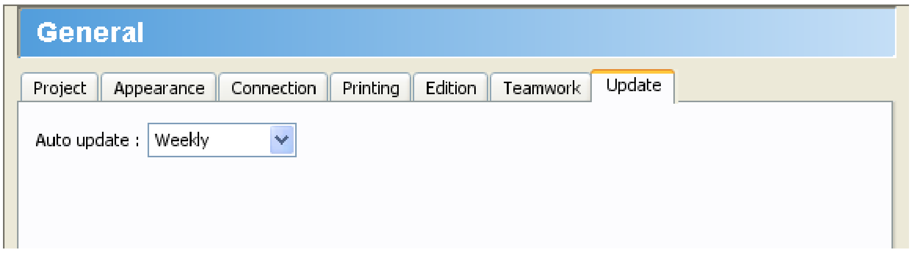
□ Show Export/Import Teamwork project

Figure 2-7 *Teamwork Options in General*

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

***Table 2-6** Teamwork Options details*

Update



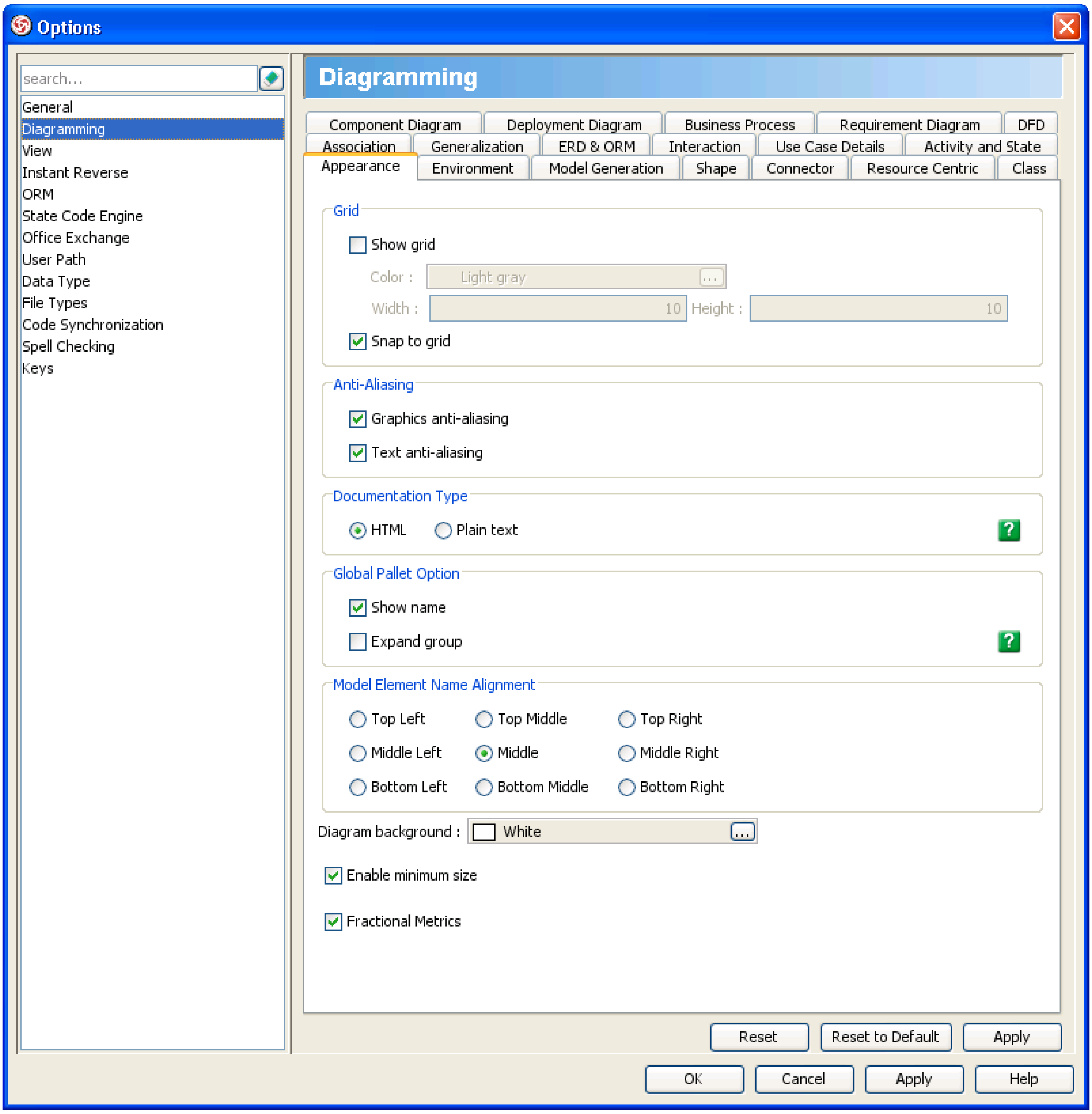
Auto update : Weekly

***Figure 2-8** Update Options in General*

Option Name	Description
Auto update	Auto Update lets VP-UML to check 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

***Table 2-7** Update Options details*

Diagramming Options



Grid

□ Show grid

Color : Light gray ...

Width : 10 Height : 10

☑ Snap to grid

Anti-Aliasing

☑ Graphics anti-aliasing

☑ Text anti-aliasing

Documentation Type

● HTML ○ Plain text

Global Pallet Option

☑ Show name

□ Expand group

Model Element Name Alignment

○ Top Left ○ Top Middle ○ Top Right

○ Middle Left ● Middle ○ Middle Right

○ Bottom Left ○ Bottom Middle ○ Bottom Right

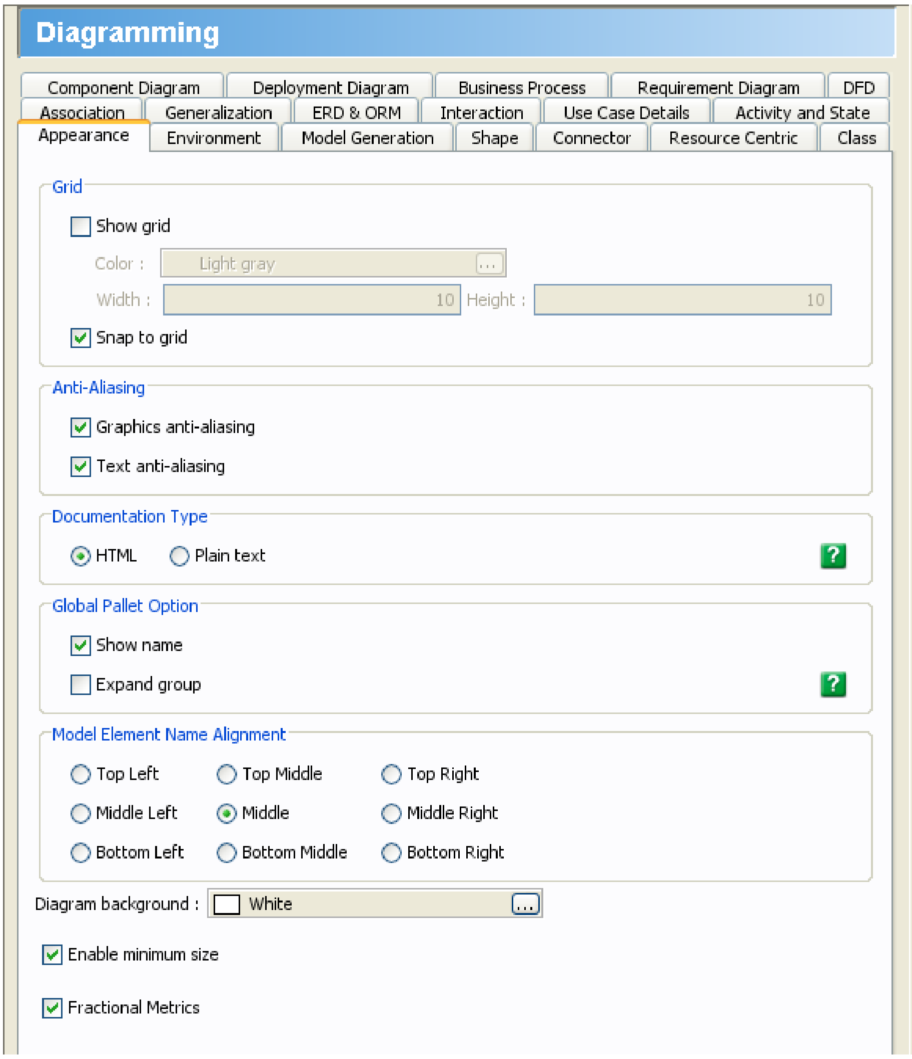
Diagram background : □ White ...

☑ Enable minimum size

☑ Fractional Metrics

Figure 2-9 Diagramming Options in Options List

Appearance



Grid

□ Show grid

Color : Light gray

Width : 10 Height : 10

☑ Snap to grid

Anti-Aliasing

☑ Graphics anti-aliasing

☑ Text anti-aliasing

Documentation Type

● HTML ○ Plain text

Global Pallet Option

☑ Show name

□ Expand group

Model Element Name Alignment

○ Top Left ○ Top Middle ○ Top Right

○ Middle Left ● Middle ○ Middle Right

○ Bottom Left ○ Bottom Middle ○ Bottom Right

Diagram background : □ White

☑ Enable minimum size

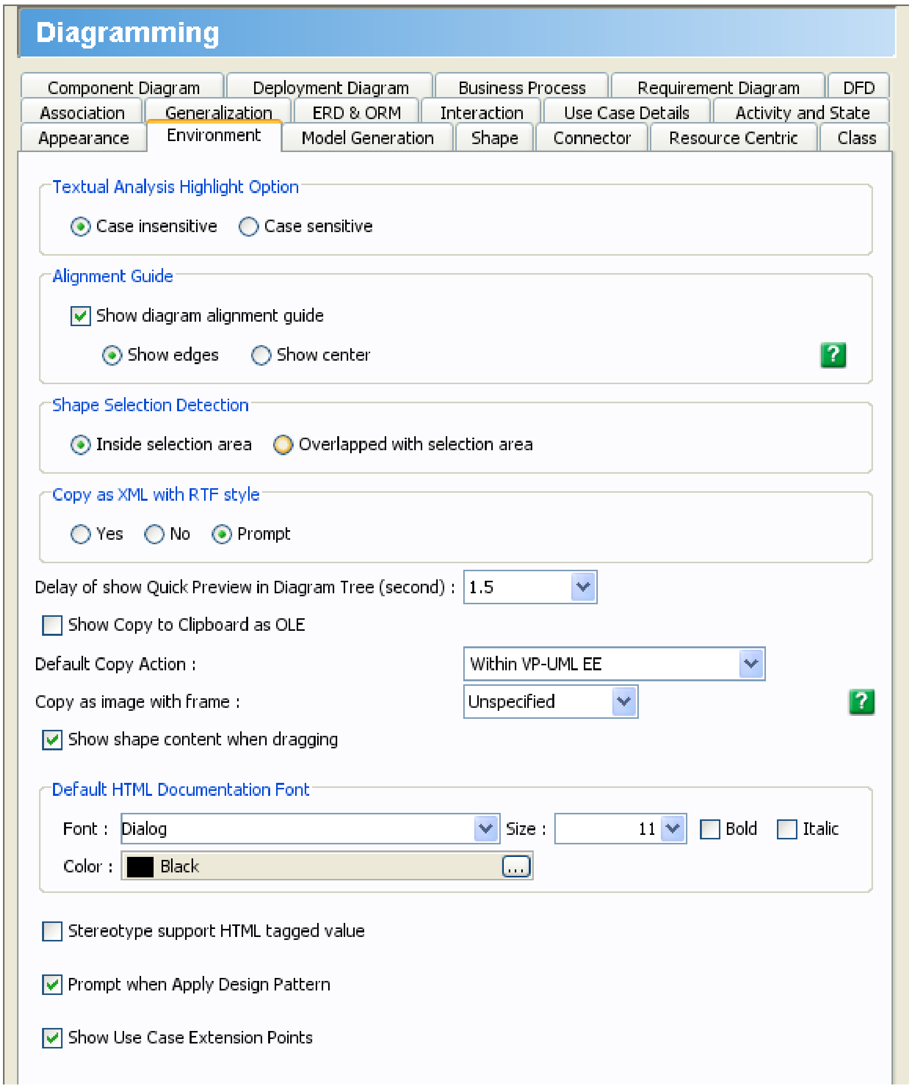
☑ Fractional Metrics

Figure 2-10 *Appearance Options in Diagramming*

Option Name	Description
Show grid	(default false) Show grid lines on diagram
Color	Color of grid lines
Width	Determines the horizontal spaces between grid lines
Height	Determines the vertical spaces between grid lines
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
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
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

Table 2-8 *Appearance Options details*

Environment



Textual Analysis Highlight Option

● Case insensitive ○ Case sensitive

Alignment Guide

☑ Show diagram alignment guide

● Show edges ○ Show center

Shape Selection Detection

● Inside selection area ○ Overlapped with selection area

Copy as XML with RTF style

○ Yes ○ No ● Prompt

Delay of show Quick Preview in Diagram Tree (second) : 1.5

□ Show Copy to Clipboard as OLE

Default Copy Action : Within VP-UML EE

Copy as image with frame : Unspecified

☑ Show shape content when dragging

Default HTML Documentation Font

Font : Dialog Size : 11 □ Bold □ Italic

Color : ■ Black

□ Stereotype support HTML tagged value

☑ Prompt when Apply Design Pattern

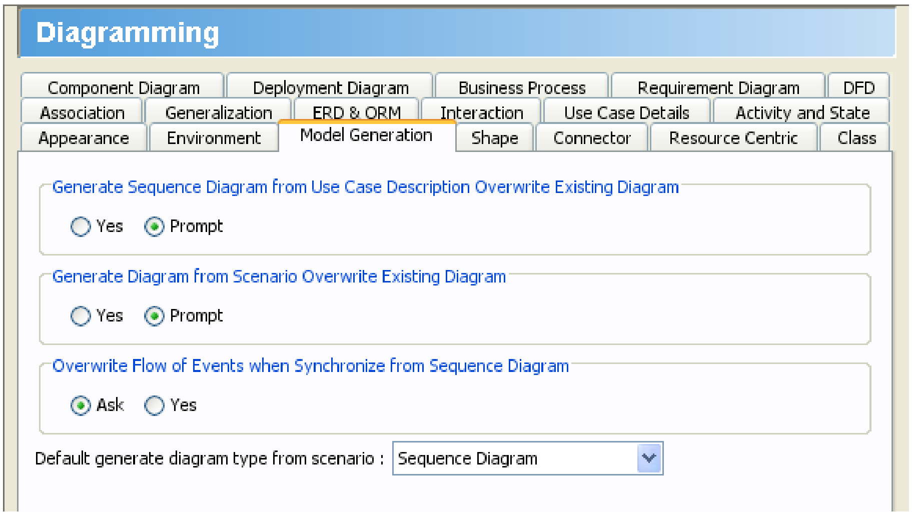
☑ Show Use Case Extension Points

Figure 2-11 *Environment Options in Diagramming*

Option Name	Description
Textual Analysis 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.
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 of show Quick Preview in Diagram Tree (second)	• 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	(default false) Determines whether the Copy to Clipboard as OLE menu is available or not
Default Copy Action	• 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
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
Prompt when Apply 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
Show Use Case Extension Points	(default true) Show Use Case Extension Points within Use Case shapes

Table 2-9 *Environment Options details*

Model Generation



Generate Sequence Diagram from Use Case Description Overwrite Existing Diagram

○ Yes ● Prompt

Generate Diagram from Scenario Overwrite Existing Diagram

○ Yes ● Prompt

Overwrite Flow of Events when Synchronize from Sequence Diagram

● Ask ○ Yes

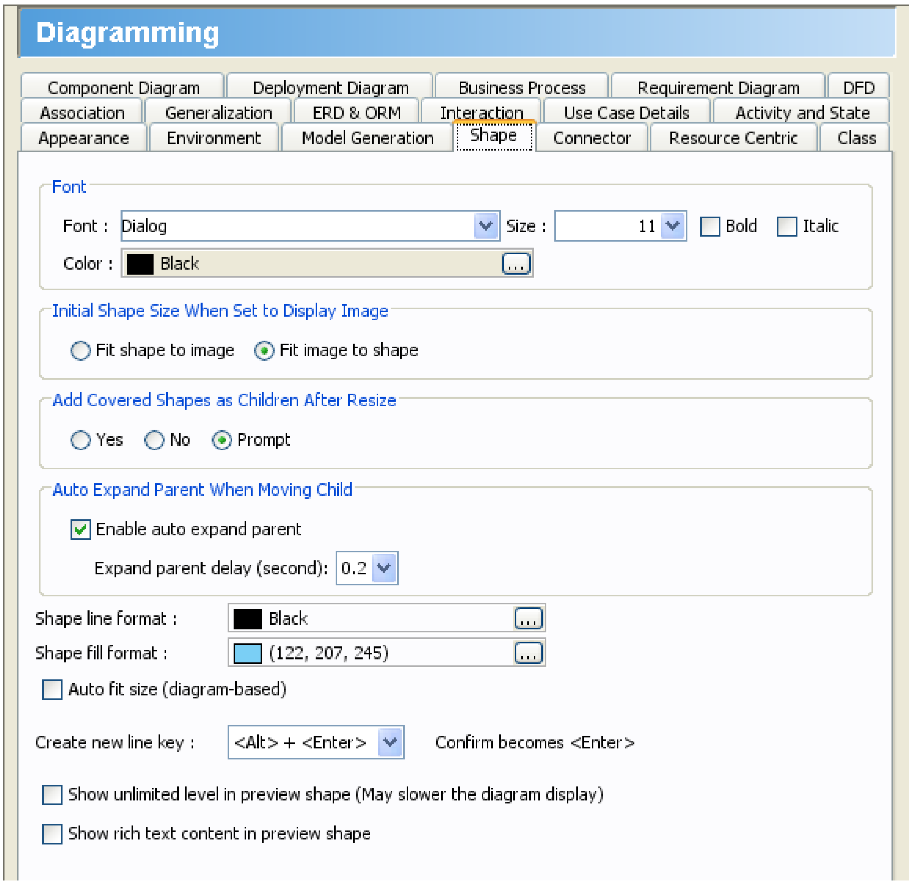
Default generate diagram type from scenario : Sequence Diagram

Figure 2-12 *Model Generation Options in Diagramming*

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
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

Table 2-10 *Model Generation Options details*

Shape



Font

Font : Dialog Size : 11 □ Bold □ Italic

Color : ■ Black

Initial Shape Size When Set to Display Image

○ Fit shape to image ● Fit image to shape

Add Covered Shapes as Children After Resize

○ Yes ○ No ● Prompt

Auto Expand Parent When Moving Child

☑ Enable auto expand parent

Expand parent delay (second): 0.2

Shape line format : ■ Black

Shape fill format : ■ (122, 207, 245)

□ Auto fit size (diagram-based)

Create new line key : <Alt> + <Enter> Confirm becomes <Enter>

□ Show unlimited level in preview shape (May slower the diagram display)

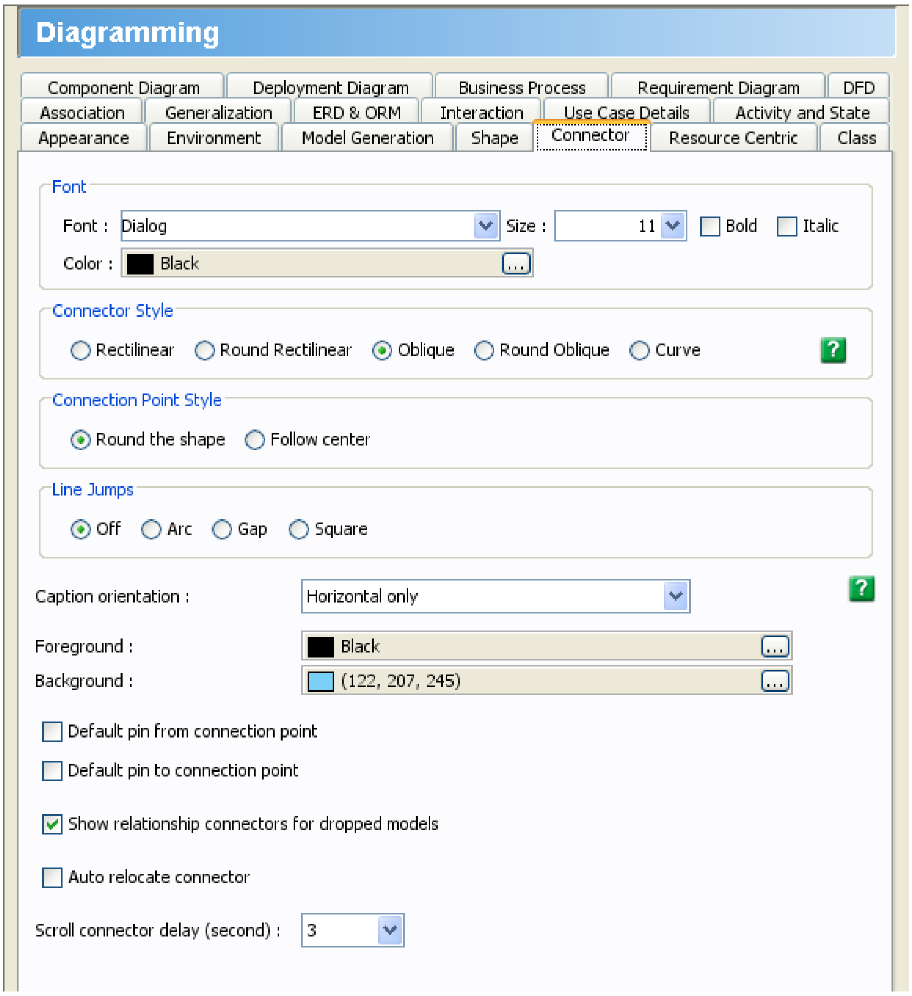
□ Show rich text content in preview shape

Figure 2-13 *Shape Options in Diagramming*

Option Name	Description
Font	Default font settings for shape content
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
Auto Expand Parent When Moving Child - Enable auto expand parent	(default true) Expand parent's width or height when a child shape is being moved around the edge of parent
Auto Expand Parent When Moving Child -Expand parent delay (second)	The time needed to response to child's movement around parent's edge
Shape line format	The default line format for shapes
Shape fill format	The default fill format for shapes
Auto fit size (diagram-based)	(default false) Determines whether shapes in diagrams will fit in size automatically
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

Table 2-11 *Shape Generation Options details*

Connector



Font

Font : Dialog Size : 11 □ Bold □ Italic

Color : ■ Black

Connector Style

○ Rectilinear ○ Round Rectilinear ● Oblique ○ Round Oblique ○ Curve

Connection Point Style

● Round the shape ○ Follow center

Line Jumps

● Off ○ Arc ○ Gap ○ Square

Caption orientation : Horizontal only

Foreground : ■ Black

Background : (122, 207, 245)

□ Default pin from connection point

□ Default pin to connection point

☑ Show relationship connectors for dropped models

□ Auto relocate connector

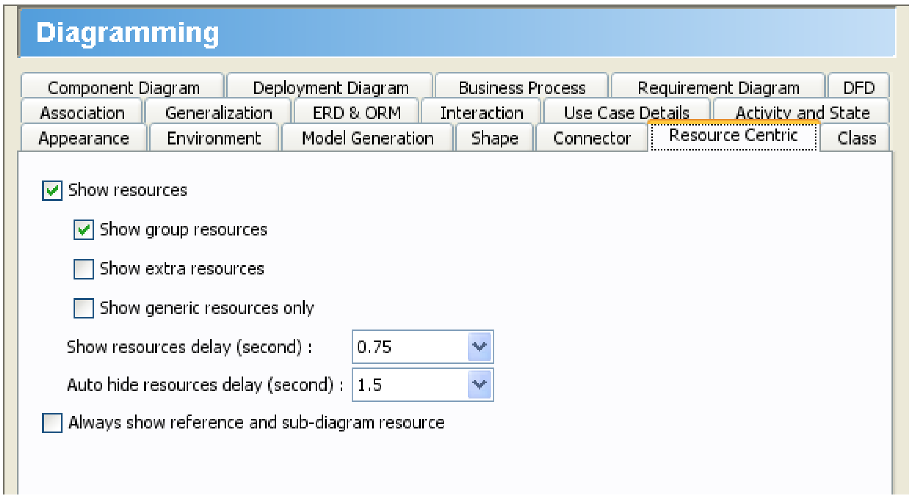
Scroll connector delay (second) : 3

Figure 2-14 *Connector Options in Diagramming*

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
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
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	(default false) Auto relocate connector when being overlapped by 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

Table 2-12 *Connector Generation Options details*

Resource Centric



☑ Show resources

☑ Show group resources

□ Show extra resources

□ Show generic resources only

Show resources delay (second) : 0.75

Auto hide resources delay (second) : 1.5

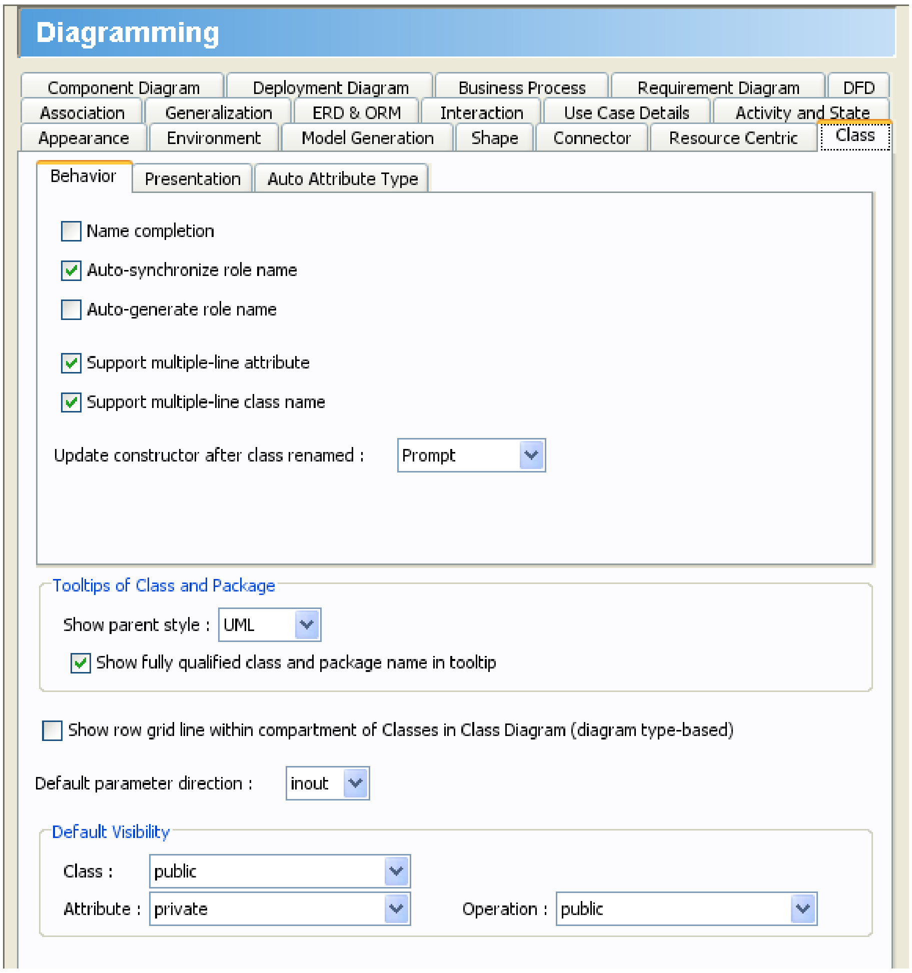
□ Always show reference and sub-diagram resource

Figure 2-15 *Resource Centric Options in Diagramming*

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 reference and sub-diagram resource	(default true) Always show the reference and subdiagram resource icon at the bottom of shape no matter whether the shape has reference and sub-diagram added.

Table 2-13 *Resource Centric Generation Options details*

Class



□ Name completion

☑ Auto-synchronize role name

□ Auto-generate role name

☑ Support multiple-line attribute

☑ Support multiple-line class name

Update constructor after class renamed : Prompt

Tooltips of Class and Package

Show parent style : UML

☑ Show fully qualified class and package name in tooltip

□ Show row grid line within compartment of Classes in Class Diagram (diagram type-based)

Default parameter direction : inout

Default Visibility

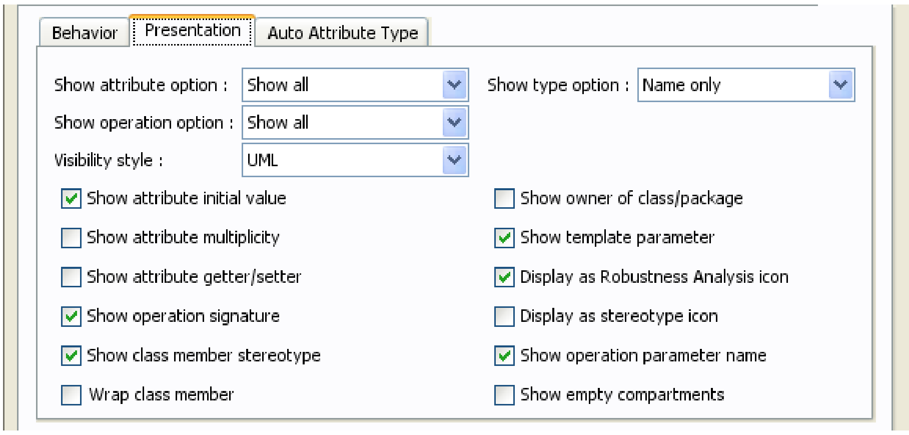
Class : public

Attribute : private

Operation : public

Figure 2-16 *Class Options in Diagramming*

Option Name	Description
Name completion	(defaultGeneralizaion true) When creating a class, you can select an existing class from a list so that a view of the selected class can be created. You can enable or disable the popup of such list.
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)
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*.
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



Show attribute option : Show all

Show type option : Name only

Show operation option : Show all

Visibility style : UML

☑ Show attribute initial value

□ Show owner of class/package

□ Show attribute multiplicity

☑ Show template parameter

□ Show attribute getter/setter

☑ Display as Robustness Analysis icon

☑ Show operation signature

□ Display as stereotype icon

☑ Show class member stereotype

☑ Show operation parameter name

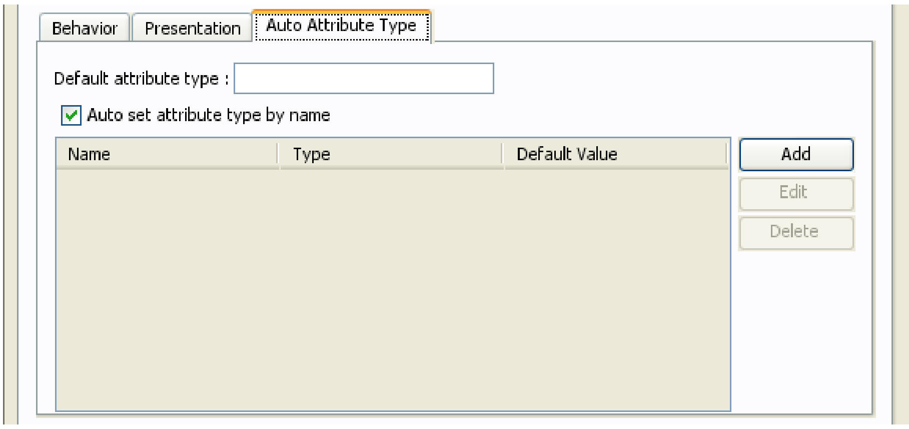
□ Wrap class member

□ Show empty compartments

Figure 2-17 *Presentation of Class Options in Diagramming*

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

Table 2-15 *Presentation of Class Options details*



Default attribute type :

☑ Auto set attribute type by name

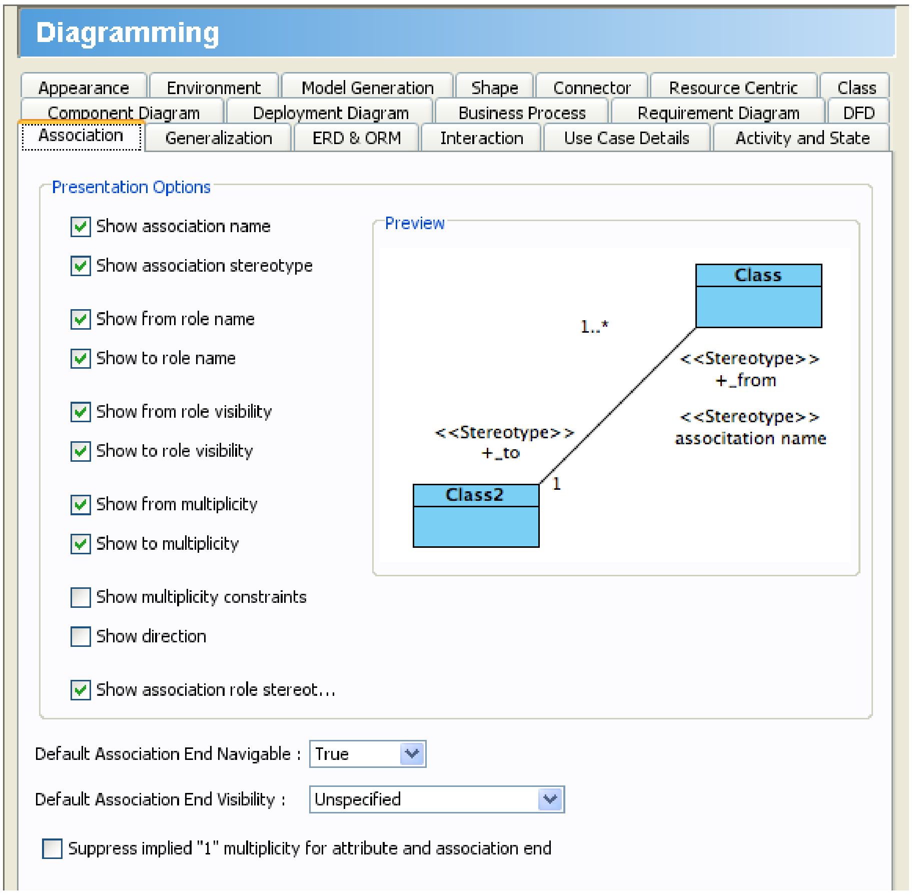
Name	Type	Default Value

Figure 2-18 *Auto Attribute Type of Class Options in Diagramming*

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.

Table 2-16 *Auto Attribute Type of Class Options details*

Association



☑ Show association name

☑ Show association stereotype

☑ Show from role name

☑ Show to role name

☑ Show from role visibility

☑ Show to role visibility

☑ Show from multiplicity

☑ Show to multiplicity

□ Show multiplicity constraints

□ Show direction

☑ Show association role stereot...

Default Association End Navigable : True

Default Association End Visibility : Unspecified

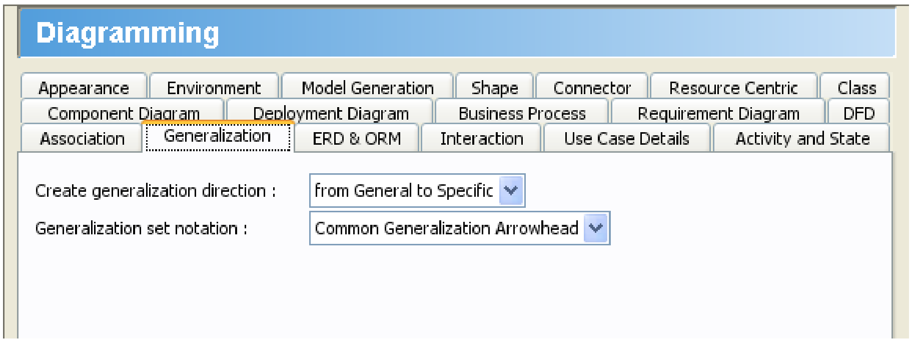
□ Suppress implied "1" multiplicity for attribute and association end

Figure 2-19 *Association Options in Diagramming*

Option Name	Description
Show association name	(default true) Show the name of association
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

Table 2-17 *Association Options details*

Generalization



Create generalization direction : from General to Specific

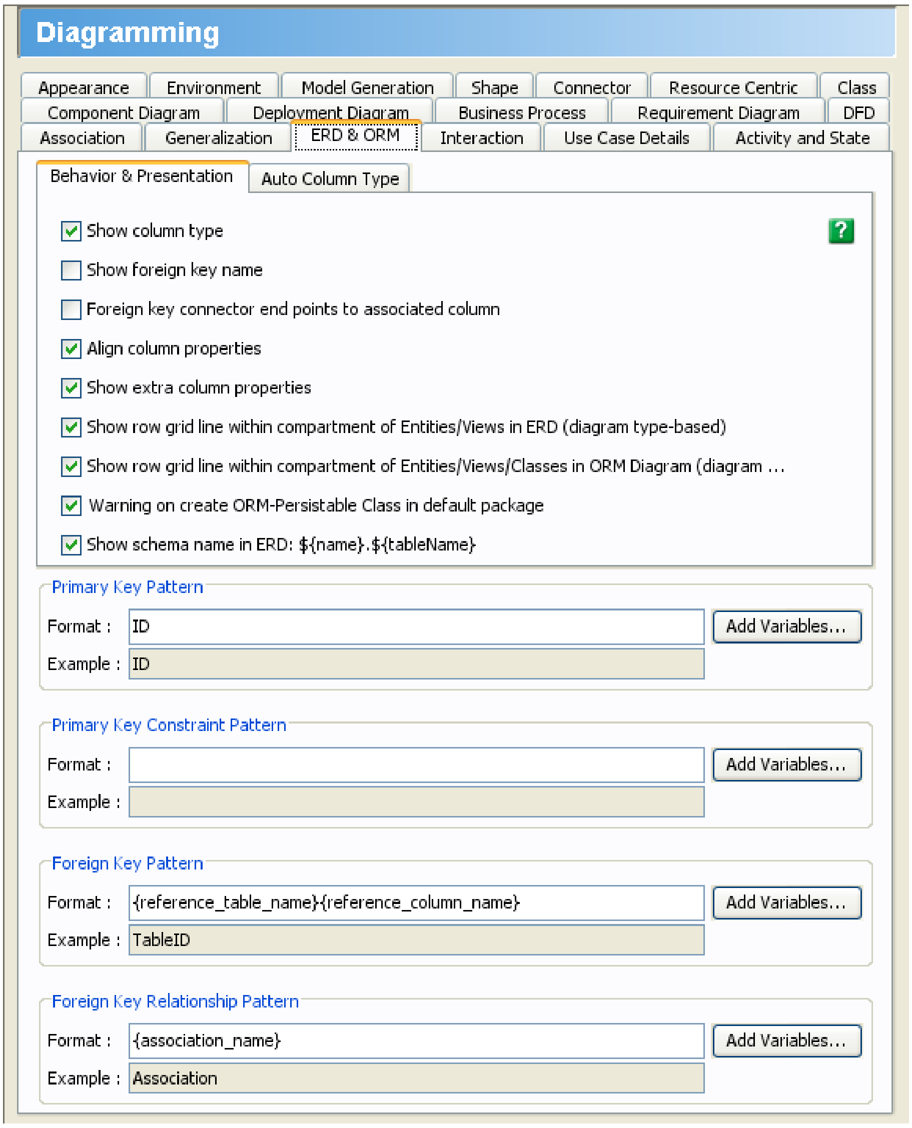
Generalization set notation : Common Generalization Arrowhead

Figure 2-20 *Generalization Options in Diagramming*

Option Name	Description
Create generalization direction	• 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 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

Table 2-18 *Generalization Options details*

ERD & ORM



☑ Show column type

☐ Show foreign key name

☐ Foreign key connector end points to associated column

☑ Align column properties

☑ Show extra column properties

☑ Show row grid line within compartment of Entities/Views in ERD (diagram type-based)

☑ Show row grid line within compartment of Entities/Views/Classes in ORM Diagram (diagram ...

☑ Warning on create ORM-Persistable Class in default package

☑ Show schema name in ERD: ${name}.${tableName}

Primary Key Pattern

Format : ID

Example : ID

Primary Key Constraint Pattern

Format :

Example :

Foreign Key Pattern

Format : {reference_table_name}{reference_column_name}

Example : TableID

Foreign Key Relationship Pattern

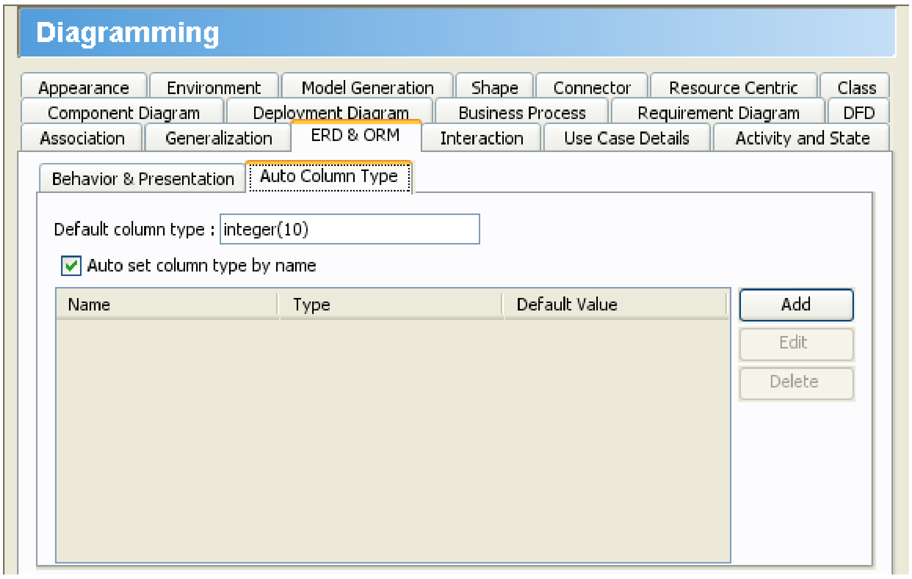
Format : {association_name}

Example : Association

Figure 2-21 *ERD & ORM Options in Diagramming*

Option Name	Description
Show column type	(default true) Show data type of column
Show foreign key name	(default false) Show
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
Primary Key Pattern Format	Pattern of primary keys that will be applied when synchronizing Class Diagram to Entity Relationship Diagram, which may create primary key
Primary Key Constraint Pattern Format	Pattern of primary key constraint that will be applied when creating Entity
Foreign Key Pattern Format	Pattern of foreign key that will be applied when creating a primary key on an entity which result in creating foreign key in connected entities
Foreign Key Relationship Pattern Format	Pattern of foreign key relationship that will be applied when creating a primary key on an entity which result in creating foreign key in connected entities

Table 2-19 *ERD & ORM Options details*



Default column type : integer(10)

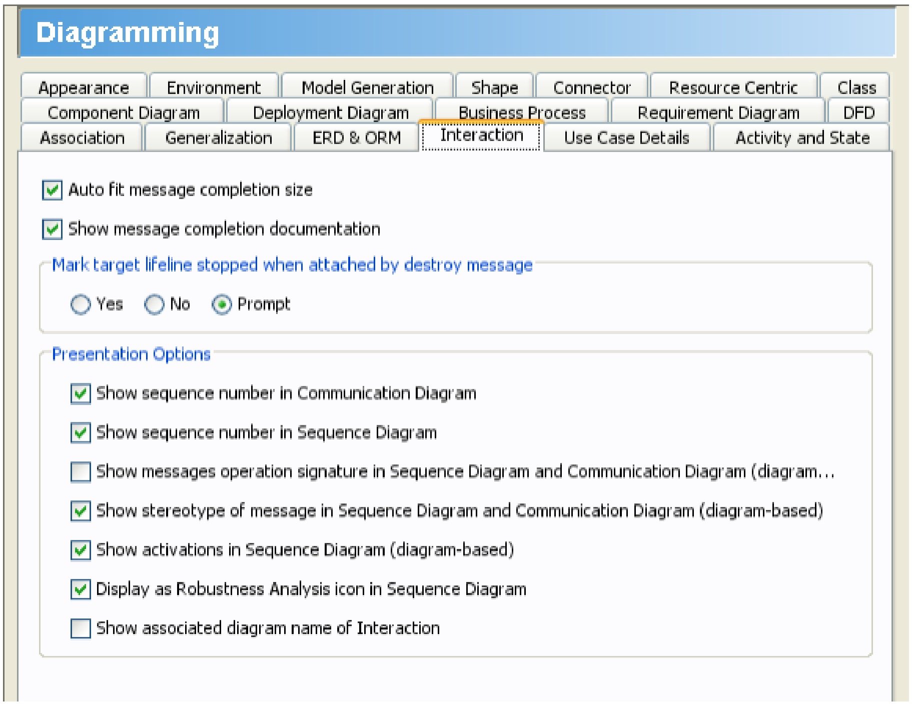
☑ Auto set column type by name

Figure 2-22 *Auto Column Type of ERD & ORM Options in Diagramming*

Option Name	Description
Default column type	Define column type that will be applied to newly created columns
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.

Table 2-20 *Auto Column Type of ERD & ORM Options details*

Interaction



☑ Auto fit message completion size

☑ Show message completion documentation

Mark target lifeline stopped when attached by destroy message

○ Yes ○ No ● Prompt

Presentation Options

☑ Show sequence number in Communication Diagram

☑ Show sequence number in Sequence Diagram

□ Show messages operation signature in Sequence Diagram and Communication Diagram (diagram...

☑ Show stereotype of message in Sequence Diagram and Communication Diagram (diagram-based)

☑ Show activations in Sequence Diagram (diagram-based)

☑ Display as Robustness Analysis icon in Sequence Diagram

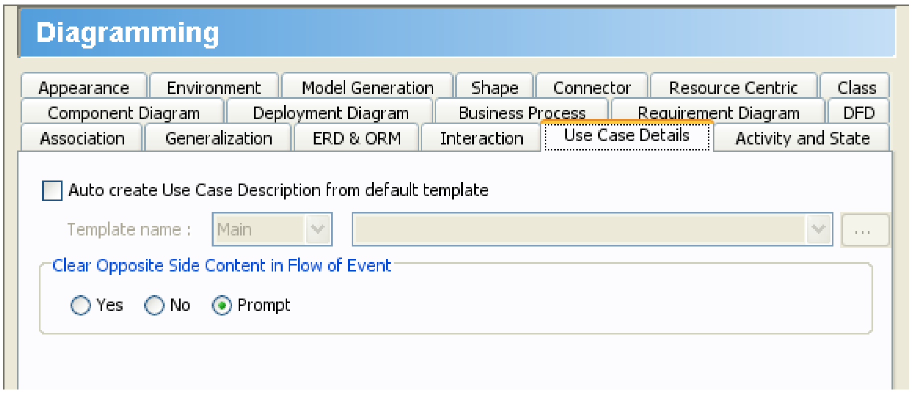
□ Show associated diagram name of Interaction

Figure 2-23 *Interaction Options in Diagramming*

Option Name	Description
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 message completion documentation	(default true) When selecting an Operation in the message completion box, the documentation of operation will appear next to the completion box
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)

Table 2-21 *Interaction Options details*

Use Case Details



□ Auto create Use Case Description from default template

Template name : Main

Clear Opposite Side Content in Flow of Event

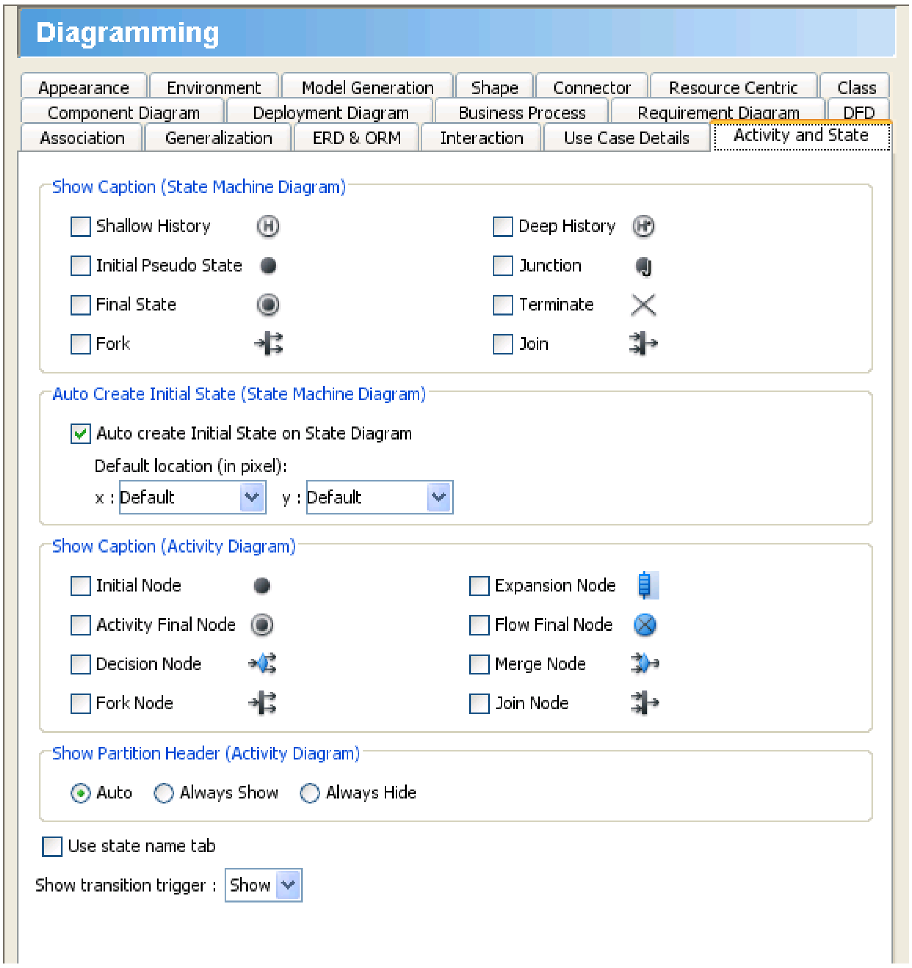
○ Yes ○ No ● Prompt

Figure 2-24 *Use Case Details Options in Diagramming*

Option Name	Description
Auto create Use Case Description from default template	(default false) When opening Use Case Description for a Use Case, a Use Case Description will be created automatically using the chosen template. You can select a template from the drop down menu under the checkbox, or select Other in the drop down menu and import the template from file.
Template name	• Main - (default) Main template will be used as Use Case Description default template • Alternative - Alternative template will be used as Use Case Description default template • Basic - Basic template will be used as Use Case Description default template • Full - Full template will be used as Use Case Description default template • Scenario - Scenario template will be used as Use Case Description default template • Other - Select Other to enable the drop down menu on the right, which is made for selecting a template file to be used for default template
Clear Opposite Side Content in Flow of Event	For every row of a flow of events table, either the Actor or the System side should be filled, but not both. This option is for controlling the effect when you try to enter content for both Actor and System cell. • Yes - Automatically clear the other side's cell content • No - Keep the other side's content • Prompt - (default) Ask for whether to clear or keep the other side's content

***Table 2-22** Use Case Details Options details*

Activity and State



Show Caption (State Machine Diagram)

□ Shallow History □ Deep History

□ Initial Pseudo State □ Junction

□ Final State □ Terminate

□ Fork □ Join

Auto Create Initial State (State Machine Diagram)

☑ Auto create Initial State on State Diagram

Default location (in pixel):

x : Default y : Default

Show Caption (Activity Diagram)

□ Initial Node □ Expansion Node

□ Activity Final Node □ Flow Final Node

□ Decision Node □ Merge Node

□ Fork Node □ Join Node

Show Partition Header (Activity Diagram)

● Auto ○ Always Show ○ Always Hide

□ Use state name tab

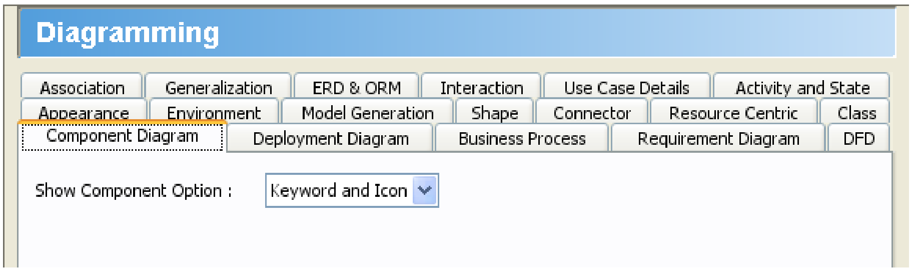
Show transition trigger : Show

***Figure 2-25** Activity and State Options in Diagramming*

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
Show Caption (Activity Diagram)	• 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 (Activity Diagram)	• 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
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

***Table 2-23** Activity and State Options details*

Component Diagram



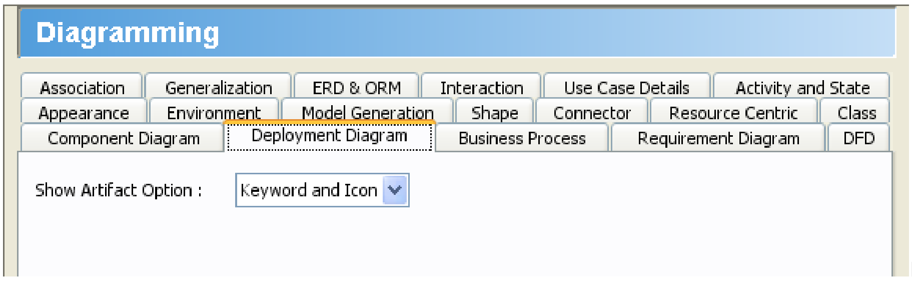
Show Component Option : Keyword and Icon

***Figure 2-26** Component Diagram Options in Diagramming*

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

***Table 2-24** Component Diagram Options details*

Deployment Diagram



Show Artifact Option : Keyword and Icon

Messag

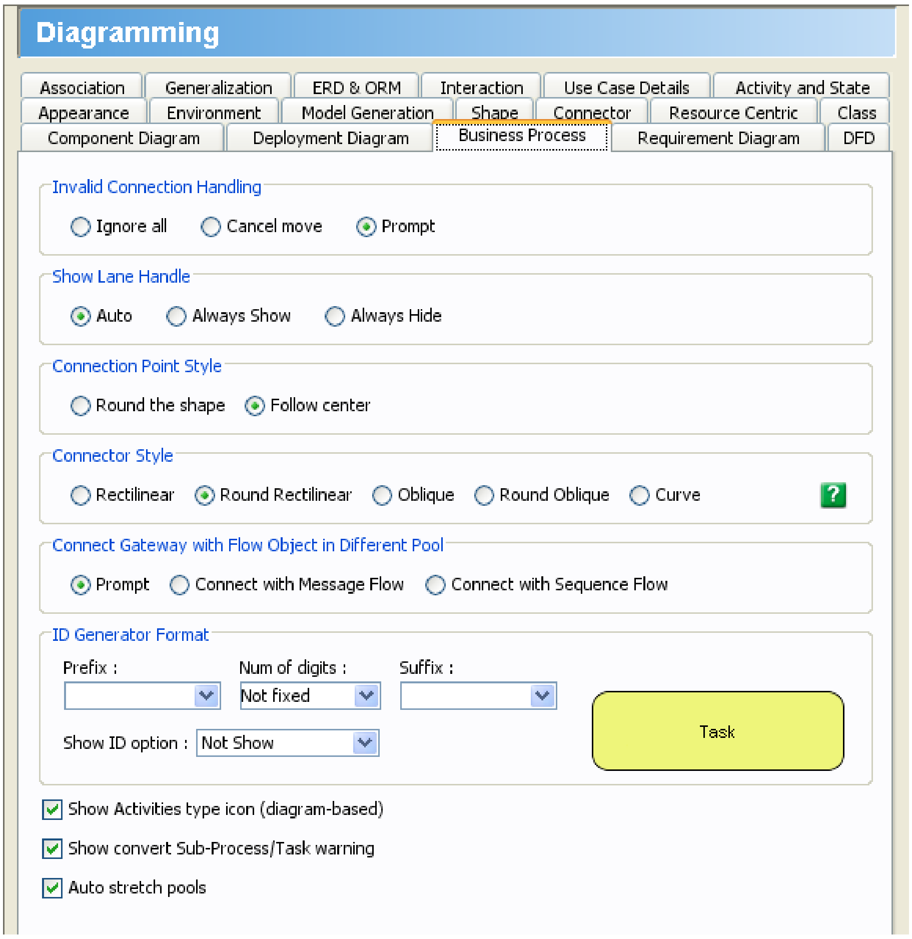
Flo

Figure 2-27 *Component Diagram Options in Diagramming*

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

Table 2-25 *Component Diagram Options details*

Business Process



Invalid Connection Handling

○ Ignore all ○ Cancel move ● Prompt

Show Lane Handle

● Auto ○ Always Show ○ Always Hide

Connection Point Style

○ Round the shape ● Follow center

Connector Style

○ Rectilinear ● Round Rectilinear ○ Oblique ○ Round Oblique ○ Curve ?

Connect Gateway with Flow Object in Different Pool

● Prompt ○ Connect with Message Flow ○ Connect with Sequence Flow

ID Generator Format

Prefix :

Num of digits : Not fixed

Suffix :

Show ID option : Not Show

☑ Show Activities type icon (diagram-based)

☑ Show convert Sub-Process/Task warning

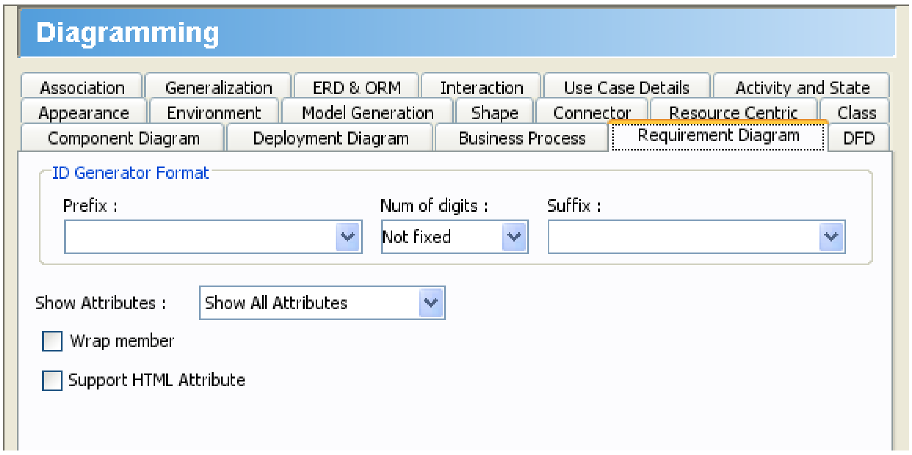
☑ Auto stretch pools

Figure 2-28 *Business Process Options in Diagramming*

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
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
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
ID Generator Format Prefix	Prefix of ID that will be automatically generated when creating BPMN shapes
ID Generator Format Num of digits	The number of digits of ID that will be automatically generated when creating BPMN shapes
ID Generator Format Suffix	Suffix of ID that will be automatically generated when creating BPMN shapes
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 Activities type icon (diagram-based)	(default true) Show icons that represent the type of Task and Sub-Process
Show convert Sub-Process/Task warning	(default true) Show warning when trying to convert between Sub-Process and Task
Auto stretch pools	(default true) Stretch Pools automatically to the reach diagram bound

Table 2-26 *Business Process Options details*

Requirement Diagram



ID Generator Format

Prefix :

Num of digits : Not fixed

Suffix :

Show Attributes : Show All Attributes

□ Wrap member

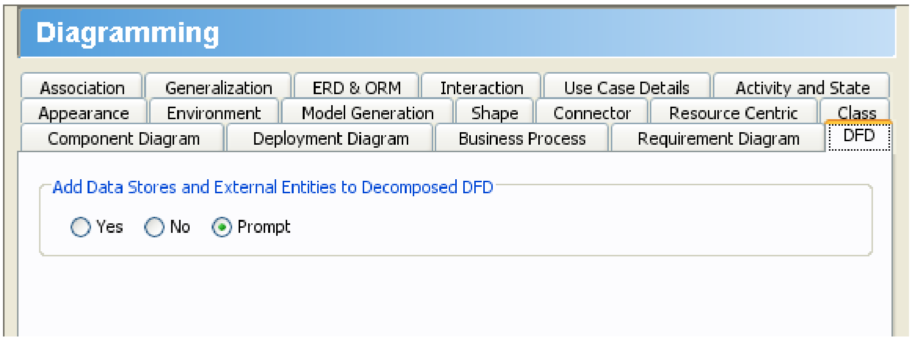
□ Support HTML Attribute

Figure 2-29 *Requirement Diagram Options in Diagramming*

Option Name	Description
ID Generator Format Prefix	Prefix of Requirement ID that will be automatically generated when creating Requirement
ID Generator Format Num of digits	The number of digits of Requirement ID that will be automatically generated when creating Requirement
ID Generator Format Suffix	Suffix of Requirement ID that will be automatically generated when creating Requirement
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

Table 2-27 *Requirement Diagram Options details*

DFD



Add Data Stores and External Entities to Decomposed DFD

○ Yes ○ No ● Prompt

Figure 2-30 *DFD Options in Diagramming*

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

Table 2-28 *DFD Options details*

View Options

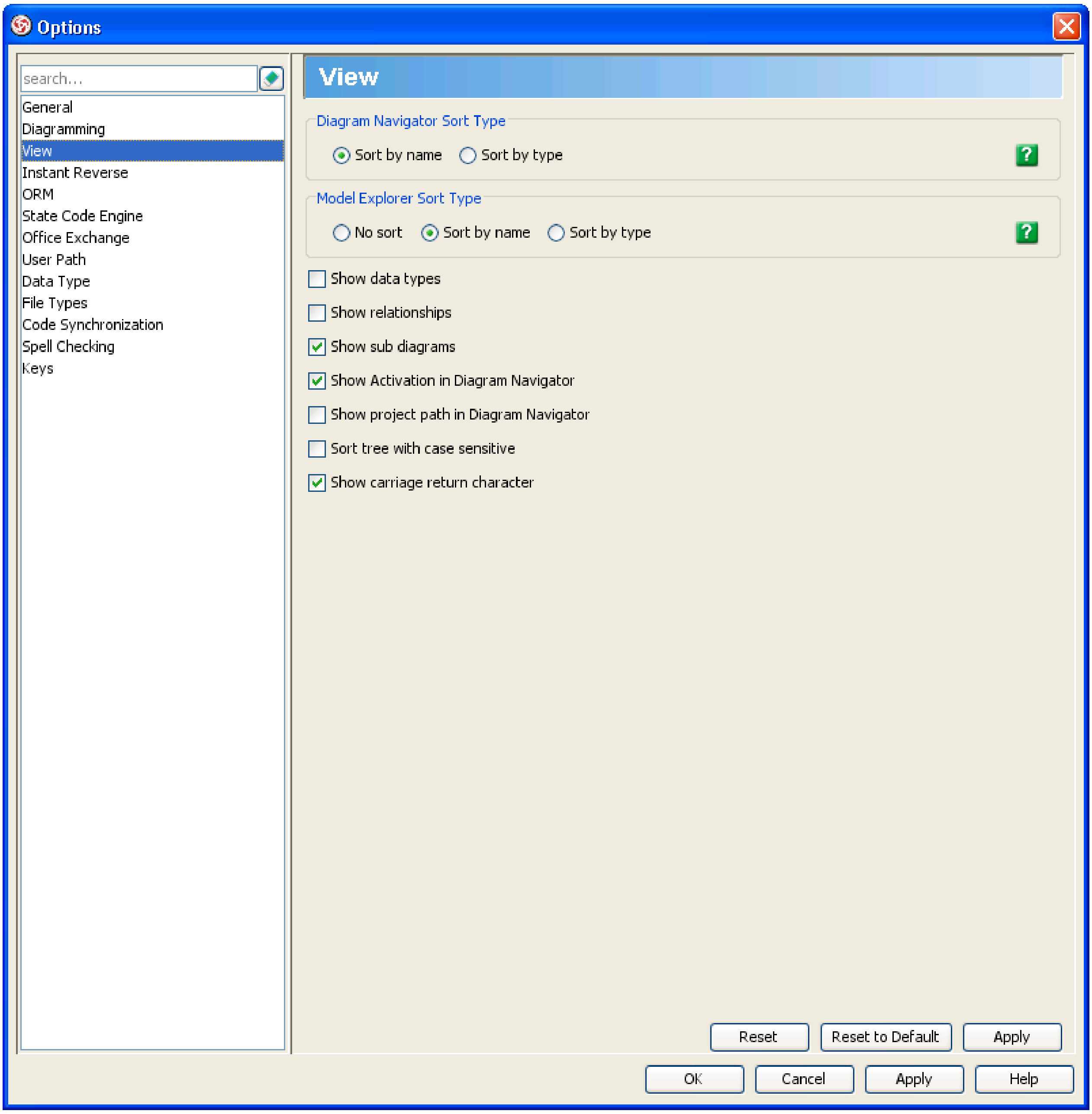


Diagram Navigator Sort Type

● Sort by name ○ Sort by type

Model Explorer Sort Type

○ No sort ● Sort by name ○ Sort by type

□ Show data types

□ Show relationships

☑ Show sub diagrams

☑ Show Activation in Diagram Navigator

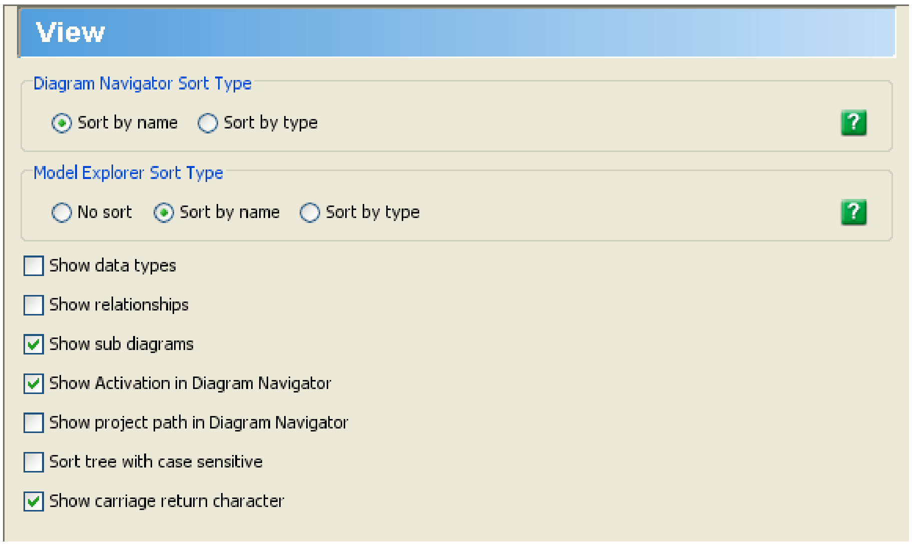
□ Show project path in Diagram Navigator

□ Sort tree with case sensitive

☑ Show carriage return character

Figure 2-31 *View Options in Option List*

Options



View

Diagram Navigator Sort Type

● Sort by name ○ Sort by type

Model Explorer Sort Type

○ No sort ● Sort by name ○ Sort by type

□ Show data types

□ Show relationships

☑ Show sub diagrams

☑ Show Activation in Diagram Navigator

□ Show project path in Diagram Navigator

□ Sort tree with case sensitive

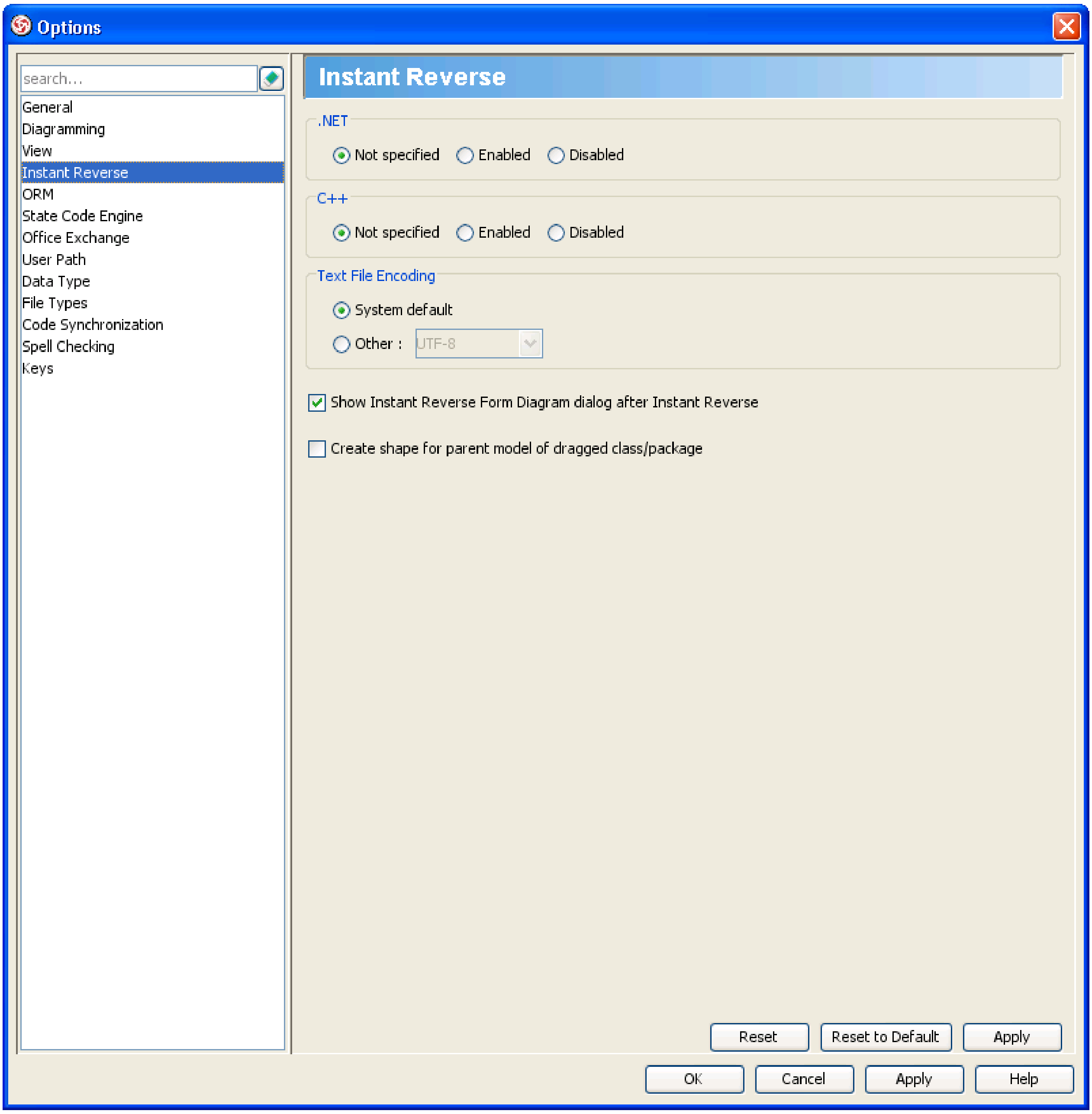
☑ Show carriage return character

Figure 2-32 *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 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

Table 2-29 *View Options details*

Instant Reverse Optons



.NET

● Not specified ○ Enabled ○ Disabled

C++

● Not specified ○ Enabled ○ Disabled

Text File Encoding

● System default

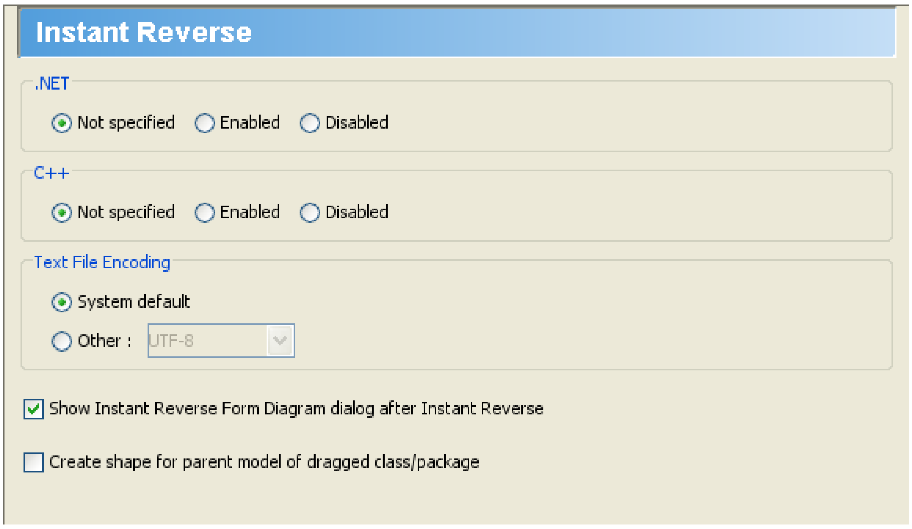
○ Other : UTF-8

☑ Show Instant Reverse Form Diagram dialog after Instant Reverse

□ Create shape for parent model of dragged class/package

Figure 2-33 *Instant Reverse Options in Option List*

Options



Instant Reverse

.NET

● Not specified ○ Enabled ○ Disabled

C++

● Not specified ○ Enabled ○ Disabled

Text File Encoding

● System default

○ Other : UTF-8

☑ Show Instant Reverse Form Diagram dialog after Instant Reverse

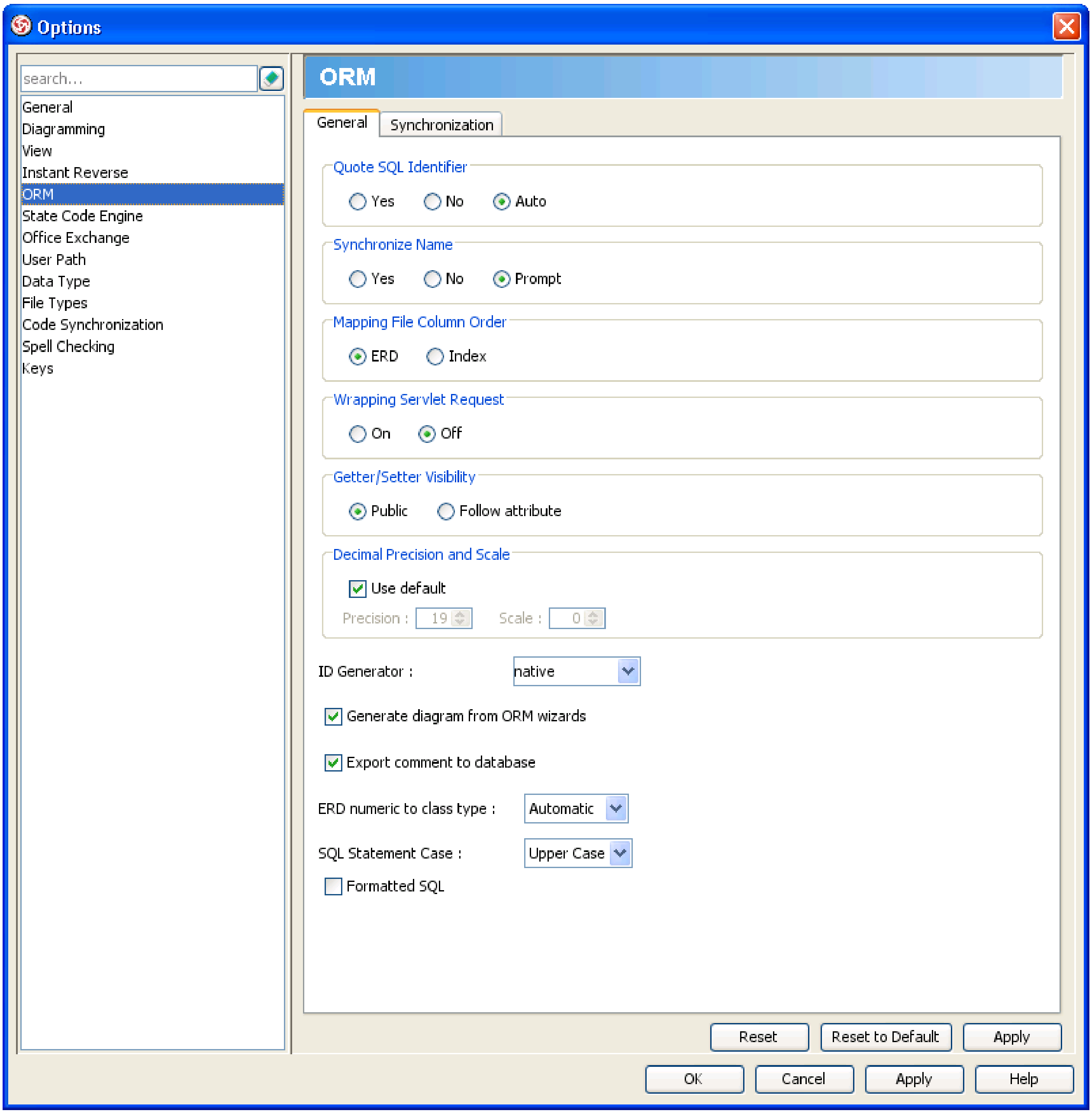
☐ Create shape for parent model of dragged class/package

Figure 2-34 *Instant Reverse Options*

Option Name	Description
.NET	• Not specified - (default) Do not specify whether Instant Reverse of .NET is enabled or not. • Enabled - Enable Instant Reverse for .NET • Disabled - Disable Instant Reverse for .NET
C++	• Not specified - (default) Do not specify whether Instant Reverse of C++ is enabled or not. • Enabled - Enable Instant Reverse for C++ • Disabled - Disable Instant Reverse for C++
Text File Encoding	• System default - (default) The default system encoding will be selected as encoding for source files that will be reversed • Other - Specify an encoding for the source files that will be reversed
Show Instant Reverse Form Diagram dialog after Instant Reverse	(default true) Show the **Instant Reverse Form Diagram** dialog box after Instant Reverse so that you can form diagram after reversing code into VP-UML
Create shape for parent model of dragged class/package	(default false) When drag and drop an element from tree to diagram, add also their parent (e.g. Package) to diagram to contain the dropped shapes

Table 2-30 *Instant Reverse Options details*

ORM Options



Quote SQL Identifier

○ Yes ○ No ● Auto

Synchronize Name

○ Yes ○ No ● Prompt

Mapping File Column Order

● ERD ○ Index

Wrapping Servlet Request

○ On ● Off

Getter/Setter Visibility

● Public ○ Follow attribute

Decimal Precision and Scale

☑ Use default

Precision : 19 Scale : 0

ID Generator : native

☑ Generate diagram from ORM wizards

☑ Export comment to database

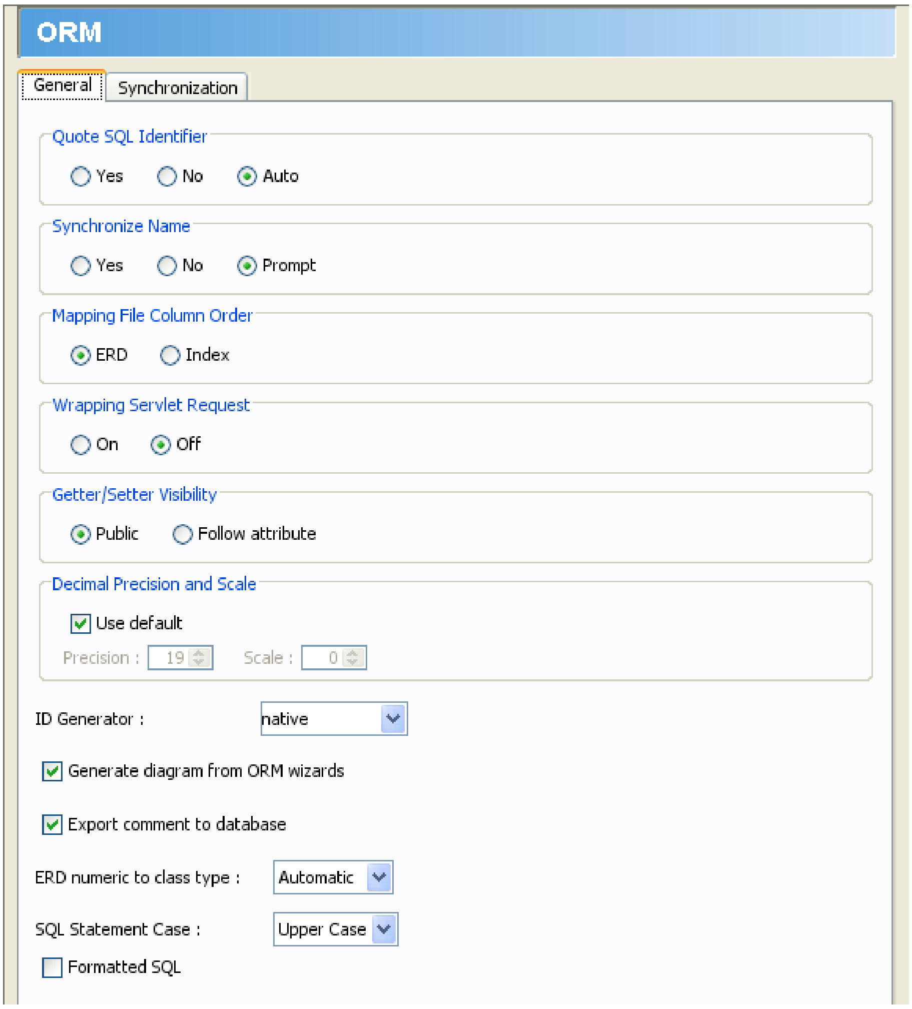
ERD numeric to class type : Automatic

SQL Statement Case : Upper Case

□ Formatted SQL

Figure 2-35 *ORM Options in Option List*

General



ORM

General | Synchronization

Quote SQL Identifier

○ Yes ○ No ● Auto

Synchronize Name

○ Yes ○ No ● Prompt

Mapping File Column Order

● ERD ○ Index

Wrapping Servlet Request

○ On ● Off

Getter/Setter Visibility

● Public ○ Follow attribute

Decimal Precision and Scale

☑ Use default

Precision : 19 Scale : 0

ID Generator : native

☑ Generate diagram from ORM wizards

☑ Export comment to database

ERD numeric to class type : Automatic

SQL Statement Case : Upper Case

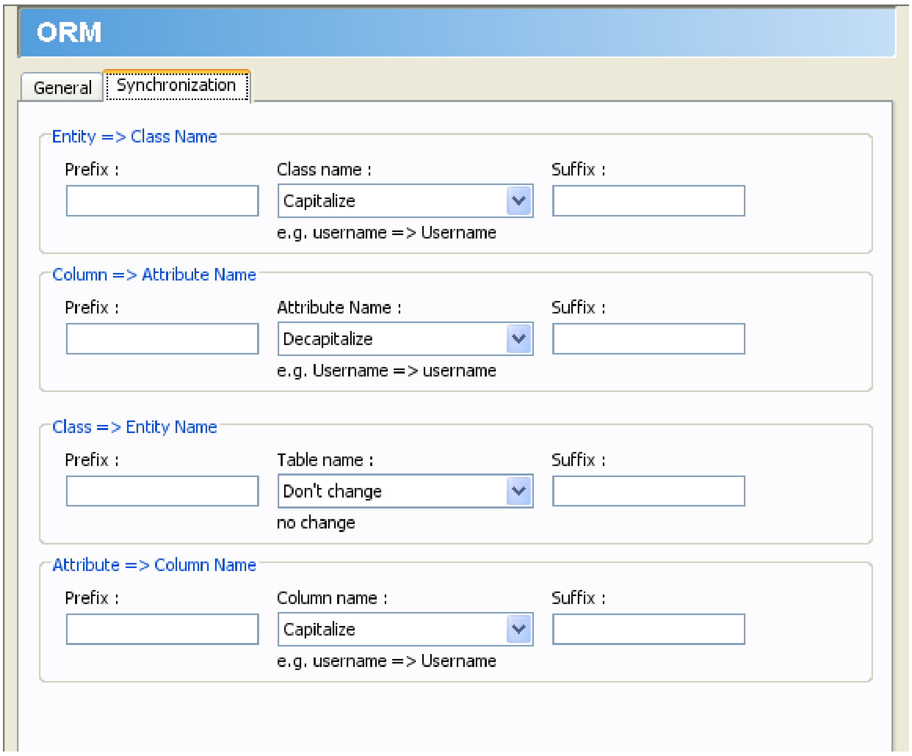
□ Formatted SQL

Figure 2-36 *General Options in ORM*

Option Name	Description
Quote SQL Identifier	• Yes - Add quotes to SQL identifier to prevent potential violation when executing SQL • No - Do not add quotes to SQL identifier • Auto - (default) Quote only reserved words
Synchronize Name	• Yes - Auto update model element name when synchronize class diagram and ERD • No - Do not update model element name when synchronize class diagram and ERD • Prompt - (default) Prompt to update model element name when synchronize class diagram and ERD
Mapping File Column Order	• ERD - (default) Generate columns in mapping file in same order as ERD • Index - Generate index columns first in mapping file
Wrapping Servlet Request	• On - Automatic lock persistable object when get by HttpSession.getAttribute() • Off - (default) Do not lock object automatically
Getter/Setter Visibility	• Public - (default) Generate public getter/setter • Follow attribute -Getter/setter visibility follow attribute's visibility
Decimal Precision and Scale - Use default	(default true) Automatic determine the most suitable precision and scale when synchronize from attribute to column as decimal
Decimal Precision and Scale - Precision	Specify the precision when synchronize from attribute to column as decimal
Decimal Precision and Scale - Scale	Specify the scale when synchronize from attribute to column as decimal
ID Generator	• assigned - lets the application to assign an identifier to the object before is called. • guid - uses a database-generated GUID string on MS SQL Server and MySQL. • hilo - uses a hi/lo algorithm to efficiently generate identifiers of type , or , given a table and column as a source of hi values. The hi/lo algorithm generates identifiers that are unique only for a particular database. • identity - supports identity columns in DB2, MySQL, MS SQL Server, Sybase and HypersonicSQL. The returned identifier is of type , or . • increment - generates identifiers of type , or that are unique only when no other process is inserting data into the same table. *Do not use in a cluster.* • native - (default) picks , or depending upon the capabilities of the underlying database. • seqhilo - uses a hi/lo algorithm to efficiently generate identifiers of type , or , given a named database sequence. • sequence - uses a sequence in DB2, PostgreSQL, Oracle. The returned identifier is of type , or
Generate diagram from ORM wizards	(default true) Generate diagram when finish ORM wizards
Export comment to database	(default true) Generate documentation to table/column
ERD numeric to class type	• Automatic - (default) Automatic select attribute type when synchronize from column numeric type • Integer - Synchronize column numeric type to attribute as integer type • Float - Synchronize column numeric type to attribute as float type • Double - Synchronize column numeric type to attribute as double type • Big Decimal - Synchronize column numeric type to attribute as big decimal type
SQL Statement Case	• Upper Case - (default) Generate upper case keyword in SQL • Lower case - Generate lower case keyword in SQL
Formatted SQL	(default false) Generate pretty formatted SQL

Table 2-31 *General Options details*

Synchronization



Entity => Class Name

Prefix :

Class name : Capitalize

Suffix :

e.g. username => Username

Column => Attribute Name

Prefix :

Attribute Name : Decapitalize

Suffix :

e.g. Username => username

Class => Entity Name

Prefix :

Table name : Don't change

Suffix :

no change

Attribute => Column Name

Prefix :

Column name : Capitalize

Suffix :

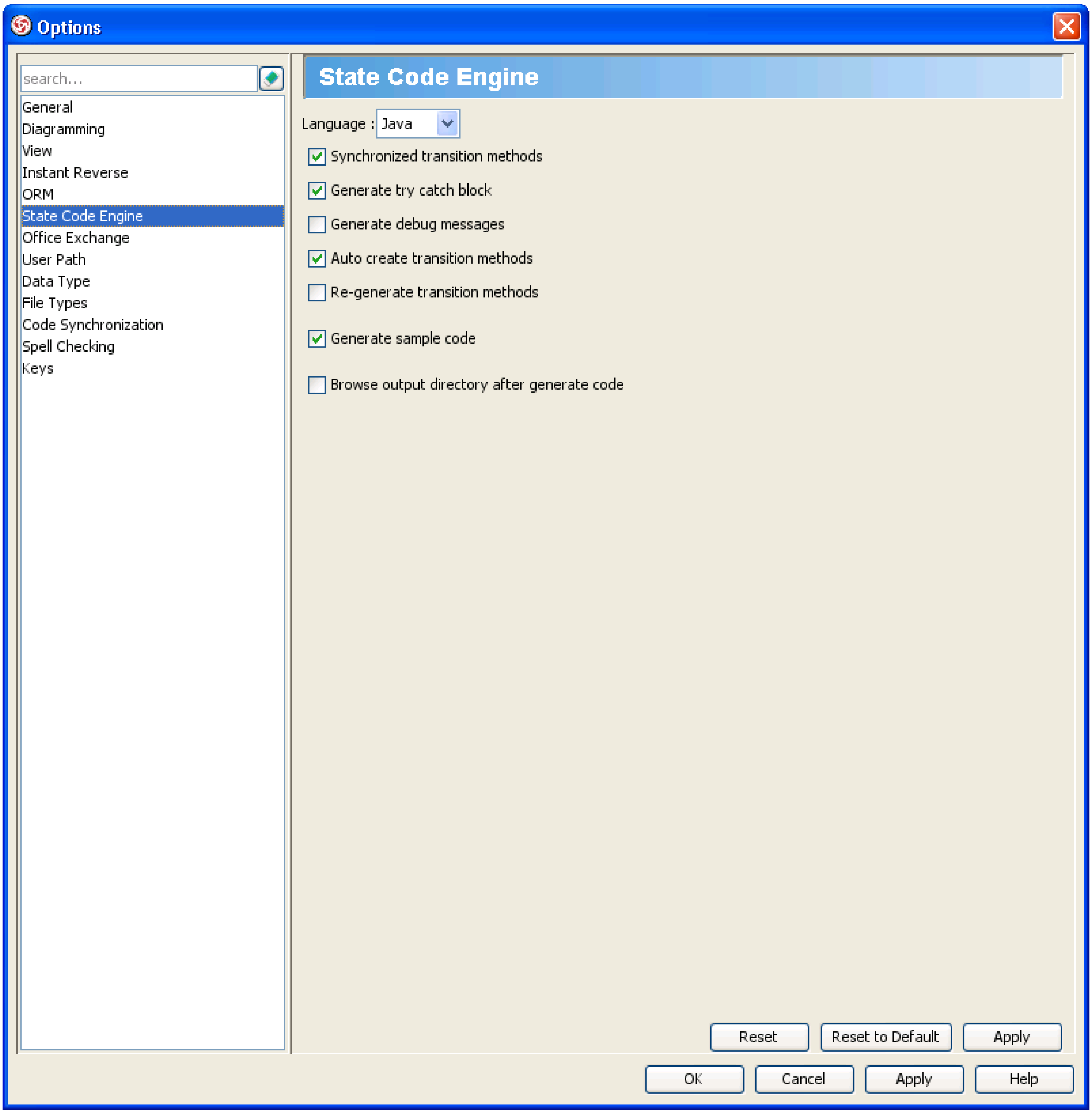
e.g. username => Username

Figure 2-37 *Synchronization Options in ORM*

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

Table 2-32 *Synchronization Options details*

State Code Engine Options



State Code Engine

Language : Java

☑ Synchronized transition methods

☑ Generate try catch block

□ Generate debug messages

☑ Auto create transition methods

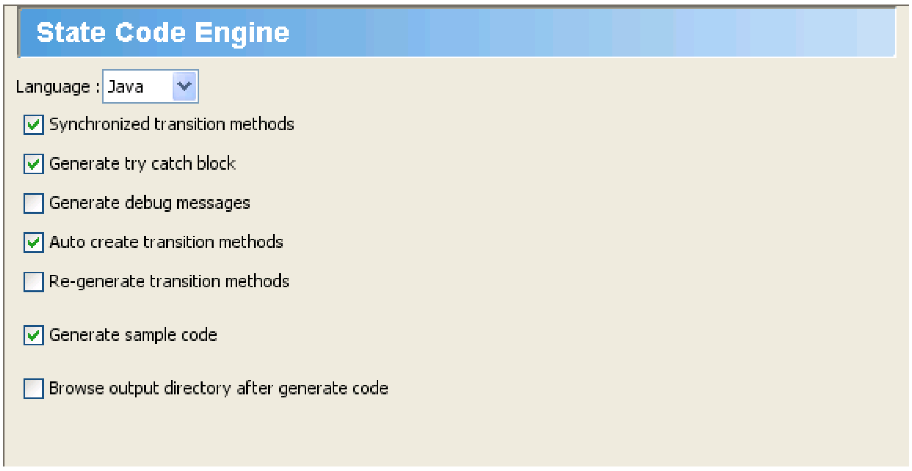
□ Re-generate transition methods

☑ Generate sample code

□ Browse output directory after generate code

Figure 2-38 *State Code Engine Options in Option List*

Options



State Code Engine

Language : Java

☑ Synchronized transition methods

☑ Generate try catch block

□ Generate debug messages

☑ Auto create transition methods

□ Re-generate transition methods

☑ Generate sample code

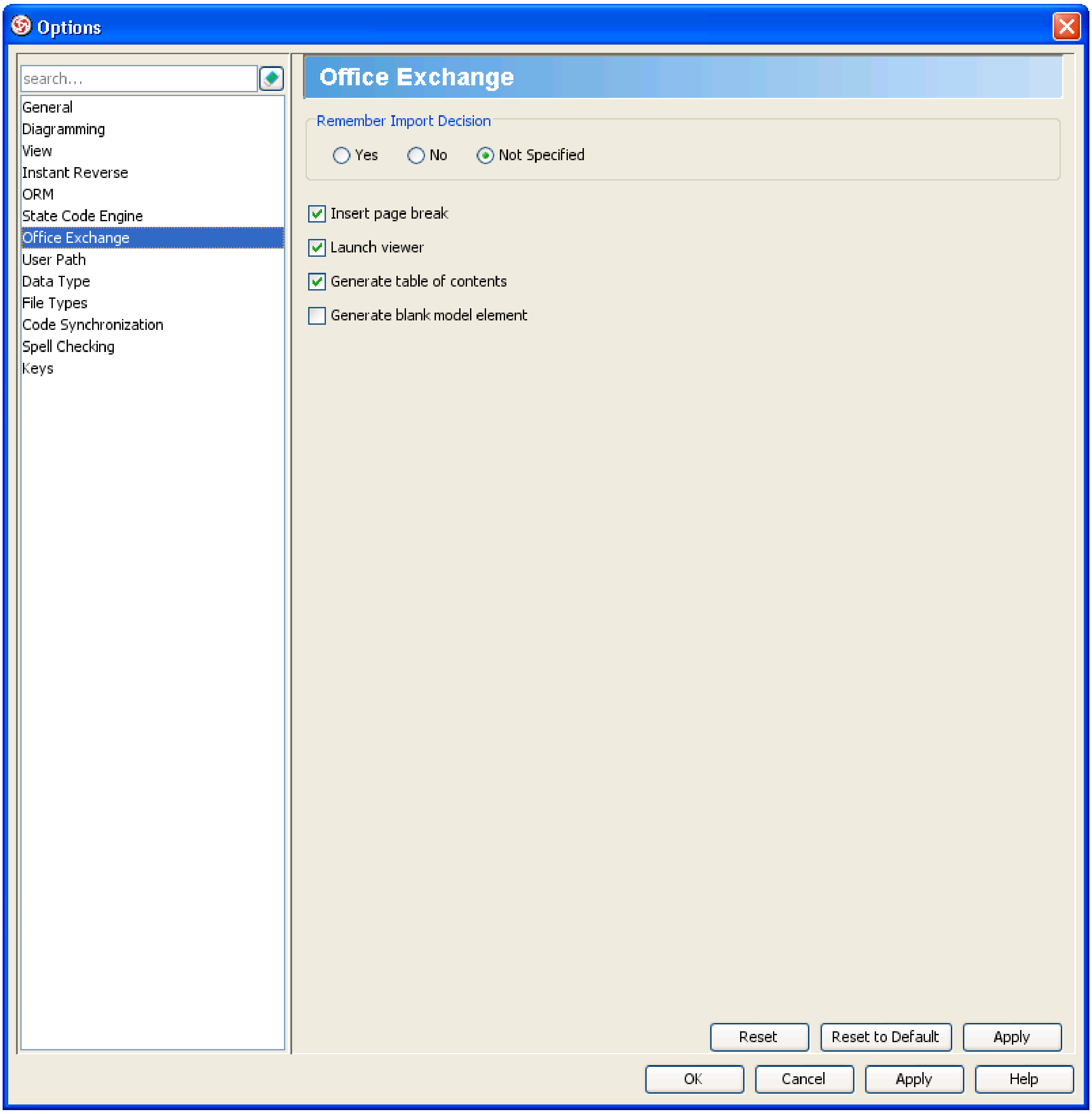
□ Browse output directory after generate code

__Figure 2-39__ State Code Engine Options

Option Name	Description
Language	• Java - (default) Generate code in Java • C# - Generate code in C# • VB .NET - Generate code in VB.NET • C++ - Generate code in C++
Synchronized transition methods	(default true) Generate synchronized keyword for transition methods
Generate try catch block	(default true) Generate try catch block for method calls that may produce exception
Generate debug messages	(default false) Generate debug message to help tracing problems that happen when running generated code
Auto create transition methods	(default true) Auto generate operation to owner class by transition
Re-generate transition methods	(default false) Overwrite the transition methods if already exists in source code
Generate sample code	(default true) Generate sample code to help you understand how to work with generated code
Browse output directory after generate code	(default false) Open the directory of generated state code

__Table 2-33__ State Code Engine Options details

Office Exchange Options



Office Exchange

Remember Import Decision

○ Yes ○ No ● Not Specified

☑ Insert page break

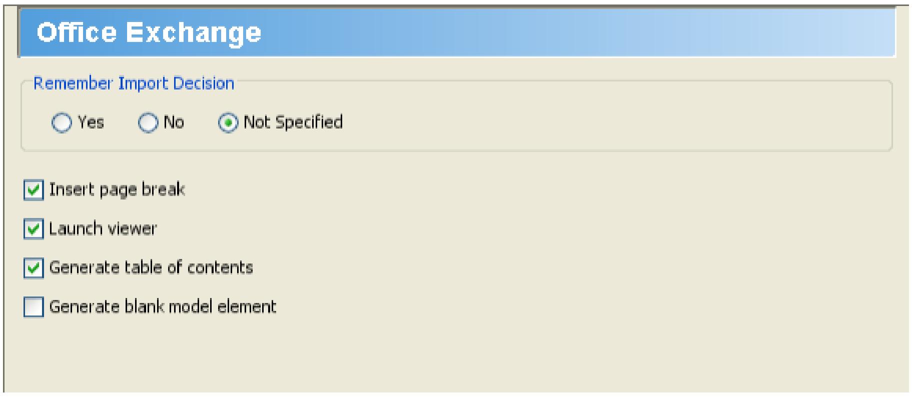
☑ Launch viewer

☑ Generate table of contents

□ Generate blank model element

Figure 2-40 *Office Exchange Options in Option List*

Options



Office Exchange

Remember Import Decision

○ Yes ○ No ● Not Specified

☑ Insert page break

☑ Launch viewer

☑ Generate table of contents

□ Generate blank model element

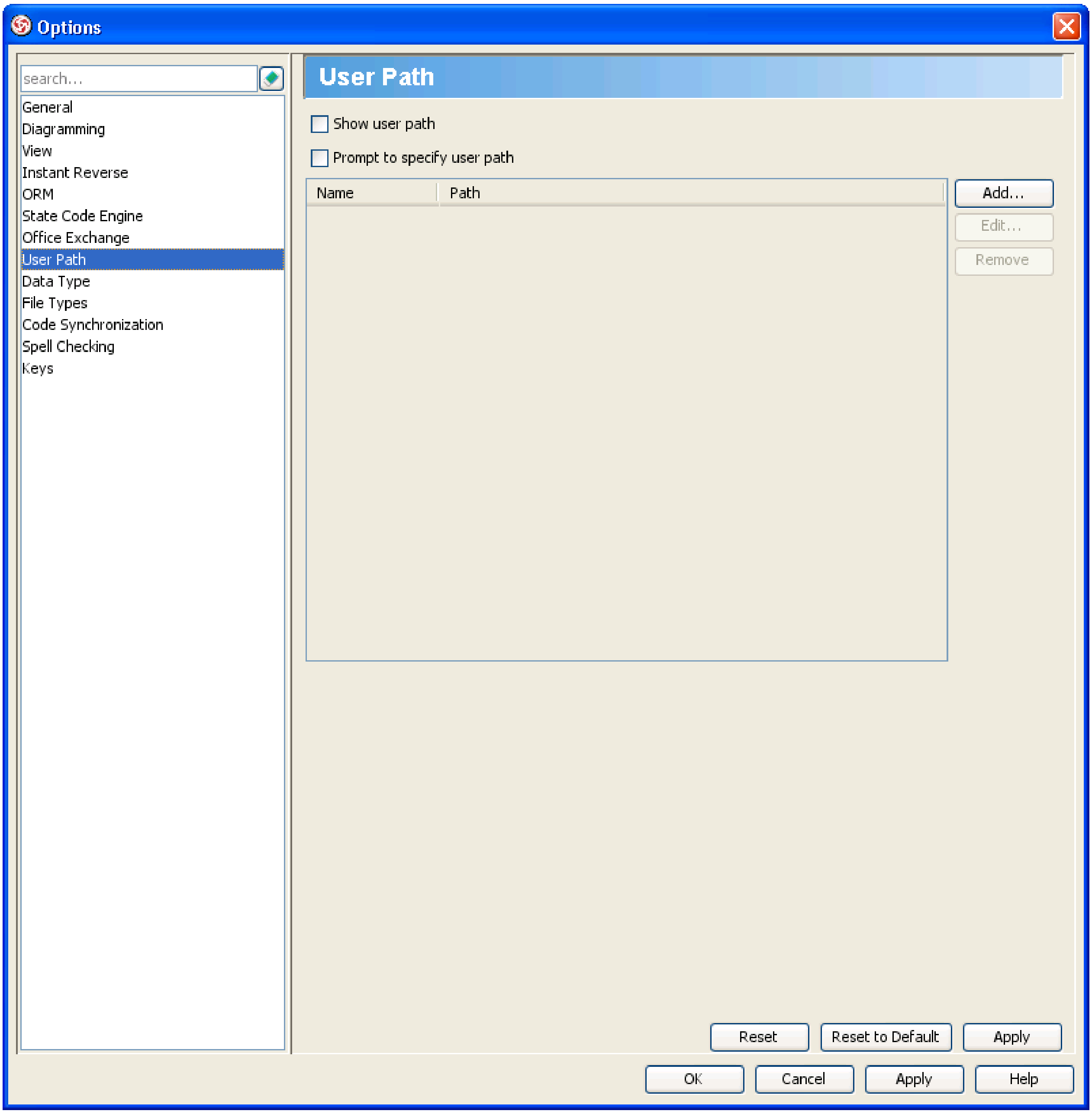
Figure 2-41 *Office Exchange Options*

Option Name	Description
Remember Import Decision	VP-UML detects changes made in exported document, and will suggest you to import changes back to VP-UML. 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-UML whenever changes are detected.
Insert page break	(default true) Insert a page break for each element
Launch viewer	(default true) Open the document after export
Generate table of contents	(default true) Generate Table of Contents in document
Generate blank model element	(default false) Export an empty form for user to create new model

Table 2-34 *Office Exchange Options details*

User Path Options

A user path is a variable that refers to a base path in user&rsquo;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.

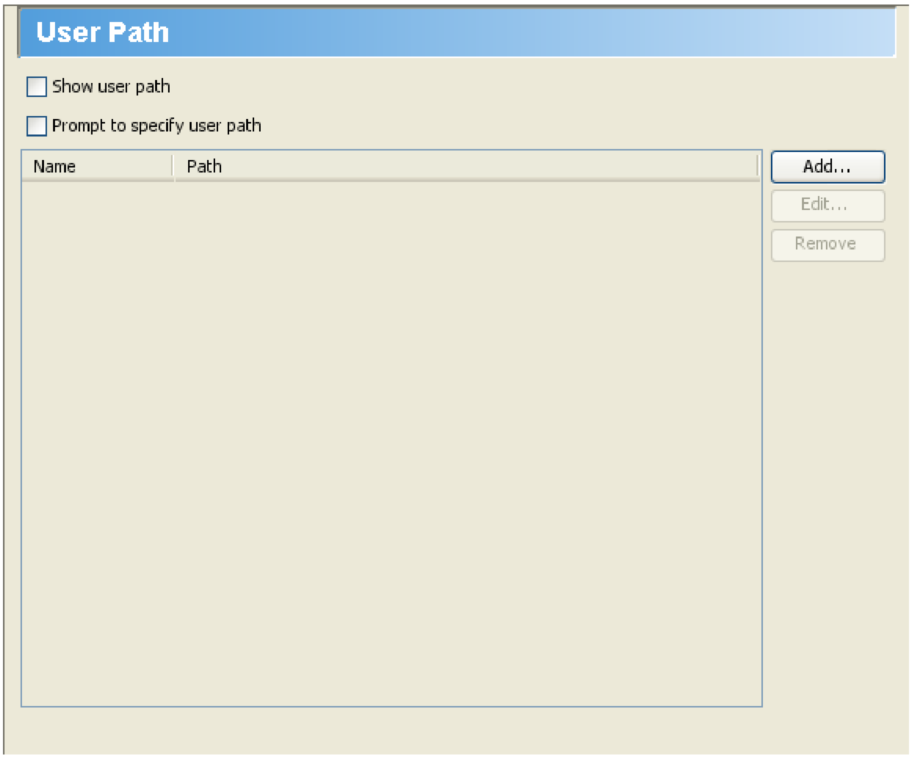


□ Show user path

□ Prompt to specify user path

Figure 2-42 *User Path Options in Option List*

Options



User Path

□ Show user path

□ Prompt to specify user path

Figure 2-43 User Path Options

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

Table 2-35 User Path Options details

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. &lsquo;boolean&rsquo;, &lsquo;bool&rsquo; and &lsquo;Boolean&rsquo; are adopted by UML and Java, C# and VB.NET respectively. But they are all referring to the same thing &ndash; 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.

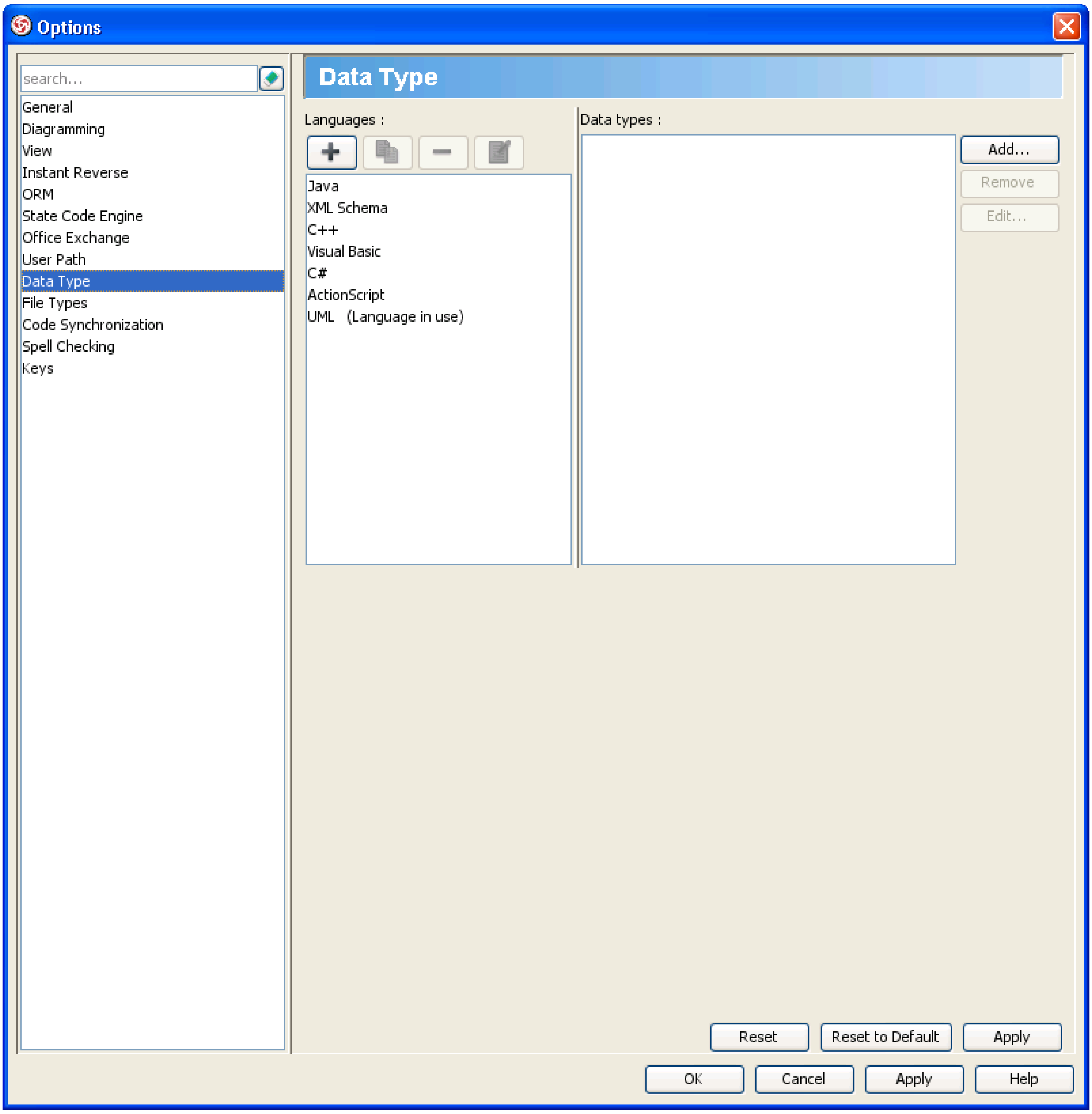


Figure 2-44 *Data Type Options in Option List*

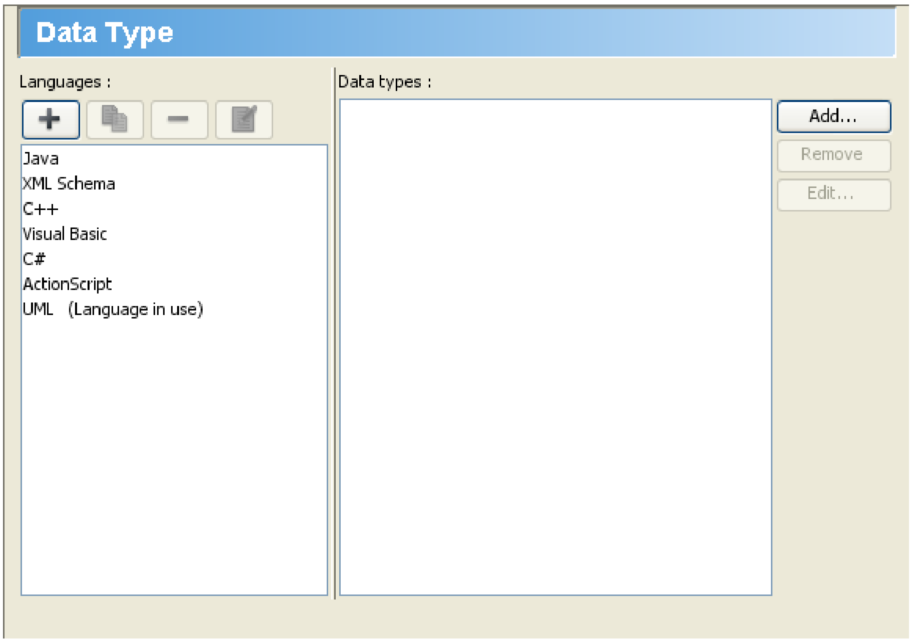
Configure Programming Language

1. Right-click on the project root node under **Diagram Navigator / Model Pane / Class Repository**
2. Select **Configure Programming Language...** from the pop-up menu. This shows the **Programming Language** dialog box.
3. Select the language to switch to.
4. 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. Press on the **plus** sign to add a language.
2. Enter its name, and press **OK** to confirm.
3. Press **Add...** to add a data-type to the chosen language.
4. 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.

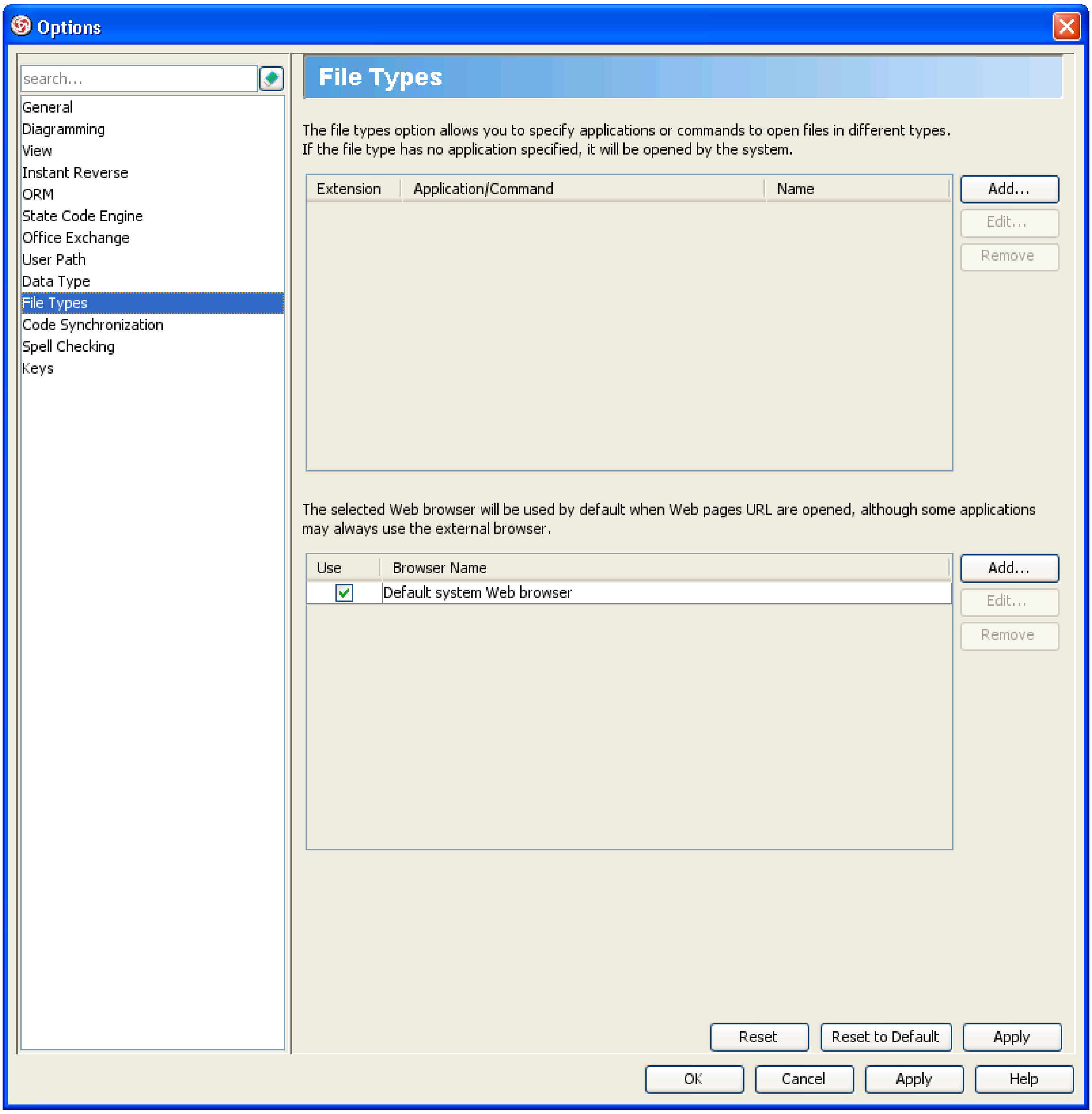


Java
XML Schema
C++
Visual Basic
C#
ActionScript
UML (Language in use)

Figure 2-45 *Data Type Options*

File Type Options

Model Reference lets you add reference(s) to external file or URL into diagram element. You can open the referenced file or 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.



The file types option allows you to specify applications or commands to open files in different types.
If the file type has no application specified, it will be opened by the system.

Extension	Application/Command	Name

The selected Web browser will be used by default when Web pages URL are opened, although some applications may always use the external browser.

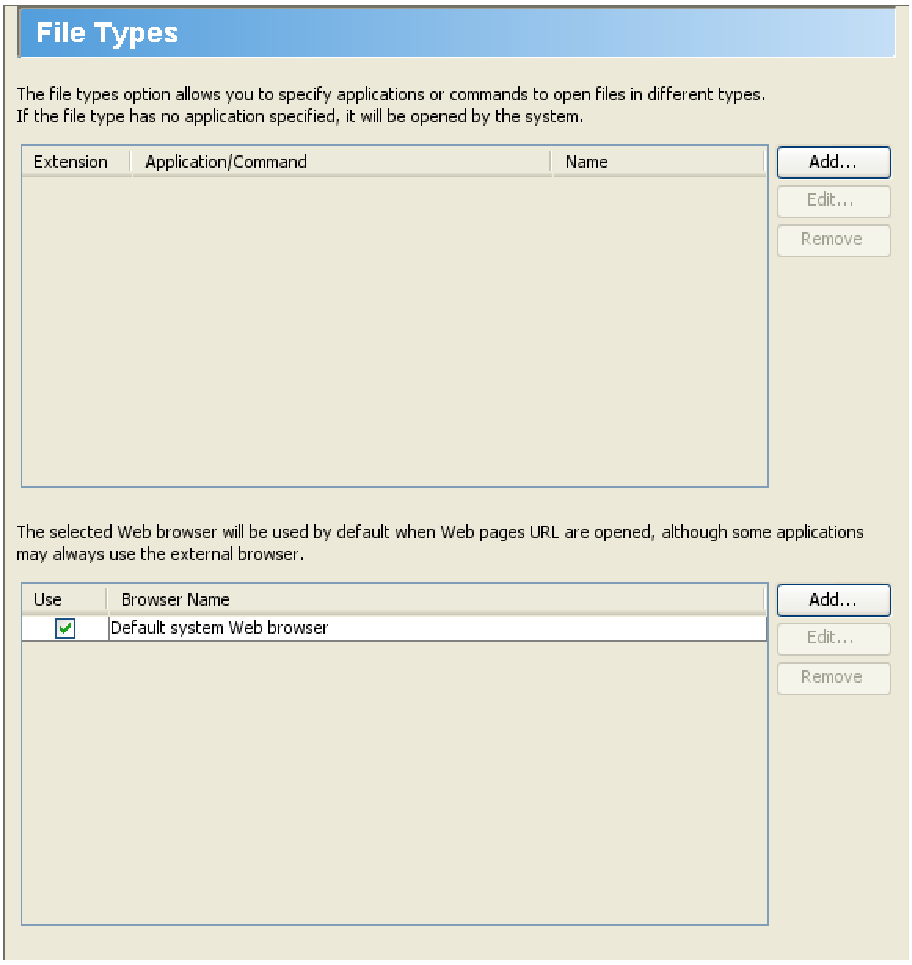
Use	Browser Name
☑	Default system Web browser

Figure 2-46 *File Type Options in Option List*

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.



File Types

The file types option allows you to specify applications or commands to open files in different types.
If the file type has no application specified, it will be opened by the system.

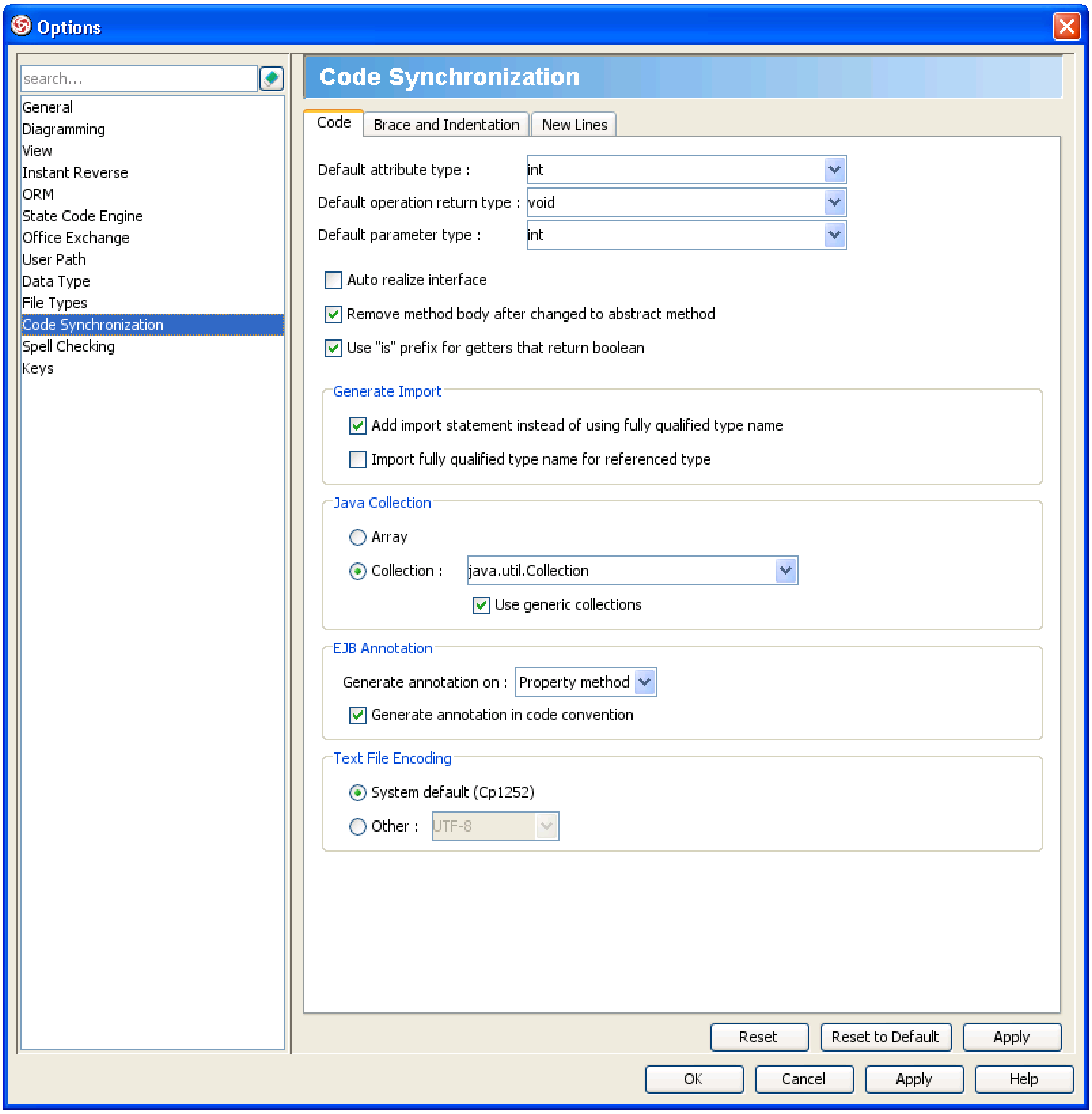
Extension	Application/Command	Name

The selected Web browser will be used by default when Web pages URL are opened, although some applications may always use the external browser.

Use	Browser Name
☑	Default system Web browser

Figure 2-47 *File Type Options*

Code Synchronization Options



Default attribute type : int

Default operation return type : void

Default parameter type : int

□ Auto realize interface

☑ Remove method body after changed to abstract method

☑ Use "is" prefix for getters that return boolean

Generate Import

☑ Add import statement instead of using fully qualified type name

□ Import fully qualified type name for referenced type

Java Collection

○ Array

● Collection : java.util.Collection

☑ Use generic collections

EJB Annotation

Generate annotation on : Property method

☑ Generate annotation in code convention

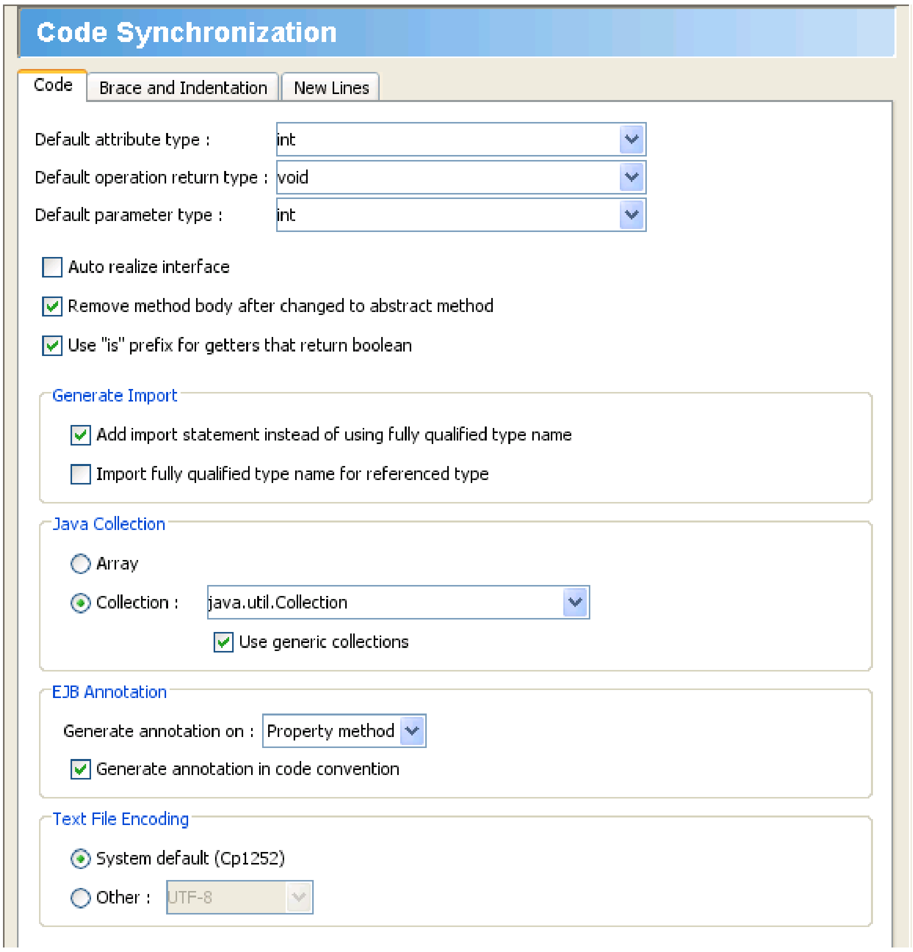
Text File Encoding

● System default (Cp1252)

○ Other : UTF-8

***Figure 2-48** Code Synchronization Options in Option List*

Code



Code Synchronization

Code | Brace and Indentation | New Lines

Default attribute type : int

Default operation return type : void

Default parameter type : int

☐ Auto realize interface

☑ Remove method body after changed to abstract method

☑ Use "is" prefix for getters that return boolean

Generate Import

☑ Add import statement instead of using fully qualified type name

☐ Import fully qualified type name for referenced type

Java Collection

○ Array

● Collection : java.util.Collection

☑ Use generic collections

EJB Annotation

Generate annotation on : Property method

☑ Generate annotation in code convention

Text File Encoding

● System default (Cp1252)

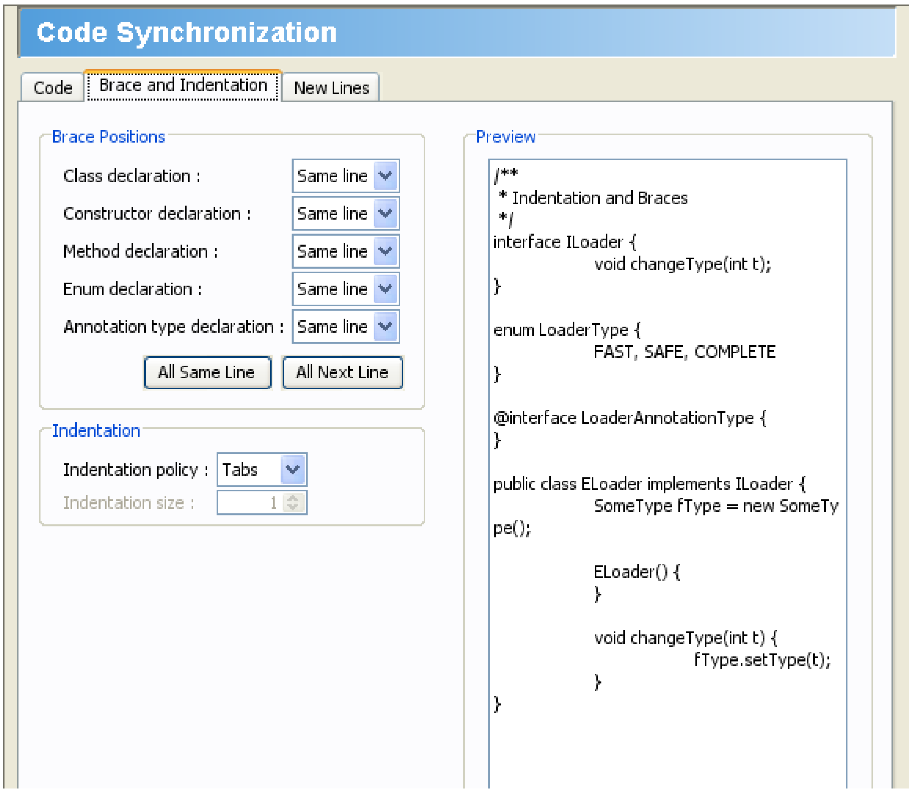
○ Other : UTF-8

Figure 2-49 *Code Options in Code Synchronization*

Option Name	Description
Default attribute type	(default int) Type that will be assigned to Attribute upon code generation when type is unspecified
Default operation return type	(default void) Return Type that will be assigned to operation upon code generation when return type is unspecified
Default parameter type	(default int) Type that will be assigned to Parameter upon code generation when type is unspecified
Auto realize interface	(default false) Generate operations defined in interface in sub-classes
Remove method body after changed to abstract method	(default true) When an operation is set from non-abstract to abstract, updating code will remove the related method's body
Use "is" prefix for getters that return boolean	(default true) Generate getter's name as isXXXX() for getters that return a boolean value
Add import statement instead of using fully qualified type name	(default true) Add import statement for referencing classes in another package/namespace instead of using fully qualified name inline
Import fully qualified type name for referenced type	(default false) Use fully qualified type name in import statements instead of using wildcard character * to represent importing all classes in package
Java Collection	• Array - Generate one-to-many relationship as array • Collection - (default) Generate one-to-many relationship as collection
Use generic collections	(default true) Allow to use generic collection
Generate annotation on	• Property method - Generate annotation on property method • Field - Generate annotation on field
Generate annotation in code convention	(default true) Generate annotation in code convention
Text File Encoding	• System default - (default) The default system encoding will be selected as encoding for source files • Other -Specify an encoding for source files

Table 2-36 Code Options details

Brace and Indentation



Class declaration : Same line
Constructor declaration : Same line
Method declaration : Same line
Enum declaration : Same line
Annotation type declaration : Same line

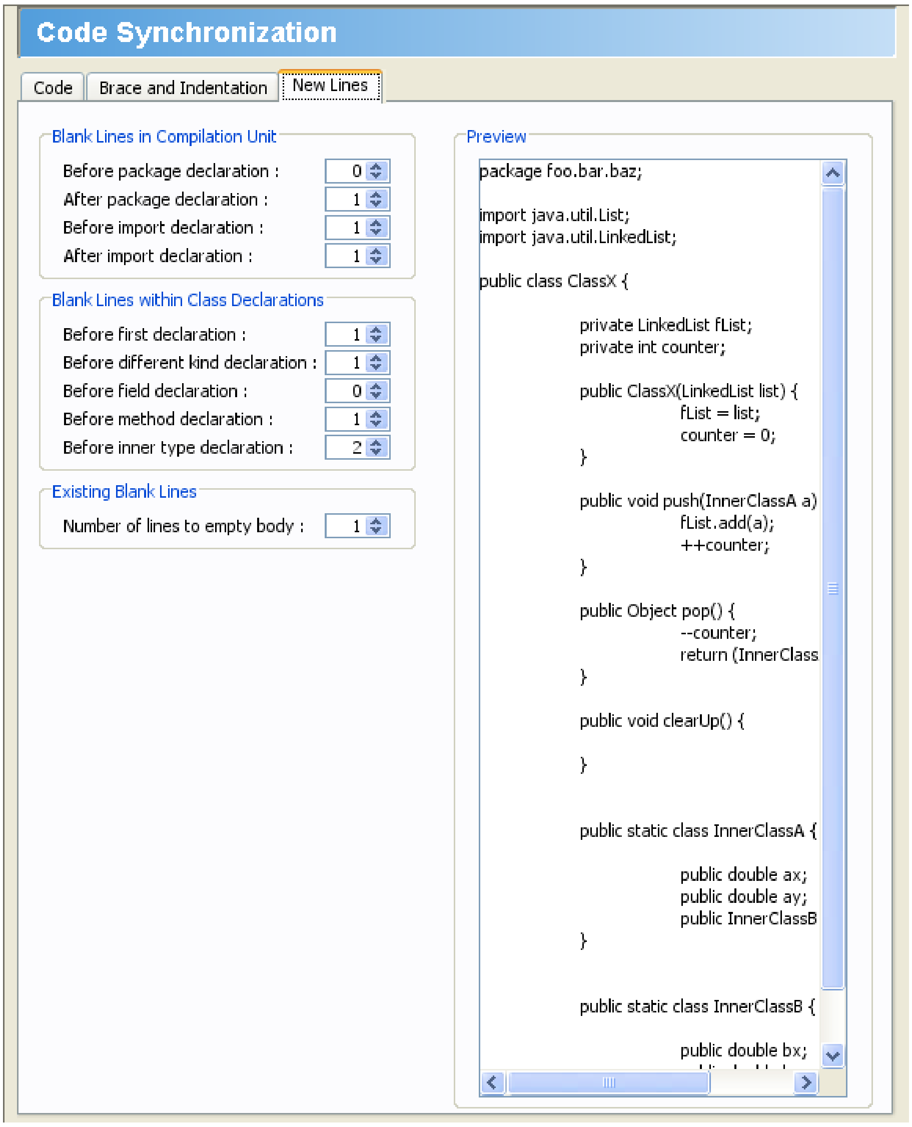
Indentation policy : Tabs
Indentation size : 1

```
/**
 * Indentation and Braces
 */
interface ILoader {
        void changeType(int t);
}

enum LoaderType {
        FAST, SAFE, COMPLETE
}

@interface LoaderAnnotationType {
}

public class ELoader implements ILoader {
        SomeType fType = new SomeTy
pe();

        ELoader() {
        }

        void changeType(int t) {
                fType.setType(t);
        }
}
```

Figure 2-50 Brace and Indentation Options in Code Synchronization

Option Name	Description
Class declaration	• Same line - (default) Brace for class declaration appear at the same line as the declaration • Next line - Brace for class declaration appear at the line after the declaration
Constructor declaration	• Same line - (default) Brace for constructor appear at the same line as the declaration • Next line - Brace for constructor appear at the line after the declaration
Method declaration	• Same line - (default) Brace for method appear at the same line as the declaration • Next line - Brace for method appear at the line after the declaration
Enum declaration	• Same line - (default) Brace for enumeration appear at the same line as the declaration • Next line - Brace for enumeration tor appear at the line after the declaration
Annotation type declaration	• Same line - (default) Brace for annotation type appear at the same line as the declaration • Next line - Brace for annotation type appear at the line after the declaration
Indentation policy	• Tabs - (default) Use a tab of space as indentation • Spaces - Use spaces as indentation. The number of spaces can be defined below
Indentation size	The number of spaces to indent

Table 2-37 Brace and Indentation Options details



Blank Lines in Compilation Unit

Before package declaration :	0
After package declaration :	1
Before import declaration :	1
After import declaration :	1

Blank Lines within Class Declarations

Before first declaration :	1
Before different kind declaration :	1
Before field declaration :	0
Before method declaration :	1
Before inner type declaration :	2

Existing Blank Lines

Number of lines to empty body :	1

Preview

```
package foo.bar.baz;

import java.util.List;
import java.util.LinkedList;

public class ClassX {

        private LinkedList fList;
        private int counter;

        public ClassX(LinkedList list) {
                fList = list;
                counter = 0;
        }

        public void push(InnerClassA a)
                fList.add(a);
                ++counter;
        }

        public Object pop() {
                --counter;
                return (InnerClass
        }

        public void clearUp() {

        }


        public static class InnerClassA {

                public double ax;
                public double ay;
                public InnerClassB
        }


        public static class InnerClassB {

                public double bx;
```

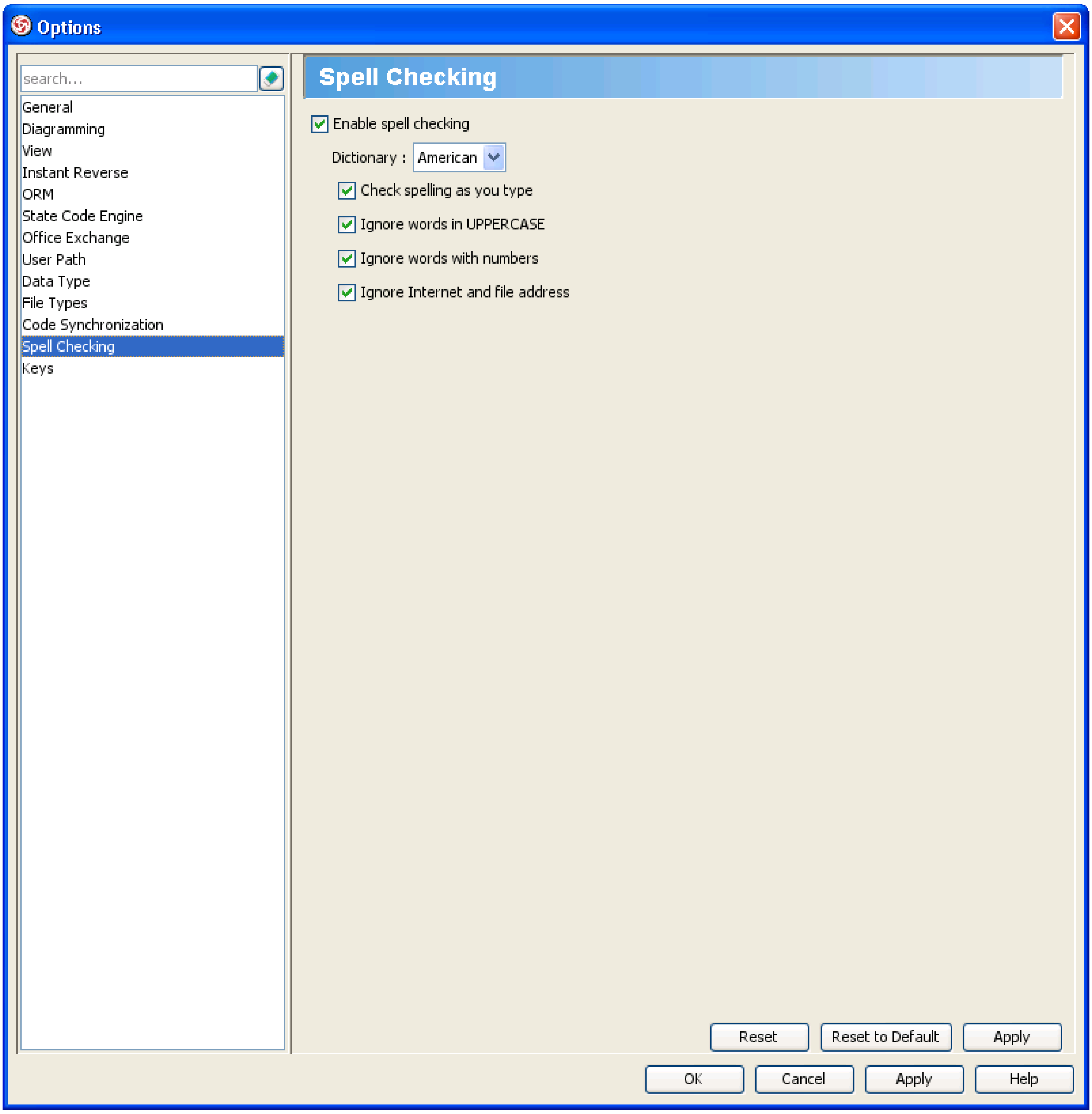
Figure 2-51 New Lines Options in Code Synchronization

Option Name	Description
Before package declaration	Number of blank lines to appear before Package declaration
After package declaration	Number of blank lines to appear after Package declaration
Before import declaration	Number of blank lines to appear before import statements
After import declaration	Number of blank lines to appear after import statements
Before first declaration	Number of blank lines to appear before the first declaration within Class declarations
Before different kind declaration	Number of blank lines to appear before a different kind of declaration
Before field declaration	Number of blank lines to appear before field declaration
Before method declaration	Number of blank lines to appear before method declaration
Before inner type declaration	Number of blank lines to appear before inner type declaration
Number of lines to empty body	Number of blank lines to appear in empty method body

Table 2-38 *New Lines Options details*

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.



☑ Enable spell checking

Dictionary : American

☑ Check spelling as you type

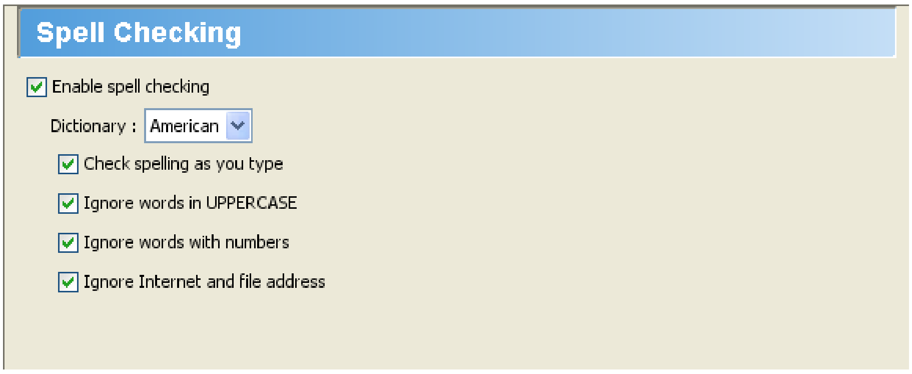
☑ Ignore words in UPPERCASE

☑ Ignore words with numbers

☑ Ignore Internet and file address

Figure 2-52 *Spell Checking Options in Option List*

Options



☑ Enable spell checking

Dictionary : American

☑ Check spelling as you type

☑ Ignore words in UPPERCASE

☑ Ignore words with numbers

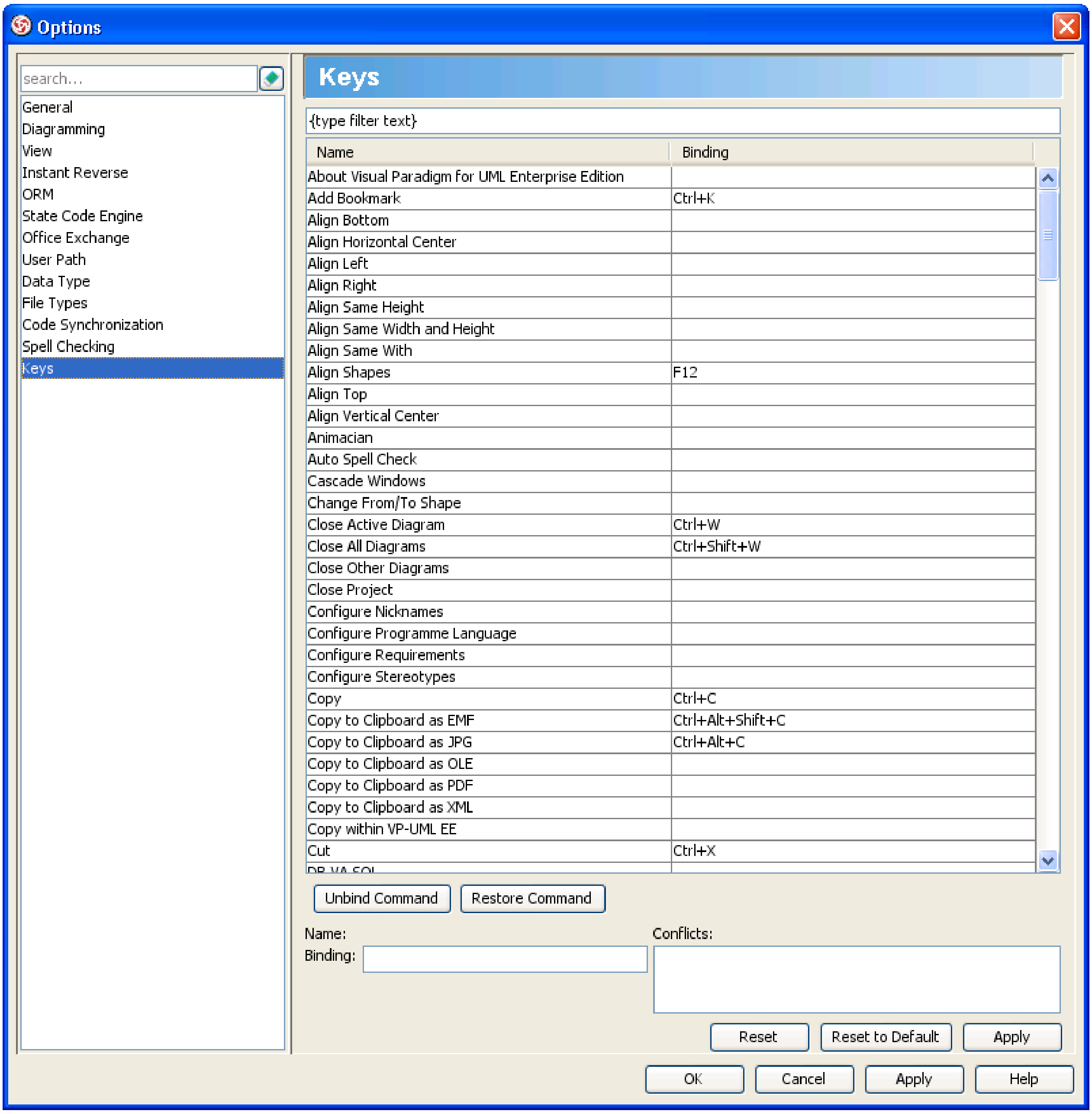
☑ Ignore Internet and file address

Figure 2-53 *Spell Checking 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

Table 2-39 *Spell Checking Options details*

Keys Options



Name	Binding
About Visual Paradigm for UML Enterprise Edition	
Add Bookmark	Ctrl+K
Align Bottom	
Align Horizontal Center	
Align Left	
Align Right	
Align Same Height	
Align Same Width and Height	
Align Same With	
Align Shapes	F12
Align Top	
Align Vertical Center	
Animacian	
Auto Spell Check	
Cascade Windows	
Change From/To Shape	
Close Active Diagram	Ctrl+W
Close All Diagrams	Ctrl+Shift+W
Close Other Diagrams	
Close Project	
Configure Nicknames	
Configure Programme Language	
Configure Requirements	
Configure Stereotypes	
Copy	Ctrl+C
Copy to Clipboard as EMF	Ctrl+Alt+Shift+C
Copy to Clipboard as JPG	Ctrl+Alt+C
Copy to Clipboard as OLE	
Copy to Clipboard as PDF	
Copy to Clipboard as XML	
Copy within VP-UML EE	
Cut	Ctrl+X

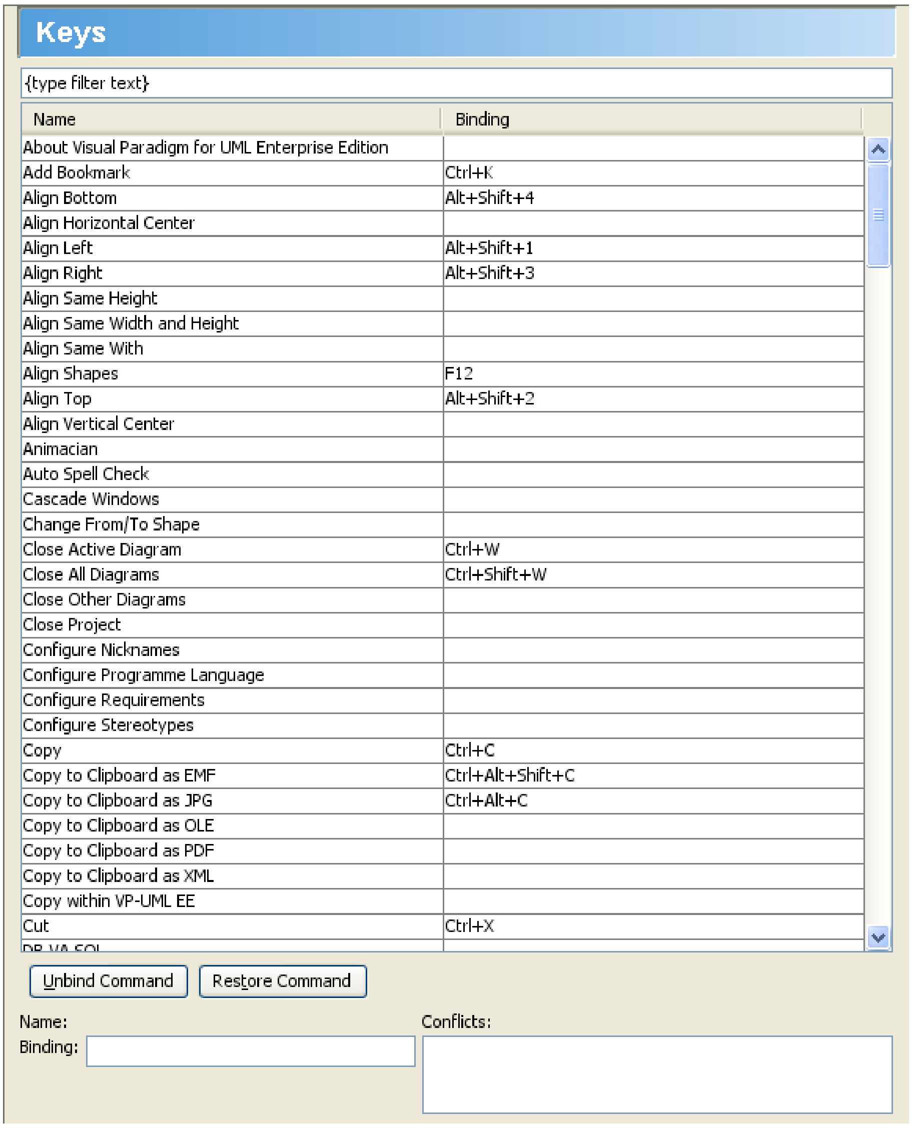
Figure 2-54 *Keys Options in Option List*

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 VP-UML.

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** 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.



Name	Binding
About Visual Paradigm for UML Enterprise Edition	
Add Bookmark	Ctrl+K
Align Bottom	Alt+Shift+4
Align Horizontal Center	
Align Left	Alt+Shift+1
Align Right	Alt+Shift+3
Align Same Height	
Align Same Width and Height	
Align Same With	
Align Shapes	F12
Align Top	Alt+Shift+2
Align Vertical Center	
Animacian	
Auto Spell Check	
Cascade Windows	
Change From/To Shape	
Close Active Diagram	Ctrl+W
Close All Diagrams	Ctrl+Shift+W
Close Other Diagrams	
Close Project	
Configure Nicknames	
Configure Programme Language	
Configure Requirements	
Configure Stereotypes	
Copy	Ctrl+C
Copy to Clipboard as EMF	Ctrl+Alt+Shift+C
Copy to Clipboard as JPG	Ctrl+Alt+C
Copy to Clipboard as OLE	
Copy to Clipboard as PDF	
Copy to Clipboard as XML	
Copy within VP-UML EE	
Cut	Ctrl+X
DB VA SQL	

Name:

Binding:

Conflicts:

Figure 2-55 *Keys Options*

Updating VP-UML with VP Suite Update

Updating of VP-UML can be done by updating Visual Paradigm Suite. To update Visual Paradigm Suite:

1. Launch VP Suite Update by any of the ways below:
   - Through the Start Menu.

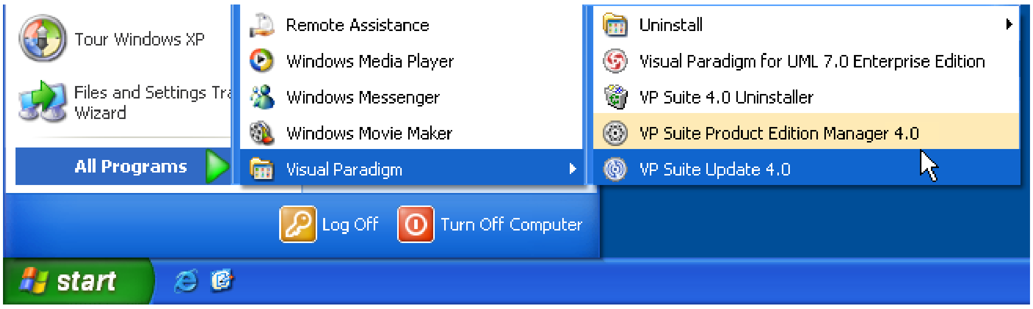


Figure 3-1 *Starting VP Suite Update via Start Menu*

   - By executing the launcher **VP Suite Update** in the launcher folder of VP Suite installation directory. Users in all operating system can start VP-UML in this way.

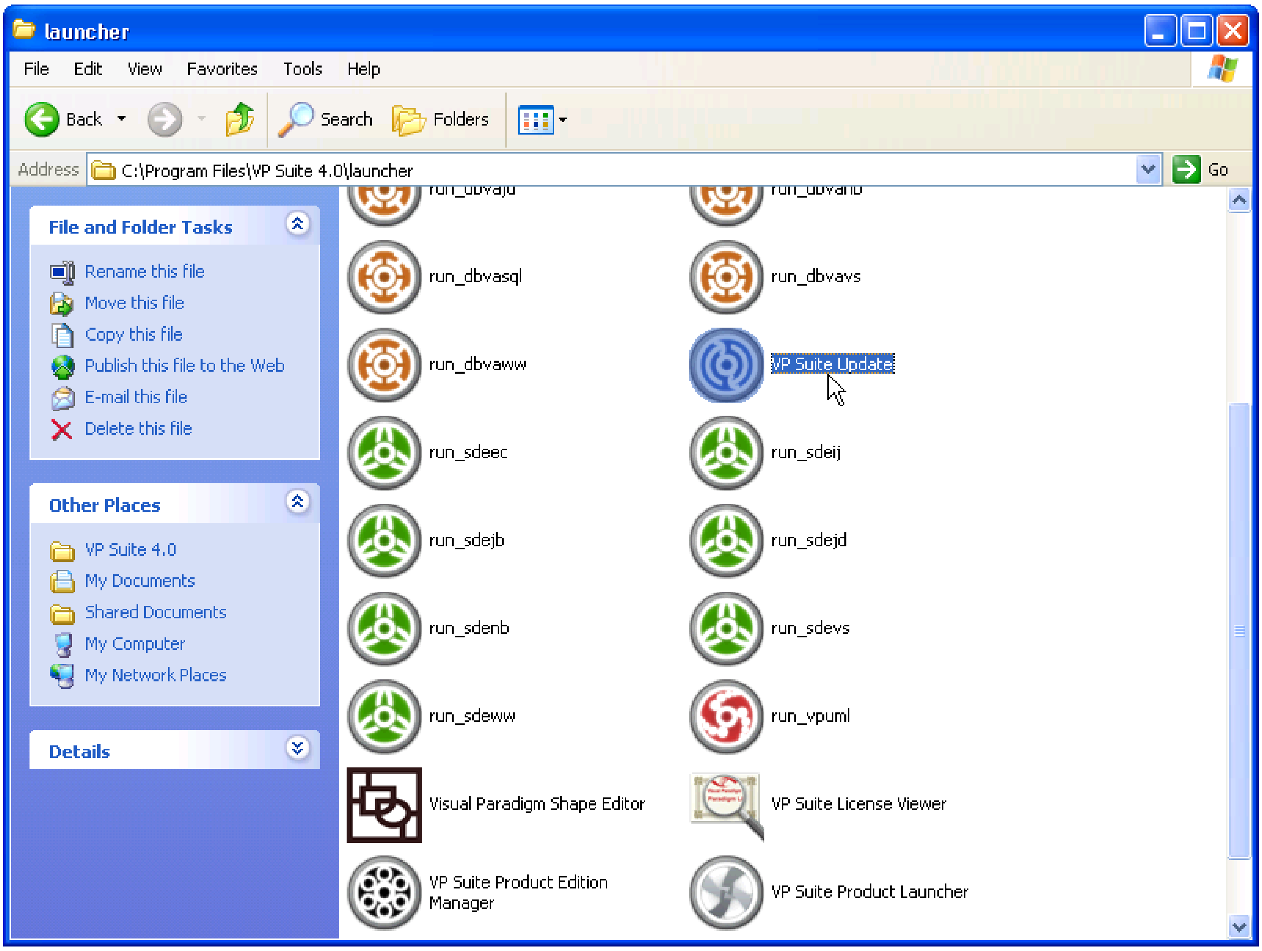
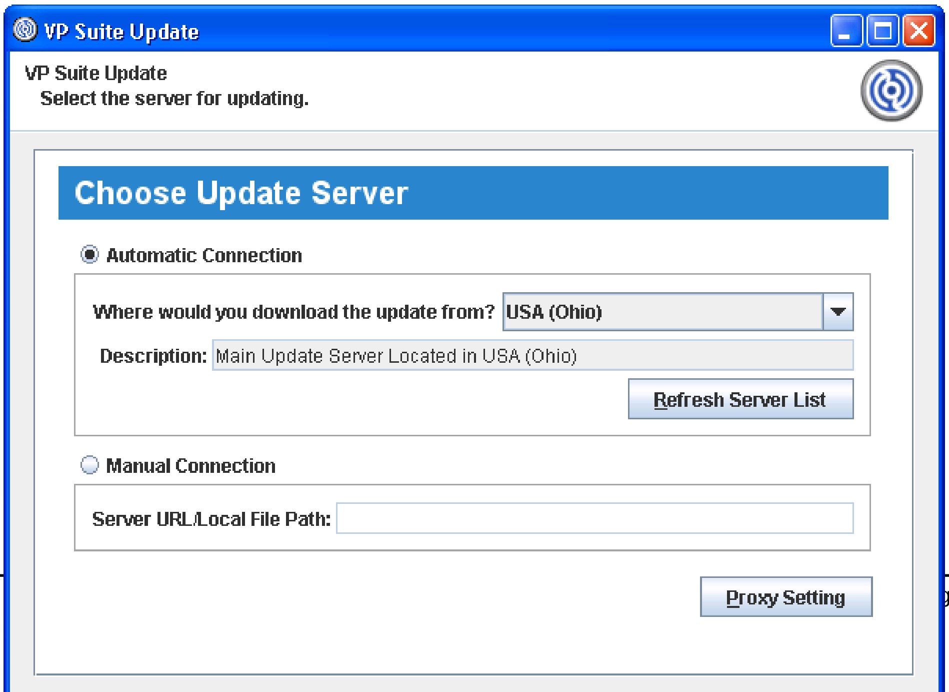


Figure 3-2 *Starting VP Suite Update with launcher*

This starts the VP Suite Update.


VP Suite Update

VP Suite Update
Select the server for updating.

Choose Update Server

● Automatic Connection

Where would you download the update from? USA (Ohio)

Description: Main Update Server Located in USA (Ohio)

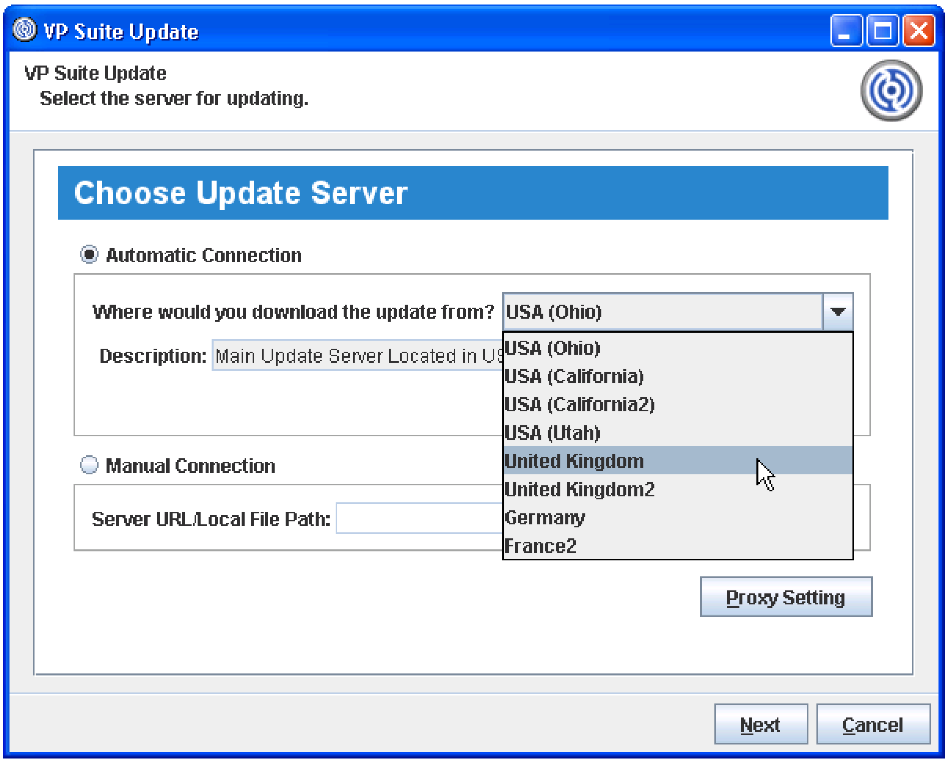
Refresh Server List

○ Manual Connection

Server URL/Local File Path:

Proxy Setting

2. Select a server from the drop-down menu for checking and downloading update files. The list of servers can be refreshed by using the **Refresh Server List** button.



VP Suite Update

VP Suite Update
Select the server for updating.

Choose Update Server

● Automatic Connection

Where would you download the update from? USA (Ohio)

Description: Main Update Server Located in US

○ Manual Connection

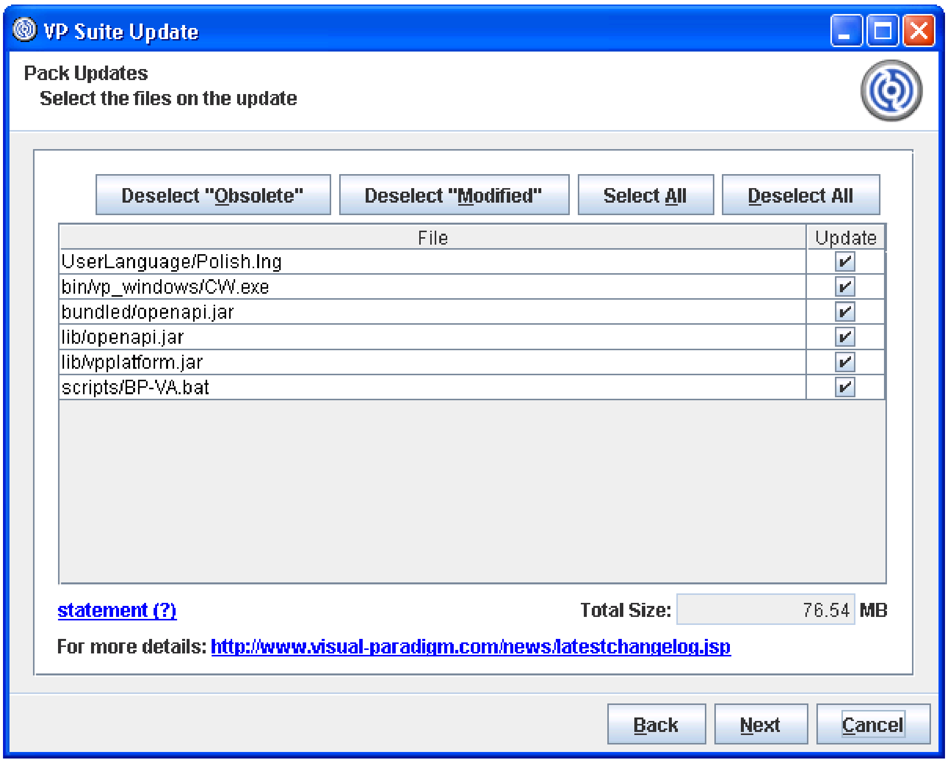
Server URL/Local File Path:

***Figure 3-4** Selecting a location to check and download updates*

NOTE: If there is a connection problem when downloading the update files from the specified server under **Automatic Connection**, VP Suite Update will connect to another server randomly instead of the specified connection to continue the update automatically.

NOTE: By using the Update Synchronizer to keep the files in the update repository up-to-date, network client can specify the or file path of the update repository using the **Manual Connection**.

3. Click **Next** to continue. After checking is completed, the updated files are listed in the **Pack Updates** page.



Pack Updates
Select the files on the update

File	Update
UserLanguage/Polish.lng	☑
bin/vp_windows/CW.exe	☑
bundled/openapi.jar	☑
lib/openapi.jar	☑
lib/vpplatform.jar	☑
scripts/BP-VA.bat	☑

Total Size: 76.54 MB

For more details: http://www.visual-paradigm.com/news/latestchangelog.jsp

Figure 3-5 *The Pack Updates page for selecting files to be updated*

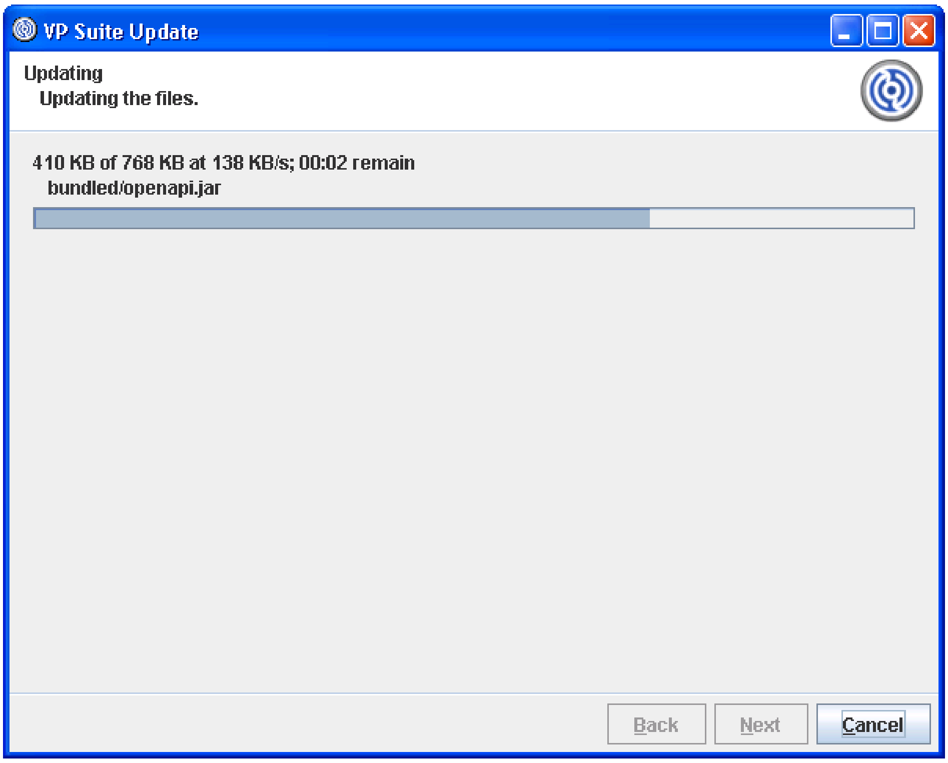
4. Review the files to be updated. You may select or deselect files to be updated by checking and unchecking individual file listing in the page, under the **Update** column. Usually, you can keep the selection unchanged. But if you see the filenames appear in blue or red, you better confirm the selection carefully. Below is a description of red and blue colored filename:

Color	Description
Blue	Modified. This refers to files which are modified by the users and will be replaced with the original files
Red	Obsolete. This refers to files which are no longer useful in the new version of the product and will be deleted.

Table 3-1 *Description of blue and red colored filename in Pack Updates page*

If you want to keep the modified and obsolete files, press the **Deselect "Obsolete"** and **Deselect "Modified"** buttons so that these files will not be updated.

5. Click **Next** to proceed with updating the files.

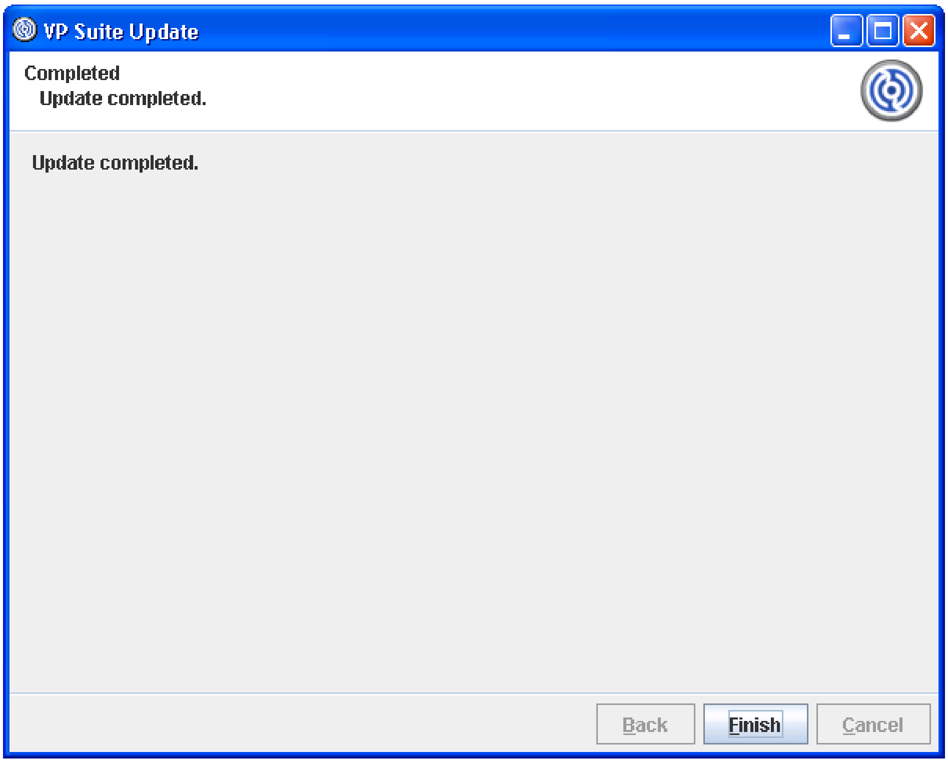


Updating
Updating the files.

410 KB of 768 KB at 138 KB/s; 00:02 remain
bundled/openapi.jar

Figure 3-6 *Update is in process*

6. Click **Finish** to confirm.



Completed
Update completed.

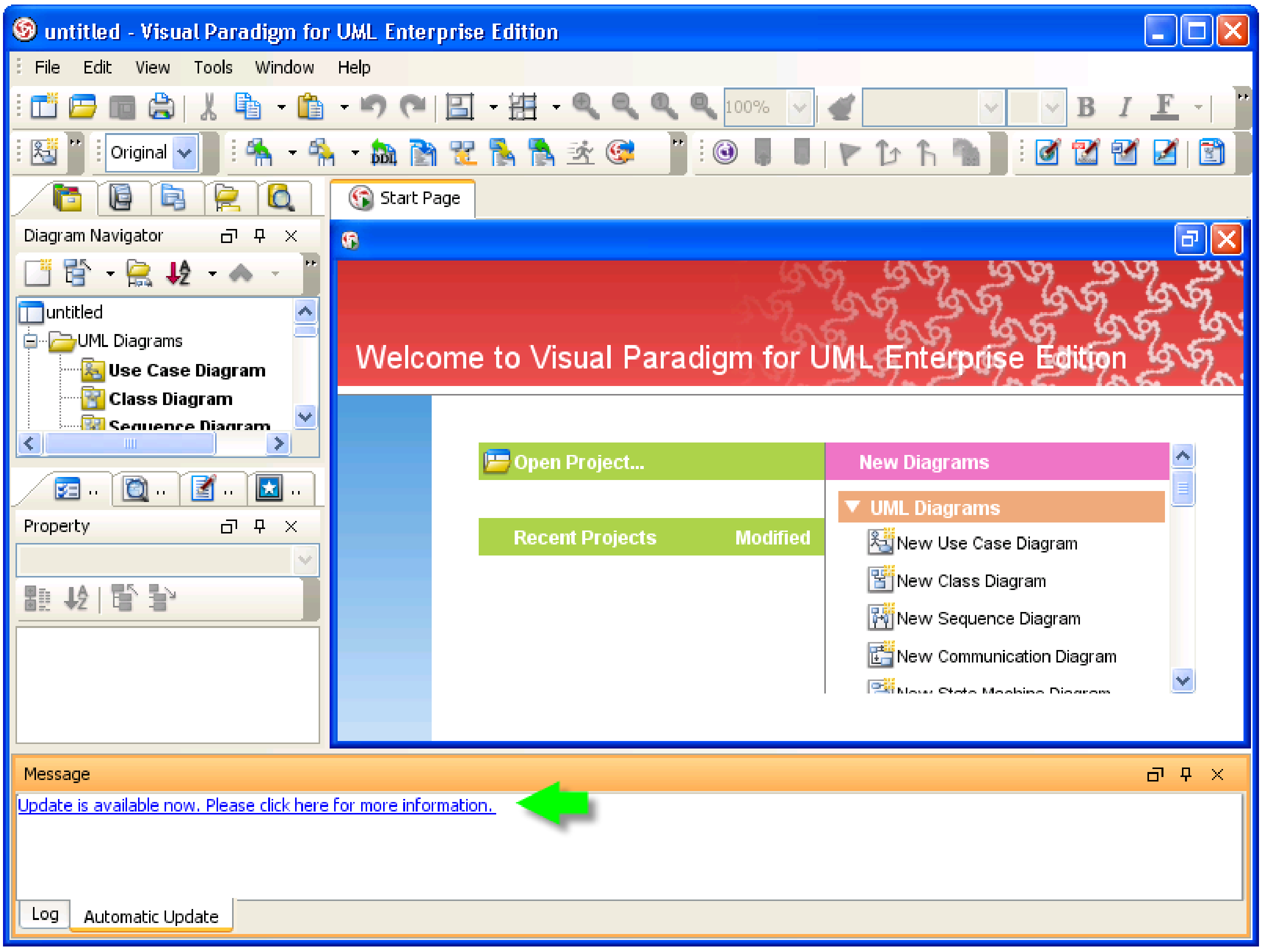
Update completed.

Figure 3-7 *Update is complete*

Automatic Update Notification

Updating when Running VP-UML

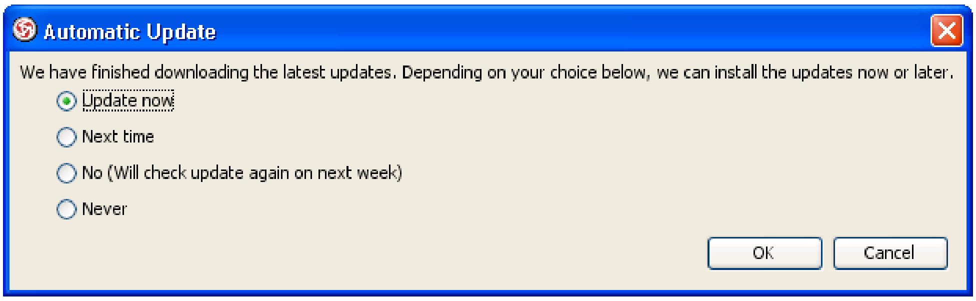
1. When running VP-UML, a message " **Update is available now. Please click here for more information**" may popup in the Message pane. This message indicates that there is a build newer than the one that you are running, and you are recommended to perform an update to advance to the latest build.



Update is available now. Please click here for more information.

Figure 3-8 Notification of available updates

2. Click on the message. This popup the **Automatic Update** dialog box.



We have finished downloading the latest updates. Depending on your choice below, we can install the updates now or later.

- ● Update now
- ○ Next time
- ○ No (Will check update again on next week)
- ○ Never

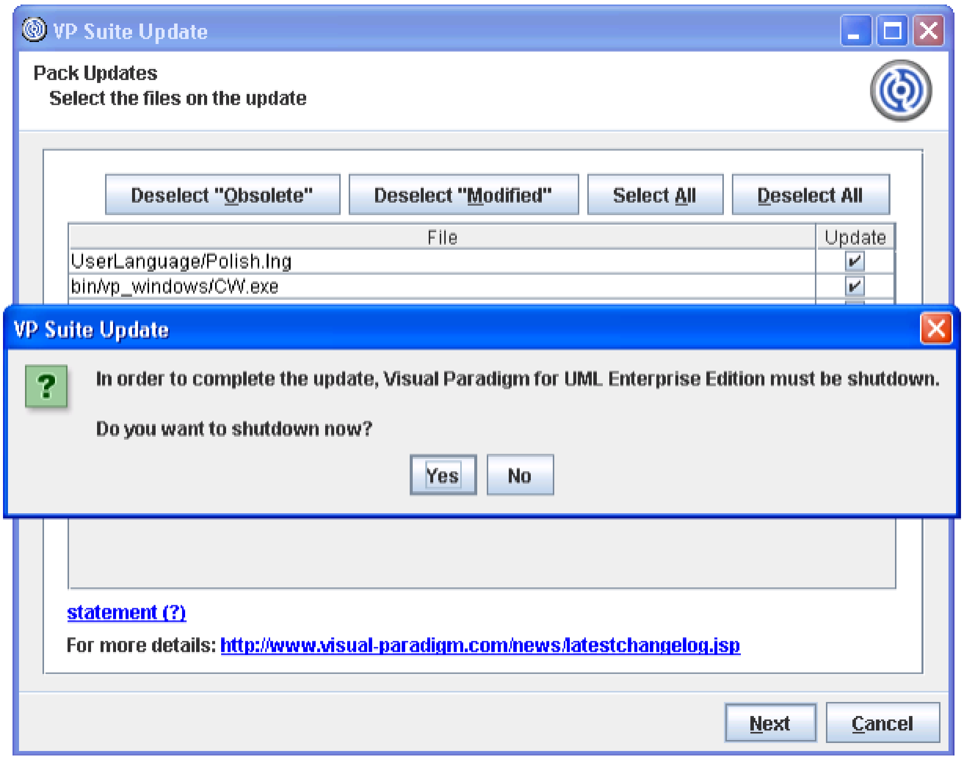
Figure 3-9 *Automatic update options*

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 VP-UML
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

Table 3-2 *Available options for automatic update*

3. Select **Update** now and click **OK** to proceed. This popup the **VP Suite Update**.



File	Update
UserLanguage/Polish.lng	☑
bin/vp_windows/CW.exe	☑

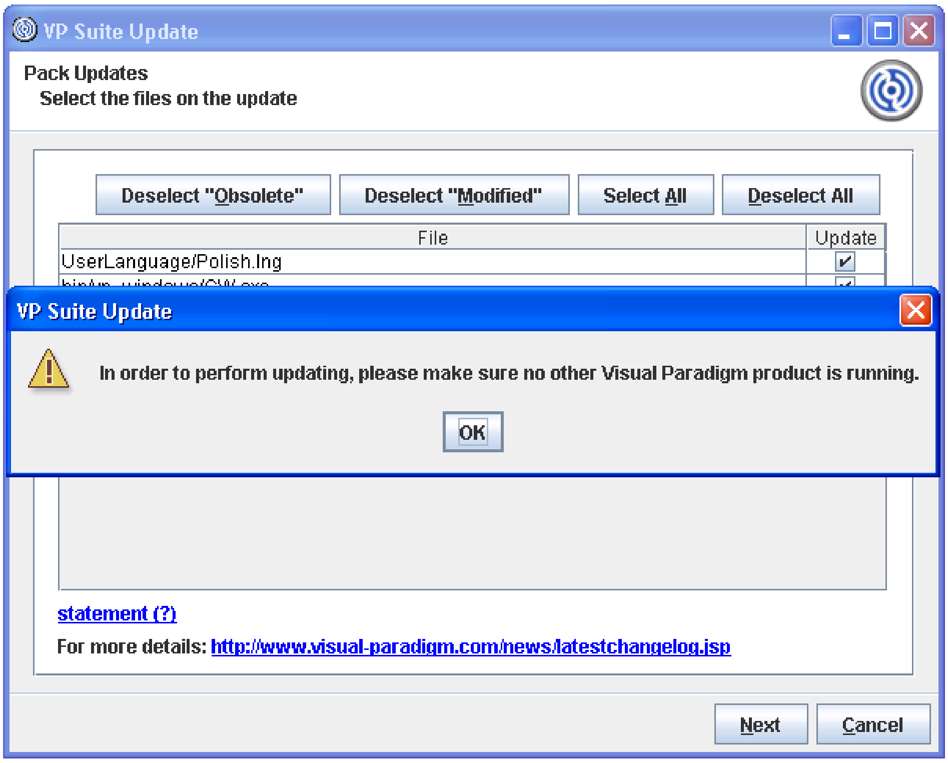
In order to complete the update, Visual Paradigm for UML Enterprise Edition must be shutdown.

Do you want to shutdown now?

For more details: http://www.visual-paradigm.com/news/latestchangelog.jsp

Figure 3-10 *To shutdown VP-UML*

4. In order to update, you need to close all the running VP application. Click **Yes** first, and then click **OK** in the dialog shown below to let VP Suite Update close the applications for you.



Pack Updates
Select the files on the update

File	Update
UserLanguage/Polish.lng	☑

In order to perform updating, please make sure no other Visual Paradigm product is running.

For more details: http://www.visual-paradigm.com/news/latestchangelog.jsp

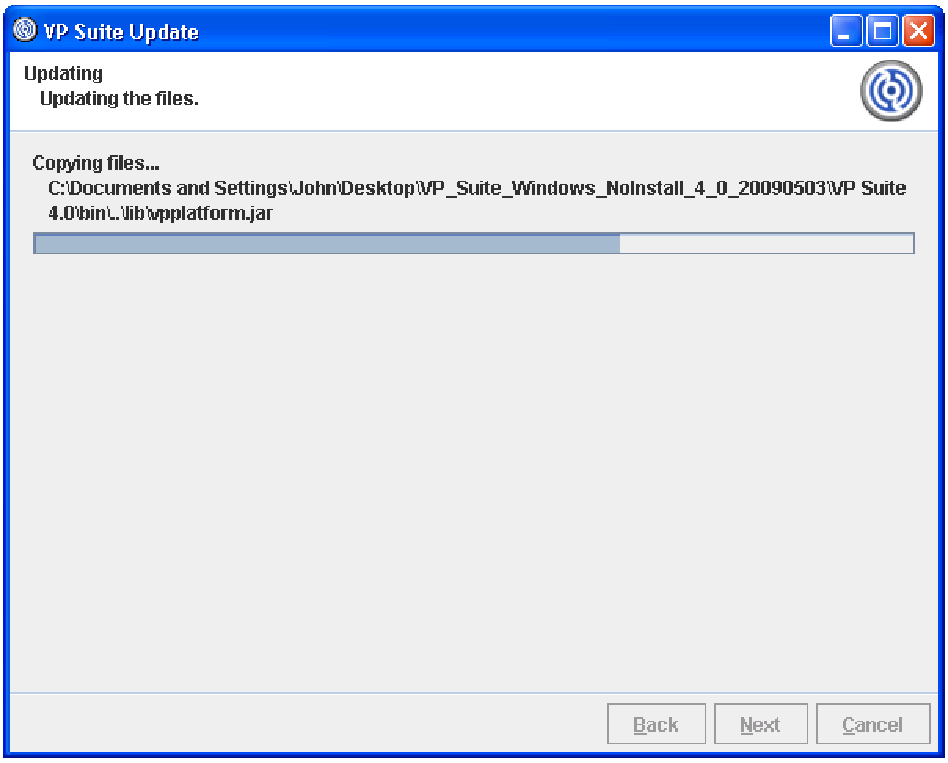
***Figure 3-11** Asked to close Visual Paradigm products*

5. Review the files to be updated. You may select or deselect files to be updated by checking and unchecking individual file listing in the page, under the **Update** column. Usually, you can keep the selection unchanged. But if you see the filenames appear in blue or red, you better confirm the selection carefully. Below is a description of red and blue colored filename:

Color	Description
Blue	Modified. This refers to files which are modified by the users and will be replaced with the original files
Red	Obsolete. This refers to files which are no longer useful in the new version of the product and will be deleted.

***Table 3-3** Description of blue and red colored filename in Pack Updates page*

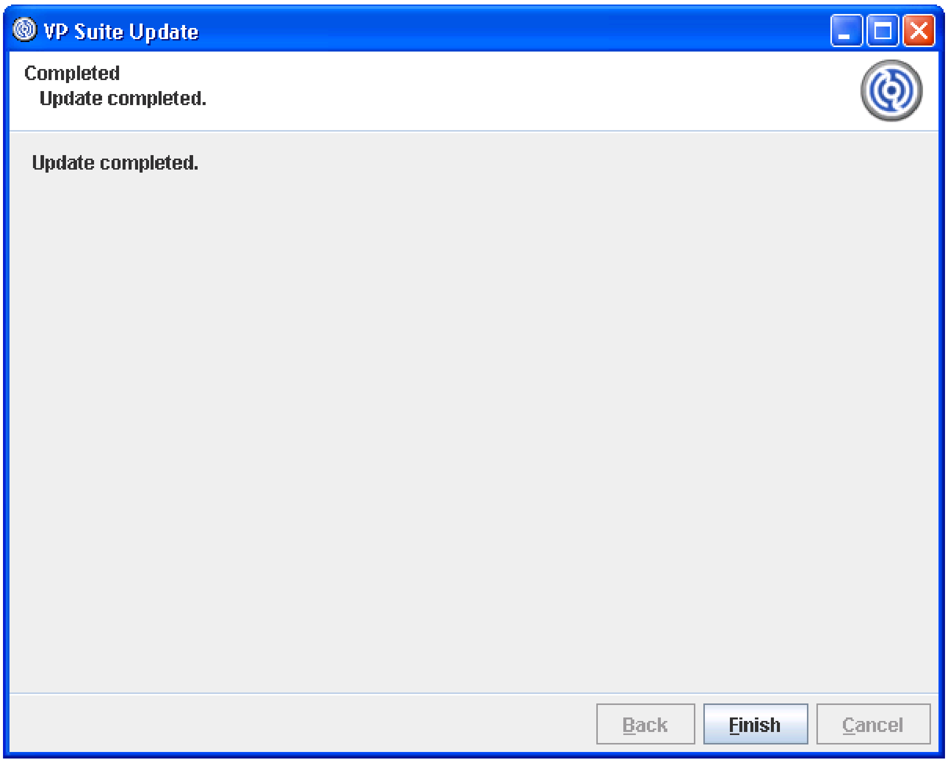
6. Click **Next** to proceed with updating the files.



Copying files...
C:\Documents and Settings\John\Desktop\VP_Suite_Windows_NoInstall_4_0_20090503\VP Suite 4.0\bin\..\lib\vpplatform.jar

Figure 3-12 *Update is in process*

7. When update is complete, click **Finish** to confirm.



Update completed.

Figure 3-13 *Update is complete*

Setting the Interval of Checking Updates

By default, update is checked weekly when starting VP-UML. You can change the interval of checking updates through the Options dialog box. To change:

1. Open the **Options** dialog box by selecting **Tools > Options...** from the main menu.

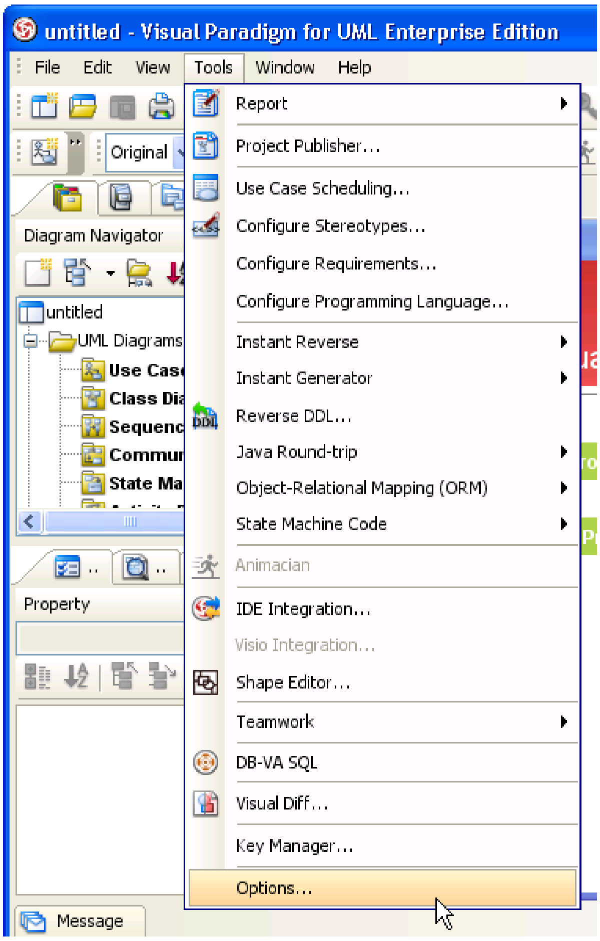
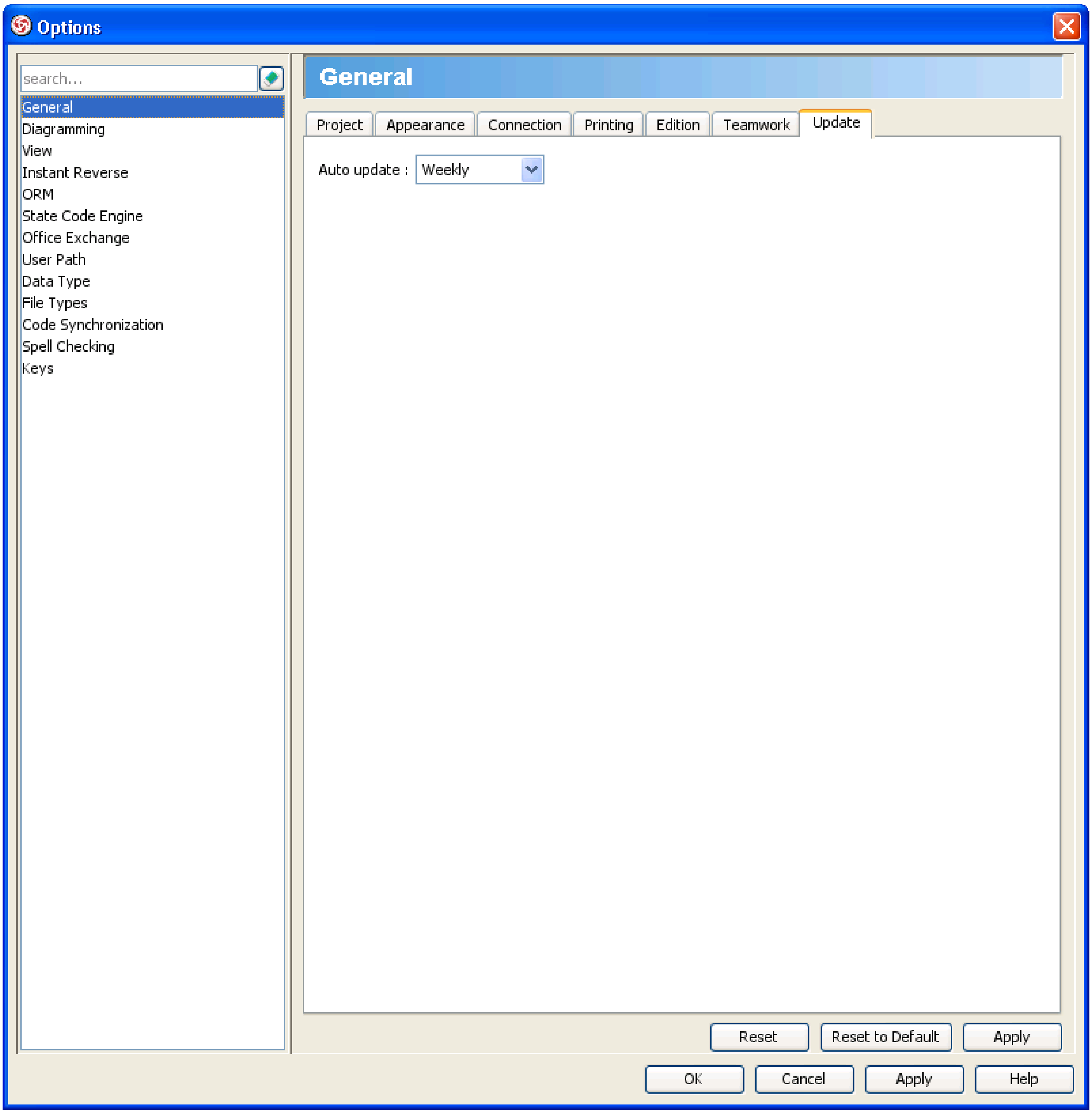


Figure 3-14 *Opening the Options dialog box*

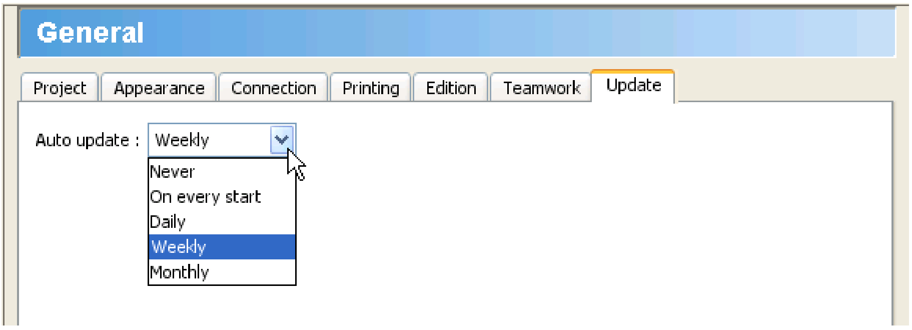
2. In the **Options** dialog box, select **General** from the list at the left hand side, then open the **Update** tab.



Auto update : Weekly

Figure 3-15 *Update page in Options dialog box*

3. Select the interval of performing auto update from the **Auto update** drop down menu.



Auto update : Weekly

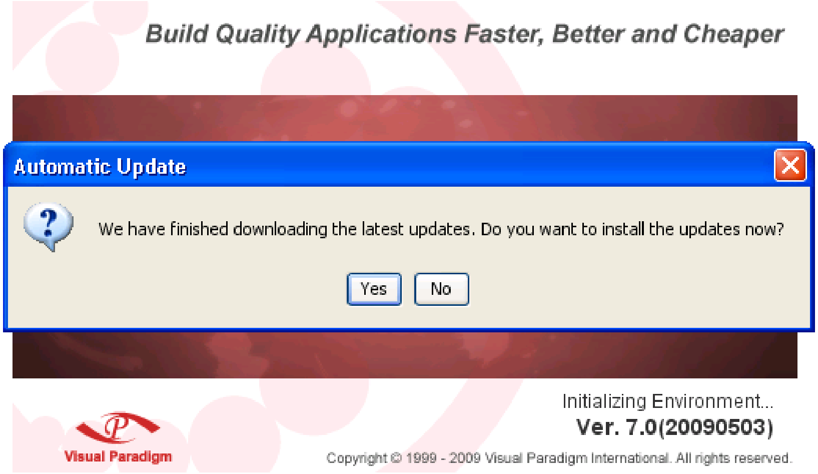
Figure 3-16 *Selecting update interval*

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 VP-UML
Daily	Check for product updates everyday, when starting VP-UML
Weekly	Check for product updates every week, when starting VP-UML
Monthly	Check for product updates every month, when starting VP-UML

Table 3-4 *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.


Build Quality Applications Faster, Better and Cheaper
Automatic Update
Yes
No
Initializing Environment...
Ver. 7.0(20090503)
Visual Paradigm
Copyright © 1999 - 2009 Visual Paradigm International. All rights reserved.

We have finished downloading the latest updates. Do you want to install the updates now?

Figure 3-17 *Prompting for update when starting VP-UML*

Introduction to Plugin Support

Plugin Support provides an interface for developers to integrate with VP-UML. Developers can develop their plugins for what they want. In this section, 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 VP-UML 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).

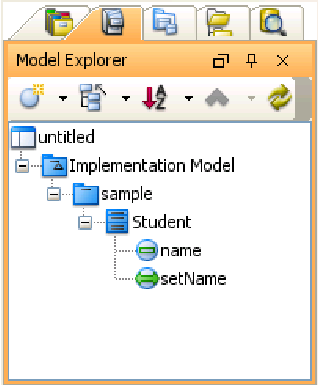


Figure 4-1 *- Model Elements under Model Explorer*

Diagram

Diagram is contains diagram elements on different domain (such as Use Case Diagram, Class Diagram, ERD, 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)

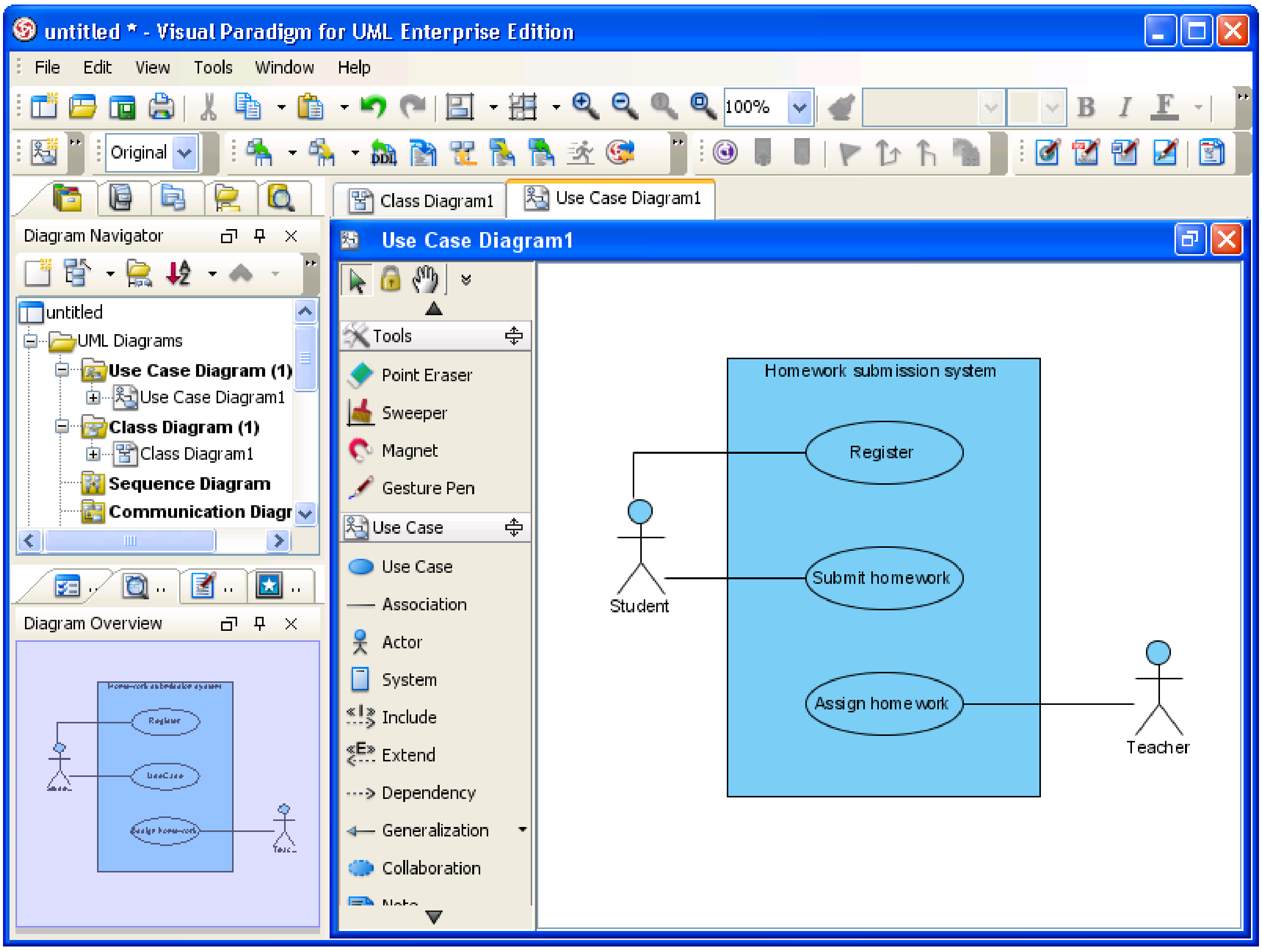


Figure 4-2 - *An opening Use Case Diagram*

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 Class). Connector represents the relationships (such as Generalization). 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).

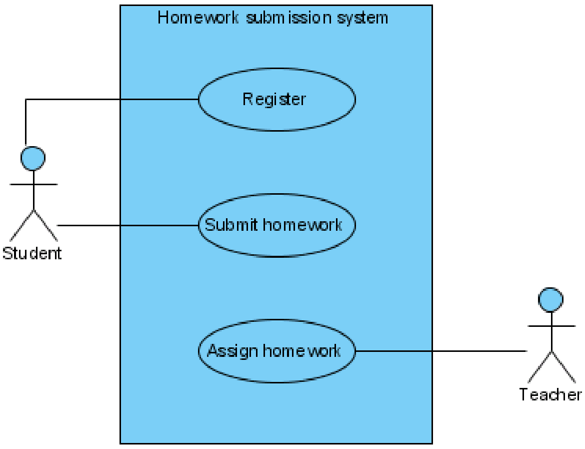


Figure 4-3 - *Shapes and connectors are both diagram elements*

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.

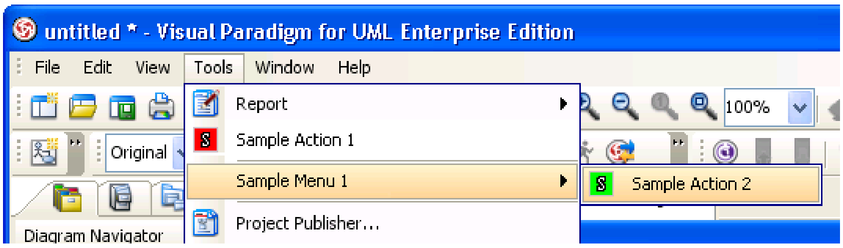


Figure 4-4 *- Menu with user-defined menus*

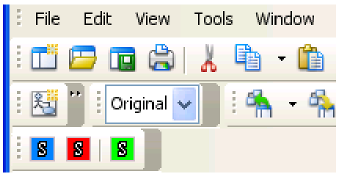


Figure 4-5 *- Toolbar with user-defined buttons*

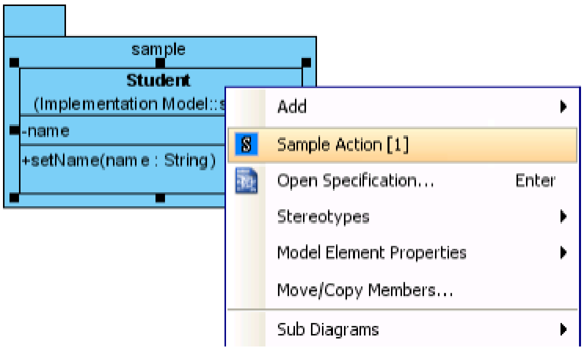


Figure 4-6 *- Popup menu with user-defined menus*

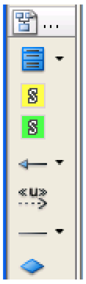


Figure 4-7 *- Diagram toolbar with user-defined buttons*

Implementing Plugin

Configuring Development Environment

Plugin Support API is placed on **%VP_SUITE%/lib/openapi.jar**. In order to working with VP-UML, 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 "VP_SUITE_HOME/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.

Table 4-1 *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%/plugins)of VP-UML from the VPPluginInfo.
        }
        public void unloaded() {
                // called when the plugin is unloaded (when the VP-UML 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.

Table 4-2 *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:

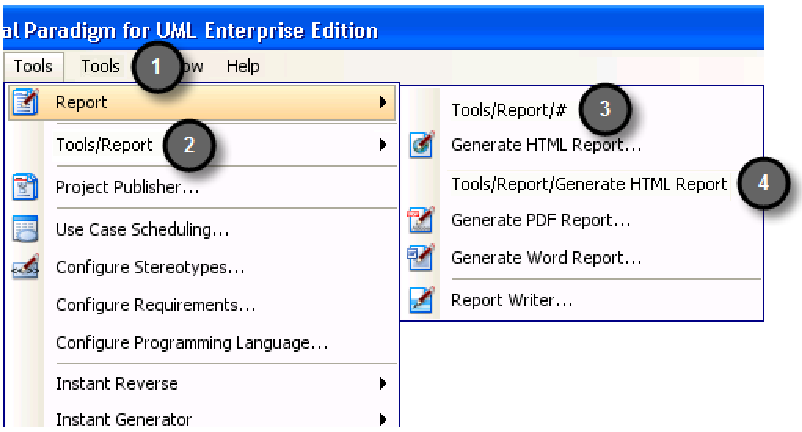


Figure 4-8 *- Custom Action on MenuBar*

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

Table 4-3 *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.VPActionCo**
separator (0..*)		Specified a separator on menu bar or toolbar.

Table 4-4 *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 updated(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**.

***Table 4-5** 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 updated(
                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.ClassDia Class" should be assigned. "com.vp.diagram.ClassDia 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,

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 VP-UML. 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 VP-UML. 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 VP-UML. 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 VP-UML. 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 VP-UML

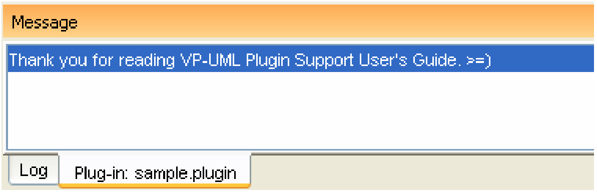
Since VP-UML may be integrated with different platforms which may not support Swing (e.g. Eclipse, Visual Studio). That may make to hang on the process if using the Swing dialog technology (e.g. JOptionPane and JDialog). So, there is necessary to use a special method to show the dialog with Swing technology.

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 VP-UML'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 VP-UML, 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.



Thank you for reading VP-UML Plugin Support User's Guide. >=)

Figure 4-9 *- Message in Message Pane*

Sample on Code:

```
// assume in a code segment
ViewManager viewManager = ApplicationManager.instance().getViewManager();
viewManager.showMessage( "Thank you for reading VP-UML 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 VP-UML 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; }
        }
}
```

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 VP-UML.

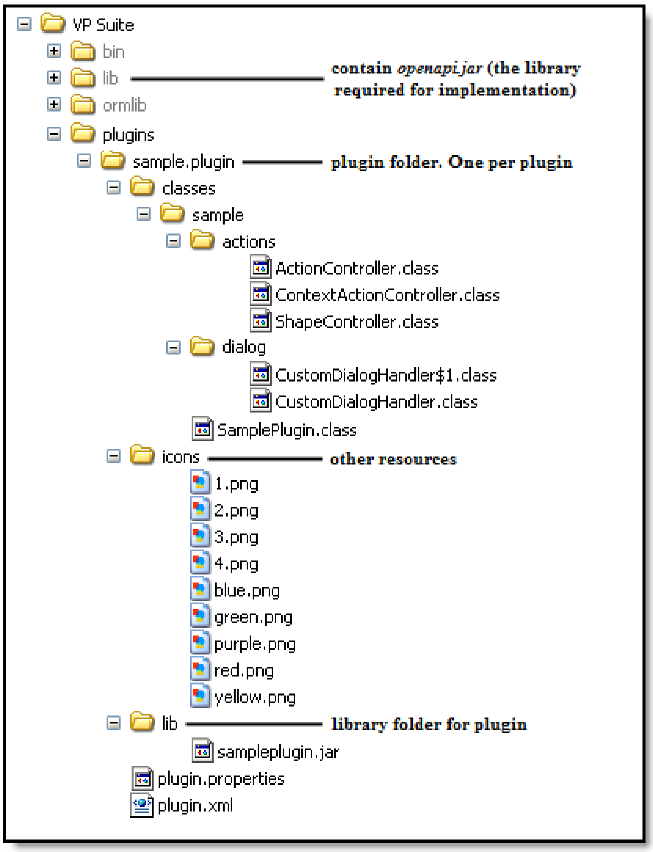
First, create a folder named **plugins** (notice the ' **s**') in the VP-Suite directory. Put the plugin files into "%VP-SUITE%\plugins\%PLUGIN_ID%\". %PLUGIN_ID% is a directory named as the plugin id (use the id as the directory name to avoid duplicated directories defined in **plugins**)

The following structure should be obtained:

%VP_SUITE%

```
        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.



```
VP Suite
  bin
  lib
  ormlib
  plugins
    sample.plugin
      classes
        sample
          actions
              ActionController.class
              ContextActionController.class
              ShapeController.class
          dialog
              CustomDialogHandler$1.class
              CustomDialogHandler.class
            SamplePlugin.class
      icons
          1.png
          2.png
          3.png
          4.png
          blue.png
          green.png
          purple.png
          red.png
          yellow.png
      lib
          sampleplugin.jar
      plugin.properties
      plugin.xml
```

contain *openapi.jar* (the library required for implementation)

plugin folder. One per plugin

other resources

library folder for plugin

Figure 4-10 *- Plugin folder structure*

After all, restart VP-UML 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.

Exporting Diagram Image

To export images from a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.

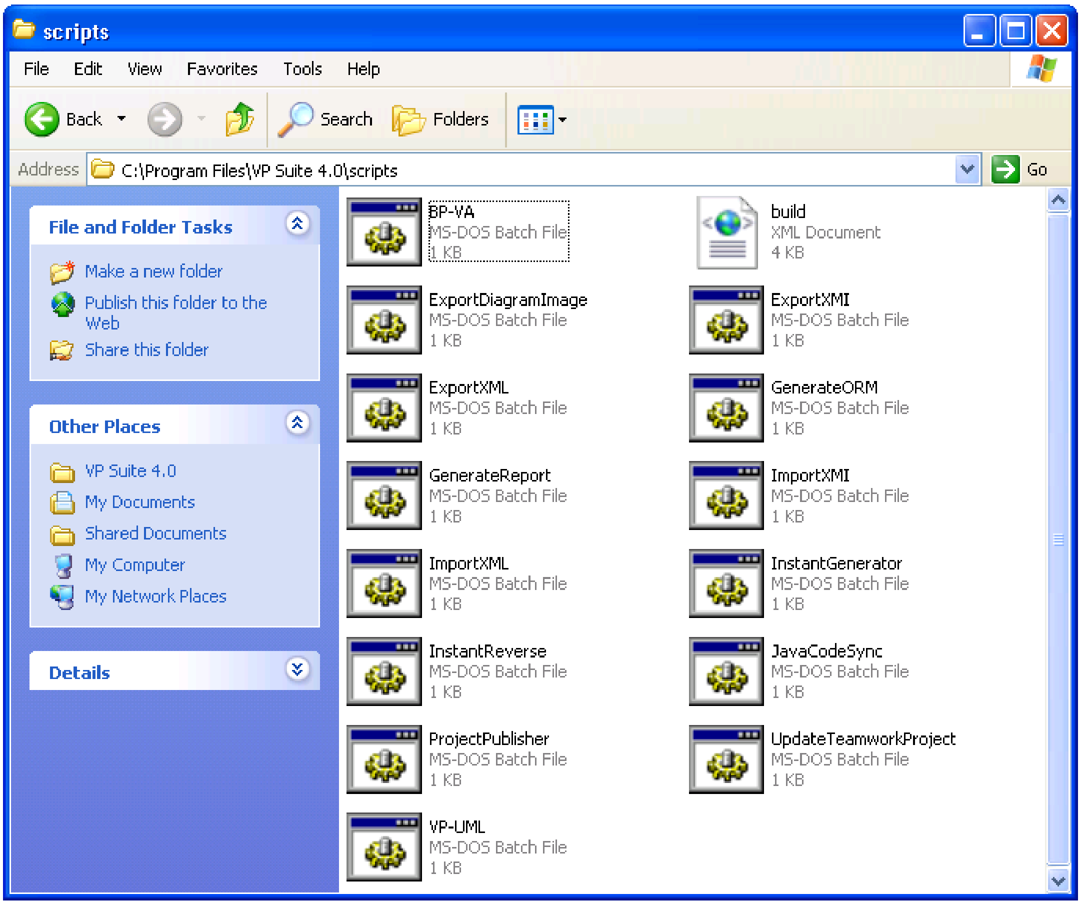

scripts
File
Edit
View
Favorites
Tools
Help
Back
Search
Folders
Address
C:\Program Files\VP Suite 4.0\scripts
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
VP Suite 4.0
My Documents
Shared Documents
My Computer
My Network Places
Details
BP-VA MS-DOS Batch File 1 KB
build XML Document 4 KB
ExportDiagramImage MS-DOS Batch File 1 KB
ExportXMI MS-DOS Batch File 1 KB
ExportXML MS-DOS Batch File 1 KB
GenerateORM MS-DOS Batch File 1 KB
GenerateReport MS-DOS Batch File 1 KB
ImportXMI MS-DOS Batch File 1 KB
ImportXML MS-DOS Batch File 1 KB
InstantGenerator MS-DOS Batch File 1 KB
InstantReverse MS-DOS Batch File 1 KB
JavaCodeSync MS-DOS Batch File 1 KB
ProjectPublisher MS-DOS Batch File 1 KB
UpdateTeamworkProject MS-DOS Batch File 1 KB
VP-UML MS-DOS Batch File 1 KB

Figure 5-1 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **ExportDiagramImage** and paste to the bin folder of VP Suite installation directory.

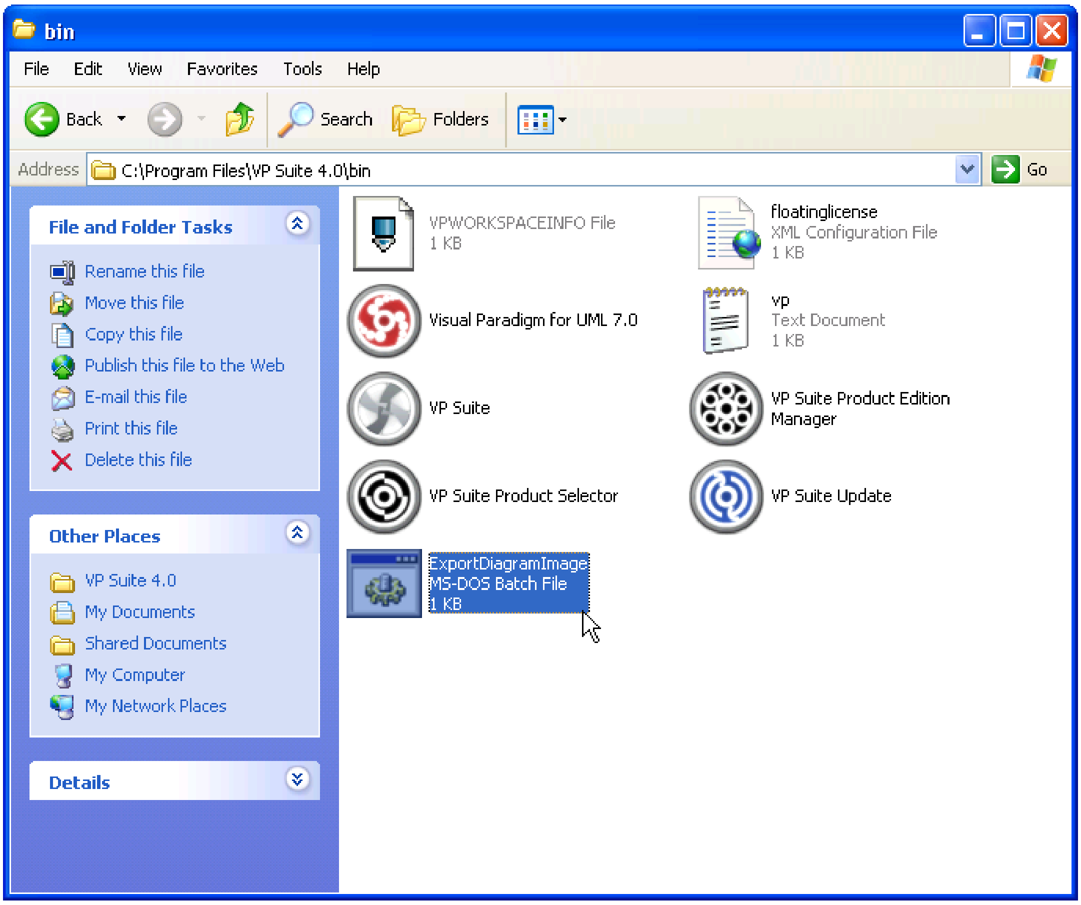
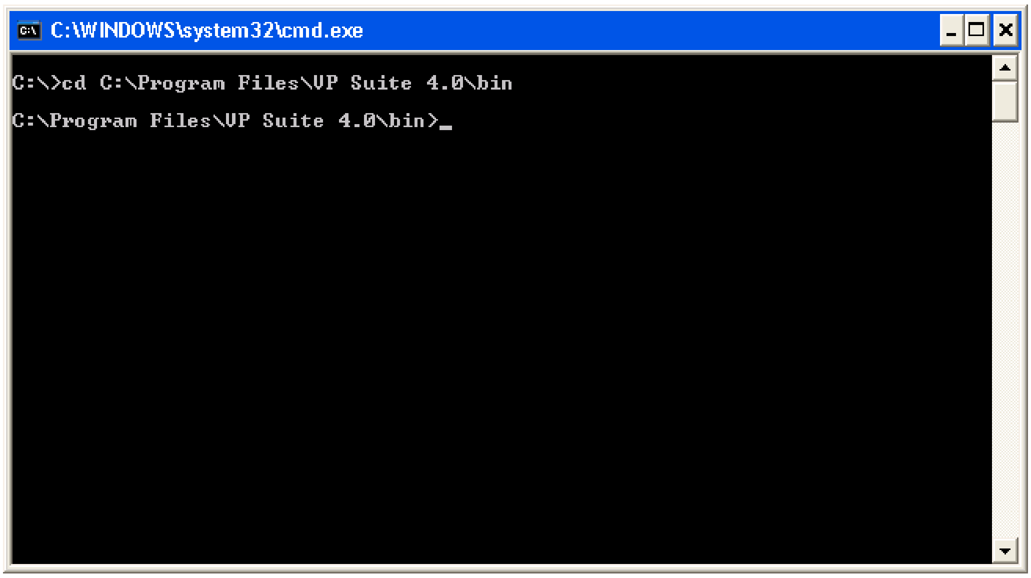


Figure 5-2 Copy and paste ExportDiagramImage from scripts folder to bin folder

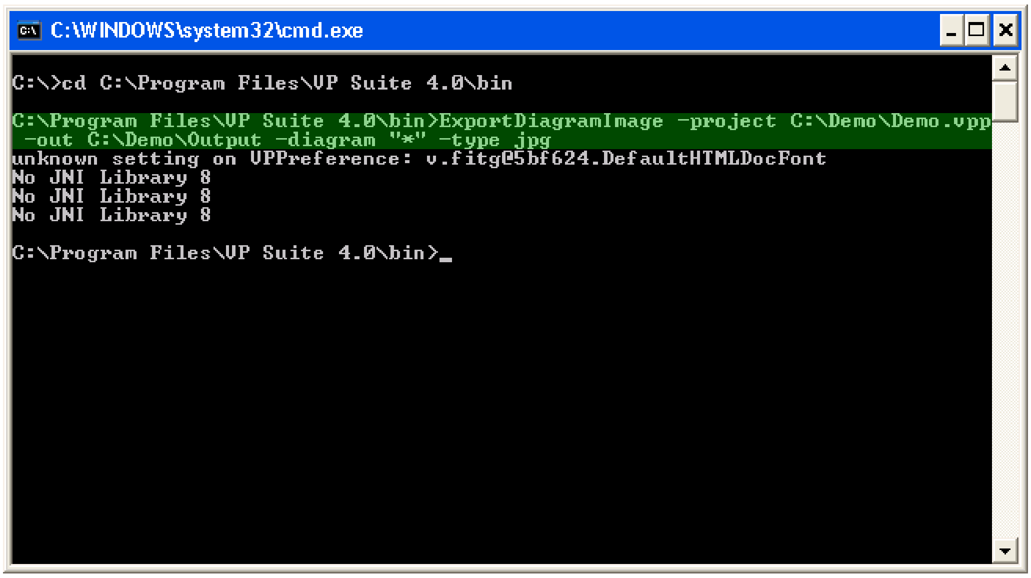
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-3 Navigate to the bin folder of VP Suite in command prompt

5. Execute the script by supplying the required parameters. For example:
ExportDiagramImage -project C:\Demo\Demo.vpp -out C:\Demo\Output -diagram "" -type jpg*



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>ExportDiagramImage -project C:\Demo\Demo.vpp
 -out C:\Demo\Output -diagram "*" -type jpg
unknown setting on VPPreference: v.fitg@5bf624.DefaultHTMLDocFont
No JNI Library 8
No JNI Library 8
No JNI Library 8

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-4 *Executing ExportDiagramImage*

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	diagram_1 diagram_2
-type [optional]	Type of diagrams. Here are the possible types: • png • png_with_background • jpg • svg • pdf	png

Table 5-1 *Parameters for ExportDiagramImage*

Exporting and Import XMI

Exporting XMI

To export XMI from a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.

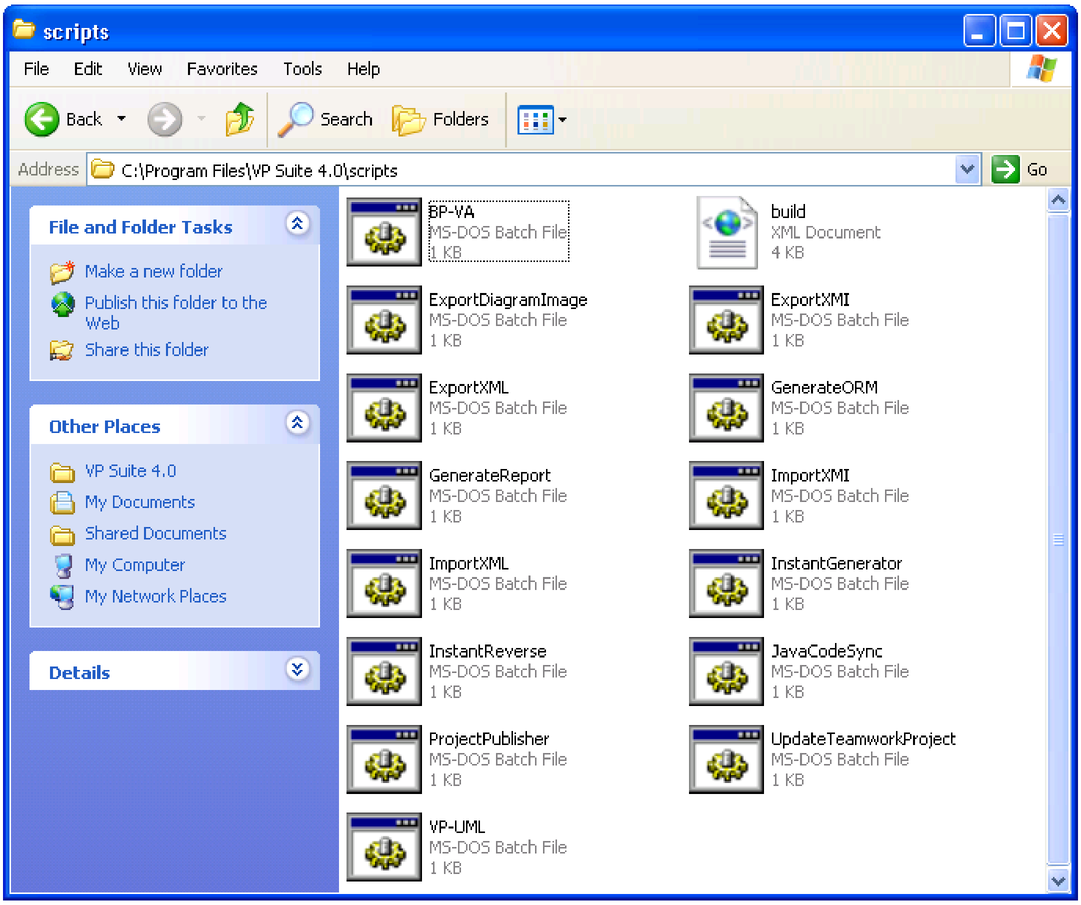


Figure 5-5 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **ExportXMI** and paste to the bin folder of VP Suite installation directory.

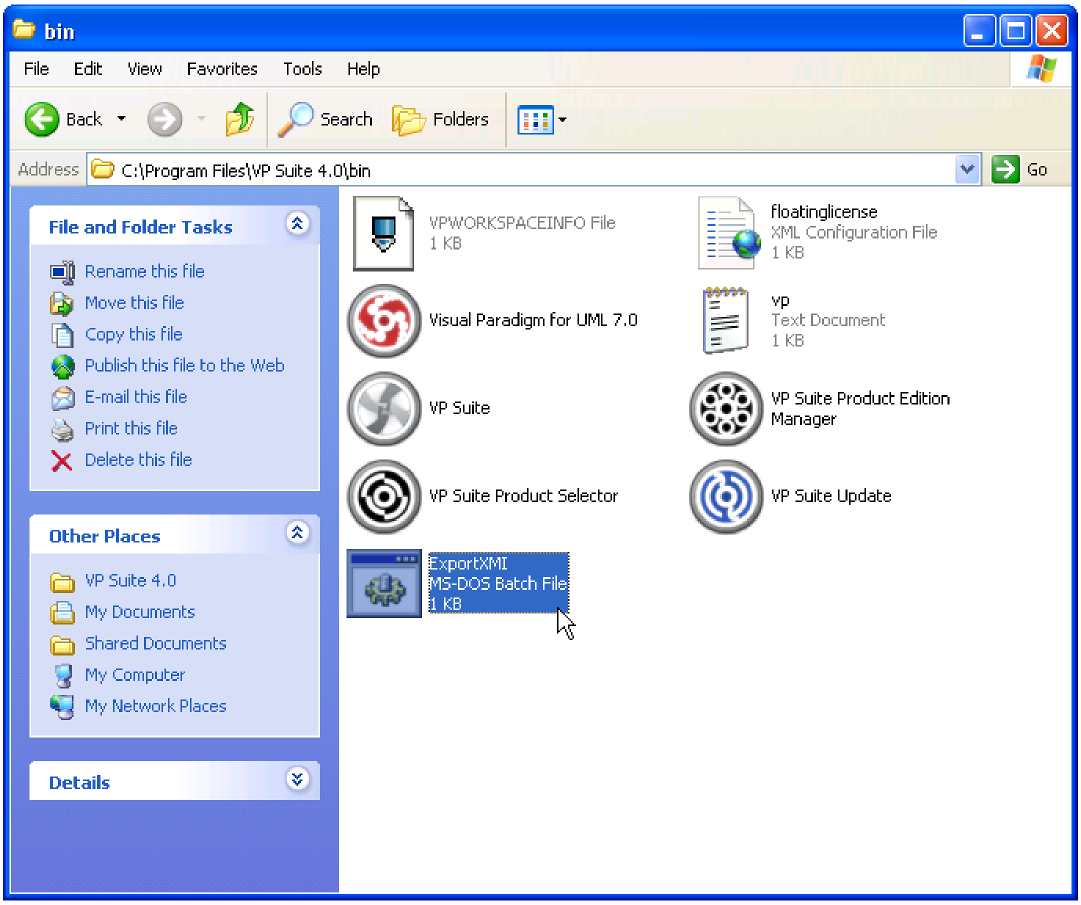
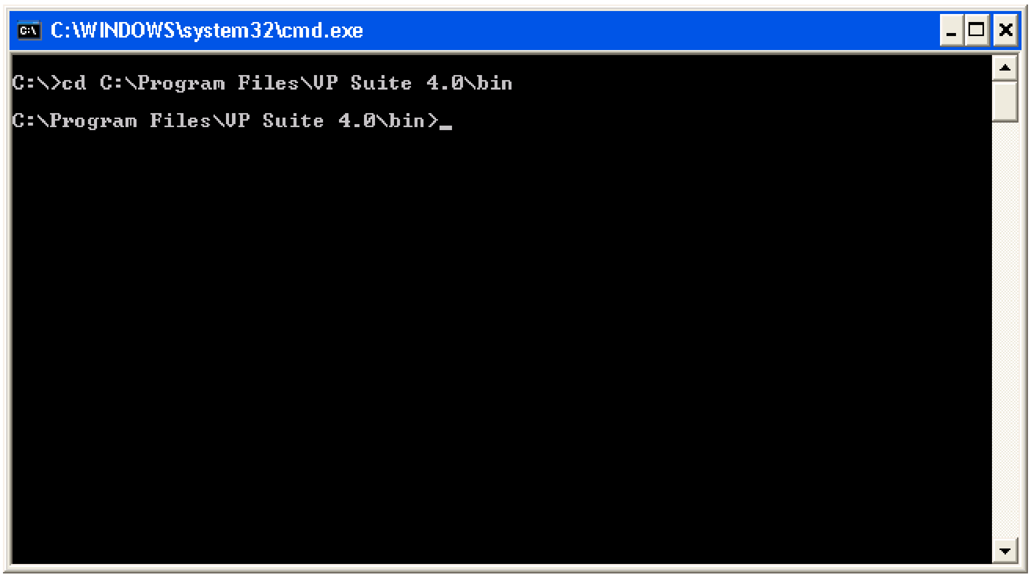


Figure 5-6 Copy and paste ExportXMI from scripts folder to bin folder

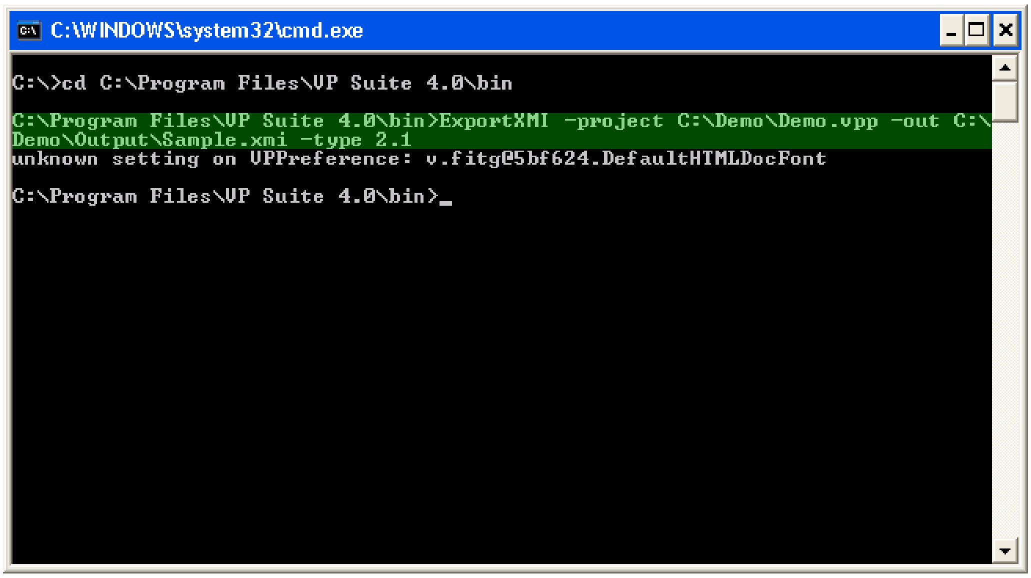
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-7 Navigate to the bin folder of VP Suite in command prompt

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



```
C:\WINDOWS\system32\cmd.exe

C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>ExportXMI -project C:\Demo\Demo.vpp -out C:\
Demo\Output\Sample.xmi -type 2.1
unknown setting on VPPreference: v.fitg@5bf624.DefaultHTMLDocFont

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-8 *Executing ExportXMI*

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	

Table 5-2 *Parameters for ExportDiagramImage*

Importing XMI

To import XMI to a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.

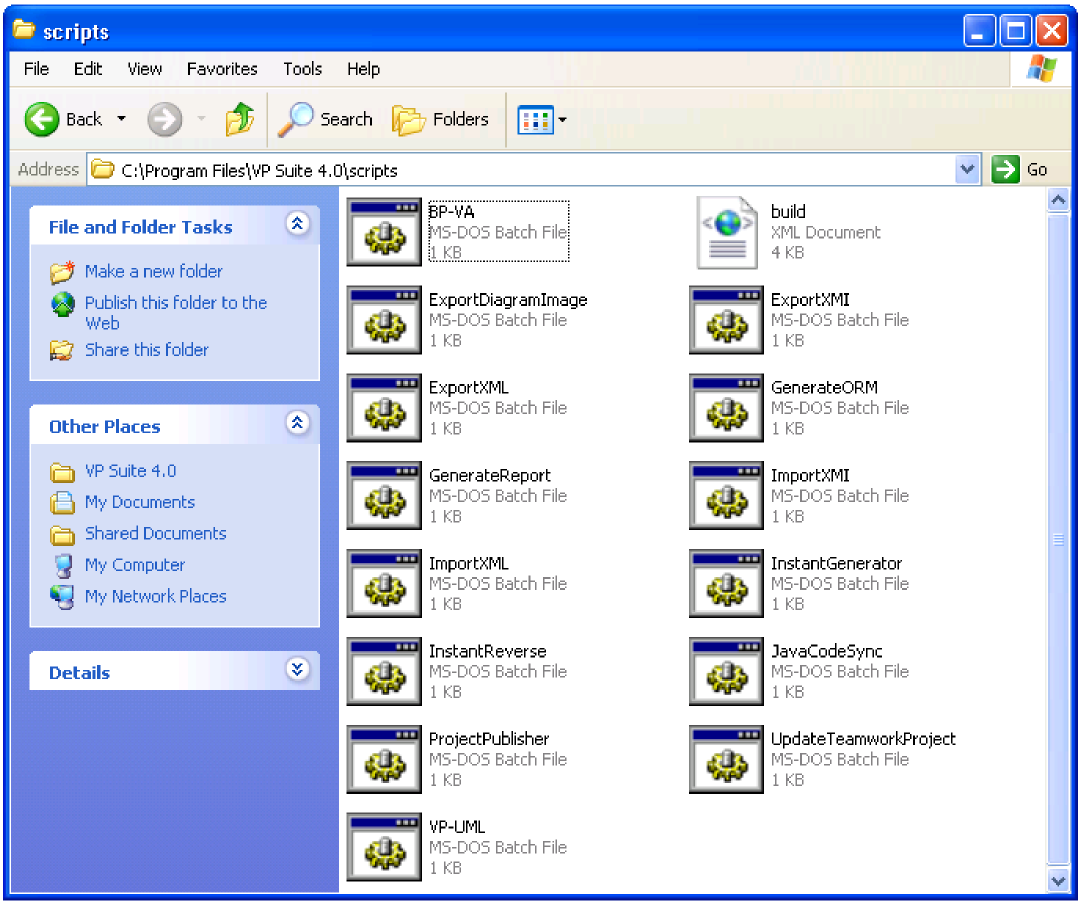


Figure 5-9 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **ImportXMI** and paste to the bin folder of VP Suite installation directory.

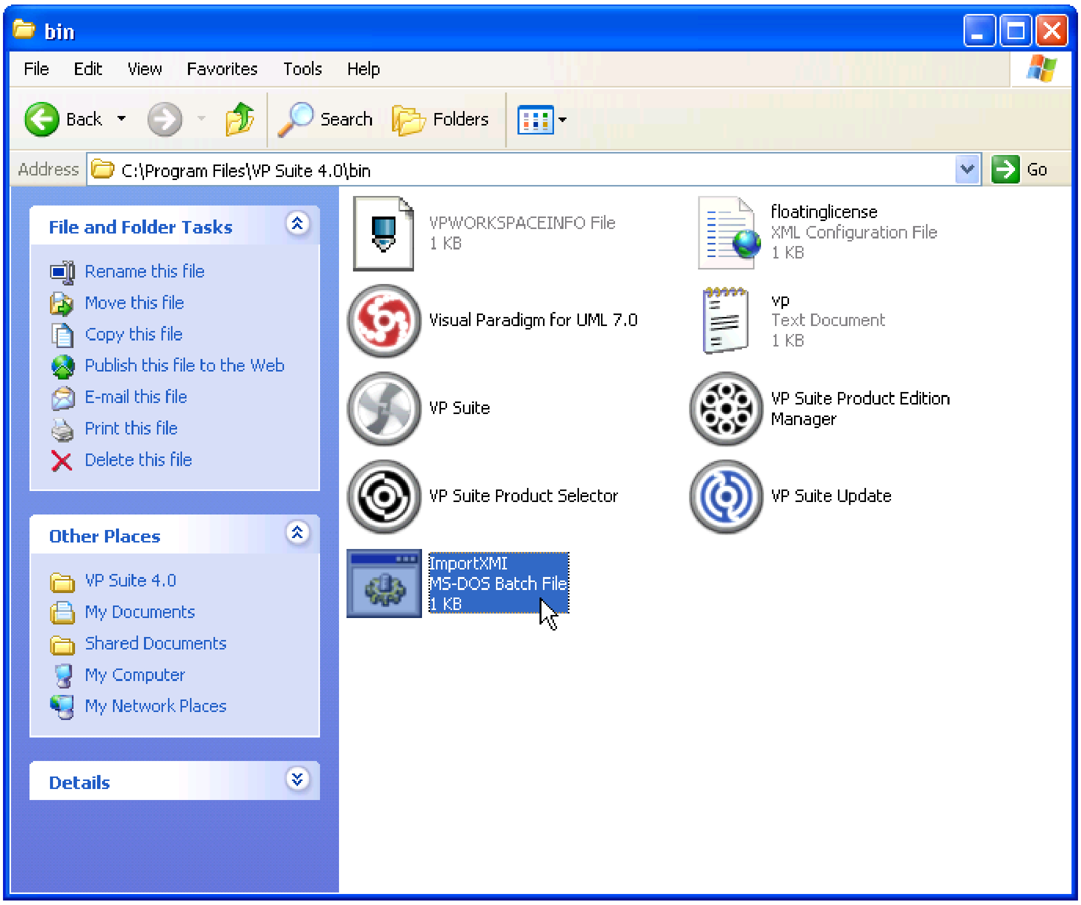
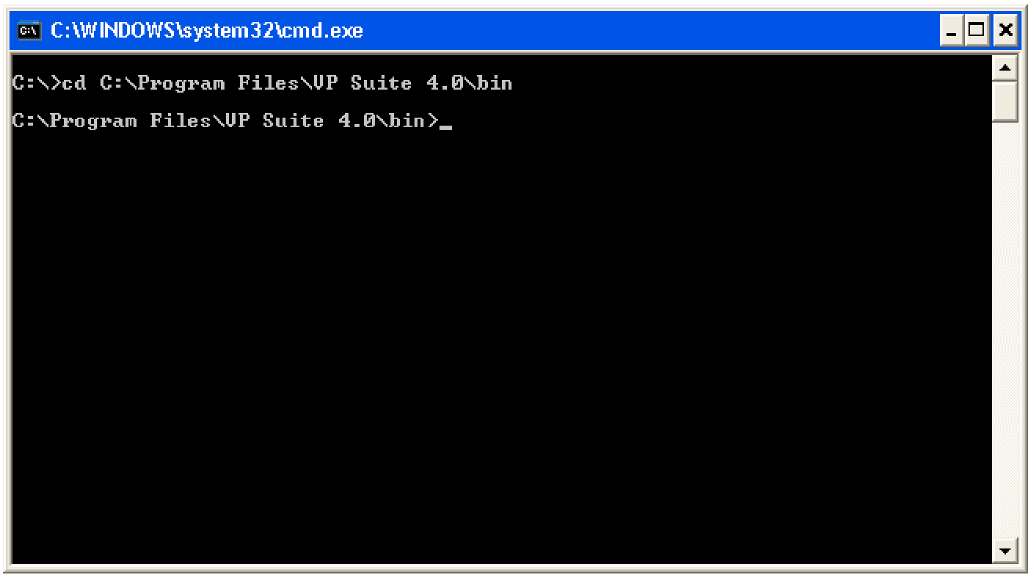


Figure 5-10 Copy and paste ImportXMI from scripts folder to bin folder

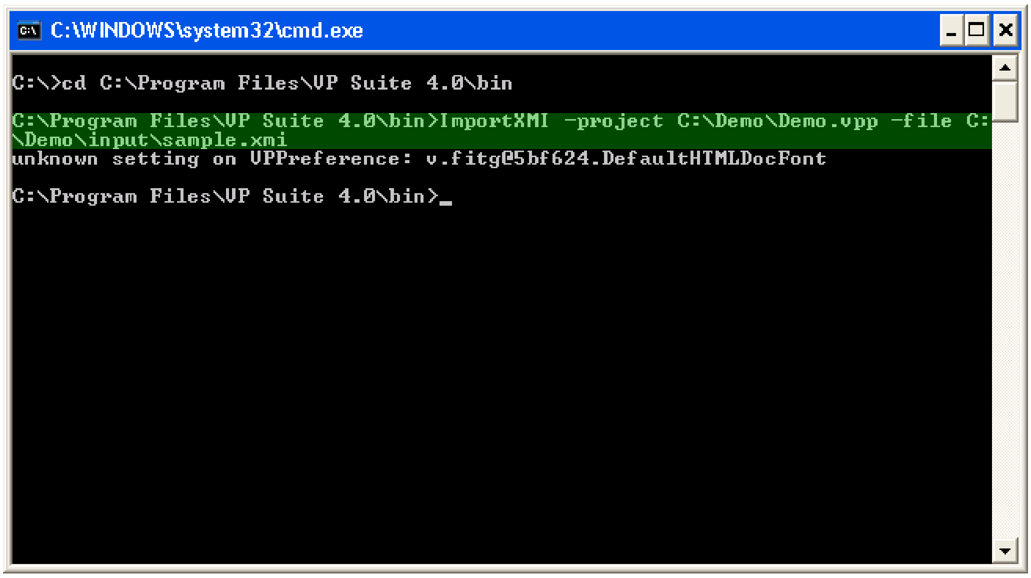
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-11 Navigate to the bin folder of VP Suite in command prompt

5. Execute the script by supplying the required parameters. For example:
ImportXMI -project C:\Demo\Demo.vpp -file C:\Demo\input\sample.xmi



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>ImportXMI -project C:\Demo\Demo.vpp -file C:
\Demo\input\sample.xmi
unknown setting on VPPreference: v.fitg@5bf624.DefaultHTMLDocFont

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-12 *Executing ImportXMI*

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

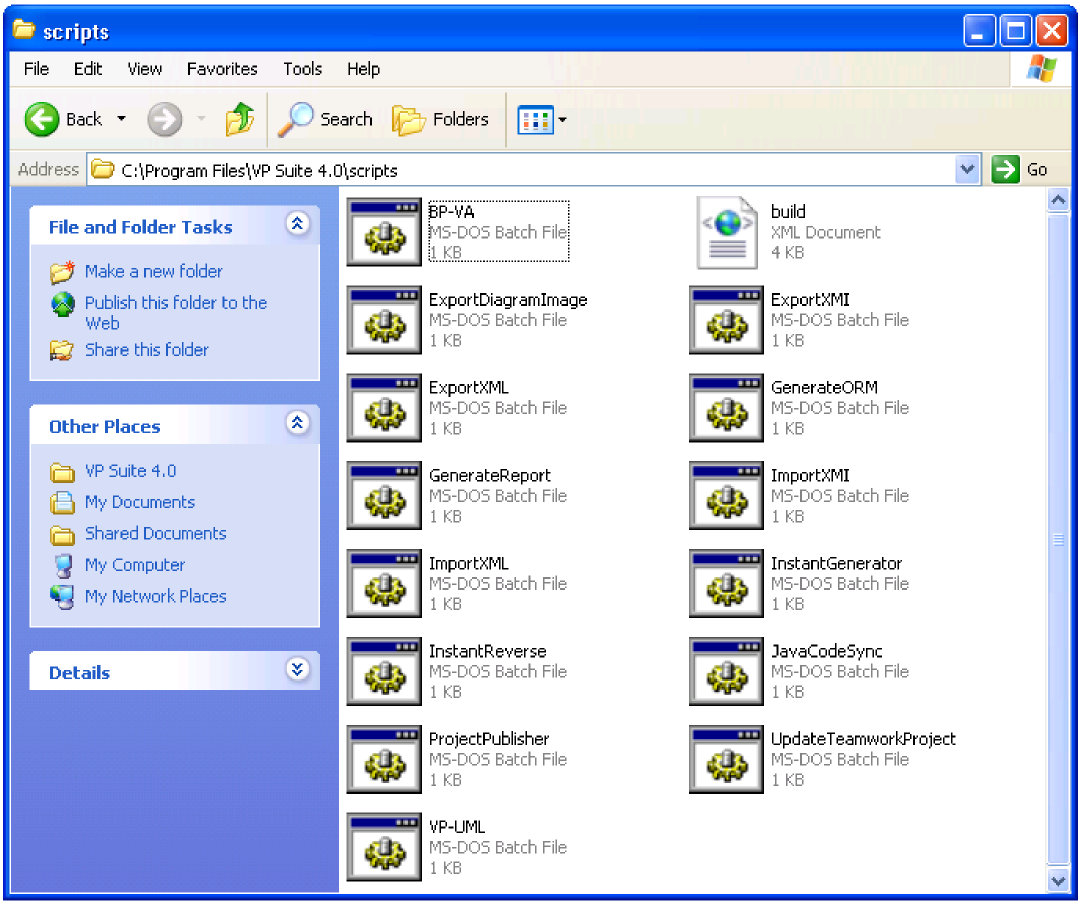
Table 5-3 *Parameters for ImportXMI*

Exporting and Import XML

Exporting XML

To export XML and images from a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.



C:\Program Files\VP Suite 4.0\scripts

Figure 5-13 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **ExportXML** and paste to the bin folder of VP Suite installation directory.

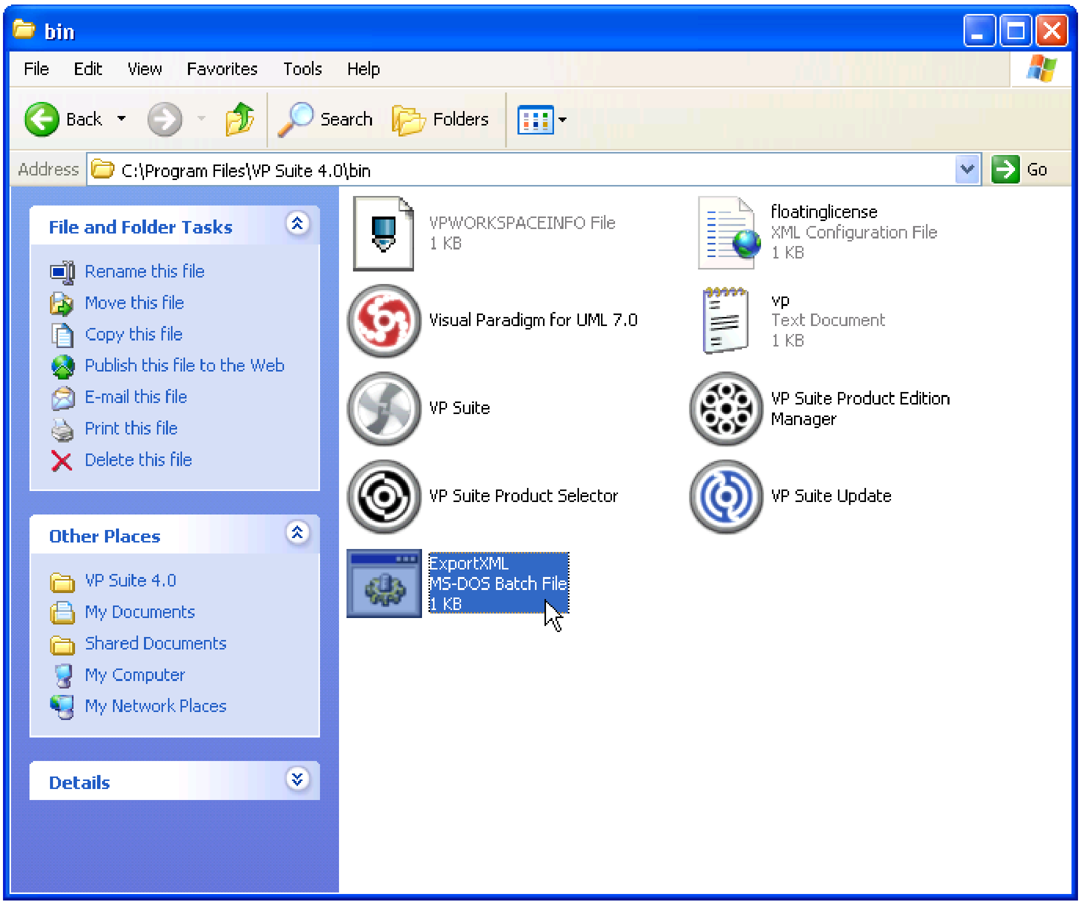
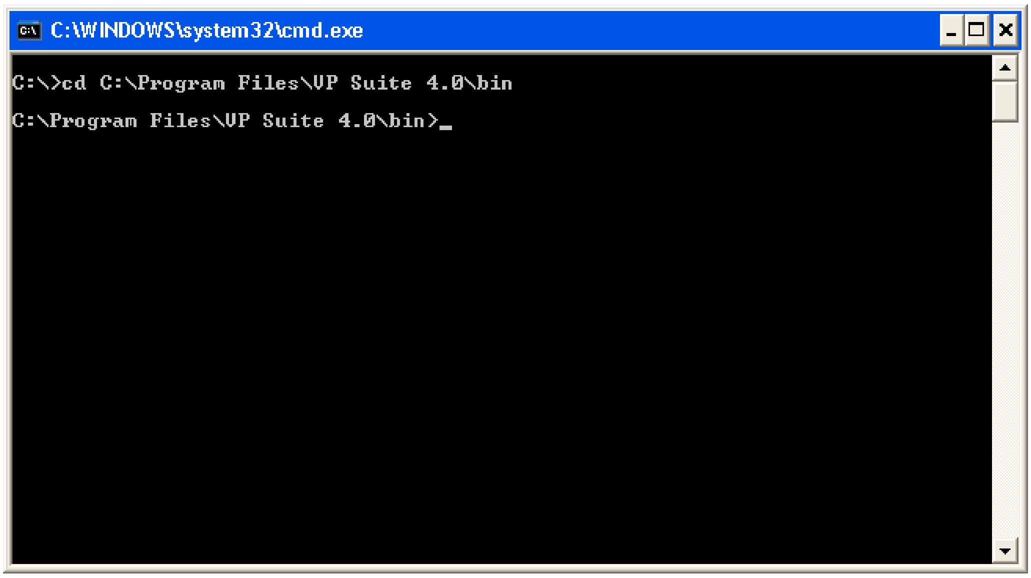


Figure 5-14 Copy and paste ExportXML from scripts folder to bin folder

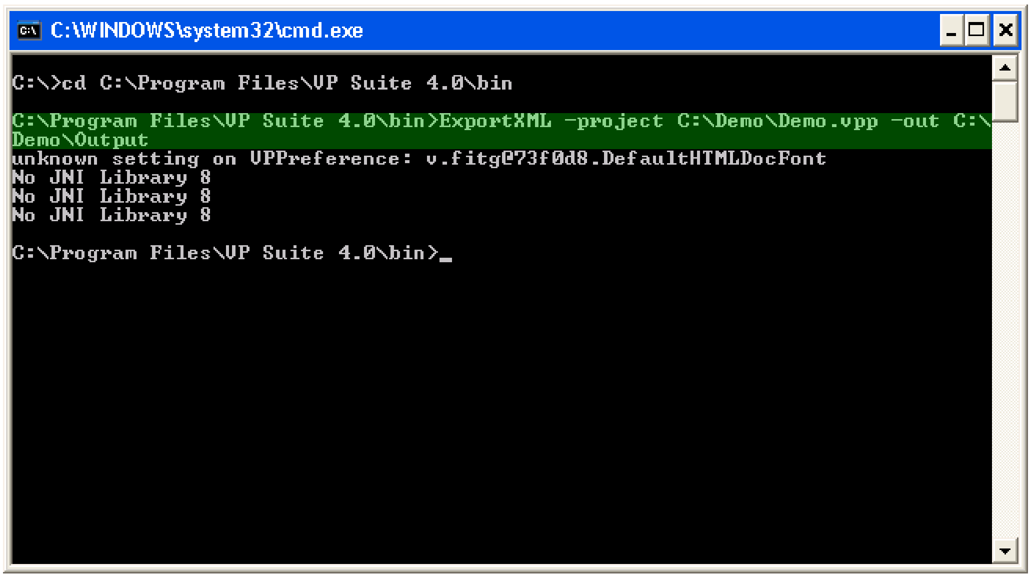
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-15 Navigate to the bin folder of VP Suite in command prompt

5. Execute the script by supplying the required parameters. For example:
ExportXML -project C:\Demo\Demo.vpp -out C:\Demo\Output



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>ExportXML -project C:\Demo\Demo.vpp -out C:\
Demo\Output
unknown setting on VPPreference: v.fitg@73f0d8.DefaultHTMLDocFont
No JNI Library 8
No JNI Library 8
No JNI Library 8

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-16 *Executing ExportXML*

Below is a description of parameters:

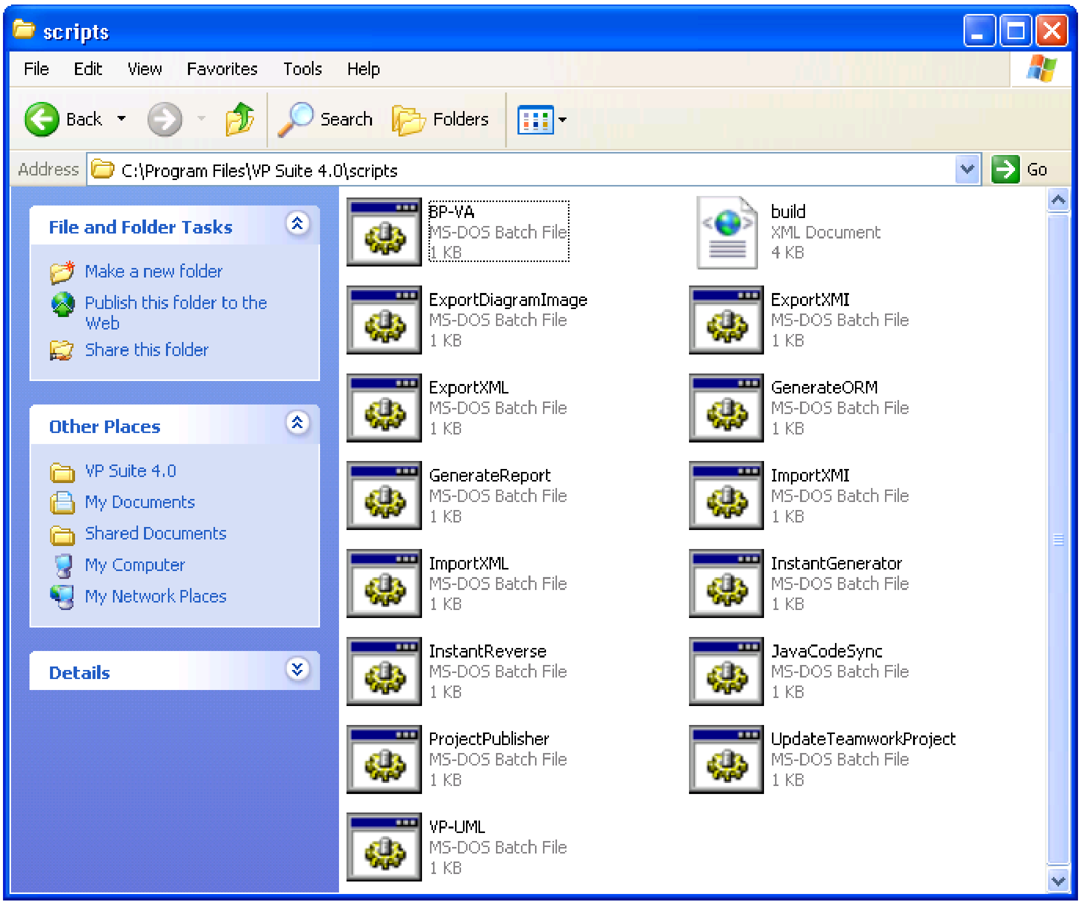
Parameter	Description	Example
-project	Project path	C:\Demo \Demo.vpp
-out	Folder for storing the exported XML and images	C:\Demo \Output

Table 5-4 *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.



Address C:\Program Files\VP Suite 4.0\scripts

BP-VA
MS-DOS Batch File
1 KB

build
XML Document
4 KB

ExportDiagramImage
MS-DOS Batch File
1 KB

ExportXMI
MS-DOS Batch File
1 KB

ExportXML
MS-DOS Batch File
1 KB

GenerateORM
MS-DOS Batch File
1 KB

GenerateReport
MS-DOS Batch File
1 KB

ImportXMI
MS-DOS Batch File
1 KB

ImportXML
MS-DOS Batch File
1 KB

InstantGenerator
MS-DOS Batch File
1 KB

InstantReverse
MS-DOS Batch File
1 KB

JavaCodeSync
MS-DOS Batch File
1 KB

ProjectPublisher
MS-DOS Batch File
1 KB

UpdateTeamworkProject
MS-DOS Batch File
1 KB

VP-UML
MS-DOS Batch File
1 KB

Figure 5-17 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **ImportXML** and paste to the bin folder of VP Suite installation directory.

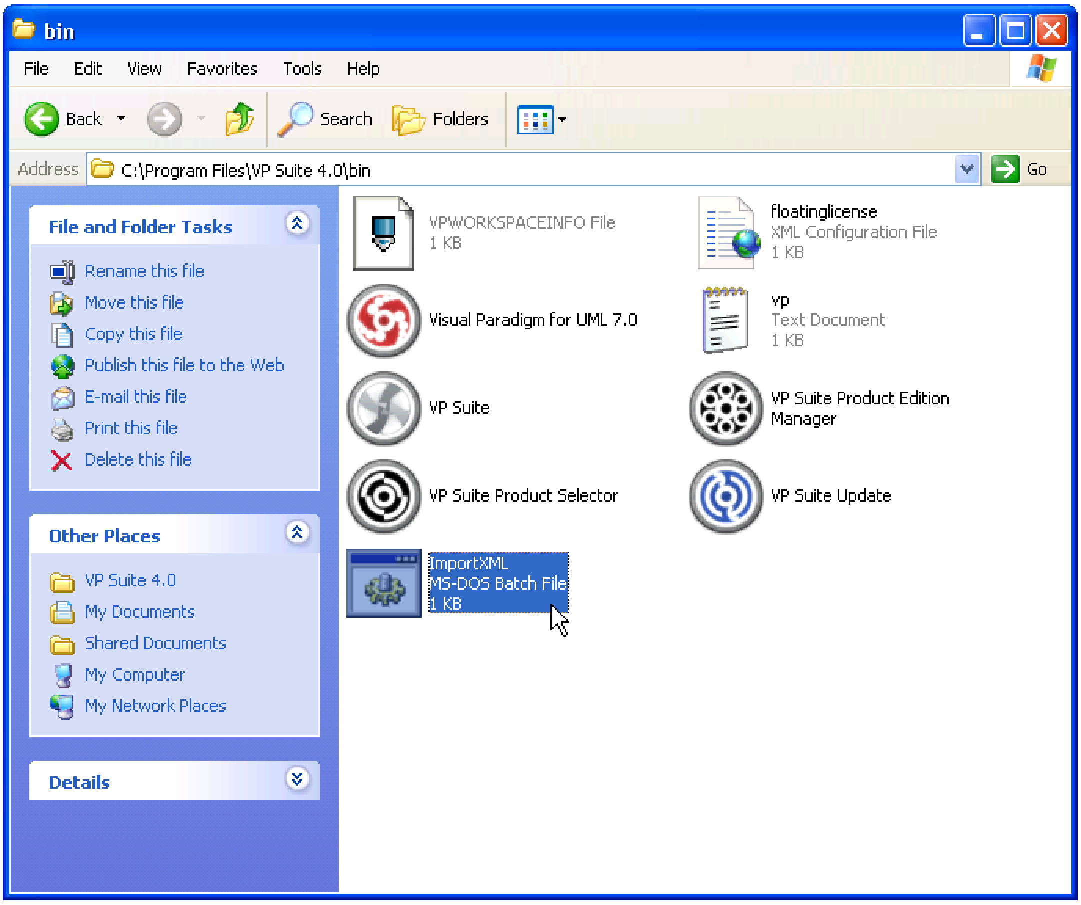
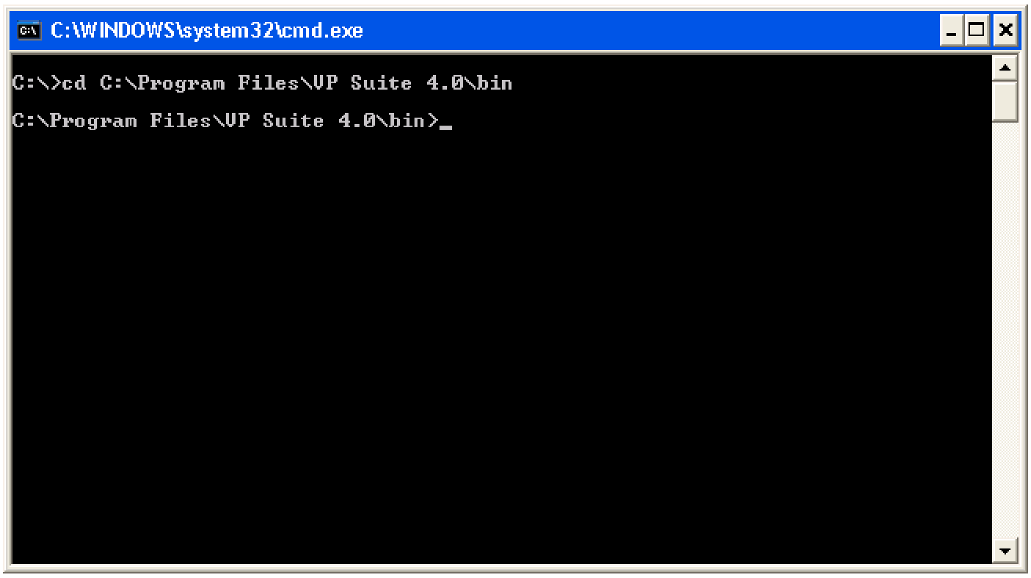


Figure 5-18 *Copy and paste ImportXML from scripts folder to bin folder*

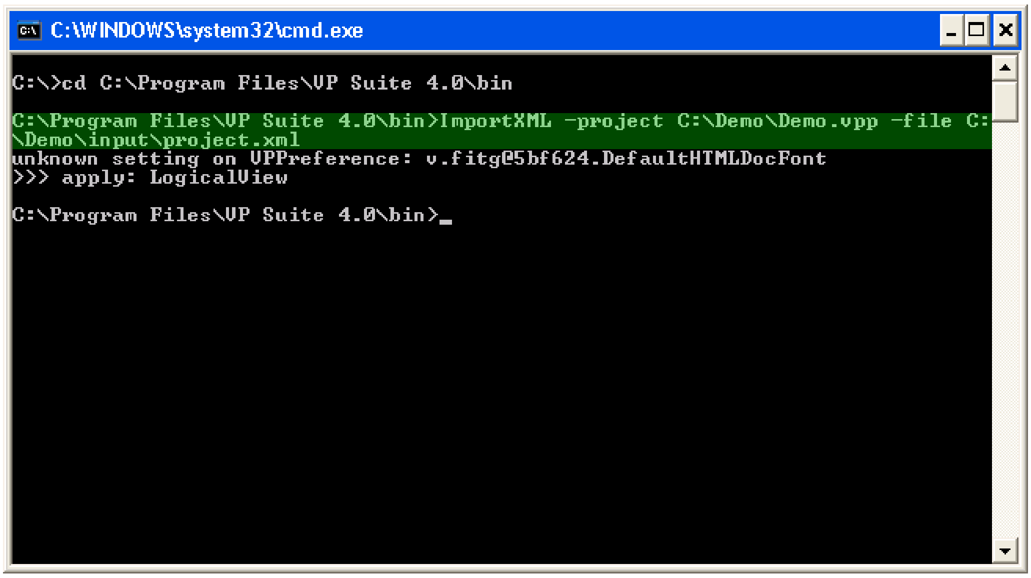
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-19 *Navigate to the bin folder of VP Suite in command prompt*

5. Execute the script by supplying the required parameters. For example:
ImportXML -project C:\Demo\Demo.vpp -file C:\Demo\input\project.xml



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>ImportXML -project C:\Demo\Demo.vpp -file C:
\Demo\input\project.xml
unknown setting on VPPreference: v.fitg@5bf624.DefaultHTMLDocFont
>>> apply: LogicalView

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-20 *Executing ImportXML*

Below is a description of parameters:

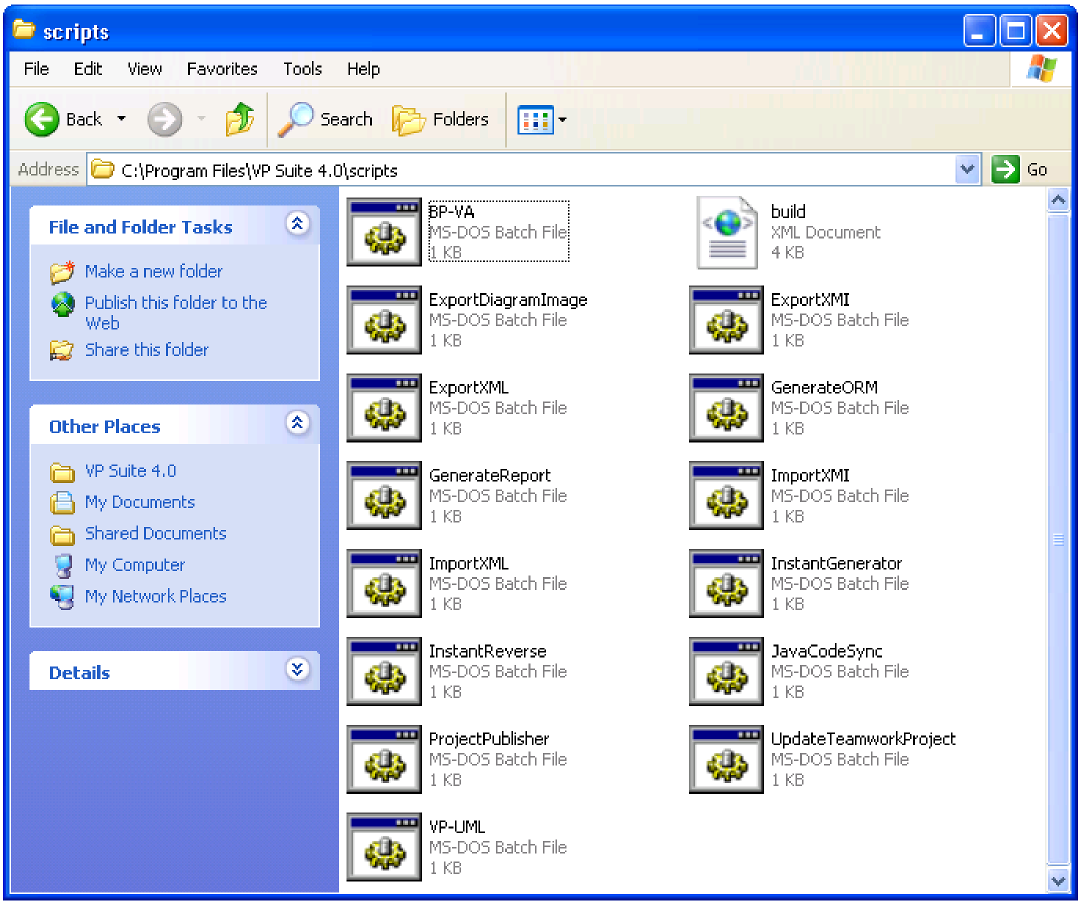
Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-file	The filepath of the XML file to import	C:\Demo\input \sample.xml

Table 5-5 *Parameters for ImportXML*

Generating ORM Code and/or Database

To generate ORM code and/or database from a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.



Address: C:\Program Files\VP Suite 4.0\scripts

Figure 5-21 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **GenerateORM** and paste to the bin folder of VP Suite installation directory.

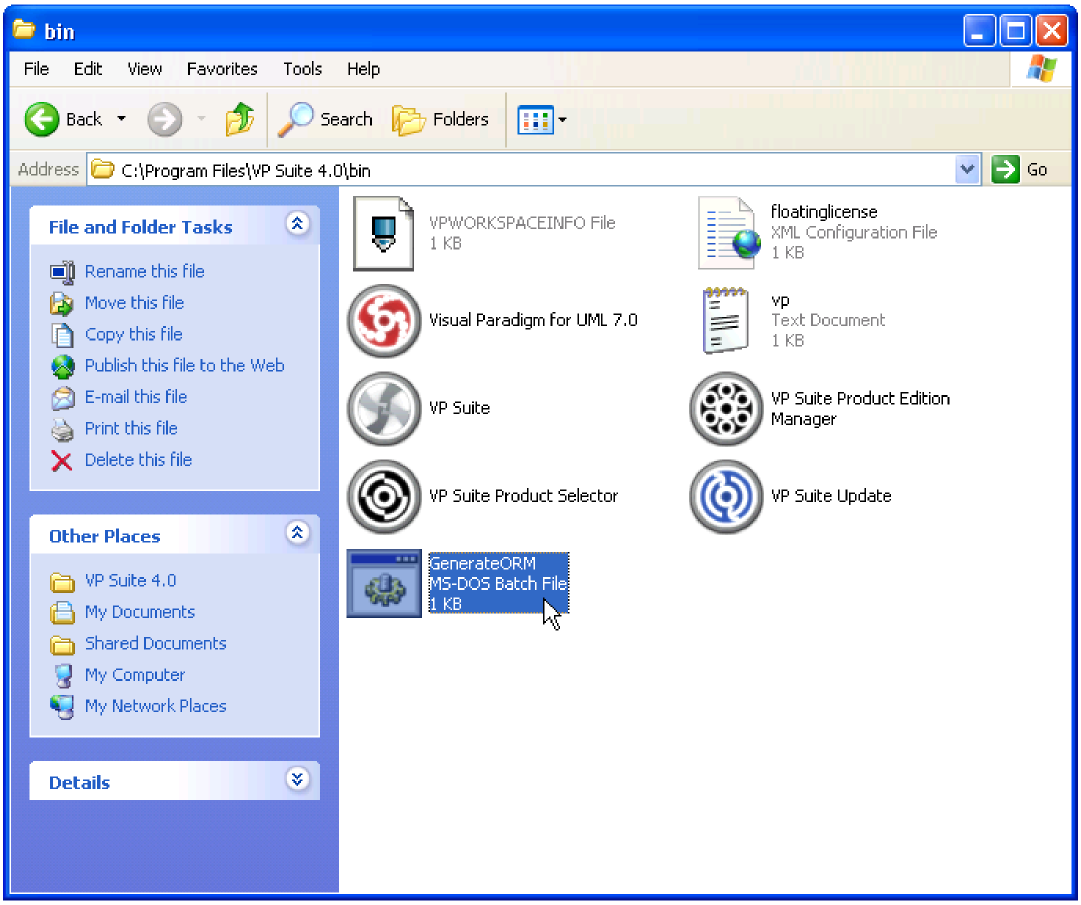
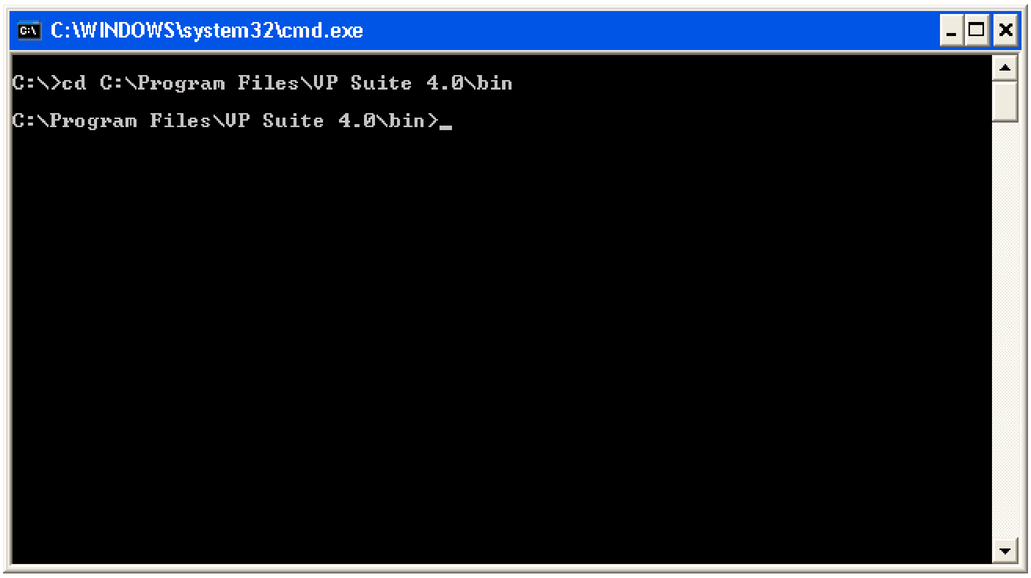


Figure 5-22 *Copy and paste GenerateORM from scripts folder to bin folder*

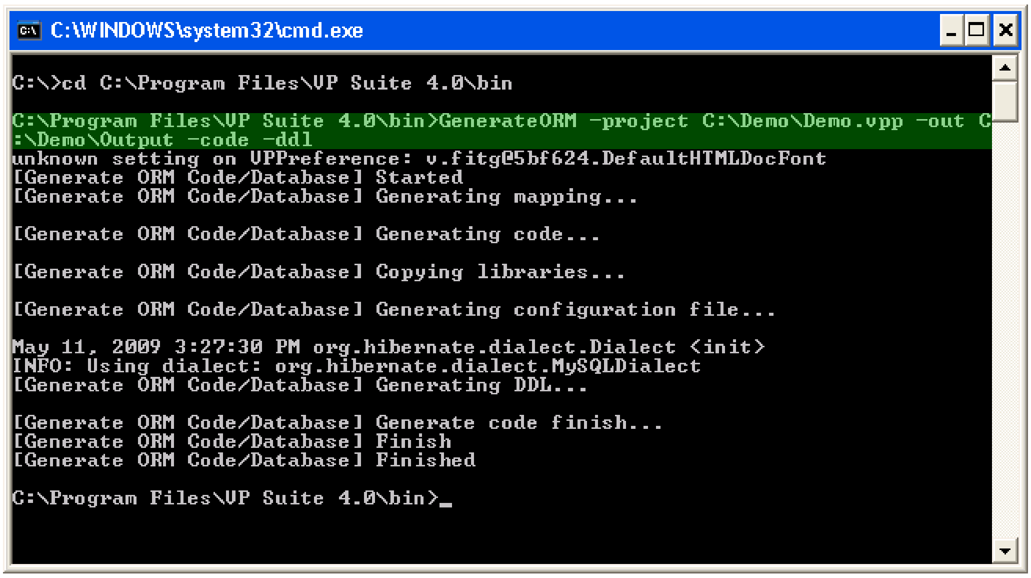
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-23 *Navigate to the bin folder of VP Suite in command prompt*

5. Execute the script by supplying the required parameters. For example:
GenerateORM -project C:\Demo\Demo.vpp -out C:\Demo\Output -code -ddl



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>GenerateORM -project C:\Demo\Demo.vpp -out C
:\Demo\Output -code -ddl
unknown setting on VPPreference: v.fitg@5bf624.DefaultHTMLDocFont
[Generate ORM Code/Database] Started
[Generate ORM Code/Database] Generating mapping...

[Generate ORM Code/Database] Generating code...

[Generate ORM Code/Database] Copying libraries...

[Generate ORM Code/Database] Generating configuration file...

May 11, 2009 3:27:30 PM org.hibernate.dialect.Dialect <init>
INFO: Using dialect: org.hibernate.dialect.MySQLDialect
[Generate ORM Code/Database] Generating DDL...

[Generate ORM Code/Database] Generate code finish...
[Generate ORM Code/Database] Finish
[Generate ORM Code/Database] Finished

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-24 *Executing GenerateORM*

Below is a description of parameters:

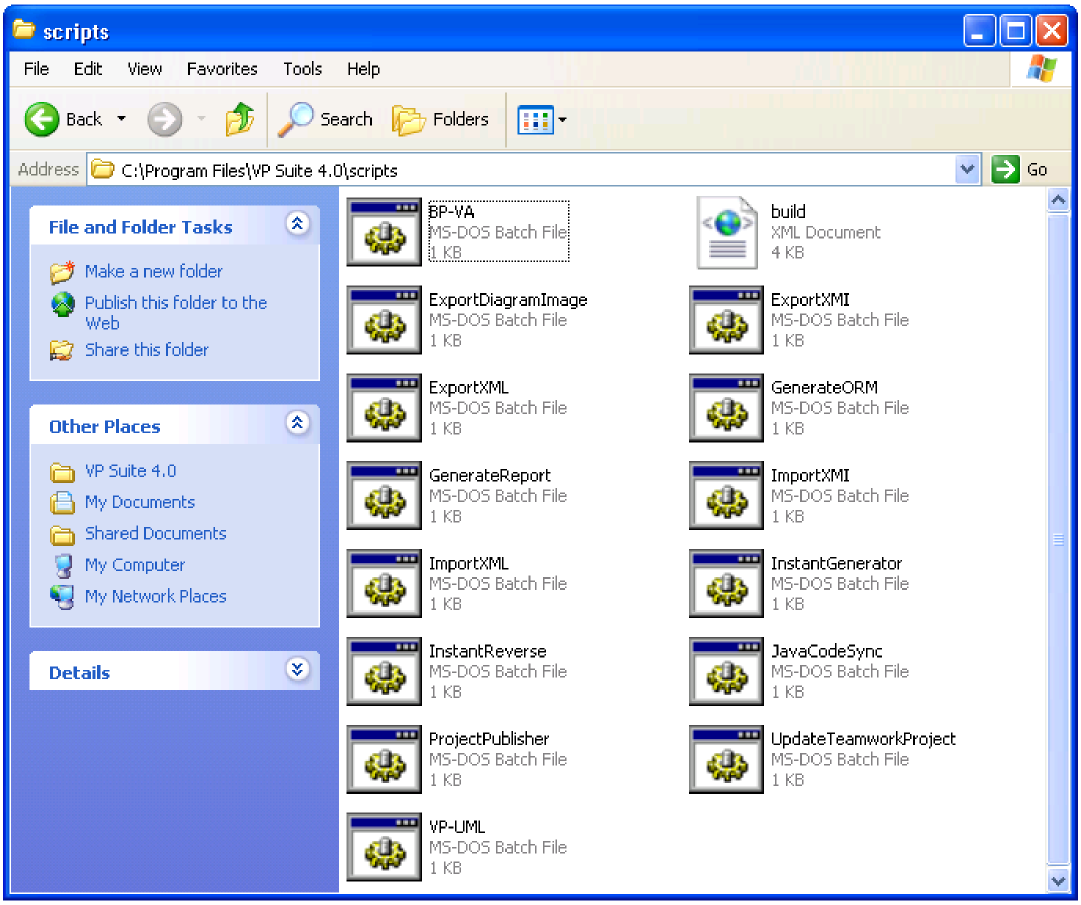
Parameter	Description	Example
-project	Project path	C: \Demo \Demo.vpp
-out	Folder for storing the generated files including the source code, required libraries and mapping XML	C: \Demo \Output
-code [optional]	Include to generate code.	- code
-ddl [optional]	Include to export DDL	- ddl
-exportdb [optional]	Include to export database	- exportdb

Table 5-6 *Parameters for GenerateORM*

Generating Report

To generate HTML/PDF/Word report from a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.



C:\Program Files\VP Suite 4.0\scripts

BP-VA
MS-DOS Batch File
1 KB

build
XML Document
4 KB

ExportDiagramImage
MS-DOS Batch File
1 KB

ExportXMI
MS-DOS Batch File
1 KB

ExportXML
MS-DOS Batch File
1 KB

GenerateORM
MS-DOS Batch File
1 KB

GenerateReport
MS-DOS Batch File
1 KB

ImportXMI
MS-DOS Batch File
1 KB

ImportXML
MS-DOS Batch File
1 KB

InstantGenerator
MS-DOS Batch File
1 KB

InstantReverse
MS-DOS Batch File
1 KB

JavaCodeSync
MS-DOS Batch File
1 KB

ProjectPublisher
MS-DOS Batch File
1 KB

UpdateTeamworkProject
MS-DOS Batch File
1 KB

VP-UML
MS-DOS Batch File
1 KB

Figure 5-25 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **GenerateReport** and paste to the bin folder of VP Suite installation directory.

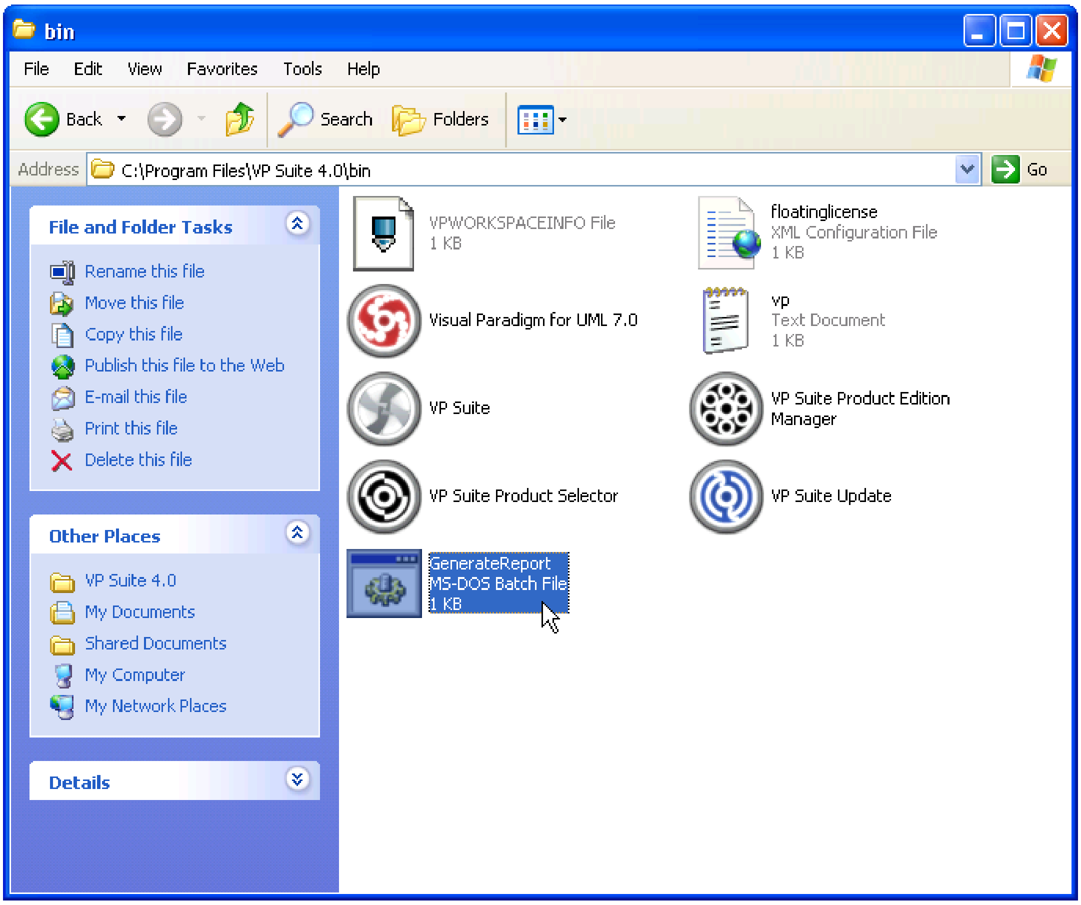
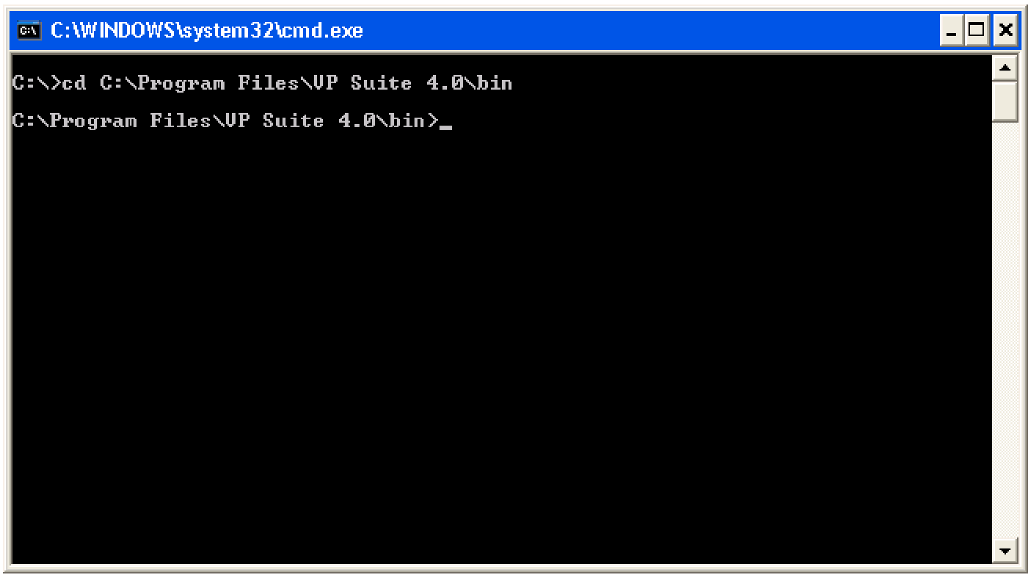


Figure 5-26 *Copy and paste GenerateReport from scripts folder to bin folder*

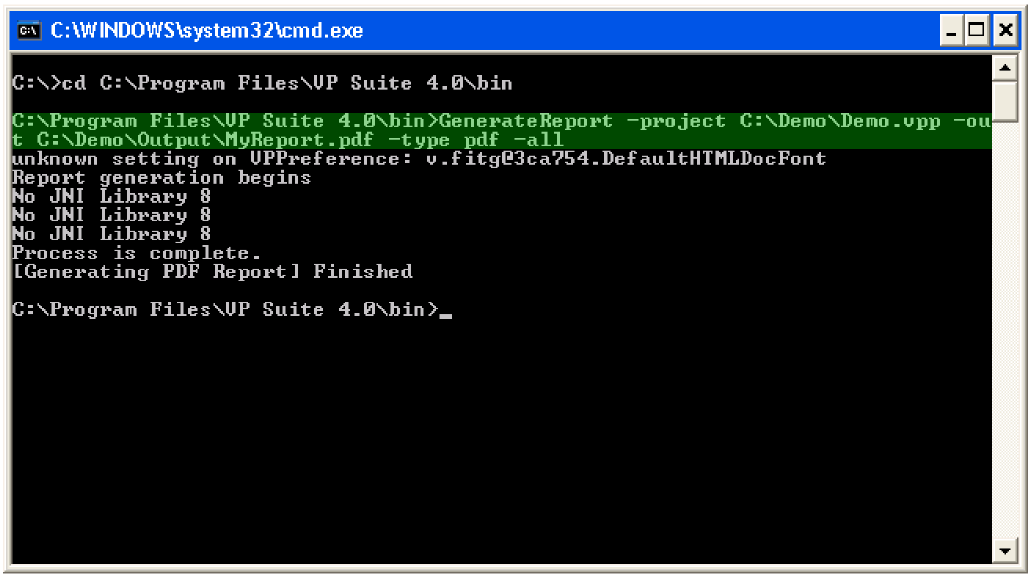
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-27 *Navigate to the bin folder of VP Suite in command prompt*

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



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>GenerateReport -project C:\Demo\Demo.vpp -ou
t C:\Demo\Output\MyReport.pdf -type pdf -all
unknown setting on VPPreference: v.fitg@3ca754.DefaultHTMLDocFont
Report generation begins
No JNI Library 8
No JNI Library 8
No JNI Library 8
Process is complete.
[Generating PDF Report] Finished

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-28 *Executing GenerateReport*

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.p
-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

Table 5-7 *Parameters for GenerateReport*

Instant Generator

To generate code from a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.

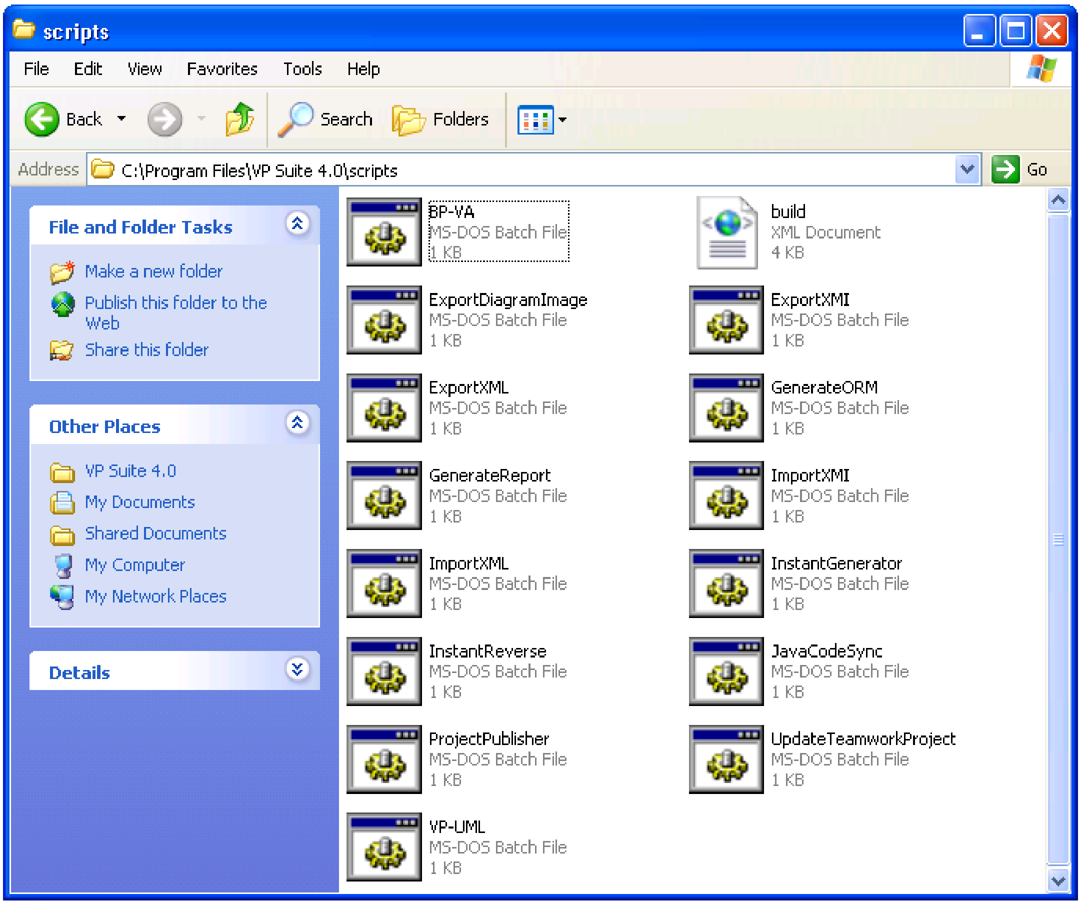


Figure 5-29 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **InstantGenerator** and paste to the bin folder of VP Suite installation directory.

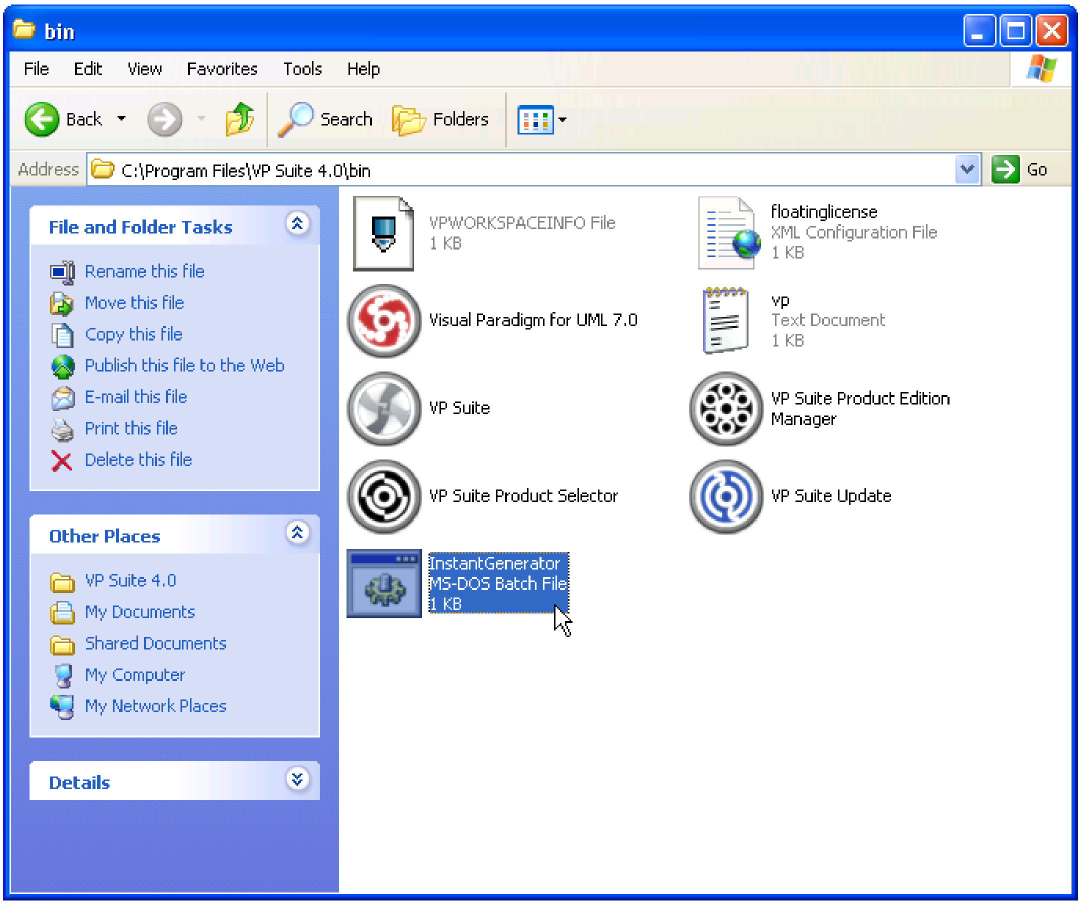
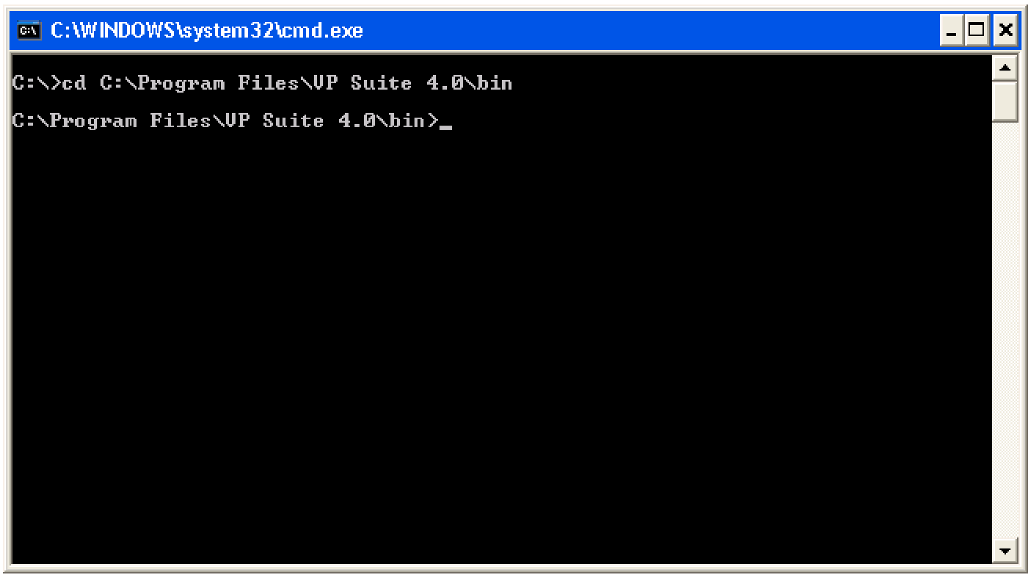


Figure 5-30 Copy and paste InstantGenerator from scripts folder to bin folder

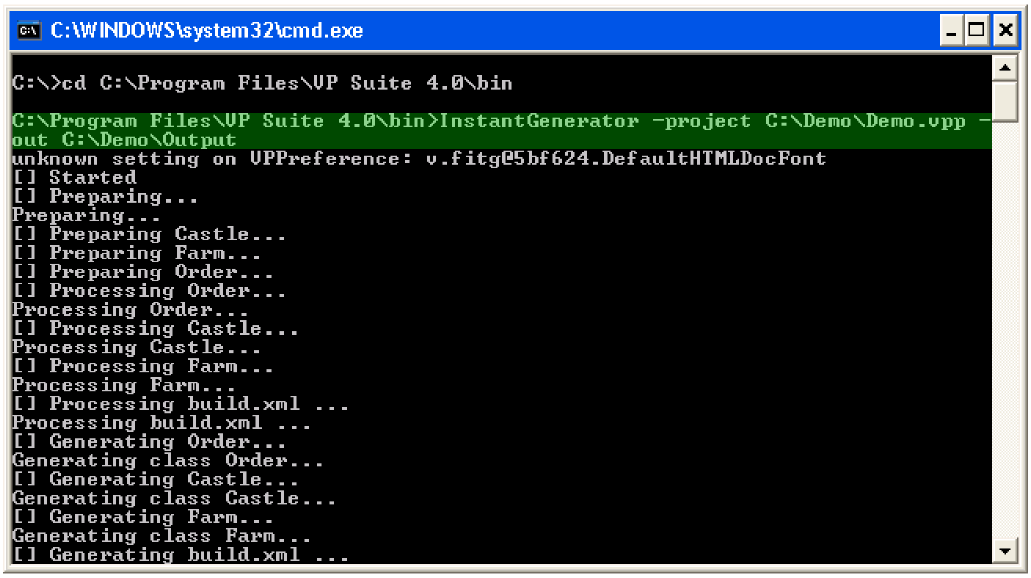
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-31 Navigate to the bin folder of VP Suite in command prompt

5. Execute the script by supplying the required parameters. For example:
InstantGenerator -project C:\Demo\Demo.vpp -out C:\Demo\Output



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>InstantGenerator -project C:\Demo\Demo.vpp -
out C:\Demo\Output
unknown setting on VPPreference: v.fitg@5bf624.DefaultHTMLDocFont
[] Started
[] Preparing...
Preparing...
[] Preparing Castle...
[] Preparing Farm...
[] Preparing Order...
[] Processing Order...
Processing Order...
[] Processing Castle...
Processing Castle...
[] Processing Farm...
Processing Farm...
[] Processing build.xml ...
Processing build.xml ...
[] Generating Order...
Generating class Order...
[] Generating Castle...
Generating class Castle...
[] Generating Farm...
Generating class Farm...
[] Generating build.xml ...
```

Figure 5-32 *Executing InstantGenerator*

Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C: \Demo \Demo.vpp
-out	The folder path of generated source files	C: \Demo \Output
-template [optiona]	The path of template folder. Unless specified, the default folder will be selected.	C: \MyTemplat
-lang [optional]	Specify the language to generate. Unless specified, the lastly selected language (saved in project file) will be generated. Here are the possible options: • Java • C# • .NET • • ODL • ActionScript • IDL • C++ • Delphi • • • Python • Objective-C • Ada95 • Ruby	C + +

Table 5-8 *Parameters for InstantGenerator*

NOTE: Code generation through command line generates only classes selected to generate when running VP-UML. In other words, you must at least generate once in VP-UML in order to make command line generation work.

Instant Reverse

To reverse source code to a project through command line:

1. Browse the scripts folder under the VP Suite installation directory.

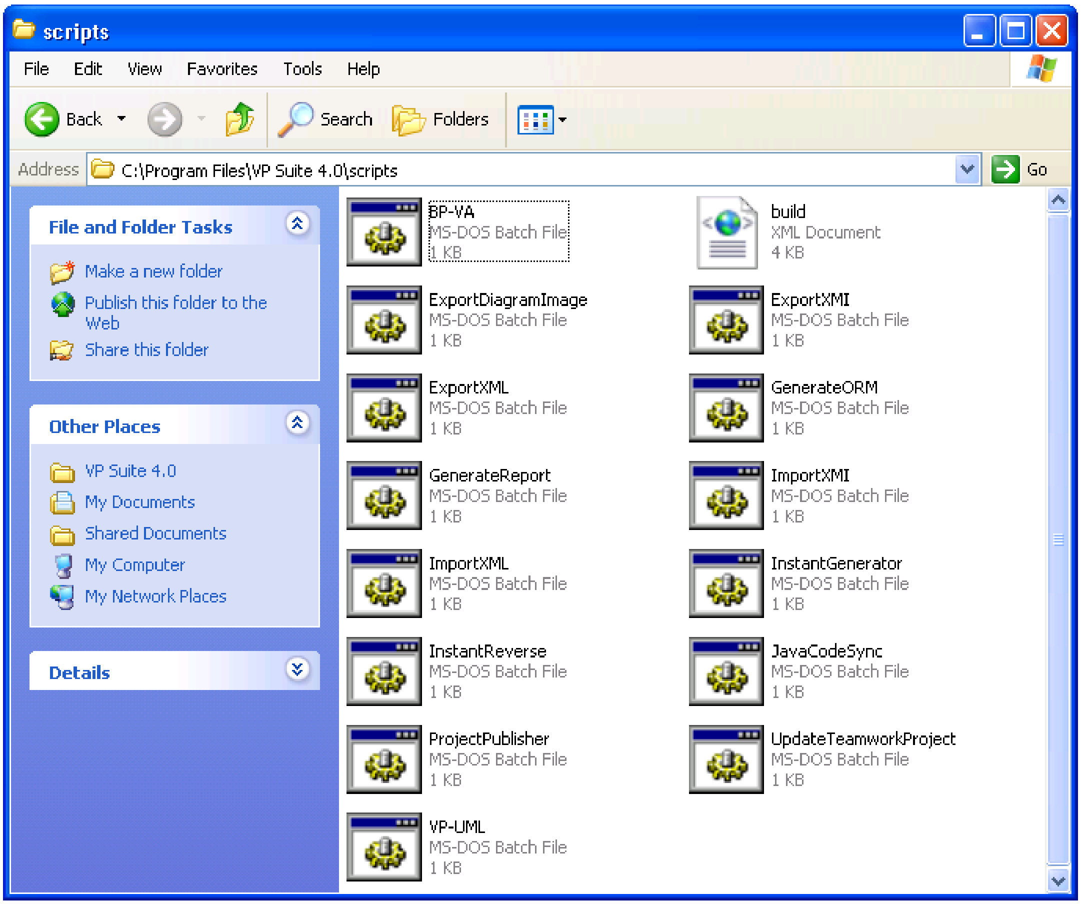


Figure 5-33 The scripts folder inside VP Suite installation directory

2. Copy the script file **InstantReverse** and paste to the bin folder of VP Suite installation directory.

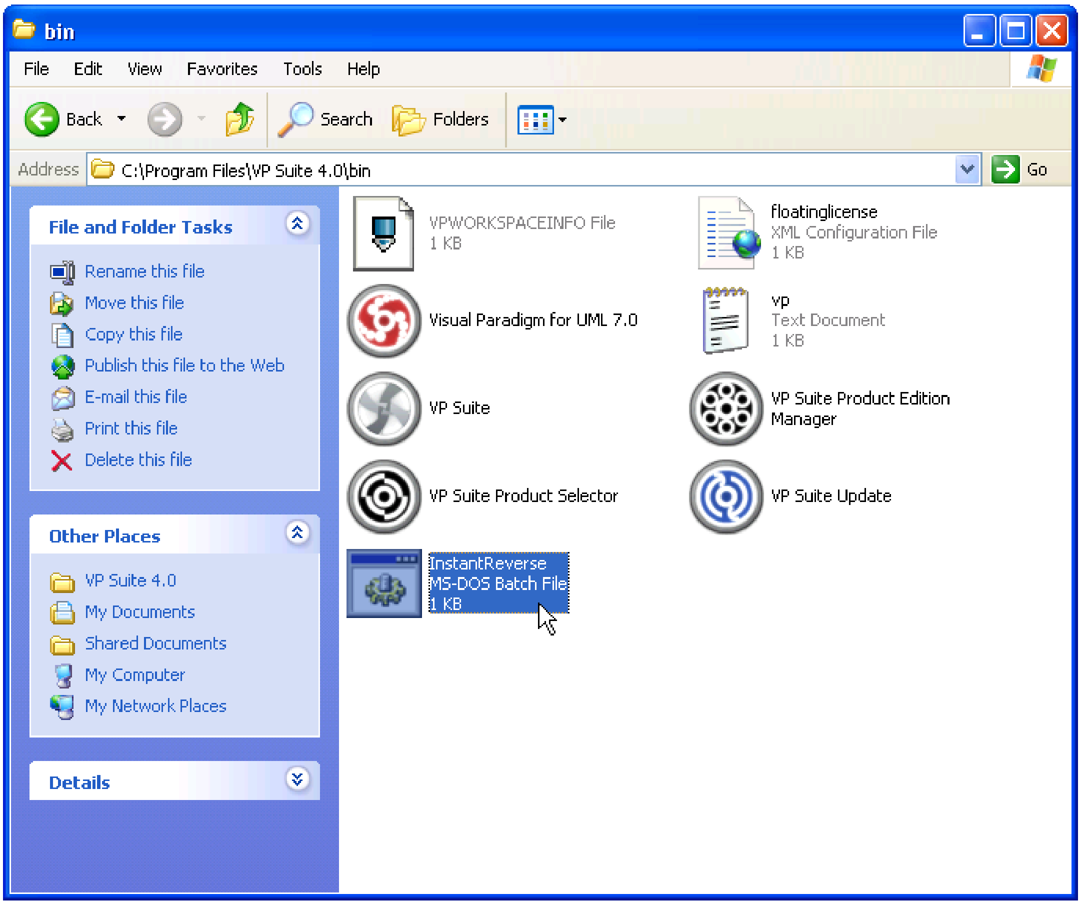
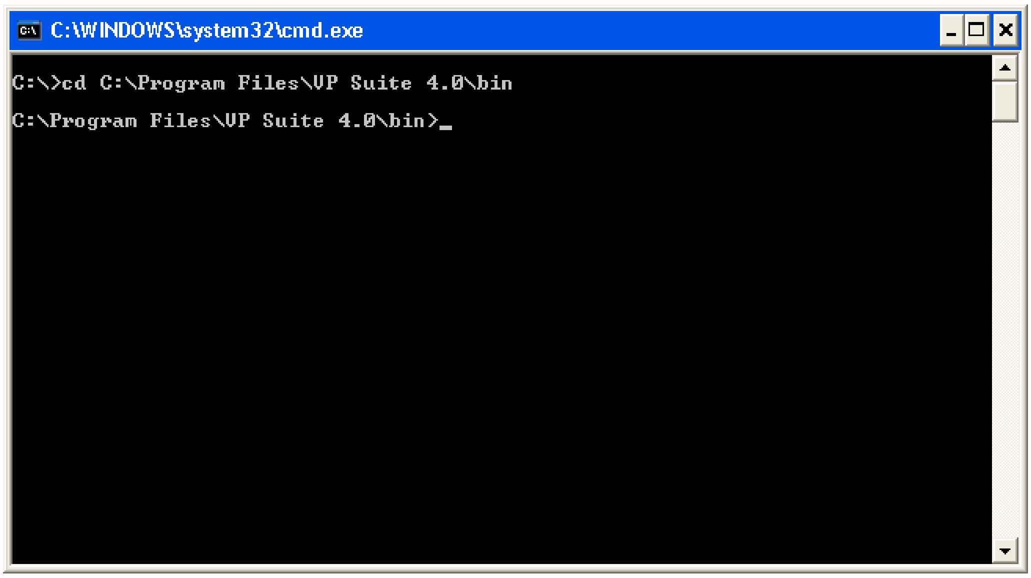


Figure 5-34 Copy and paste InstantReverse from scripts folder to bin folder

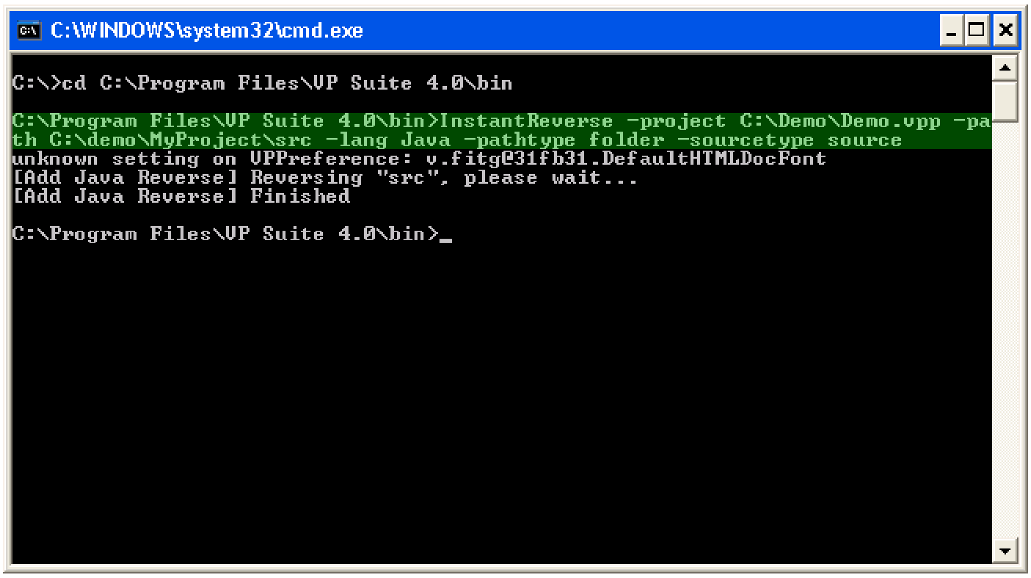
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-35 Navigate to the bin folder of VP Suite in command prompt

5. Execute the script by supplying the required parameters. For example:
InstantReverse -project C:\Demo\Demo.vpp -path C:\Demo\MyProject\src -lang Java -pathtype folder -sourcetype source



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>InstantReverse -project C:\Demo\Demo.vpp -pa
th C:\demo\MyProject\src -lang Java -pathtype folder -sourcetype source
unknown setting on VPPreference: v.fitg@31fb31.DefaultHTMLDocFont
[Add Java Reverse] Reversing "src", please wait...
[Add Java Reverse] Finished

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-36 *Executing InstantReverse*

Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C:\Demo\Demo.vpp
-path	The file or folder path of the source files to be reversed	C:\Demo\Output\src
-lang	Specify the language of the source code to reverse. Here are the possible options: • Java • "C++ Source" • ".NET dll or exe files" • "CORBA IDL Source" • Ada 9x Source" • XML • "XML Schema" • Hibernate • "PHP 5.0 Source" • "Python Source" • Objective-C	Java
pathtype	Useful only for Java, pathtype defines the type of the path supplied for -path. Here are the possible options: • file • folder • zip	file
sourcetype	Useful only for Java, sourcetype defines the type of source to reverse. Here are the possible options: • source • class	source
-overwrite | -update	Overwrite or update model from code	-overwrite

Table 5-9 *Parameters for InstantReverse*

Java Code Synchronization

To perform synchronization between model and Java code through command line:

1. Browse the scripts folder under the VP Suite installation directory.

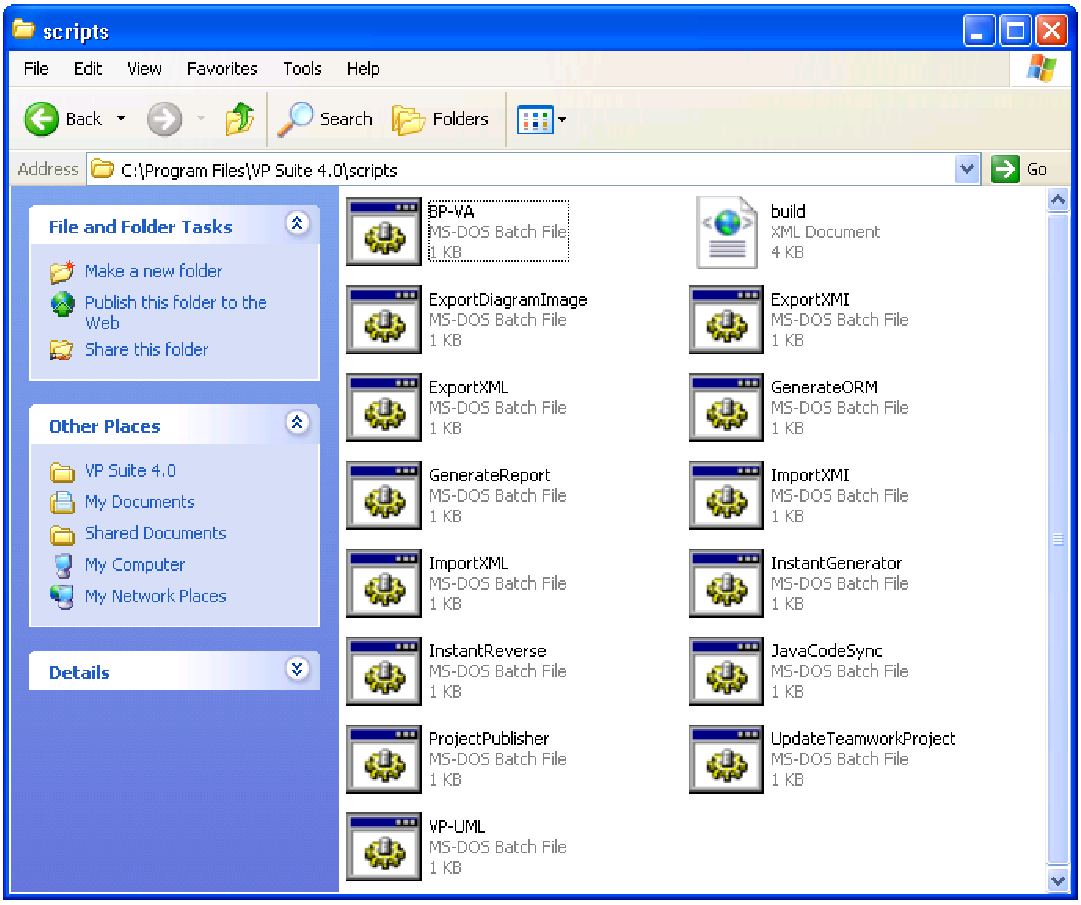


Figure 5-37 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **JavaCodeSync** and paste to the bin folder of VP Suite installation directory.

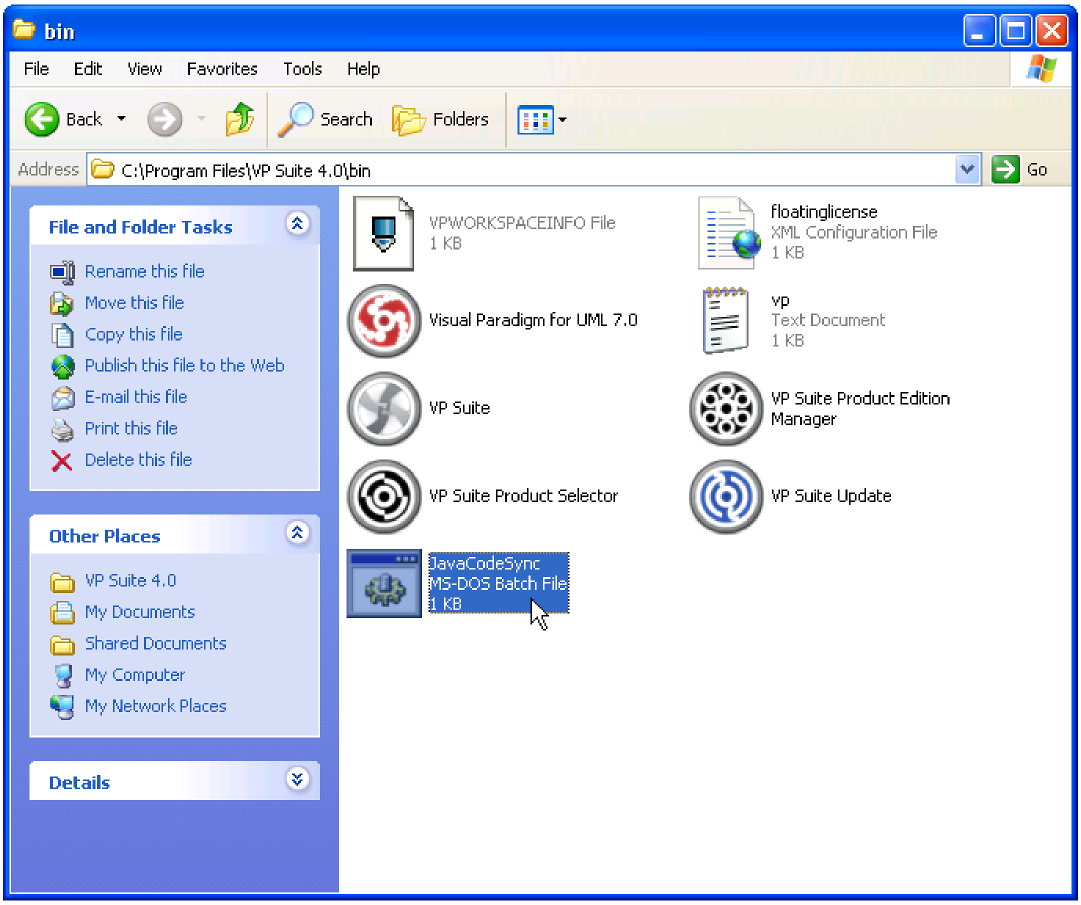
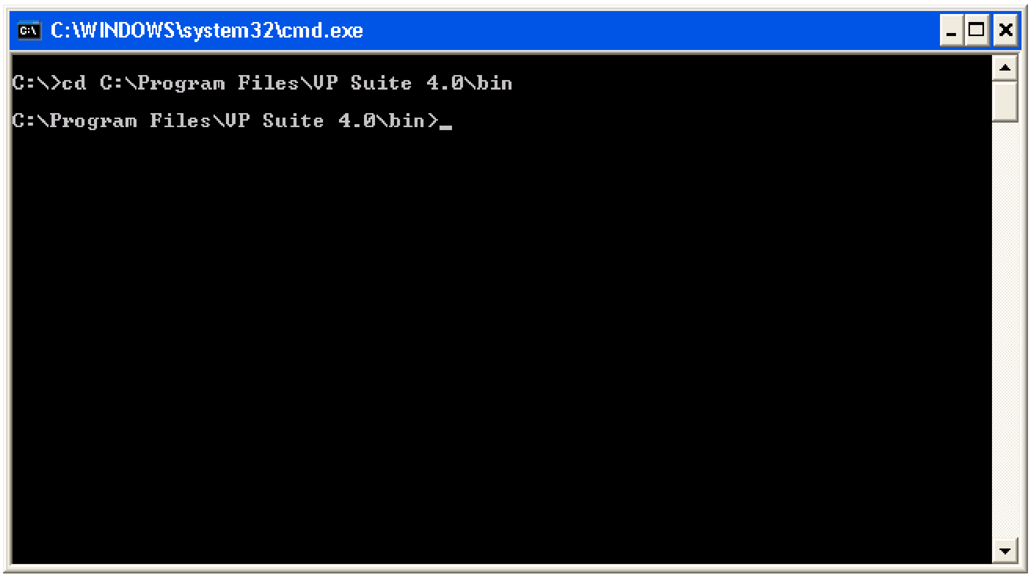


Figure 5-38 Copy and paste JavaCodeSync from scripts folder to bin folder

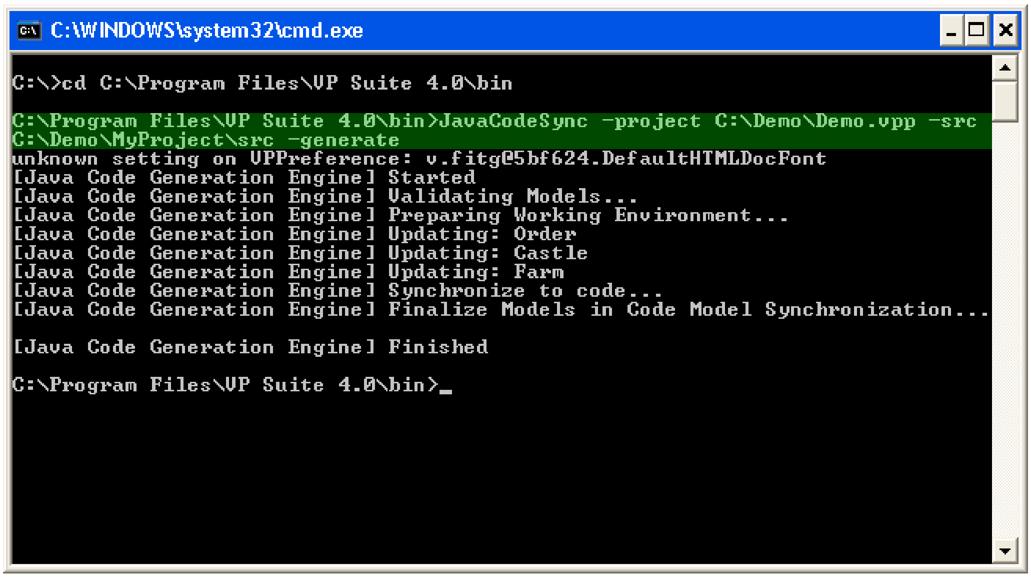
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-39 Navigate to the bin folder of VP Suite in command prompt

5. Execute the script by supplying the required parameters. For example:
JavaCodeSync -project C:\Demo\Demo.vpp -src C:\Demo\MyProject\src -generate



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>JavaCodeSync -project C:\Demo\Demo.vpp -src
C:\Demo\MyProject\src -generate
unknown setting on VPPreference: v.fitg@5bf624.DefaultHTMLDocFont
[Java Code Generation Engine] Started
[Java Code Generation Engine] Validating Models...
[Java Code Generation Engine] Preparing Working Environment...
[Java Code Generation Engine] Updating: Order
[Java Code Generation Engine] Updating: Castle
[Java Code Generation Engine] Updating: Farm
[Java Code Generation Engine] Synchronize to code...
[Java Code Generation Engine] Finalize Models in Code Model Synchronization...

[Java Code Generation Engine] Finished

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-40 *Executing JavaCodeSync*

Below is a description of parameters:

Parameter	Description	Example
-project	Project path	C: \Demo \Demo.vpp
-src	The folder path of the source file	C: \Demo \Output \src
-generate | -reverse	Action to perform. Include -generate to indicate the update of code from model. Include -reverse to indicate the update of model from code.	- generate

Table 5-10 *Parameters for JavaCodeSync*

Project Publisher

To publish project through command line:

1. Browse the scripts folder under the VP Suite installation directory.

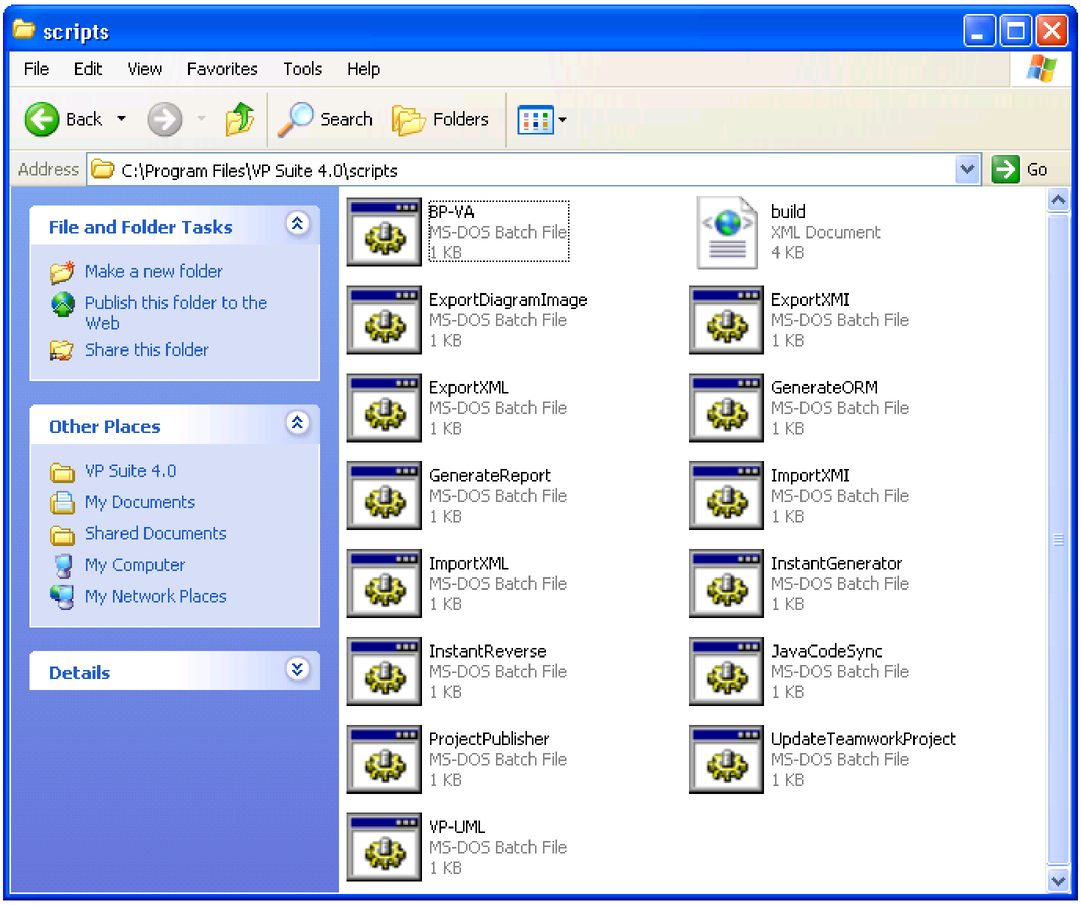


***Figure 5-41** The scripts folder inside VP Suite installation directory*

2. Copy the script file **ProjectPublisher** and paste to the bin folder of VP Suite installation directory.

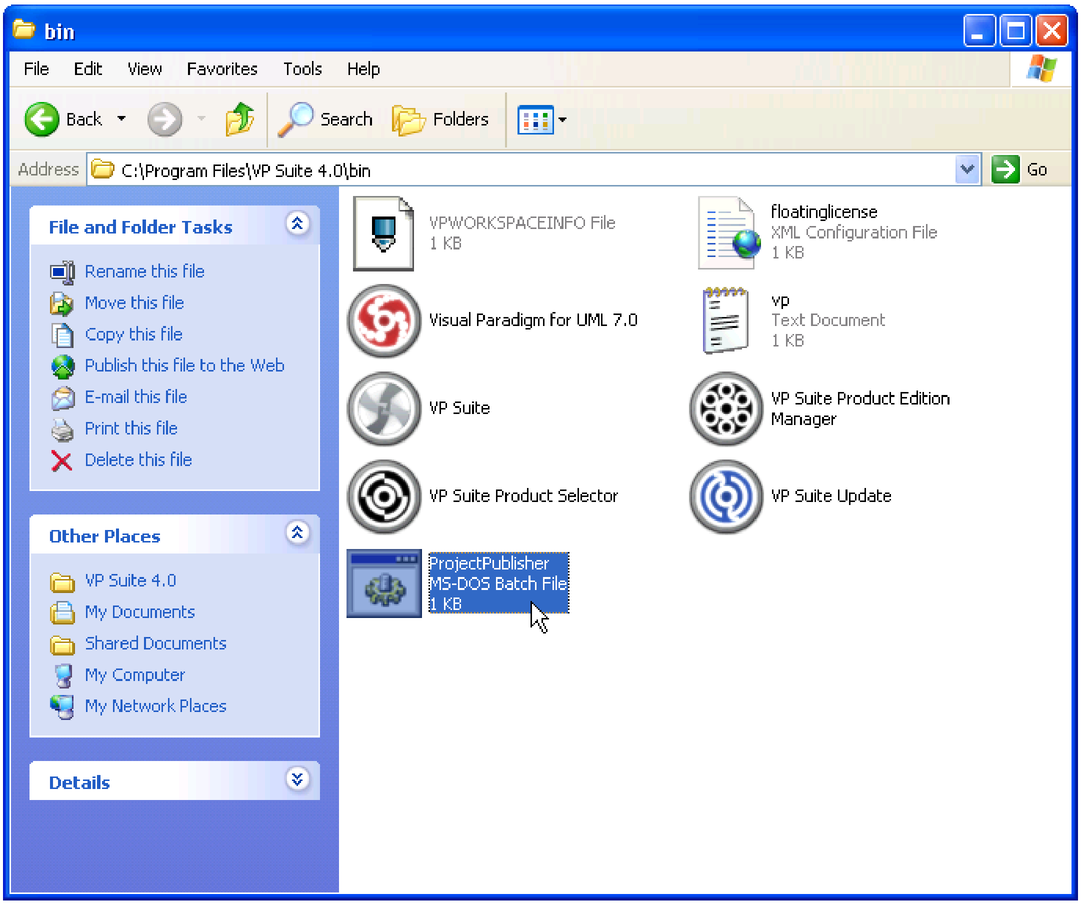
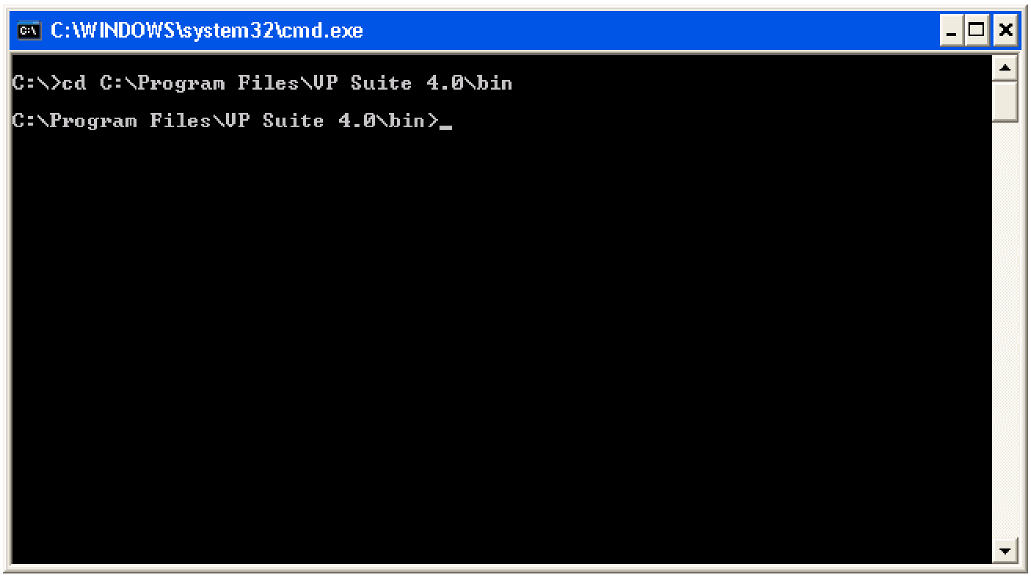


Figure 5-42 *Copy and paste ProjectPublisher from scripts folder to bin folder*

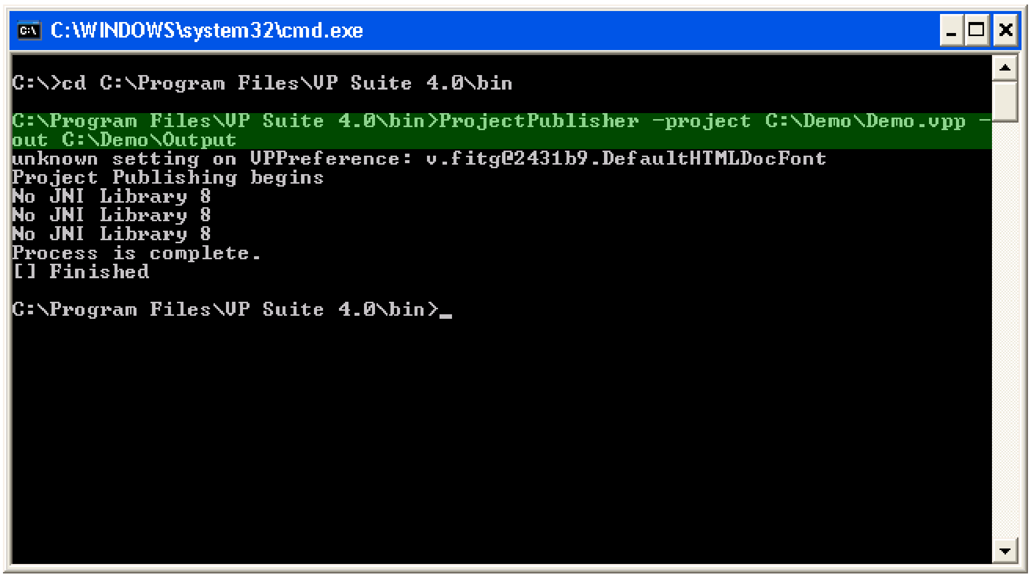
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-43 *Navigate to the bin folder of VP Suite in command prompt*

5. Execute the script by supplying the required parameters. For example:
ProjectPublisher -project C:\Demo\Demo.vpp -out C:\Demo\Output



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>ProjectPublisher -project C:\Demo\Demo.vpp -
out C:\Demo\Output
unknown setting on VPPreference: v.fitg@2431b9.DefaultHTMLDocFont
Project Publishing begins
No JNI Library 8
No JNI Library 8
No JNI Library 8
Process is complete.
[] Finished

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-44 *Executing ProjectPublisher*

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

Table 5-11 *Parameters for ProjectPublisher*

Updating Teamwork Project from Server

To update Teamwork project from server through command line:

1. Browse the scripts folder under the VP Suite installation directory.

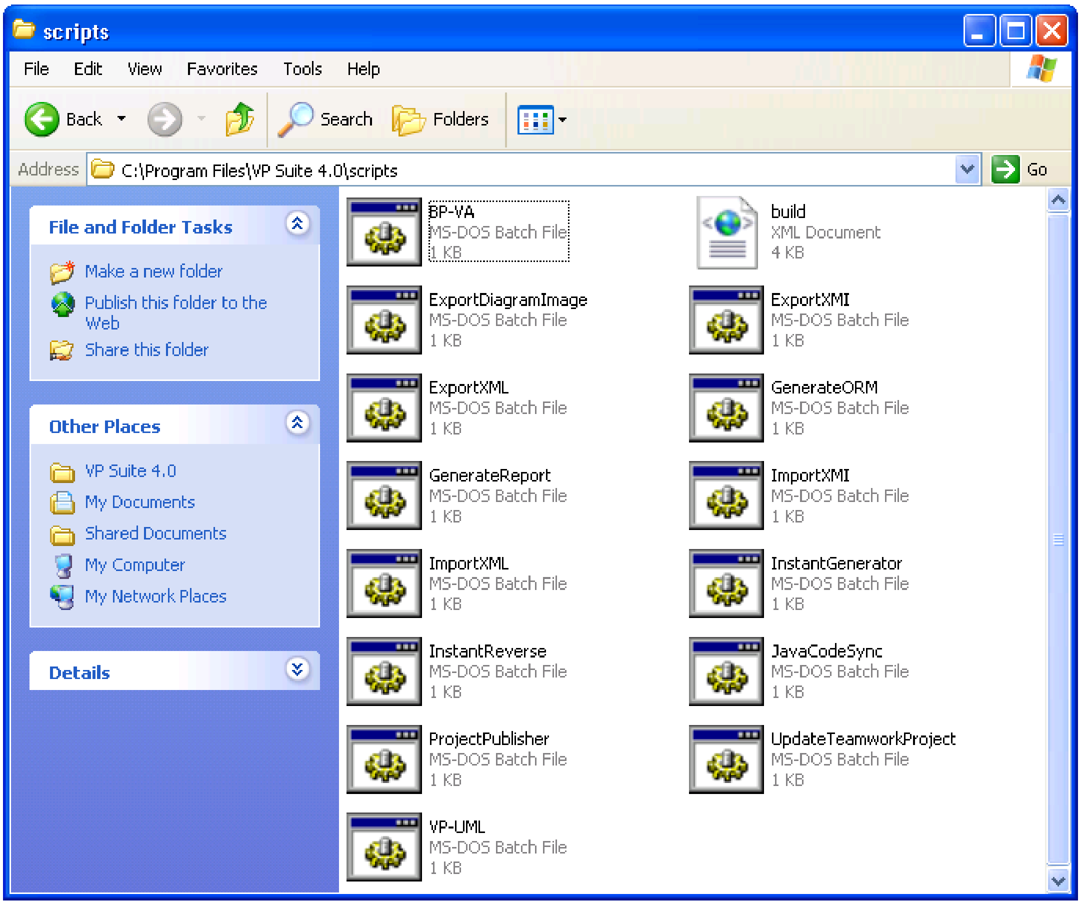


Figure 5-45 *The scripts folder inside VP Suite installation directory*

2. Copy the script file **ProjectPublisher** and paste to the bin folder of VP Suite installation directory.

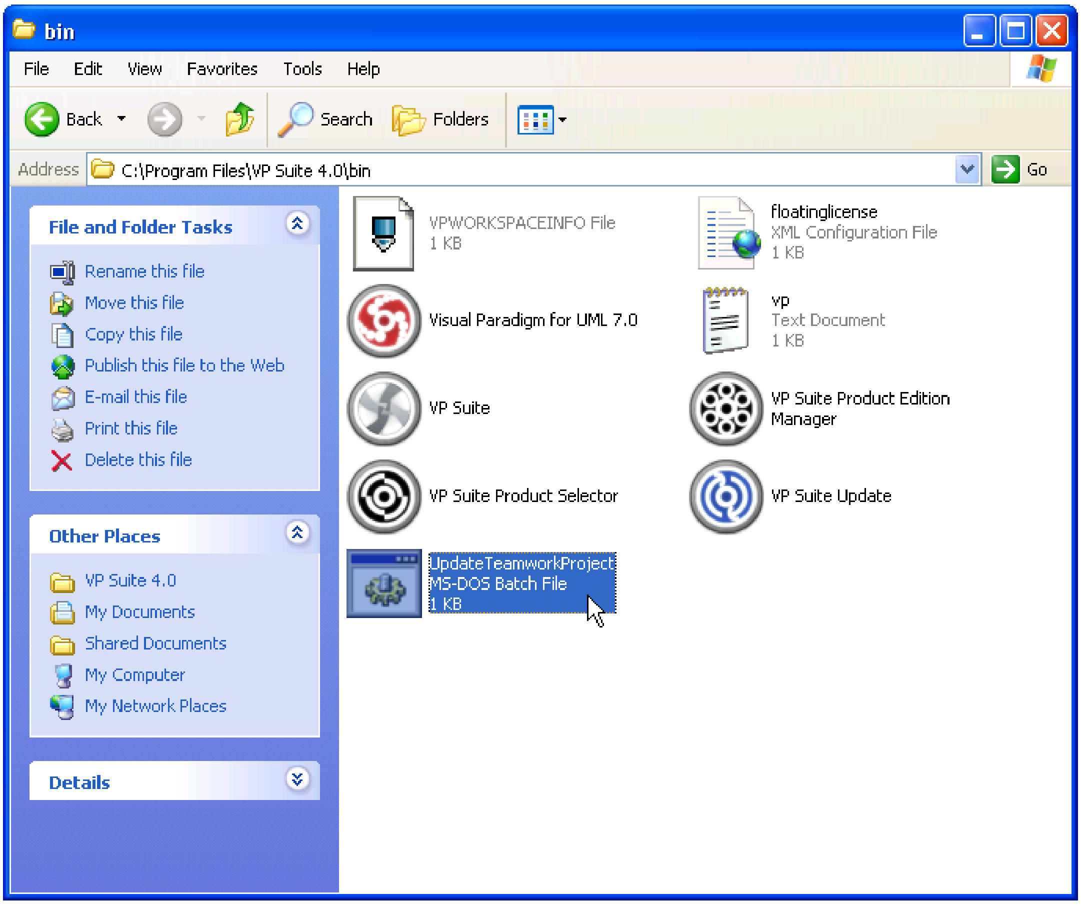
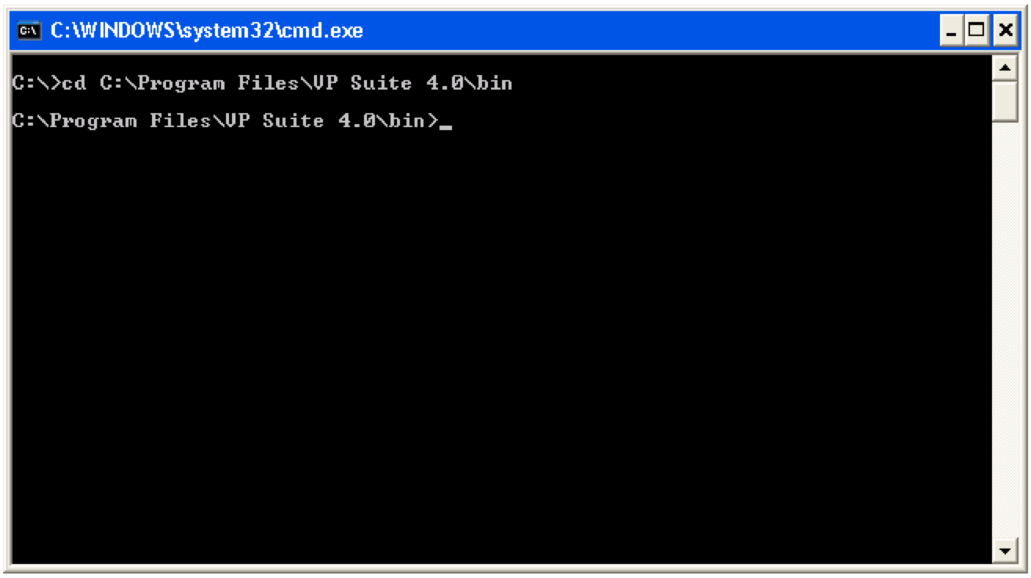


Figure 5-46 Copy and paste UpdateTeamworkProject from scripts folder to bin folder

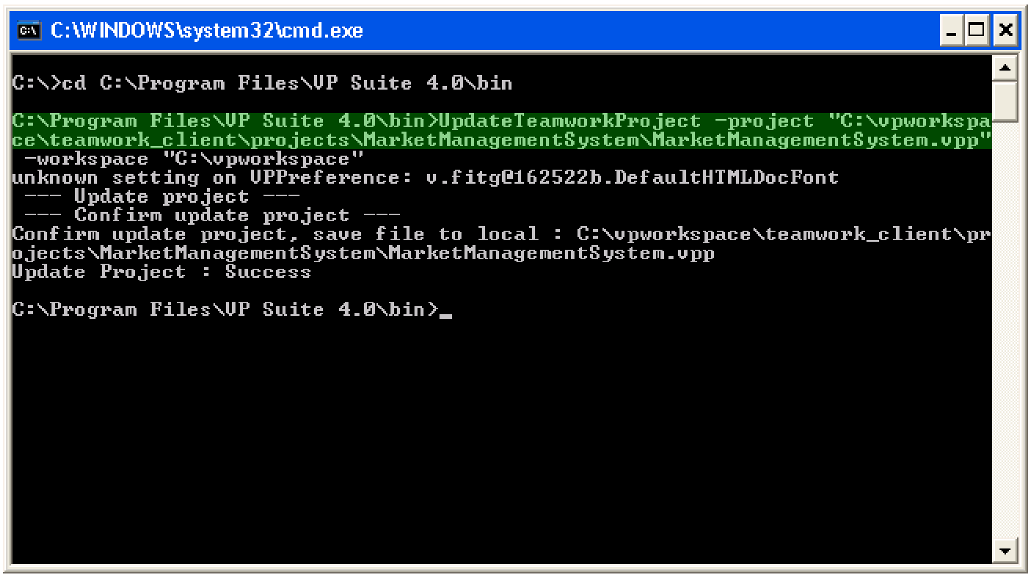
3. Start the command prompt.
4. Navigate to the bin folder of VP Suite installation directory.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-47 Navigate to the bin folder of VP Suite in command prompt

5. Execute the script by supplying the required parameters. For example:
UpdateTeamworkProject -project "C:\vpworkspace\teamwork_client\projects\MarketManagementSystem\MarketManagementSystem.vpp" -workspace "C:\vpworkspace"



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>UpdateTeamworkProject -project "C:\vpworkspa
ce\teamwork_client\projects\MarketManagementSystem\MarketManagementSystem.vpp"
 -workspace "C:\vpworkspace"
unknown setting on VPPreference: v.fitg@162522b.DefaultHTMLDocFont
 --- Update project ---
 --- Confirm update project ---
Confirm update project, save file to local : C:\vpworkspace\teamwork_client\pr
ojects\MarketManagementSystem\MarketManagementSystem.vpp
Update Project : Success

C:\Program Files\VP Suite 4.0\bin>_
```

Figure 5-48 *Executing UpdateTeamworkProject*

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

Table 5-12 *Parameters for UpdateTeamworkProject*

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 VP Suite installation directory.

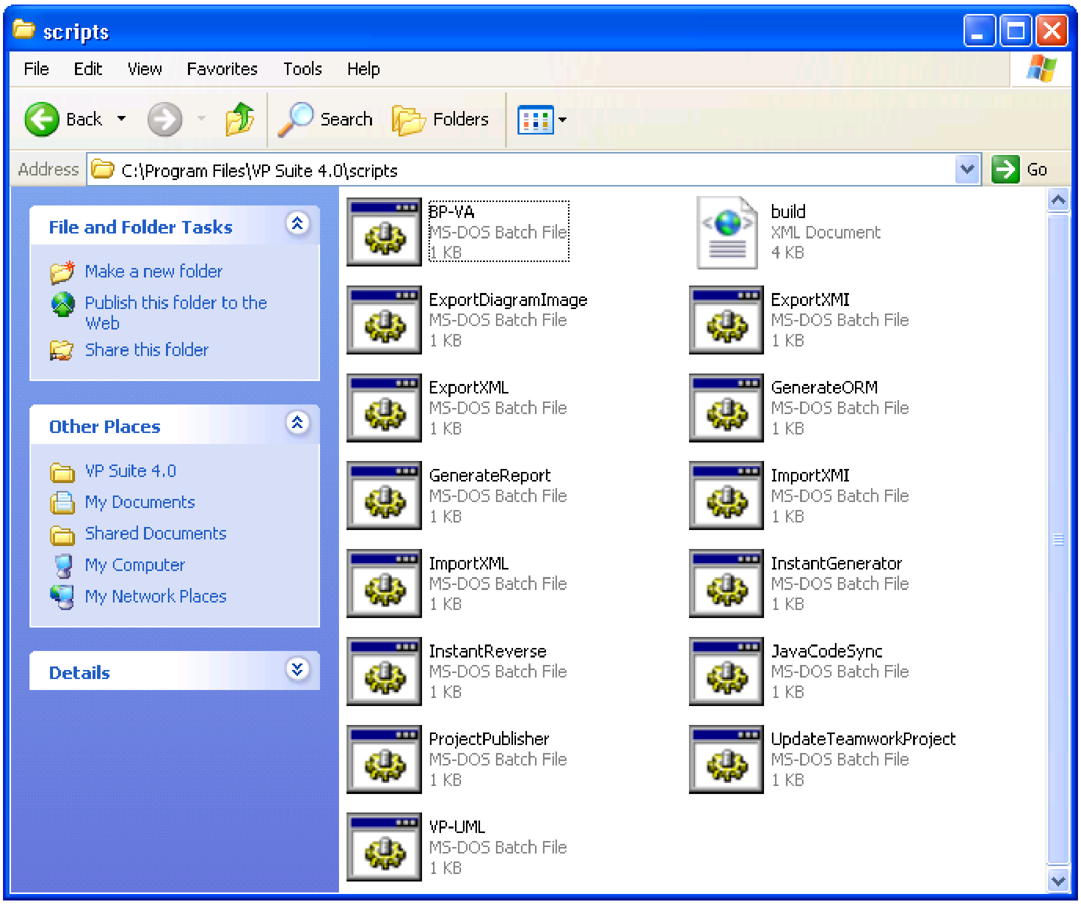


***Figure 5-49** The scripts folder inside VP Suite installation directory*

2. Copy the script file **build.xml** and paste to the bin folder of VP Suite installation directory.

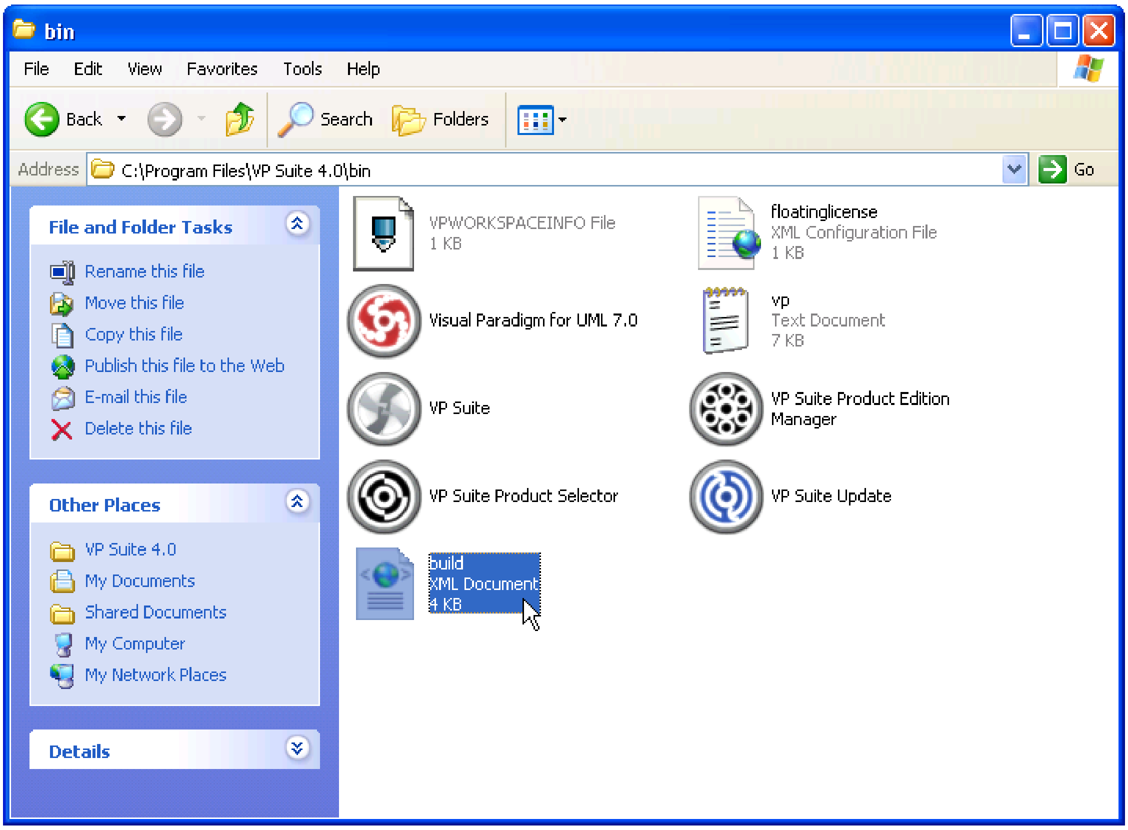
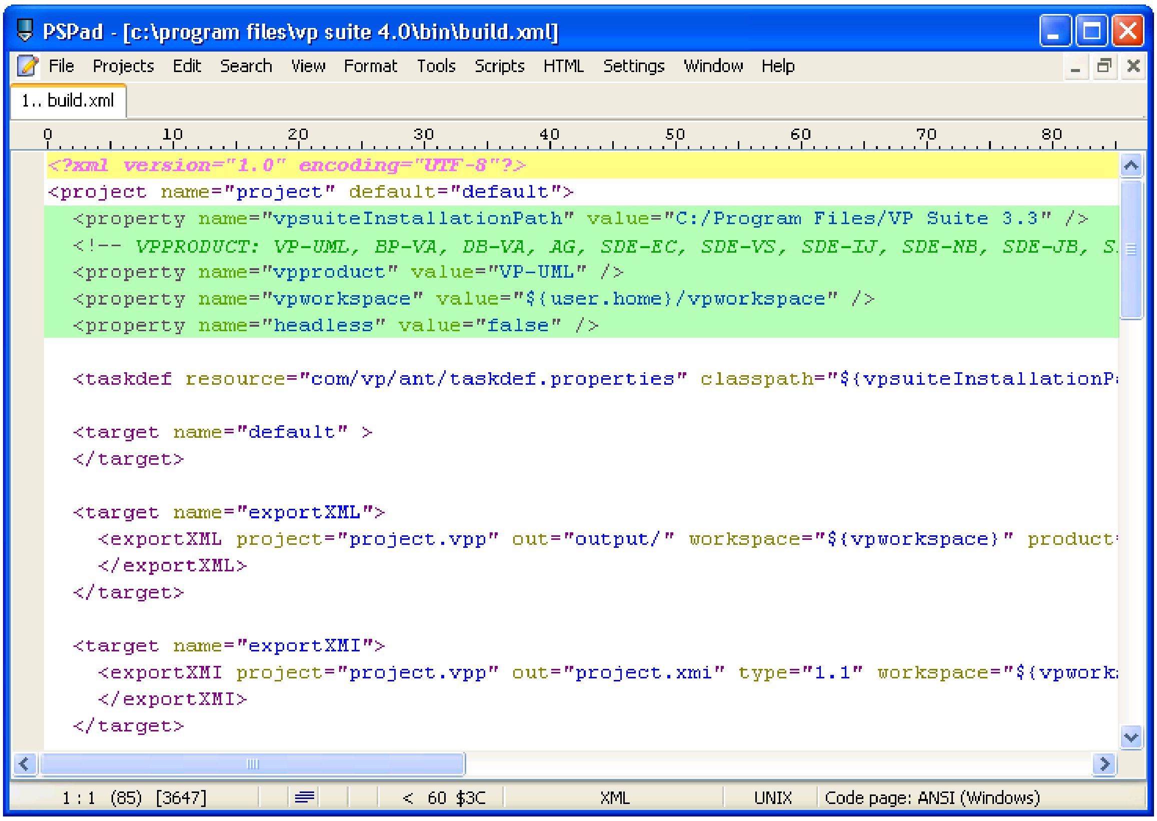


Figure 5-50 *Copy and paste build.xml from scripts folder to bin folder*

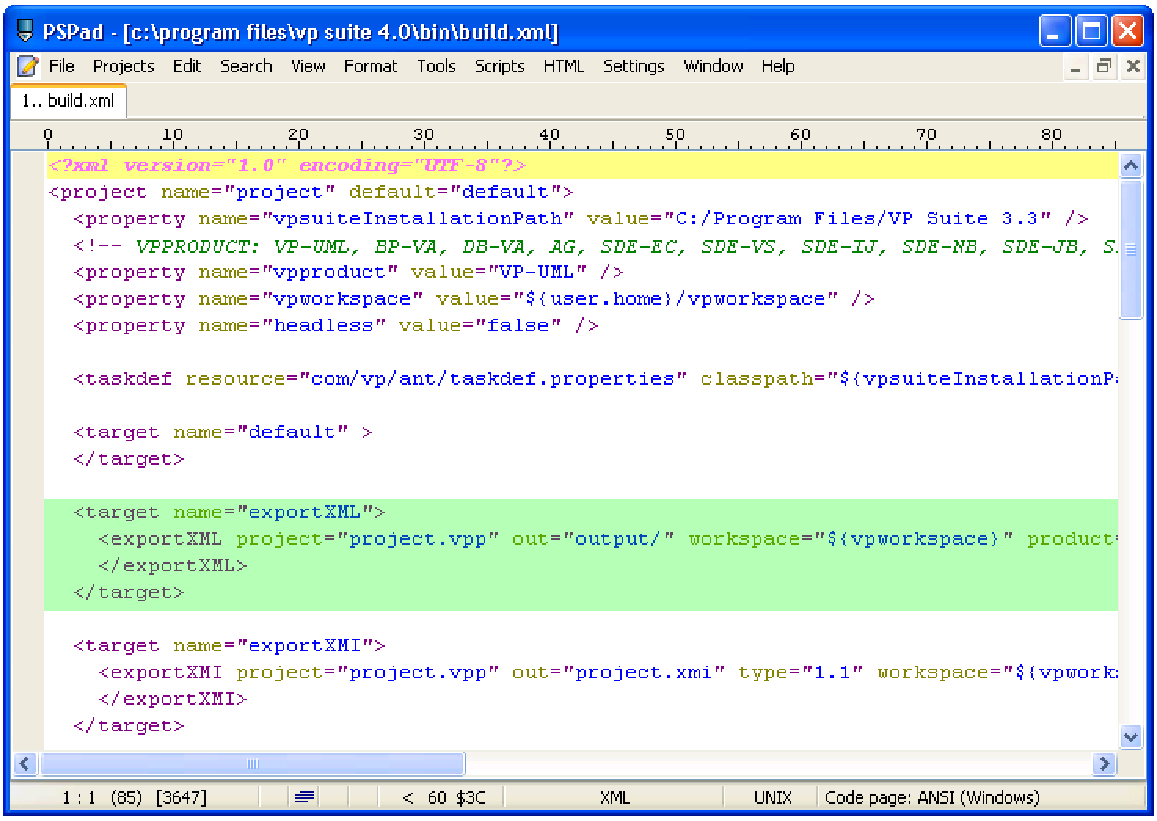
3. Open the build file in any text editor. Modify the properties **vpsuiteInstallationPath**, **vpproduct**, **vpworkspace** and **headless** to suit your environment.


PSPad - [c:\program files\vp suite 4.0\bin\build.xml]
File Projects Edit Search View Format Tools Scripts HTML Settings Window Help
1.. build.xml
1 : 1 (85) [3647]
< 60 $3C
XML
UNIX
Code page: ANSI (Windows)

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="project" default="default">
  <property name="vpsuiteInstallationPath" value="C:/Program Files/VP Suite 3.3" />
  <!-- VPPRODUCT: VP-UML, BP-VA, DB-VA, AG, SDE-EC, SDE-VS, SDE-IJ, SDE-NB, SDE-JB, S
  <property name="vpproduct" value="VP-UML" />
  <property name="vpworkspace" value="${user.home}/vpworkspace" />
  <property name="headless" value="false" />

  <taskdef resource="com/vp/ant/taskdef.properties" classpath="${vpsuiteInstallationP

  <target name="default" >
  </target>

  <target name="exportXML">
    <exportXML project="project.vpp" out="output/" workspace="${vpworkspace}" product
    </exportXML>
  </target>

  <target name="exportXMI">
    <exportXMI project="project.vpp" out="project.xmi" type="1.1" workspace="${vpwork
    </exportXMI>
  </target>
```

Figure 5-51 *To modify basic properties in build.xml*

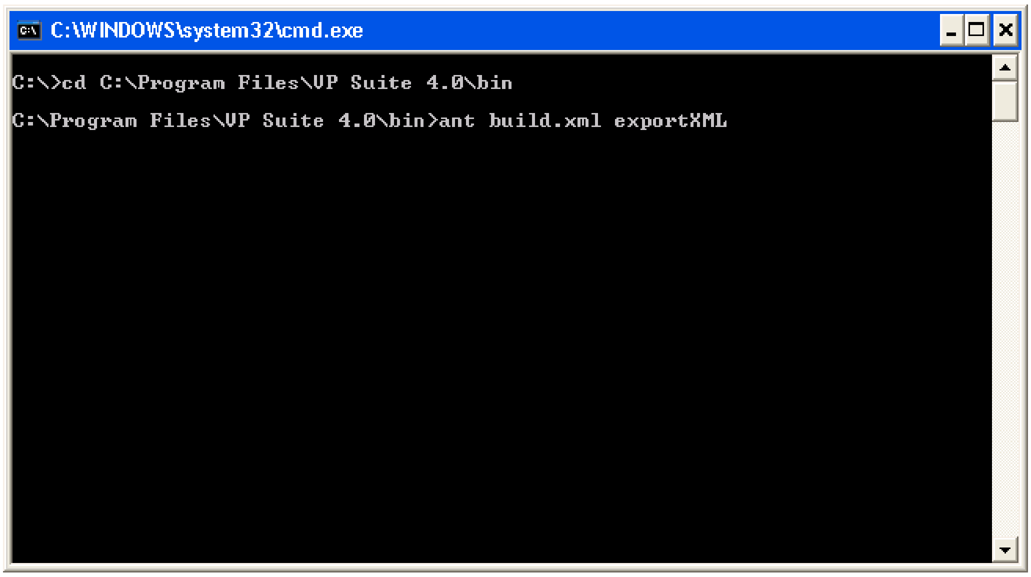
4. Modify task(s) specific parts by changing the values of parameters. For details about the parameters, refer to previous sections.



```
<?xml version="1.0" encoding="UTF-8"?>
<project name="project" default="default">
  <property name="vpsuiteInstallationPath" value="C:/Program Files/VP Suite 3.3" />
  <!-- VPPRODUCT: VP-UML, BP-VA, DB-VA, AG, SDE-EC, SDE-VS, SDE-IJ, SDE-NB, SDE-JB, S
  <property name="vpproduct" value="VP-UML" />
  <property name="vpworkspace" value="${user.home}/vpworkspace" />
  <property name="headless" value="false" />

  <taskdef resource="com/vp/ant/taskdef.properties" classpath="${vpsuiteInstallationPa

  <target name="default" >
  </target>

  <target name="exportXML">
    <exportXML project="project.vpp" out="output/" workspace="${vpworkspace}" product
    </exportXML>
  </target>

  <target name="exportXMI">
    <exportXMI project="project.vpp" out="project.xmi" type="1.1" workspace="${vpwork
    </exportXMI>
  </target>
```

Figure 5-52 *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.



```
C:\>cd C:\Program Files\VP Suite 4.0\bin

C:\Program Files\VP Suite 4.0\bin>ant build.xml exportXML
```

Figure 5-53 *Execute an ant script*