# Visual Paradigm User’s Guide

**From**: <http://www.visual-paradigm.com>

# Part I. Getting Started

# Introduction to Visual Paradigm

## Visual Paradigm user’s Guide Trademark Disclaimer

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

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

**Related Resources**

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

* [Visual Paradigm official website](https://www.visual-paradigm.com/)
* [Contact us if you need any help or have any suggestion](https://www.visual-paradigm.com/support/?product=vpuml)

## Visual Paradigm Product Overview

Visual Paradigm is a powerful, cross-platform and yet easy-to-use design and management tool for IT systems. Visual Paradigm 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 learning 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 diagram in Visual Paradigm’s UML tools, 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

### Agile development

Apply UeXceler, an agile development methodology in your software project. UeXceler comes with both the theory and principles, and a set of applicable agile tools that help you build software system in an agile way. It features agile tools like user story map and Affinity Table, plus UX design tools like storyboard and wireframes, etc.

### Database design

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

* Entity Relationship Diagram (ERD)
* 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 an ERD from scratch, you can reverse engineer a diagram from an existing database.

Besides diagramming, you can also synchronize between class diagram and entity relationship diagram to maintain the consistency between them. SQL generation and 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

### Object-relational mapping

Object-Relational Mapping enables you to access relational database in an object relational approach when coding. Visual Paradigm 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:

* VPository (Cloud)
* VP Teamwork Server (Need to by Visual Paradigm Teamwork Server separately)

### Documentation production

Develop system specification quickly with Doc. Composer. A full set of customizable document templates are ready for you.

## Visual Paradigm Products

Bellows are the kinds of features that can be found in each Visual Paradigm product. For details, please visit: <https://www.visual-paradigm.com/features/>.

|  |
| --- |
| Jedi: 原文是一个表格，这里只把特性点给列举出来。 |

* [UML & SysML](https://www.visual-paradigm.com/features/uml-tool/)
* [BPMN](https://www.visual-paradigm.com/features/bpmn-diagram-and-tools/)
* Entity Relationship Diagram
* Data Flow Diagram (DFD)
* Business Modeling
* Team Collaboration
* Task Management
* Glossary Management
* User Story 3C’s
* User Story Scenario
* Wireframe and Storyboard
* High-Fidelity Mockup Tools
* Code Engineering (Java, REST, C++, C#, PHP, XSD, etc.)
* Database Generation & Reversal
* Hibernate (ORM)
* Document Production
* RACI and CRUD Chart
* Mind Mapping
* Advanced Modeling Management
* Impact Analysis
* User Story Mapping
* User Story Affinity Estimation
* Sprint Management
* Customer Journey Mapping
* On-Demand Model ETL
* Project Management Tools
* Decisions & Rules Management
* Process Animation
* CMMN
* SoaML
* Zachman Frameworks & BMM Diagram
* ArchiMate 3.0
* TOGAF ADM process Guide-through
* Project Management Lifecycle Guide-through
* Strategic Planning with Business Motivation Model (BMM) Guide-through
* Just-in-Time Process Map
* Just-in-Time TOGAF Templates
* Just-in-Time PMBOK Templates
* Just-In-Time Project Management Templates

The filled circle indicates the support of certain feature in certain product. A summary of feature supported by Visual Paradigm.

# PART II. UML modeling

# Use case diagram

## How to Draw Use Case Diagram ?

Use case diagram is a kind of UML diagram that enables you to model system functions (i.e. goals) as well as the actors that interact with those functions. You can draw use case diagrams in Visual Paradigm as well as to document the use case scenario of use cases using the flow-of-events editor. In this page, you will see how to use case diagram with the UML tool.

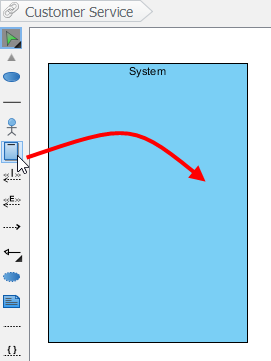
### Creating a use case diagram

Perform the steps below to create a UML use case diagram in Visual Paradigm.

1. Select **Diagram > New** from the application toolbar.
2. In the **New Diagram** window, select **Use Case Diagram**.
3. Click **Next**.
4. Enter the diagram name and description. The **Location** field enables you to select a model to store the diagram.
5. Click **OK**.

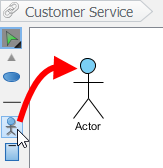
### Drawing a system

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



### Drawing an actor

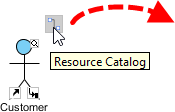
To draw an actor in use case diagram, select Actor on the diagram toolbar and then click it on the diagram pane. Finally, name the newly created actor when it is created.



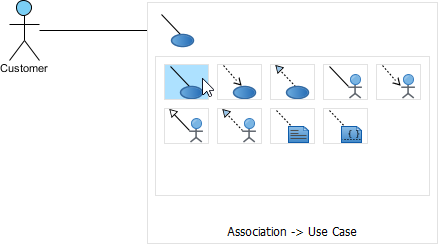
### Drawing a use case

Besides creating a use case through diagram toolbar, you can also create it through Resource Catalog:

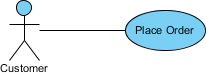
1. Move the mouse over a source shape (e.g. an actor).
2. Press on the Resource Catalog button and drag it out.



1. Release the mouse button until it reaches your preferred place.
2. Select Association -> Use case from Resource Catalog.



1. The source shape and the newly created use case are connected. Finally, name the newly created use case.



### Line wrapping use case name

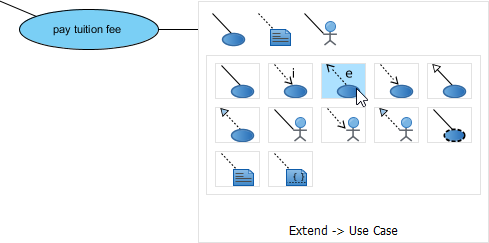
If a use case is too wide, you may resize it by dragging the filled selectors for a better outlook. As a result, the name of use case will be line-wrapped automatically.



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

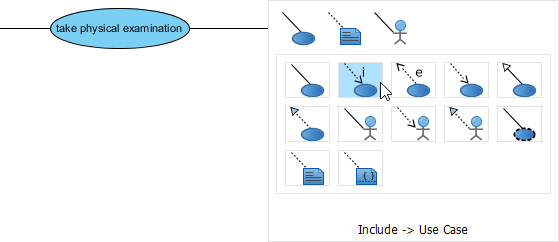
### Drawing <<Extend>> relationship

To create an extend relationship, move the mouse over a use case, press and drag out its Resource Catalog button. Then, release the mouse button at the preferred place and select Extend-> Use Case from Resource Catalog. The use case with extension points and a newly created use case are connected. After you name the newly created use case, you can name the extension point.



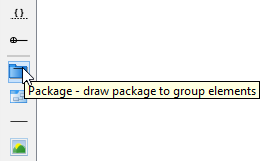
### Drawing <<Include>> relationship

To create an include relationship, move the mouse over a use case, press and drag out its Resource Catalog button. Then, release the mouse button at the preferred place and select Include -> Use case from Resource Catalog. A new use case together with an include relationship is created. Finally, name the newly created use case.

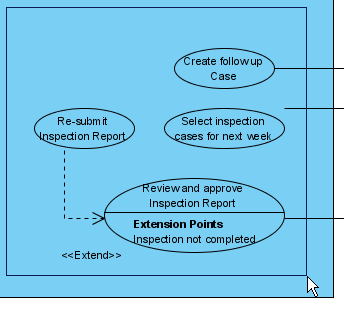


### Structuring use case with package

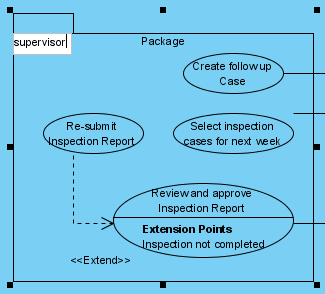
You can organize use cases with package when there are many of them on the diagram. Select Package on the diagram toolbar.



Drag the mouse to create a package surrounding those use cases.



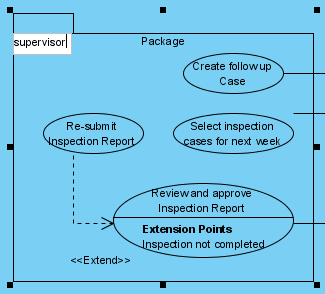
Finally, name the package.



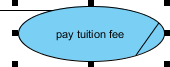
### Drawing business use case

The UML diagram tool also supports the representation actor and use case. To show an ordinary use case as business use case:

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



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

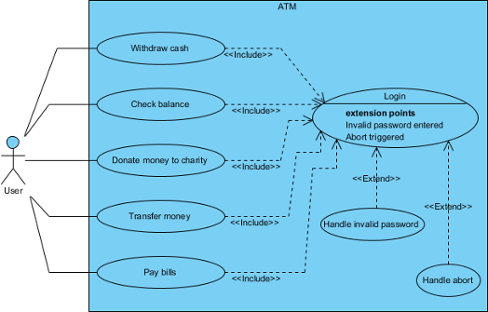


### Related Resources

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

* What is Use Case Diagram? – An introductory guide to Use Case Diagram
* The 10-tips to create a professional use case diagram
* User’s Guide – Drawing a use case diagram
* Visual Paradigm on YouTube

## UML Use Case Diagram Notations Guide



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

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

List of UML notations available in UML use case diagram

### Use Case



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

**OMG UML Specification**

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

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

### Association



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

**OMG UML Specification**

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

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

### Actor