Visual Paradigm Quick Start

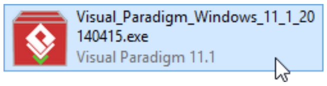
Last update: Apr 15, 2014

Table of Contents

Getting Started

Installing Visual Paradigm

1. Execute the Visual Paradigm installer.

   Visual_Paradigm_Windows_11_1_20140415.exe
   Visual Paradigm 11.1

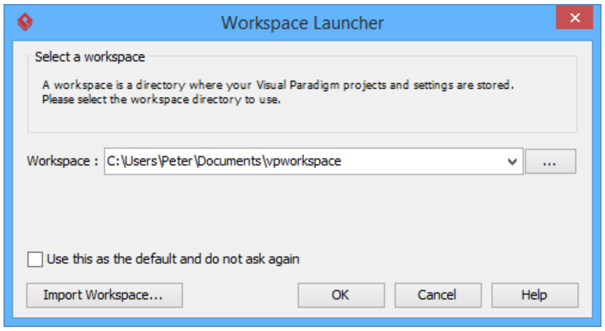
2. Click **Next** to proceed to the **License Agreement** page.
3. Read through the license agreement. Choose **I accept the agreement** if you fully understand and accept the terms. Click **Next**.
4. Specify the directory for installing Visual Paradigm. Click **Next** to continue.
5. Specify the name 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 continue.
6. In the **Select File Associations** page, keep **Visual Paradigm Project (*.vpp)** checked if you want your system able to open the project file upon direct execution (i.e. double click). Click **Next** to start the file copying process.
7. Once the file copying is finished, you can choose to start Visual Paradigm immediately, or just finish the installation without starting Visual Paradigm. Choose the option **Don't Start** and click **Finish**. This ends the installation of Visual Paradigm.

Starting Visual Paradigm

Windows users can start Visual Paradigm via the Start screen (for Windows 8/8.1) or Start menu (for Windows 7 or earlier). If you selected not to create an entry in the Start menu (during the installation), you can look under the installation folder of Visual Paradigm (the same path specified in step 4 in the section above) and start Visual Paradigm by running **Visual Paradigm.exe** in the **bin** folder.

Selecting Workspace

When you start Visual Paradigm, you need to specify a folder for workspace. A workspace is a folder in which application preferences, like the look-and-feel settings, are stored.



Workspace Launcher

Select a workspace

A workspace is a directory where your Visual Paradigm projects and settings are stored.
Please select the workspace directory to use.

Workspace : C:\Users\Peter\Documents\vpworkspace

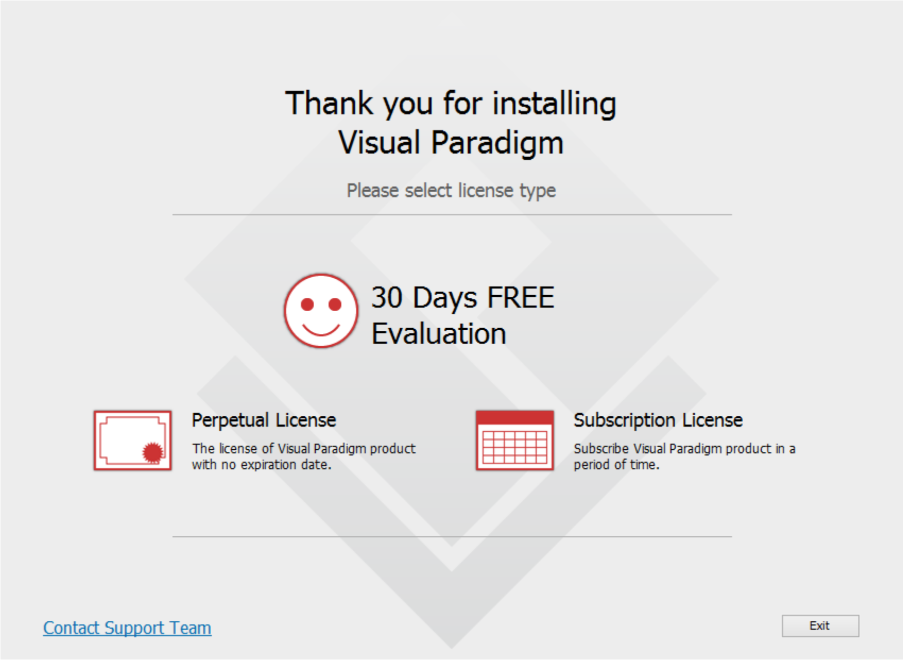
☐ Use this as the default and do not ask again

You need to select a workspace every time you start Visual Paradigm but the workspace selected need not always be the same. If you want to keep applying the same set of preferences, always start Visual Paradigm with the same workspace folder. If you want a fresh working environment, select a new folder and proceed. Now, select any folder you like and click **OK** to continue. Here we choose *C:\Users\Peter\Documents\vpworkspace* as the workspace folder.

If you are moving to a new machine and want to keep the application preferences, simply copy the workspace folder over and choose it as workspace when starting Visual Paradigm on the new machine.

Selecting License Type

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



Thank you for installing Visual Paradigm

Please select license type

30 Days FREE Evaluation

Perpetual License

The license of Visual Paradigm product with no expiration date.

Subscription License

Subscribe Visual Paradigm product in a period of time.

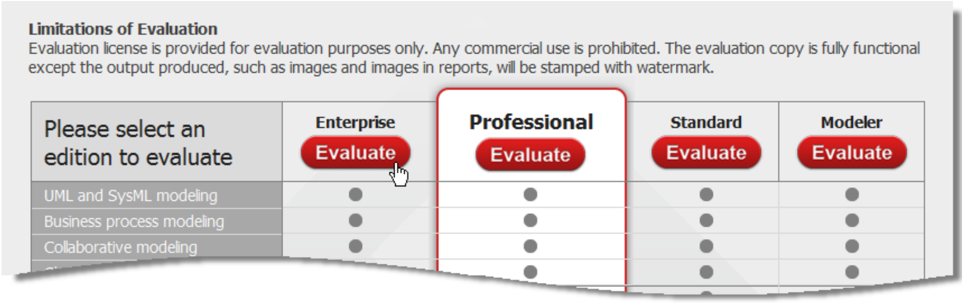
Depending on whether you own a purchased copy or an evaluation copy of Visual Paradigm, you can proceed by following the steps below:

For Customers

You should receive a notification Email with an activation code. The same activation code can also be found from the license key listed in your customer account. Copy the activation code first. Then, click on **Perpetual License**, paste the copied activation code and click **Activate** continue.

For Evaluators

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

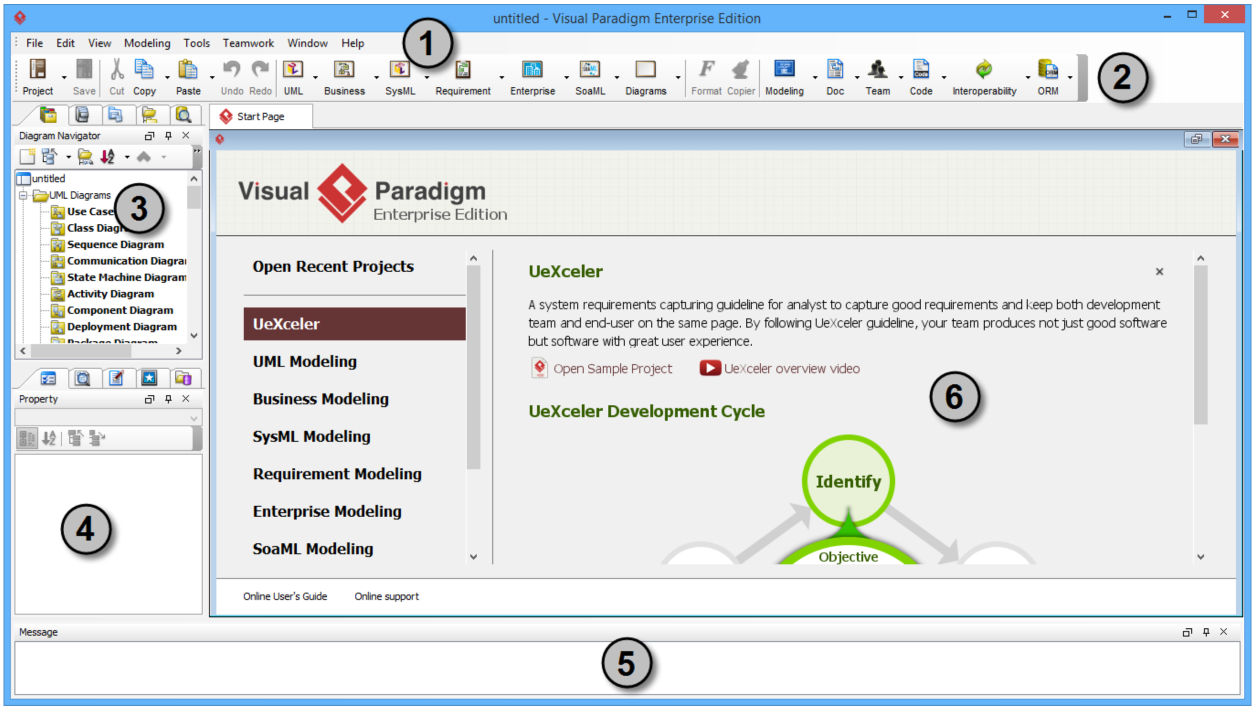


Limitations of Evaluation

Evaluation license is provided for evaluation purposes only. Any commercial use is prohibited. The evaluation copy is fully functional except the output produced, such as images and images in reports, will be stamped with watermark.

Please select an edition to evaluate	Enterprise Evaluate	Professional Evaluate	Standard Evaluate	Modeler Evaluate
UML and SysML modeling	●	●	●	●
Business process modeling	●	●	●	●
Collaborative modeling	●	●	●	●

The Environment



UeXceler

A system requirements capturing guideline for analyst to capture good requirements and keep both development team and end-user on the same page. By following UeXceler guideline, your team produces not just good software but software with great user experience.

No.	Name	Description
1	Menu bar	A string of pull-down menus at the top of the window which allows you to select and perform various operations in Visual Paradigm.
2	Toolbar	Located below the menu bar, it is an extension of the main menu. All buttons open up to groups of icons that perform various functions.
3	Diagram Navigator	A place where supported diagram types and existing diagrams are listed accordingly. It allows you to create or access existing diagrams by their types.
4	Property Pane	A display of properties of the model element/shape currently being selected in the diagram pane.
5	Message Pane	All available information or warnings will be shown here.
6	Diagram Pane	The diagram will be displayed in diagram pane.

Saving and Opening Project

To save your work, select either **File > Save Project** or **File > Save Project as…**. When you are saving a project for the first time, you will be asked to specify its location.

To open an existing project, select **File > Open Project…** from the main menu and select the project to open.

Basic Diagramming Techniques

This section will go through the steps of creating diagrams, creating shapes and connecting them. You will also learn how to document model elements and make diagrams more readable by coloring shapes.

Creating Diagram

Let's take use case diagram as an example. To create a use case diagram:

1. Click on **UML** in the toolbar and select **Use Case Diagram** from the drop-down menu.
2. Enter the diagram name.

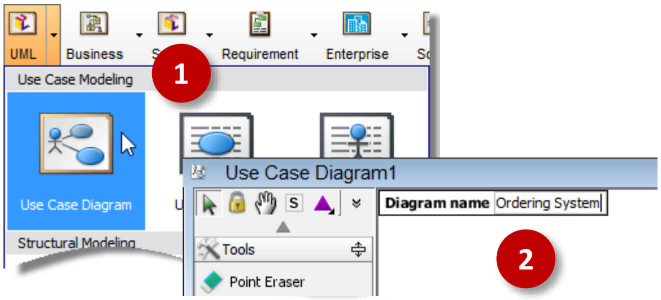


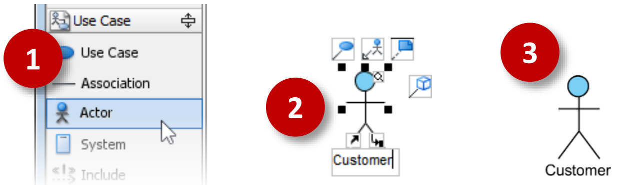
Diagram name Ordering System

Creating and Connecting Shapes

Using the Diagram Toolbar

Let's create an actor from the Diagram Toolbar for now.

1. Click on **Actor** in the diagram toolbar.
2. Click on the diagram to create an actor, and enter its name.
3. Click on the diagram or press **Enter** to confirm.

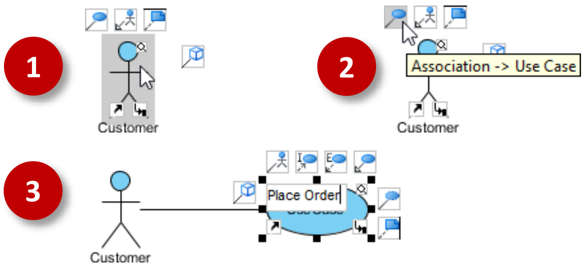


Using the Resource-Centric Interface

If you move your mouse pointer over a shape, you will see a number of icons surrounding it. Those are known as the resource icons. They together formed the resource-centric interface.

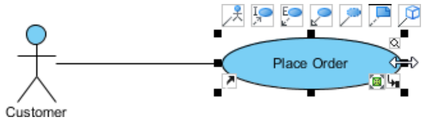
Resource-centric interface allows you to create a new shape that connects with an existing one. You can also use the resource-centric interface to create connector between two shapes. Let's create a use case from actor.

1. Move your mouse pointer over the actor shape.
2. Press on the resource icon **Association -> Use Case** and drag to the right.
3. Release the mouse button to create the use case, and enter its name. This will create a use case that associates with the actor.



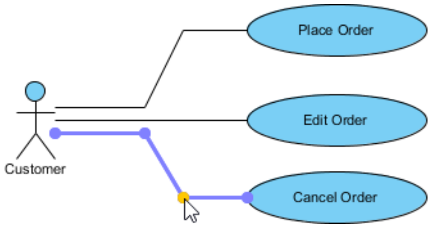
Resizing Shape

When you click on a shape, you will see several resize handlers appear around the shape. You can drag on these handlers to make the shape larger or smaller.



Adding Control Points to Connector

For most diagram types, ‘oblique connector’ is chosen as the default connector type, meaning that when you draw a connector, it will be an oblique connector. To route such connector, you can add control points to it. To add control points, simply drag on the connector to create a point, and then keep dragging to adjust its position.



There are totally five connector types. If you want to apply another connector type on a connector, right click on that connector and select **Styles and Formatting > Connector Styles > [TYPE]** from the popup menu. If you want to update all the connectors in a diagram, right click on the background of diagram and select **Connectors > [TYPE]** from the popup menu.

Documenting Model Element

Generally speaking, naming model elements alone would not suffice to describe their details. To elaborate, you can enter additional details in the **Documentation Pane**. Simply go to the diagram and select the shape you need to document. At the bottom left of the application, open the **Documentation Pane** and fill in the details.

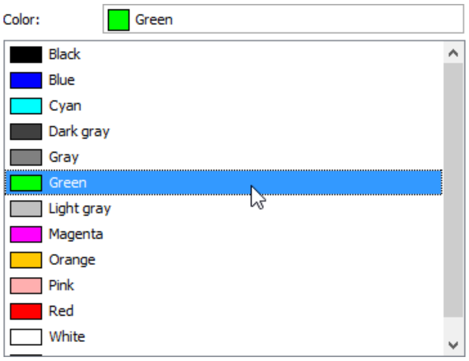
In addition to textual description, voice recording is also available. If your machine supports microphone usage, click the Record button at the bottom of the **Documentation Pane**. In the **Record Voice** window, click the red circle button to start recording. To stop, click the button with a square inside. To save your recording, click **OK**.

Make sure your recording device is available when applying this feature.

Editing Shape Color

Make your diagram more expressive by formatting shapes based on their contexts. Let’s change the color of a use case shape.

1. Right click on the use case shape and select **Styles and Formatting > Formats...** from the popup menu.
2. Open the **Background** tab in the **Formats** window. Select *Green* for color. Click **OK** to confirm the change.



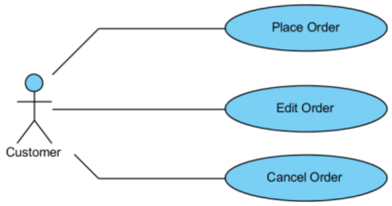
Color: 🟩 Green

UML Modeling

All Editions

Use Case Diagram

A use case diagram is used to model and identify the functional requirements of a software system. In a use case diagram, stakeholders and user goals are represented by actor and use cases, respectively.



An actor is any person or external system that interacts with the system to achieve a user goal (i.e. use case). The use case diagram above illustrates the use cases of an order processing system. Customer, an actor, interacts with the system to accomplish the goal of order placement, as modeled by the use case *Place Order*. There are other goals that the customer wants to accomplish, such as *Edit Order* and *Cancel Order*. You can apply the diagramming techniques described in the previous section to create such diagram.

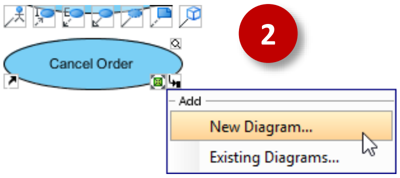
Sequence Diagram

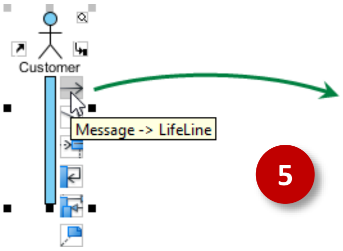
A sequence diagram is primarily used to show the interaction between objects that are represented as lifelines, in a sequential order. More importantly, lifelines show all of their interaction points with other objects in events.

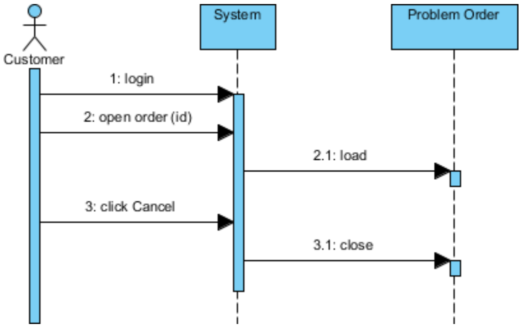
As always, you can create a new diagram via the toolbar, same for sequence diagram. To create sequence diagram, choose **UML > Sequence Diagram** from the toolbar. Here let's create a sequence diagram with an alternative approach – create as a sub-diagram of use case.

The reason of creating a sequence diagram as a sub-diagram of use case is to visualize the interaction between user and system, in accomplishing the use case. Let's see how it works. We will carry on with the use case diagram created in the previous section.

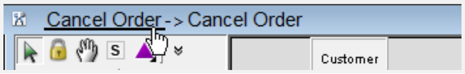
1. Move the mouse pointer over the use case *Cancel Order*.
2. Click on the tiny **Sub Diagrams** button at bottom right of the shape and select **New Diagram…** from the popup menu.
3. This shows the **New Diagram** window. Keep **Sequence Diagram** selected and then click **OK**.
4. This creates a blank sequence diagram. Create an actor *Customer* in the diagram. You can find *Actor* from the diagram toolbar.
5. Let's model the interaction of how customer can cancel an order. Move the mouse pointer over the actor. Press on the **Message -> LifeLine** resource and drag it out.
6. Release the mouse button. Name the lifeline *System*, and the message *login*.
7. Complete the interaction by adding other messages and the *Order* lifeline.







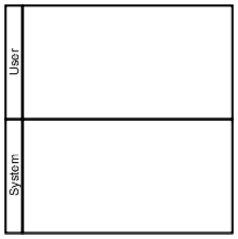
To go back to the parent use case, you can click on the shortcut on top of the diagram:



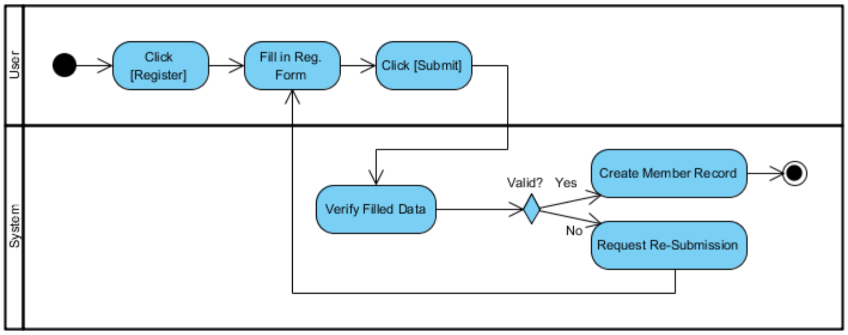
Activity Diagram

An activity diagram is essentially a flowchart, showing flow of control from one activity to another. Unlike a traditional flowchart, it can model the dynamic aspects of a system because it involves modeling the sequential steps in a computational process. Let's make use of activity diagram to model the registration process.

1. Create an empty activity diagram named *Registration*. You can create an activity diagram by selecting **UML > Activity Diagram** from the toolbar.
2. Create a swimlane for grouping actions based on the participants. Select **Horizontal Swimlane** from the diagram toolbar and click on the diagram to create one.
3. Double click on the headers and rename them to *User* and *System*, respectively.



4. Create initial nodes, actions, decision node and activity final node inside the swimlane. Do remember to resize the swimlane to accommodate the shapes.

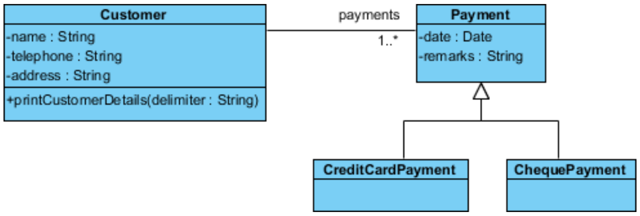


The decision node represent the moment where a decision has to be made. Multiple flows will be produced based on different results of decision making.

The **Control Flow -> Action** resource helps you create action shapes rapidly.

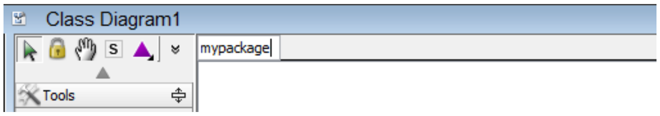
Class Diagram

A class diagram models the blueprints of objects required by a system and the relationships between them. Let's make use of class diagram to model the domain classes of the order processing system.

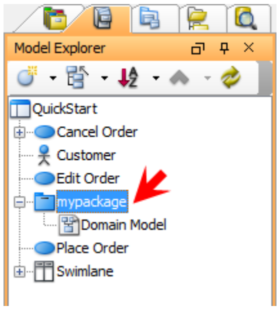


Class Diagram and Package Header

When you create a class diagram, you are prompted to enter the package header (see the image below). Package header is a quick way to create new class diagram in a package. The diagram and the shapes created in the diagram will all be grouped by the package specified. Let's say if you want the class diagram be grouped by a package *mypackage*, enter *mypackage* in package header.



By doing so, a new package will be created, with the new class diagram put inside. You can inspect the structuring in the **Model Explorer**.

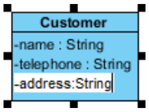


If the package you specified exists, the new class diagram will be put in that existing package. If the specified package does not exist, a new package will be created.

You may enter fully qualified name – for example, *com.example.mypackage*. By doing so, multiple packages will be created (if not exist), nested package hierarchy will be formed.

Attributes and Operations

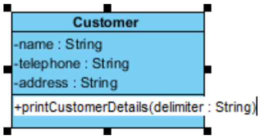
Properties of a class are represented by attributes. To add an attribute:



1. Right click on a class and select **Add > Attribute** from the popup menu.
2. Type the name and type of attribute inline - for example, *name : String*.
3. Press **Enter** to confirm.
4. You can now create the next attribute by repeating step 2. To stop, press **Esc**.

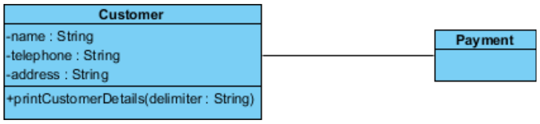
Features of a class are represented by operations. To add an operation:

1. Right click on a class and select **Add > Operation** from the popup menu.
2. Type the name and optionally the method signature (i.e. parameters and return type) - for example, *printCustomerDetails(delimiter : String)*.
3. Press **Enter** to confirm.
4. You can now create the next operation by repeating step 2. To stop, press **Esc**.



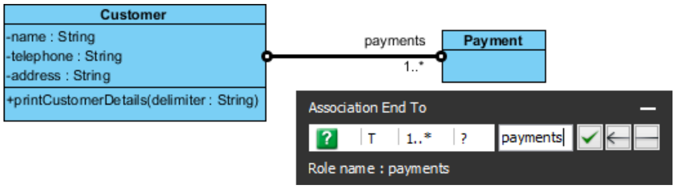
Classes with Association

To associate two classes:



1. Move the mouse pointer over the source class.
2. Press on the **Association -> Class** resource icon and drag it out.
3. Release the mouse button to create a class with an association.
4. Name the class and press **Enter**.

To edit an association end, double-click on the end to open the **Association Editor**. Enter a name for the role of the association in the middle text box and adjust properties like multiplicity and navigability as needed.



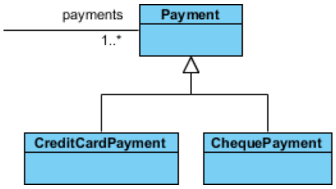
Association End To

? T 1..* ? payments

Role name : payments

Classes with Generalization

Generalization models “a-kind-of” relationship among classes. In Visual Paradigm, generalization can be created easily via the resource-centric interface, from super-class to sub-class.



1. Move the mouse pointer over the super class.
2. Press on the **Generalization -> Class** resource icon and drag it out.
3. Release the mouse button to create a class with a generalization.
4. Name the sub-class and press **Enter**.

Business Process Modeling (with BPMN)

Business Process Diagram (BPD)

All Editions

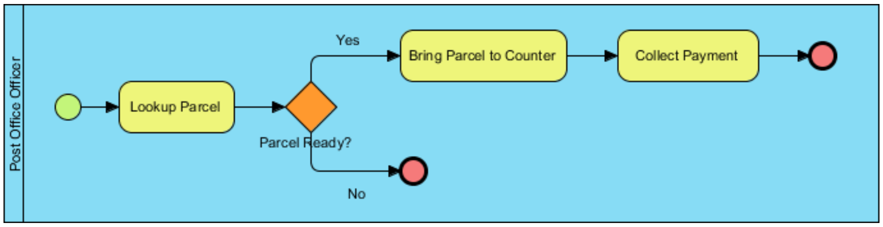
You draw Business Process Diagram (BPD) when you want to represent operational workflow. A BPD is mainly composed of activity elements such as tasks and sub-processes. They both represent work that an organization performs in a business process.

Here is a list of frequently-seen elements in a BPD:

Symbol	Name	Description
●	Start event	Where the process begins, and under what condition
◉	Intermediate event	Drive business flow based on the event it specifies. Intermediate event can be attached to an activity for modeling an event that may happen during the execution of that activity. It may also be connected by a connecting object for modeling an event that may happen after the execution of the flow element before.
●	End event	Indicate where a business process completes
Task	Task	Atomic work which cannot be further broken down.
Process ⊞	Sub-process	Non-atomic, complex work that can be elaborated into smaller works
◆	Gateway	The diagram will be displayed in diagram pane.

Let's make use of BPD to model a parcel collection process.

1. Create an empty BPD named *Parcel Collection*. You can create a BPD by selecting **Business > Business Process Diagram** from the toolbar.
2. Create a pool for the post office officer, the person who execute the process. Select **Horizontal Pool** from the diagram toolbar and click on the diagram to create one. Name the pool *Post Office Officer*.
3. Create start event, tasks, gateway and end event inside the pool.

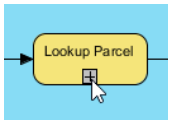


Working with Sub-Process

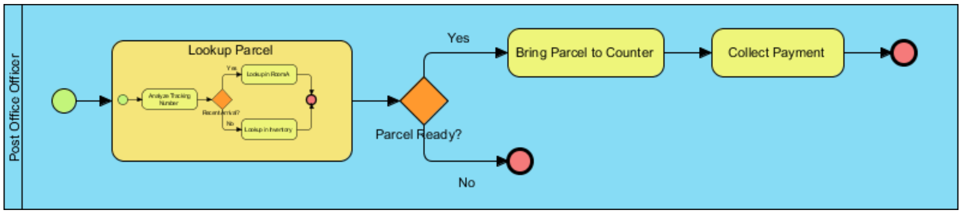
Converting Task to Sub-Process

When your business expanded, you may need to revise your business process design by including more details. In such case you may need to convert a task into a sub-process, and to detail its workflow in another business process diagram. To convert a task to a sub-process, right click on the desired task and select **Convert to Sub-Process** from the popup menu.

Drilling-Down Sub-Process



A sub-process can be expanded to show a lower-level process. To expand a sub-process, click on the "+" symbol at the bottom of the sub-process shape. By doing this, a new BPD will be opened. You can then model the internal details using activities, gateways, events, etc. When you go back to the parent level BPD, you will see the sub-level details appear in the sub-process shape.

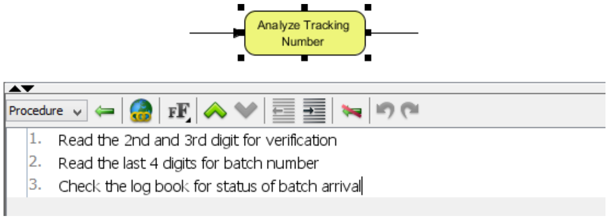


To collapse an expanded sub-process shape, click "-" at its bottom. To expand it, click "+" again.

Documenting Working Procedure

All Editions

The necessary procedure to handle a task/sub-process can be specified by filling in the operating procedure. To specify working procedure of a task/sub-process, simply select the desired task/sub-process and enter the procedure in the **Procedure Editor**, below the diagram. If you don't see the editor opened, right click on the background of diagram and select **Show Procedure Editor** from the popup menu.



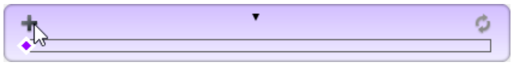
1. Read the 2nd and 3rd digit for verification
2. Read the last 4 digits for batch number
3. Check the log book for status of batch arrival

Animating Business Process

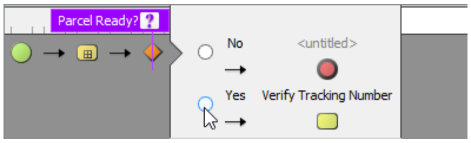
Professional Edition +

When you present your business process design to someone, like your clients or manager, having a way to visualize the workflow dynamically can help you express the design more effectively, and that's what the animation feature can help. The animation feature allows you to animate your design, from the very first element, which is typically the start event, until the final element. To animate a BPD:

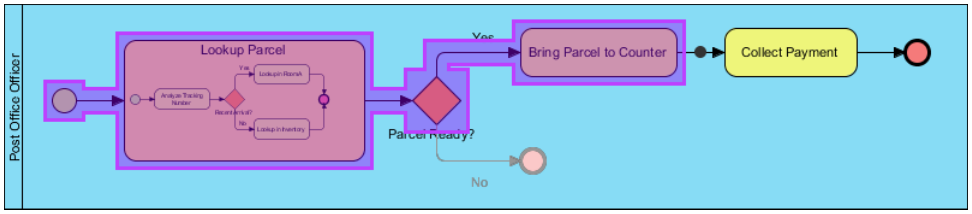
1. Right click on the background of BPD and select **Show Animation Panel** from the popup menu.
2. Click "+" at the top of the **Animation Panel**.



3. Enter the name of the path to animate – for example, *Collect parcel (success)*.
4. If your process design contains gateway (or any other kind of elements for decision making), you may need to construct the path manually.



5. Click ▶ to play the path. Shapes having been passed are highlighted in purple by default. And whenever the animation is progressing through a sequence or message flow, a dark ball would appear and move along the line to guide your eye.



Requirements Gathering

Professional Edition +

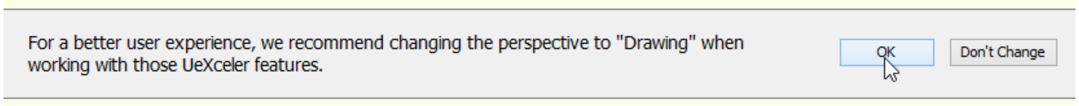
Visual Paradigm helps you capture the right software requirements at the very beginning of your project. Let's see how the requirements gathering features work.

Managing Use Cases in Use Case List

A use case represents a high level function that yields a measurable result of values. The Use Case List provides a high level view of what the user want to achieve by using the software.

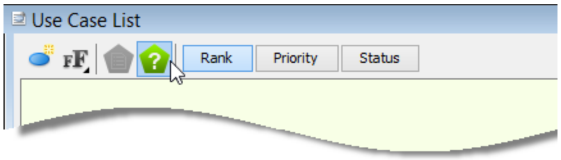
Let's open the **Use Case List** and create some use cases in it.

1. Create a new project first. You can create a new project by selecting **File > New Project** from the main menu. (We will go back to *QuickStart.vpp* later on.)
2. Select **Modeling > Use Case List** from the main menu.
3. You are recommended to change perspective to **Drawing**. Click **Keep Change** to confirm.

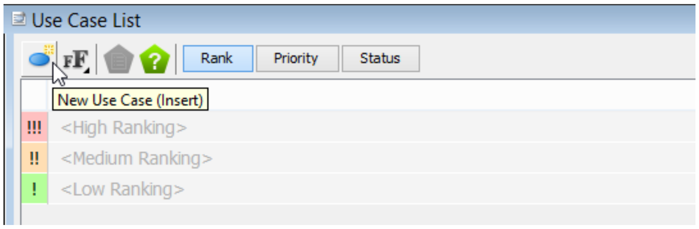


For a better user experience, we recommend changing the perspective to "Drawing" when working with those UeXceler features.

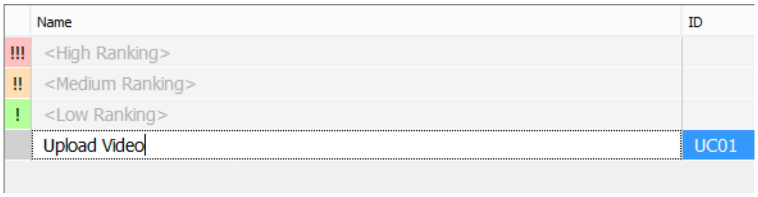
4. On top of the **Use Case List**, click the button with a green pentagon to close the tips popup.



5. Let's start by creating use cases. Click on the **New Use Case** button on top of the **Use Case List**.

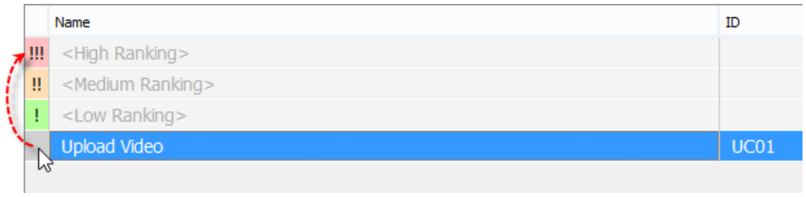


6. Enter *Upload Video* as the name of the use case.



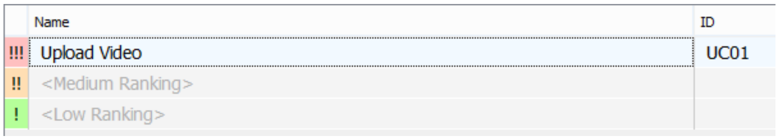
	Name	ID
!!!	<High Ranking>	
!!	<Medium Ranking>	
!	<Low Ranking>	
	Upload Video	UC01

7. Assuming that this use case has a high business value. Drag it to the **<High Ranking>** section.



	Name	ID
!!!	<High Ranking>	
!!	<Medium Ranking>	
!	<Low Ranking>	
	Upload Video	UC01

8. Release the mouse button. You should see this.



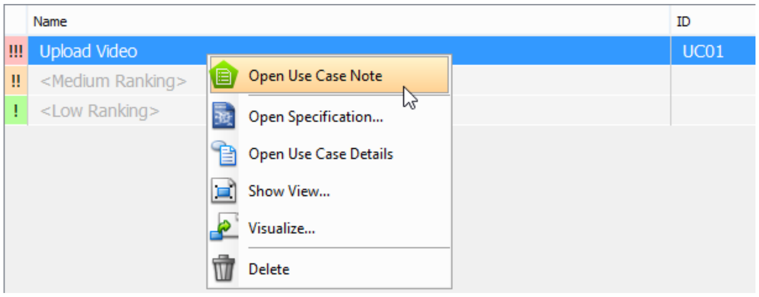
	Name	ID
!!!	Upload Video	UC01
!!	<Medium Ranking>	
!	<Low Ranking>	

Noting Down Use Case -Related Ideas

When you meet with user in knowing his/her needs, you will be told a lot of information that is relevant and important to the development of the system. The **Use Case Note** provides you with a notepad for writing down what he/she said in point form.

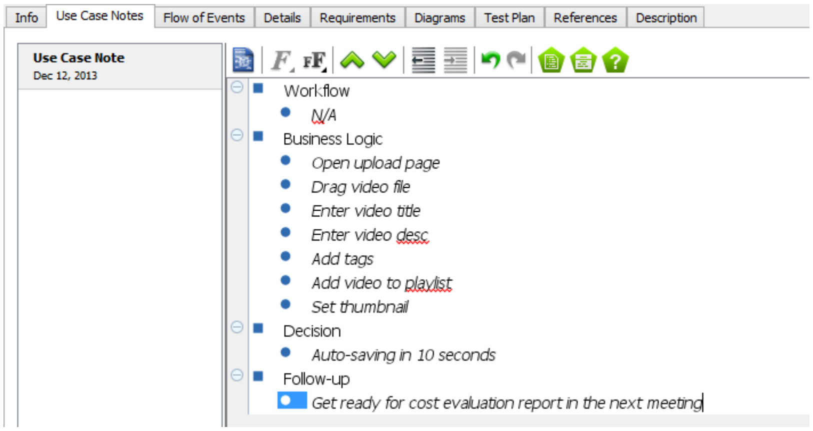
Let's enter notes for the *Upload Video* use case you created in the previous section.

1. Right click on the *Upload Video* use case in the **Use Case List** and select **Open Use Case Note** from the popup menu.



	Name	ID
!!!	Upload Video	UC01
!!	<Medium Ranking>	
!	<Low Ranking>	

2. This opens the **Use Case Notes** editor. There is a predefined template that guides you in noting use case – related ideas. Let's click on the green text and replace with note text. You can enter a new line by pressing **Enter**.



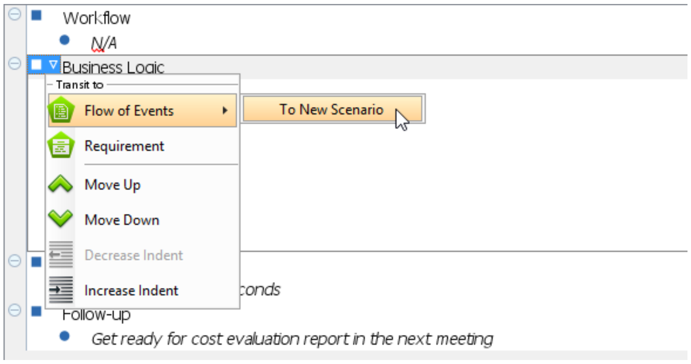
Use Case Note
Dec 12, 2013

- Workflow
  - N/A
- Business Logic
  - Open upload page
  - Drag video file
  - Enter video title
  - Enter video desc
  - Add tags
  - Add video to playlist
  - Set thumbnail
- Decision
  - Auto-saving in 10 seconds
- Follow-up
  - Get ready for cost evaluation report in the next meeting

Writing Use Case Scenario

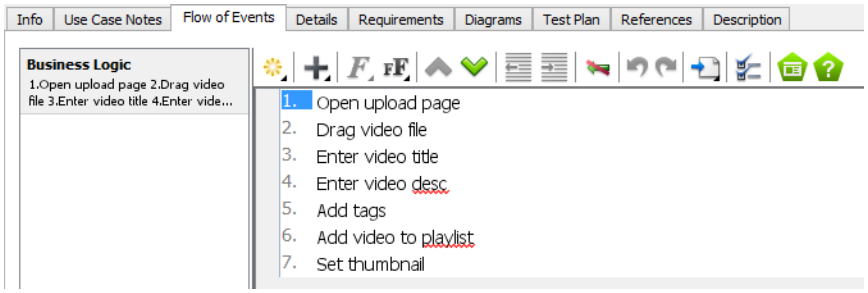
Now, you have some notes collected from stakeholder. Some of the notes are about the preferred behavior of software. You can make use of these notes in forming a standard use case scenario.

1. Move the mouse pointer to the left hand side of the note Business Logic. Click on the down arrow next to the bullet point and select **Flow of Events > To New Scenario** from the popup menu.



- Workflow
  - N/A
- Business Logic
- ...conds
- Follow-up
  - Get ready for cost evaluation report in the next meeting

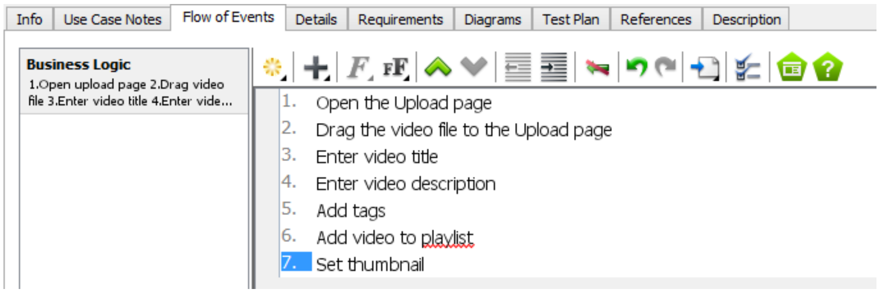
2. The **Flow of Events** tab is opened, with the note items under *Business Logic* added as steps of the scenario.



Business Logic
1.Open upload page 2.Drag video file 3.Enter video title 4.Enter vide...

1. Open upload page
2. Drag video file
3. Enter video title
4. Enter video desc
5. Add tags
6. Add video to playlist
7. Set thumbnail

3. Rephrase the scenario text with proper use of English and more clear description.



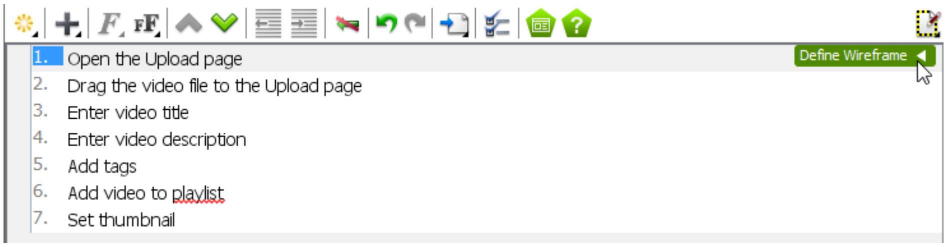
Business Logic
1.Open upload page 2.Drag video file 3.Enter video title 4.Enter vide...

1. Open the Upload page
2. Drag the video file to the Upload page
3. Enter video title
4. Enter video description
5. Add tags
6. Add video to playlist
7. Set thumbnail

Scenario-based wireframing

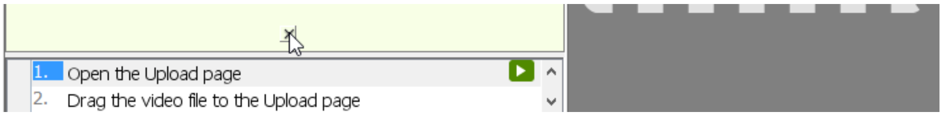
Wireframe is a sketch of the system to be developed. We use wireframe to show users how the system will look like, with minimal cost and effort. In Visual Paradigm, wireframe can be drawn and associated with scenario. Let's try.

1. Click on step 1.
2. On the right hand side, click on the button that has a triangle on it.



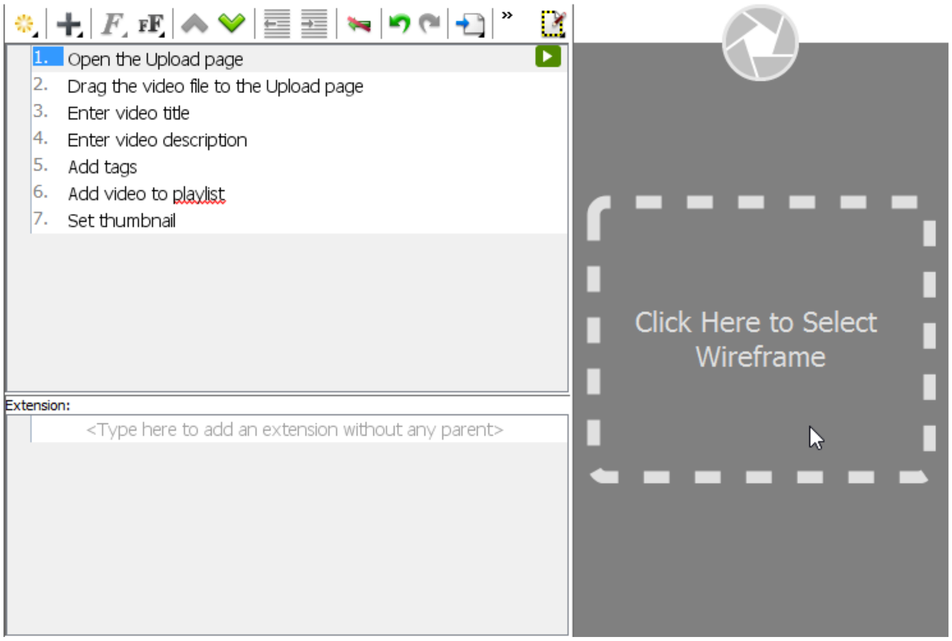
1. Open the Upload page
2. Drag the video file to the Upload page
3. Enter video title
4. Enter video description
5. Add tags
6. Add video to playlist
7. Set thumbnail

3. You are presented the help contents. Click on the close button to hide it.



1. Open the Upload page
2. Drag the video file to the Upload page

4. You see the gray pane appear on the right hand side? Click on it to select a kind of wireframe to create.

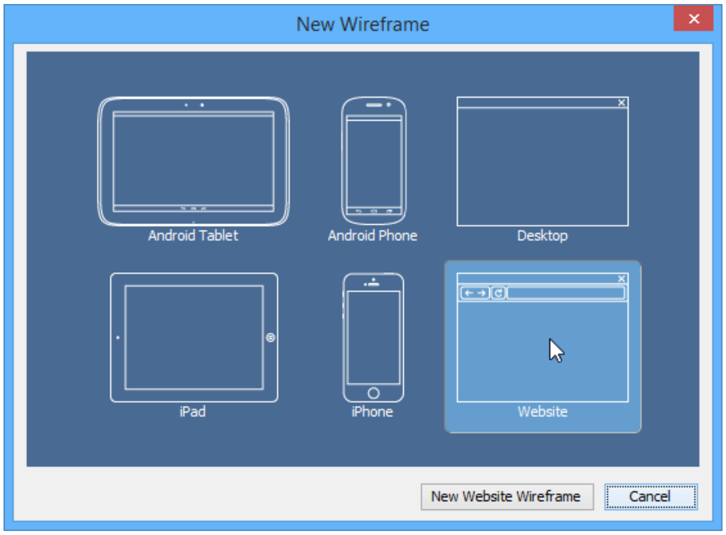


1. Open the Upload page
2. Drag the video file to the Upload page
3. Enter video title
4. Enter video description
5. Add tags
6. Add video to playlist
7. Set thumbnail

Extension:

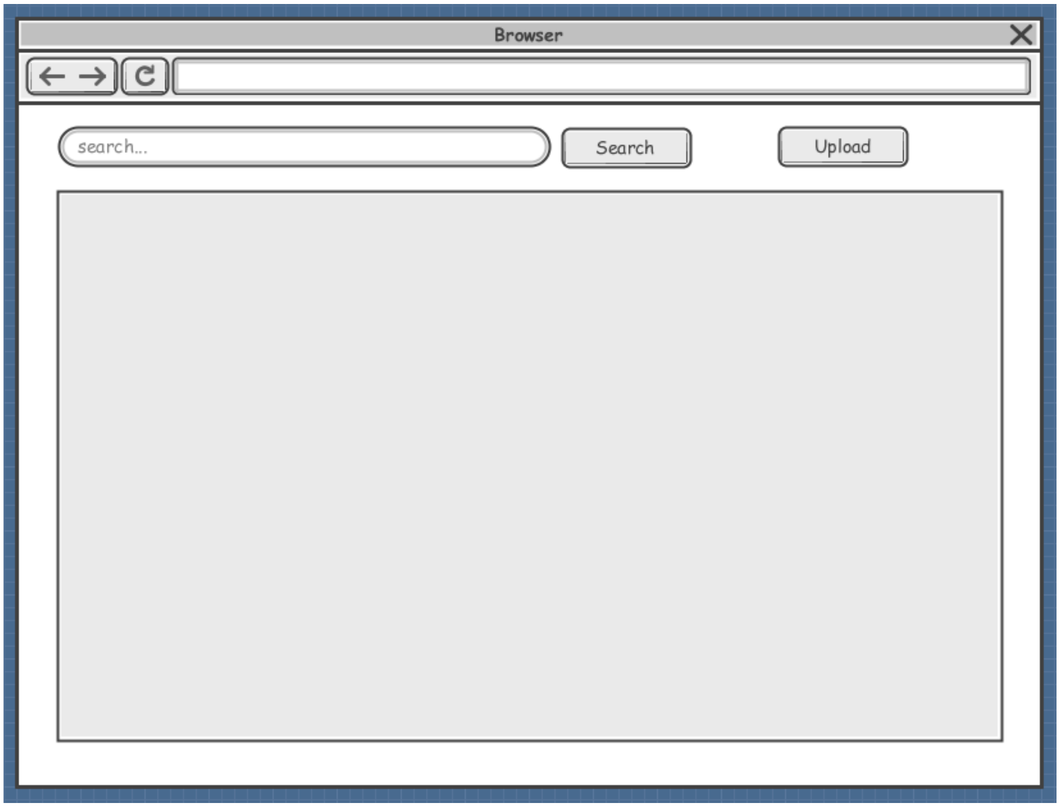
<Type here to add an extension without any parent>

5. In the popup window, select **Website**.
6. Click **New Website Wireframe**.

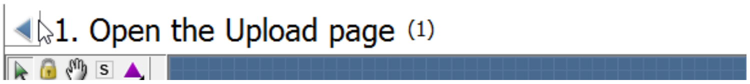


7. A new wireframe appear, with an empty browser window in it. Apply the diagramming techniques described above to draw a wireframe like the following. About the search field, it’s actually a text field.

Just create a text field, right click on a text field and select **Search Field** to convert it into a search field.



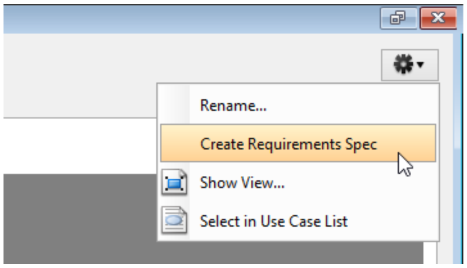
8. Go back to the flow of events by clicking on the triangle button next to the step title.



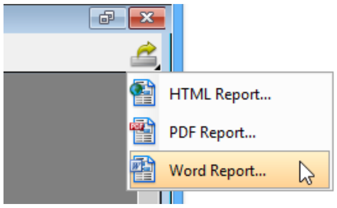
Producing Requirement Specification

You can generate a functional specification for one use case which includes usage scenarios, wireframe design requirements and more. The stakeholder can easily verify the specification to development plans. Let's try.

1. At the top right corner of the **Flow of Events** editor, click on the gear button.
2. Select **Create Requirements Spec** from the drop down menu.



3. This creates a requirement specification in Report Composer. Now, you may edit the report, click export it into HTML/PDF/Word file by clicking on the **Export** button at the top right corner of Report Composer.



Code Engineering

Standard Edition +

Java Round-Trip

Round-trip engineering enables you to produce source code from UML model (and the other way round) and keep the source code and UML model synchronized. With Java round-trip, you can generate Java source code from class model, or the reverse a Java code-base to class diagram.

To generate Java source code from UML model:

1. Select **Tools > Code Engineering > Java Round-trip > Generate Code…** from the main menu.
2. Edit the output folder in the **Generate Code** window.
3. Click **OK** to generate code.

To reverse engineer class model from Java source code:

1. Select **Tools > Code Engineering > Java Round-trip > Reverse Code…** from the main menu.
2. Edit the source folder in the **Reverse Code** window.
3. Click **OK** to reverse code.
4. You can then create class diagram with the reversed classes. Simply open the **Model Explorer** to find the classes, and then drag them to class diagram to visualize them.

C++ Round-Trip

Round-trip engineering enables you to produce source code from UML model (and the other way round) and keep the source code and UML model synchronized. With C++ round-trip, you can generate C++ source code from class model, or the reverse a C++ code-base to class diagram.

To generate C++ source code from UML model:

1. Select **Tools > Code Engineering > C++ Round-trip > Generate Code…** from the main menu.
2. Edit the source and Cpp folder in the **Generate Code** window.
3. Click **OK** to generate code.

To reverse engineer class model from C++ source code:

1. Select **Tools > Code Engineering > C++ Round-trip > Reverse Code…** from the main menu.
2. Edit the source and Cpp folder in the **Reverse Code** window.
3. Click **OK** to reverse code.
4. You can then create class diagram with the reversed classes. Simply open the **Model Explorer** to find the classes, and then drag them to class diagram to visualize them.

Instant Reverse

Instant Reverse allows you to reverse different types of source into UML class models, such as Java, C++, CORBA, Ada, PHP, Python, Objective-C, etc. To reverse, select **Tools > Code Engineering > Instant Reverse** from the main menu, then select the kind of programming language to reverse. Select the source files and proceed.

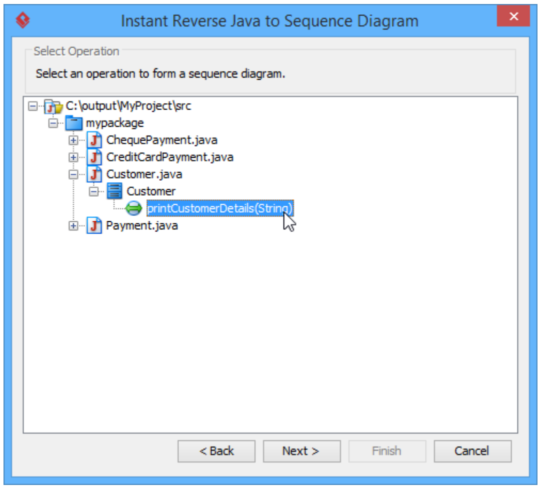
Instant Generator

Instant generator produces source code from your model at a particular instant. Unlike the code generation support in round-trip engineering, instant generator is a one-off. To generate code, select **Tools > Code Engineering > Instant Generator** from the main menu, then select the programming language in which to generate.

Reverse Engineer Sequence Diagram from Java

Sequence diagram can help represent interactions between objects in runtime. Visual Paradigm enables you to reverse your Java source code to sequence diagram. You can gain a better understanding of a piece of Java source code by reading its corresponding diagram, instead of looking at possibly a thousand lines of code. To reverse Java code to sequence diagram:

1. Select **Tools > Code Engineering > Instant Reverse > Java to Sequence Diagram…** from the main menu.
2. Add the folder that contains the source code. Click **Next**.
3. Expand the tree to select the class and its operation to be reversed. Click **Next**.
4. In the **Choose Diagram** screen, select **Create new sequence diagram**.
5. Click **Finish**



Instant Reverse Java to Sequence Diagram

Select Operation

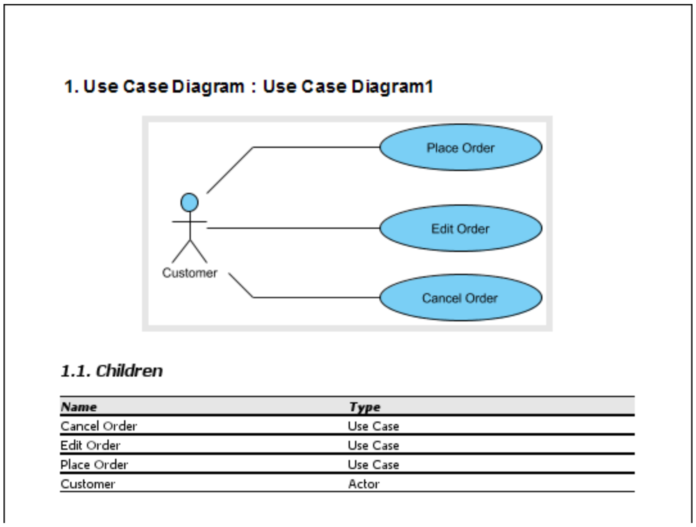
Select an operation to form a sequence diagram.

Reporting

Standard Edition +

Report Composer

Report Composer is a reporting creation tool. You develop a report by dragging a dropping templates to the report editor, forming a complete report. The result can then be exported to MS Word, HTML and PDF file.



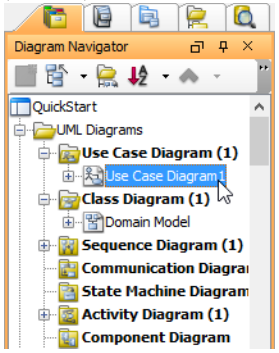
1. Use Case Diagram : Use Case Diagram1

1.1. Children

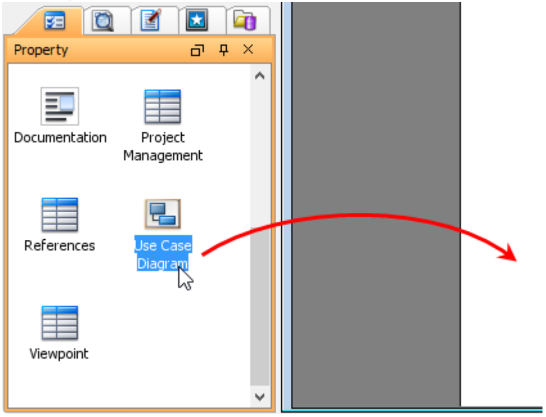
Name	Type
Cancel Order	Use Case
Edit Order	Use Case
Place Order	Use Case
Customer	Actor

To use Report Composer:

1. Create a new report via the toolbar (**Diagrams > Report**).
2. Select in **Diagram Navigator**, **Model Explorer** or **Class Repository** the model element/diagram that you want to show in your report.



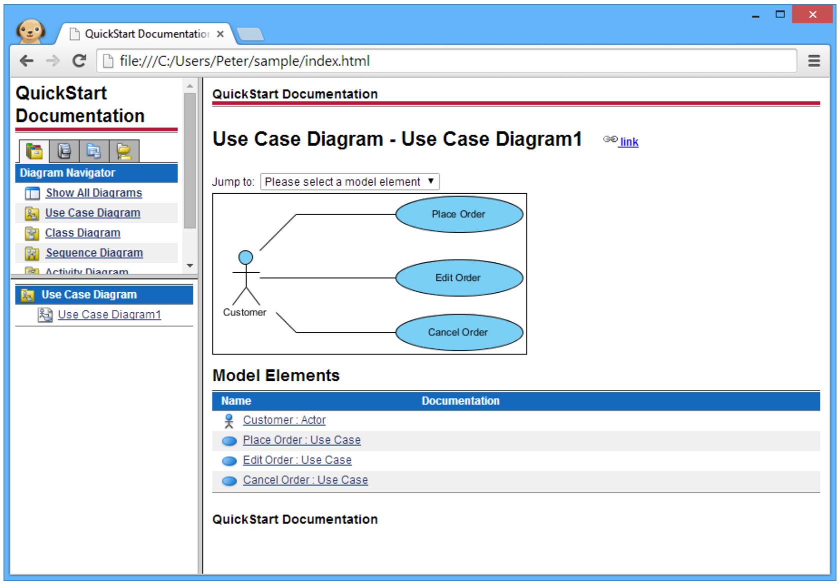
3. The **Property Pane** is updated to list out the available templates based on your selection. Choose the appropriate template and drag it to the report editor.



To export report as MS Word/HTML/PDF file, click on at the top right corner of Report Composer.

Project Publisher

Project Publisher helps produce Web contents from your project file.



QuickStart Documentation

Use Case Diagram - Use Case Diagram1

link

Jump to: Please select a model element

Model Elements

Name	Documentation
Customer : Actor	
Place Order : Use Case	
Edit Order : Use Case	
Cancel Order : Use Case	

QuickStart Documentation

To publish project with Project Publisher:

1. Select **Tools > Project Publisher** from the main menu.
2. In the **Project Publisher** window, enter the output folder.
3. Click **OK** to start publishing.

Collaborative Modeling

All Editions

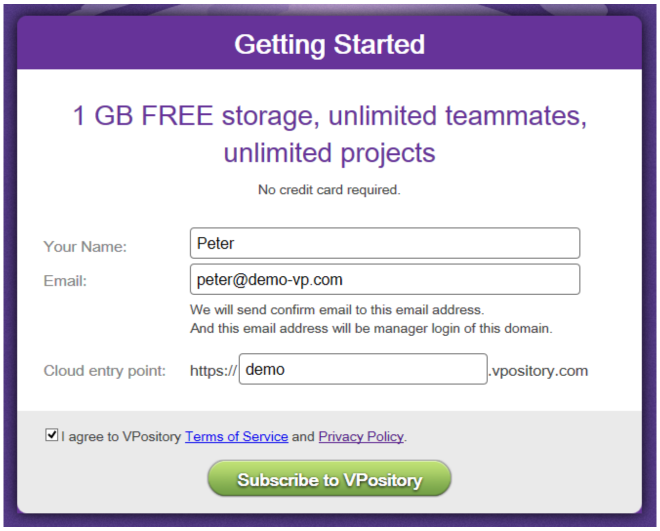
If you work as a team and need to work together on the same design, or to share your design with your teammates, you need to make use of VPository.

VPository is the cloud-based repository where designs can be stored. You can upload your project to VPository, and let your teammates to open and edit it.

Subscribing to VPository

In order to use VPository, you have to subscribe to it first. Subscription of VPository is made inside Visual Paradigm. To subscribe to VPository for your team:

1. Select **Team > Open Teamwork Client** from the toolbar of Visual Paradigm.
2. In the **Teamwork Client** window, click **Subscribe to VPository…** at bottom left.
3. You should see the VPository subscription page opened in your web browser. Enter your name, email address and the cloud entry point. The cloud entry point is the unique URL of your VPository. Your team will access the entry point to enter VPository for managing and opening the projects stored in VPository.
4. Read through the Terms of Service and Privacy Policy. Check **I agree to VPository Teams of Service and Privacy Policy** if you agree to them.
5. Click **Subscribe to VPository**.



Getting Started

1 GB FREE storage, unlimited teammates, unlimited projects

No credit card required.

Your Name: Peter

Email: peter@demo-vp.com

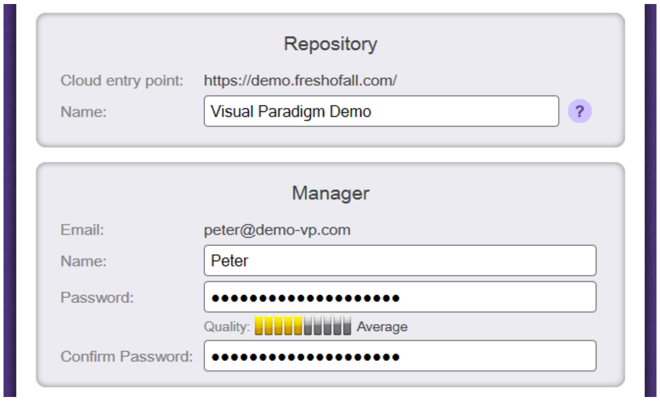
We will send confirm email to this email address.
And this email address will be manager login of this domain.

Cloud entry point: https:// demo .vpository.com

☑ I agree to VPository Terms of Service and Privacy Policy.

Subscribe to VPository

6. Check your Email client. You should receive an Email with **VPository subscription confirmation** as subject.
7. Click on the hyperlink **Confirm your VPository Subscription** in the Email.
8. This opens another web page, asking you for details about the repository and you, the manager of the VPository account. Enter the information and click **Start My VPository**.



Repository

Cloud entry point: https://demo.freshofall.com/

Name: Visual Paradigm Demo ?

Manager

Email: peter@demo-vp.com

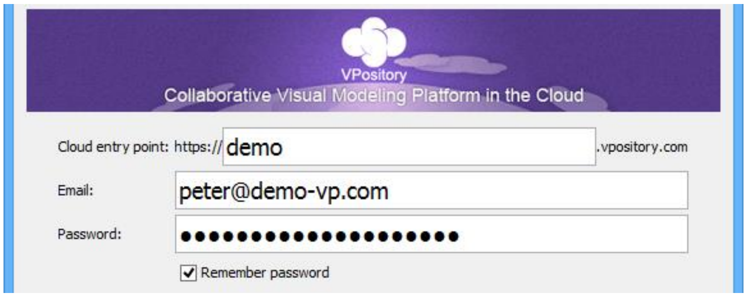
Name: Peter

Password: •••••••••••••••••••

Quality: Average

Confirm Password: •••••••••••••••••••

9. You've finished subscribing to VPository. Now, go back to Visual Paradigm. You should see that the login details have been filled for you automatically.



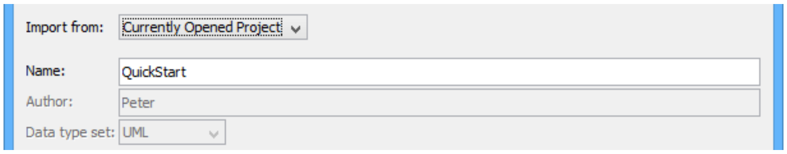
Cloud entry point: https:// demo .vpository.com

Email: peter@demo-vp.com

Password: ●●●●●●●●●●●●●●●●●●●●●

☑ Remember password

10. Click **Login**. Because your VPository is newly setup, you are suggested to create a project in it. Just click **OK** to continue.
11. This shows the **Import Project to VPository** screen. If you are opening the Quick Start project, the option **Currently Opened Project** should be selected for **Import from**. Just keep it selected. In practice, you can change to import a new project, or to import an existing .vpp project file to VPository.



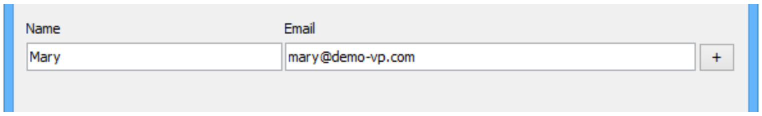
Import from: Currently Opened Project

Name: QuickStart

Author: Peter

Data type set: UML

12. At the bottom of the window, click **Add Project Member**.
13. Let's invite your teammate to work on this project. Click **Invite New Member…** at the bottom.
14. Enter the name and Email address of your teammate and click **OK**.



Name	Email
Mary	mary@demo-vp.com

15. Click **Import**. Invitees will receive invitation Emails. By accepting the invitation, they can login VPository from Visual Paradigm, with the cloud entry point and their own password (provided during the acceptance of invitation).
16. Click **Open** at the bottom right of the **Teamwork Client** window to open the imported project.

Importing a Project

To import a new project to server:

1. Select **Team > Open Teamwork Client** from the toolbar of Visual Paradigm.
2. In the **Teamwork Client** window, select **Project > Import Project to Repository** from its menu.
3. Specify the project source in the **Import from** field, edit the **project name**, **author** and **description**.
4. Invite your teammate(s) by clicking **Add Project Member** at the bottom of the window.
5. Click **Import**.

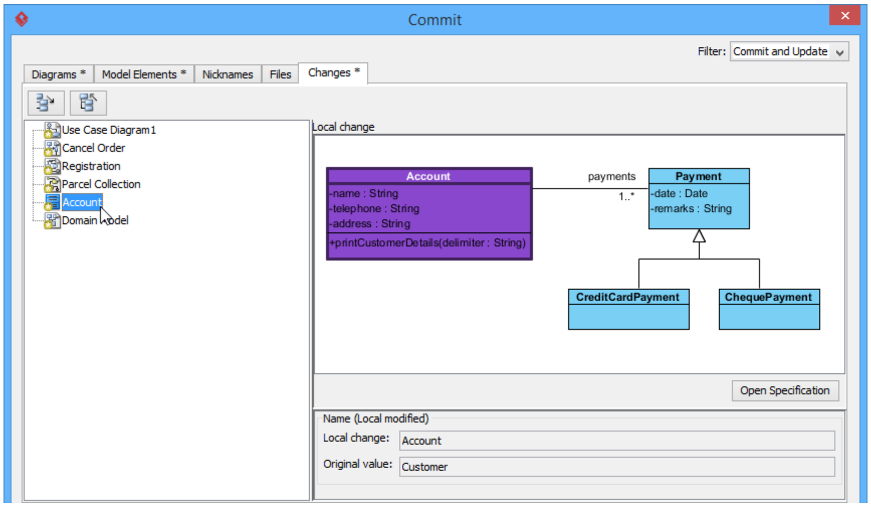
Committing

Commit is the process of uploading changes made in the working copy back to server. As you, the team member, make changes in a project, you can share your works by committing those changes to the VPository, and let others to update the changes from VPository.

To commit changes:

1. Select **Team > Commit…** from the toolbar.

2. Review the changes in the **Commit** window, which involves both the changes you made and the changes made by others and will be updated to your project.



Filter: Commit and Update

Name (Local modified)

Local change: Account

Original value: Customer

3. Click **Commit**.

When finish committing, the changes you made will be uploaded to server and, at the same time, latest changes in server are updated to your local project copy.

Updating

Update is the process of refreshing the working copy by merging changes that others have made and committed to server.

To update changes from VPository:

1. Select **Team > Update…** from the toolbar.
2. Review the changes in the **Update** window.
3. Click **Update**.

Sharing Your Design with PostMania

When you finish your design, you might want to share it with someone, to ask him/her for verification, approval or take certain action based on your design (e.g. to execute a process plan modeled with a BPD). In Visual Paradigm, an effective way to share your design is by using PostMania.

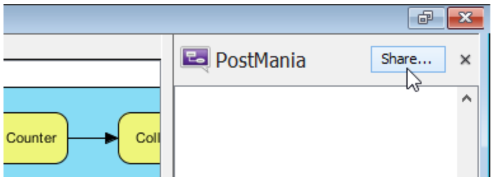
PostMania is a tool built into VPository. It allows you to share your design with someone. It also allows the reader known as a “viewer” in PostMania, to share his comments with you.

There is an important concept that you must be clear about when using PostMania. In PostMania, the people who can draw diagrams (in Visual Paradigm) and share them with someone else are known as “member”, while the people who can **only** view the diagrams are known as “viewer”. To make it simpler, members are usually the designer, while viewers are typically the clients, stakeholders or end users.

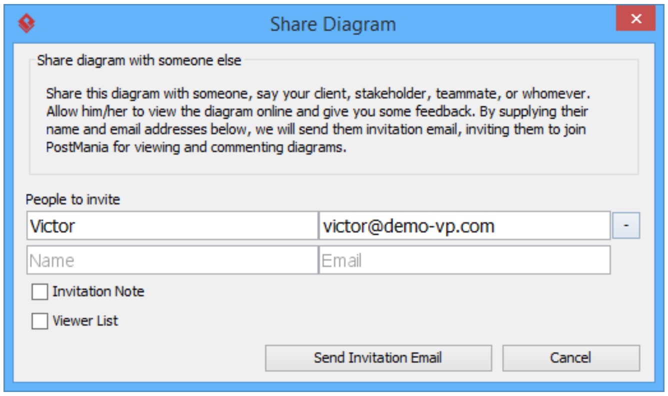
Sharing Diagram

To share your design with PostMania:

1. If you have connected to VPository, you should see the **PostMania Topic Pane** on the right hand side of each of your diagrams in Visual Paradigm. If you do not see the **PostMania Topic Pane**, open it by selecting **Team > PostMania Topic Pane** from the toolbar.
2. Open the diagram you want to share with others. Click **Share...** at the top of the **PostMania Topic Pane**.



3. In the **Share Diagram** window, enter the name and Email address of the people to share with.



Share Diagram

Share diagram with someone else

Share this diagram with someone, say your client, stakeholder, teammate, or whomever. Allow him/her to view the diagram online and give you some feedback. By supplying their name and email addresses below, we will send them invitation email, inviting them to join PostMania for viewing and commenting diagrams.

People to invite

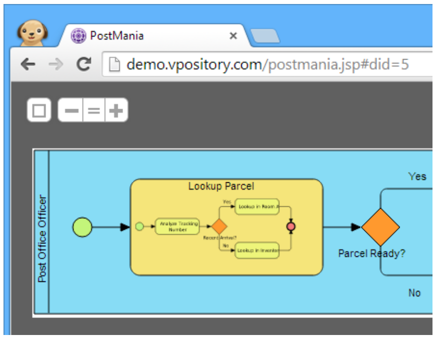
Victor	victor@demo-vp.com	-
Name	Email	

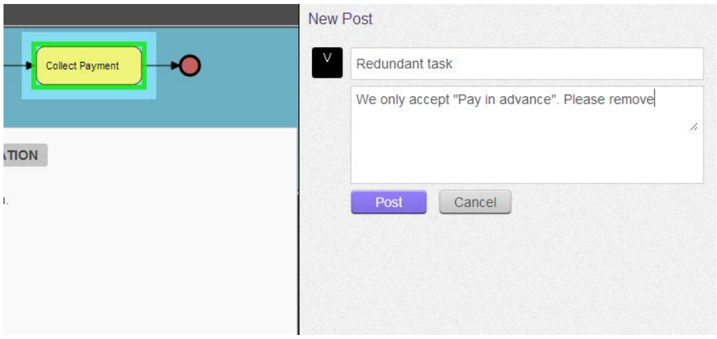
□ Invitation Note

□ Viewer List

4. Click **Send Invitation Email**. In order for these people to view the shared diagram, they have to join PostMania. Here we are going to send them invitation Emails.

Invitation Emails will be sent out in three minutes. Invitees can then accept the invitation to join PostMania. After that, they can view the diagram you shared in web browser. They can also post comments to diagrams, to shapes or to properties like documentation, flow of events (for use case), etc.





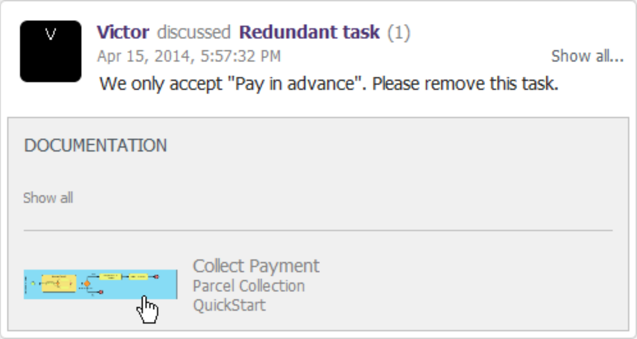
Redundant task

We only accept "Pay in advance". Please remove

Reacting to Comments

When a viewer has posted a comment, as a member, you will see a number appear on the **Team** button in the toolbar. The number indicates the number of unread posts.

To view the unread posts, select **Team > PostMania Page** from the toolbar. In the **PostMania Page**, you can see the posts listed. By clicking on a post...

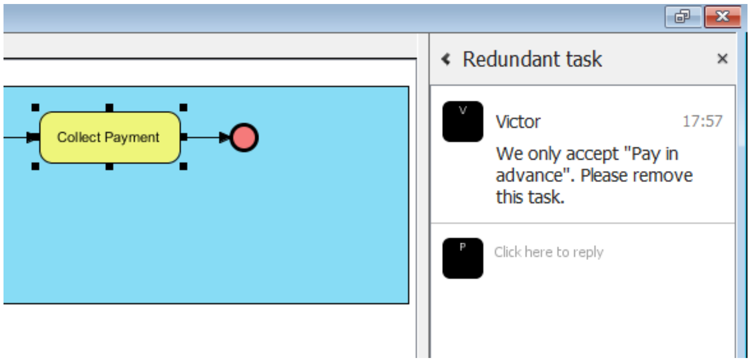


Victor discussed Redundant task (1)

Apr 15, 2014, 5:57:32 PM

We only accept "Pay in advance". Please remove this task.

The related diagram will be opened, with highlight to the discussed area. You can then refine the design, commit the change and add a reply in the **PostMania Topic Pane** to ask the viewer to verify the change.



Victor 17:57

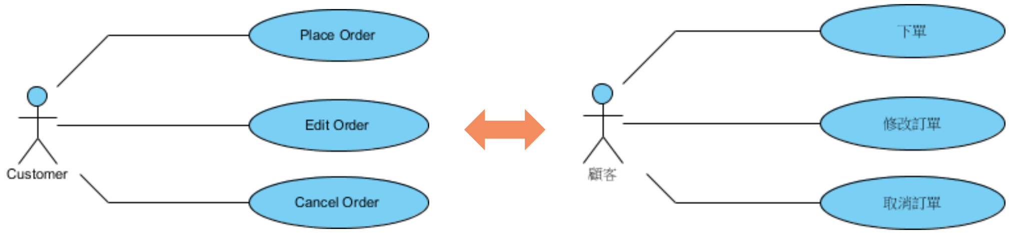
We only accept "Pay in advance". Please remove this task.

Advanced Modeling

Standard Edition +

Nickname

If you are working in or working for multinational corporations, you may need to maintain your model in multiple languages. The Nickname feature helps you achieve that. It allows you to maintain multiple languages in one single project, without having to keep multiple project files for same content.



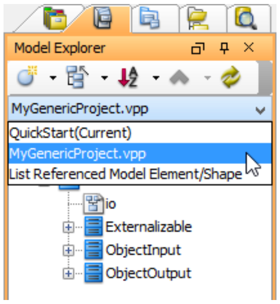
To add a new language:

1. Select **View > Nicknames > Configure Nicknames…** from the main menu.
2. In the **Configure Nickname** window, click **Add User Language…**.
3. In the **Add User Language** window, select the language to add.
4. The newly added language is selected in the **Configure Nicknames** window. Keep it selected and click **OK**.
5. Now, you can rename the shapes and diagrams to “translate” the model to the language added. To switch the model back to the original language, select **View > Nicknames > Original** from the main menu.

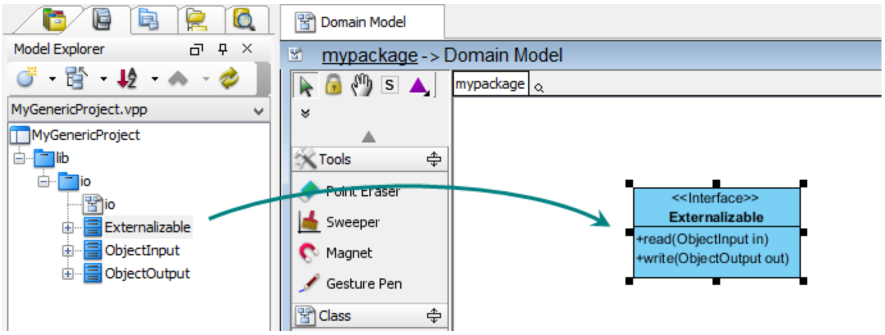
Project Reference

To avoid creating the same things (e.g. a class) over and over again, it would be useful to have a generic library to keep components for reuse. When you make any changes to the components in the library, those changes will ripple down to where the components are actually used. In Visual Paradigm, we call this generic library a “Reference Project”. To add project reference and re-use the data in referenced project:

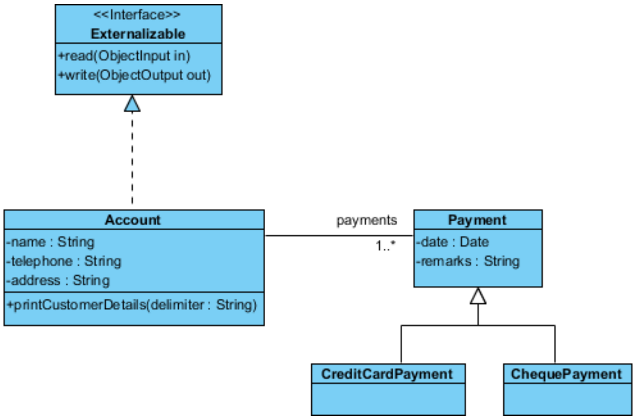
1. Select **File > Manage Referenced Project** from the main menu.
2. In the **Manage Referenced Projects** window, click **Add**.
3. Choose the .vpp project file to reference to and click **Open**.
4. Close the **Manage Referenced Projects** window by clicking **Close**.
5. Open the **Model Explorer**. The referenced project(s) are listed in the drop down menu at the top of the **Model Explorer**. You can switch between the current project and the referenced project through the drop down menu to see the elements in them.



6. You can re-use a model element from referenced project by drag-and-drop.



7. You can connect referenced project data with the data of the current editing project.



Using Mirror

The benefit of using referenced model is to prevent your working project from becoming oversized as the information of referenced model will not be stored. However, since the referenced model is read-only from its source project, you cannot create a child to it. To deal with this problem, you can create mirror for parent-type elements such as package (Right click on a referenced element and select **Create Mirror Model Element**). The mirrored model element is also read-only on its properties, however, you can add a child model to it.

Impact Analysis

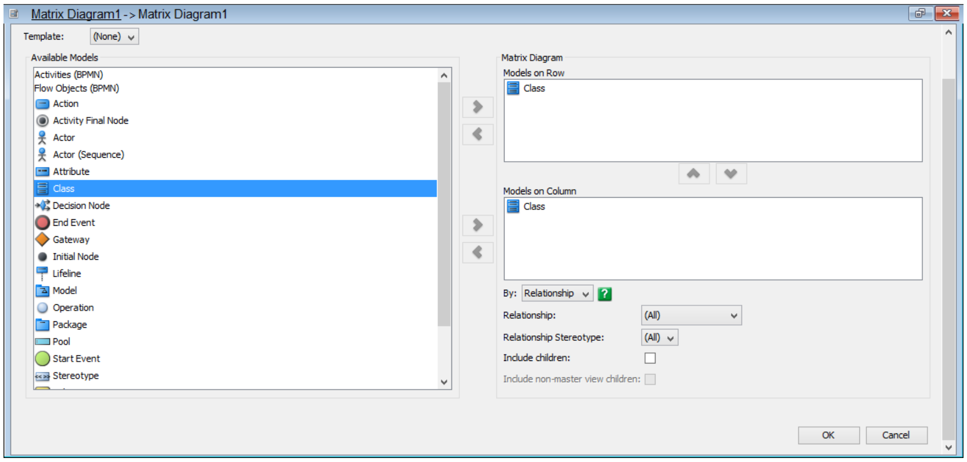
Professional Edition +

If you were to make a change to some model elements, it would be important to know which other elements will get affected because of it. Impact Analysis can help you with that.

Matrix Diagram

Matrix Diagram helps you identify the relationship between model elements of specific type(s), so as to study the consequence of making certain changes. To create a matrix:

1. Create a matrix by selecting **Diagrams > Matrix Diagram** from the toolbar.
2. Let's say if you want to inspect the relationship among all the classes in the project. Add class to both **Models on Row** and **Models on Column** by first selecting class on the left hand side and clicking **>** to add it into row and column list respectively.
3. Select **Relationship** for **By**. By here means to compare row and column items by the selected criterion.



Matrix Diagram1 -> Matrix Diagram1

Template: (None)

Available Models: Activities (BPMN), Flow Objects (BPMN), Action, Activity Final Node, Actor, Actor (Sequence), Attribute, Class, Decision Node, End Event, Gateway, Initial Node, Lifeline, Model, Operation, Package, Pool, Start Event, Stereotype

Matrix Diagram

Models on Row: Class

Models on Column: Class

By: Relationship

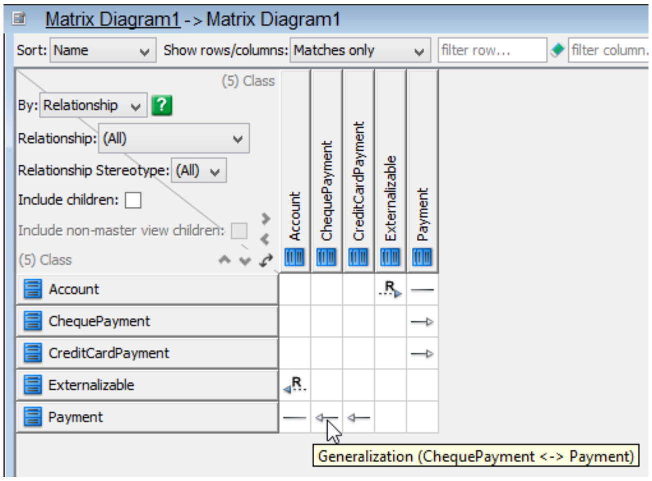
Relationship: (All)

Relationship Stereotype: (All)

Include children: □

Include non-master view children: □

4. Click **OK**. This produces a matrix which lists the classes in rows and columns, showing their relationships in cells.



Matrix Diagram1 -> Matrix Diagram1

Sort: Name   Show rows/columns: Matches only   filter row...   filter column...

By: Relationship

Relationship: (All)

Relationship Stereotype: (All)

Include children: □

Include non-master view children: □

(5) Class	Account	ChequePayment	CreditCardPayment	Externalizable	Payment
Account				R	—
ChequePayment					—▷
CreditCardPayment					—▷
Externalizable	R				
Payment	—	◁—	◁—		

Generalization (ChequePayment <-> Payment)

Let's say you are thinking about deleting the *Payment* class. By reading the matrix, you realize that the *Payment* class is a super class of *CreditCardPayment* and *ChequePayment*. Deleting the super class *Payment* may risk losing data. So you'd probably need to consider withdrawing the deletion or to move the data from super class to sub class, etc.