IDERA ER/Studio Software Architect

Evaluation Guide

Version 18.0
Published March 2019

March 31, 2019

Contents

Introduction to ER/Studio Software Architect

ER/Studio Software Architect, the design-driven environment for modeling applications. ER/Studio Software Architect includes features such as support for UML 2.0, OCL, patterns, Quality Assurance audits and metrics, XMI format import and export, and automated documentation generation.

ER/Studio Software Architect is an Eclipse-based RCP application, and the primary components are a Model Navigator, Diagram Editor, Palette, and Properties View.

Organizations can leverage ER/Studio Software Architect technologies to:

- Improve requirements elaboration definition processes through the use of visualization and supported modeling notations such as Unified Modeling Language (UML).
- Ensure applications and/or integrations are designed and developed clearly with maximum adaptability and reusability.
- Ensure application requirements, architectures, and designs can be well understood and communicated effectively to enterprise teams without ambiguity.

Product Benefits by Audience

Data Modelers: This product assists data modelers in designing logical and physical data models and exporting them into databases, as well as visualizing existing databases into ER models.

IT or Application Analysts: Assists with the definition and analysis of requirements through UML use case analysis - the process of visualizing requirements through use cases, activity diagrams, and interaction diagrams.

Architects and Toolsmiths: Assist with the definition, implementation, customization, and deployment of Domain Specific Modeling Languages, consisting of MetaModels, Graphical Editors, Transformations, Reports and additional functions to support the need to express models with tailored tooling.

Development Team: Assists with communication through common language and visual representation of the requirements, architecture, and code.

Architects and Developers: Assists with improving and monitoring application quality through audits and metrics for models.

About this Evaluation Guide

This evaluation guide is intended to help you get started using ER/Studio Software Architect (SA) and gives you the foundation you need to explore its many features and benefits. You will learn how to create a new project and class diagram, create a pattern, use the UML Profile, and generate a report. You will also become familiar with some frequently used tasks and commands to make you more productive.

This guide is divided into short, easy to learn sessions. Do them all at once, or complete them individually as you have time.

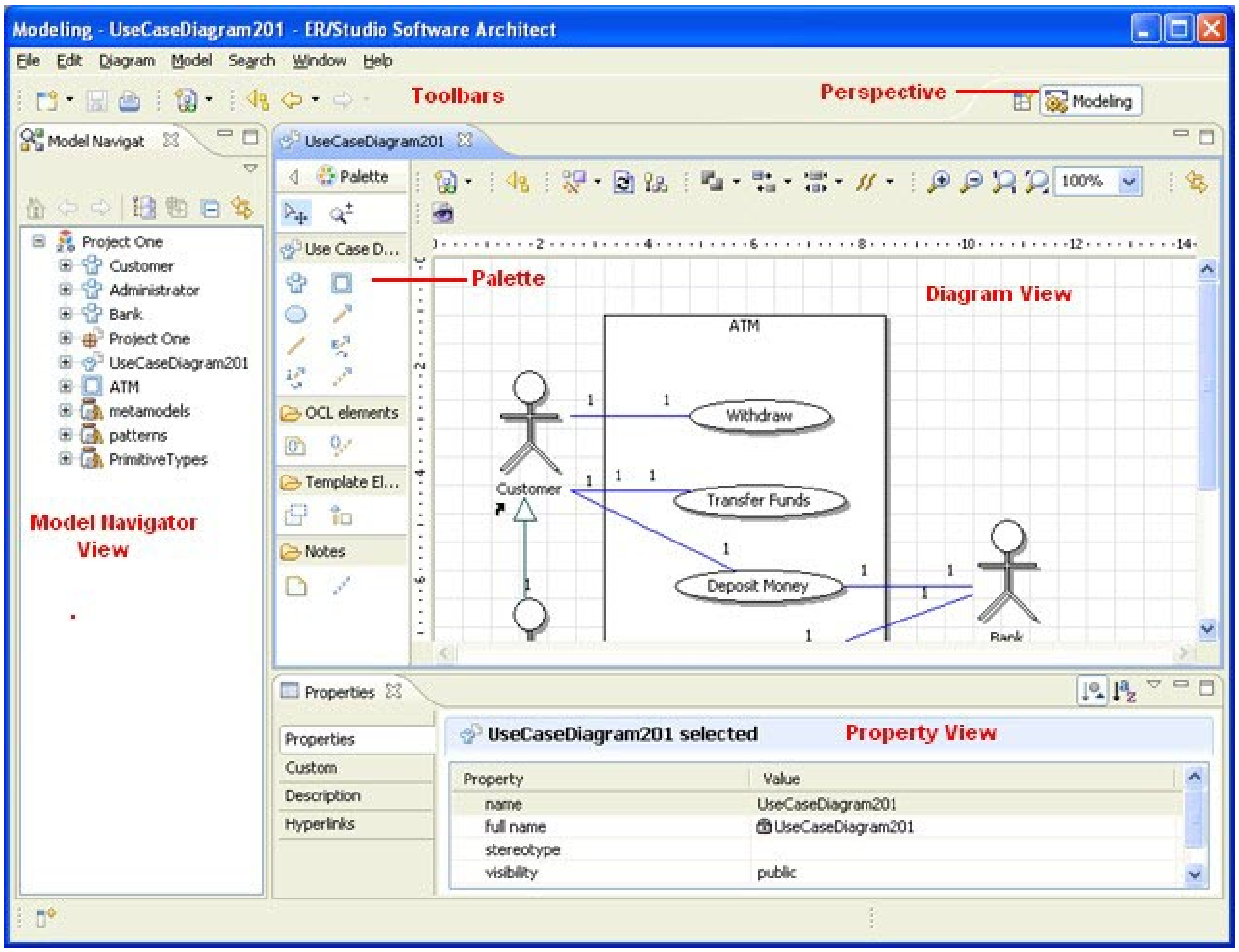
- Session 1: Getting Started with ER/Studio SA
- Session 2: Understanding the ER/Studio SA Infrastructure
- Session 3: Creating a Use Case Diagram
- Session 4: Create a Pattern from Existing Elements
- Session 5: Define the UML Profile
- Session 6: Deploy the UML Profile
- Session 7: Apply the UML Profile
- Session 8: Generating HTML Documentation
- Session 9: Using ER/Studio SA Tools

You can use this basic guide as a roadmap of product highlights, but it also is provided to help you find your own path in exploring ER/Studio Software Architect.

Once you've started, you can select **Help** from the menu to find many additional resources that complement and build on many of the activities shown in this brief guide.

Session 1: Getting Started with ER/Studio SA

The following graphic illustrates the main elements of the ER/Studio SA Application User Interface. The first graphic shows a project, and a use case diagram. The available palette is for the Use Case diagram.



UseCaseDiagram201 selected

Property	Value
name	UseCaseDiagram201
full name	UseCaseDiagram201
stereotype	
visibility	public

Note: The available palette depends on the type of Model, i.e., if it is a Conceptual model, only the Conceptual Objects palette is available.

Session 2: Understanding the ER/Studio SA Infrastructure

Before we begin, let's cover some of the basic constructs that help organize and display the information in ER/Studio SA:

- **Workspace**. The workspace is the top level container or owner of your projects. You can only have one workspace open at a time.
- **Project**. Work in ER/Studio SA is done in the context of a **project**. A project is a logical structure that holds all resources required for your work. All projects located in the selected Workspace are listed in the Model Navigator. You can set up project properties when the project is being created, and modify them further, using the **Properties** dialog.

  The following projects can be created in ER/Studio SA:

  - **Pattern Definition project** is a profiled UML 2.0 modeling project that allows creation of new patterns.
  - **Profile Definition project** is a profiled modeling project that allows you to create new profiles.
  - **UML 2.0 project** is a design project with no source code support.
- **Packages**. The notion of a package has two facets: logical and physical.

  Logically, a model consists of one or more packages. A package is a model element used to group elements, and provides a namespace for the grouped elements. A package can contain packageable elements (the elements that can be directly owned by a package) and the other packages. A model itself is a package.

  Physically, a package is a folder containing the files that store diagrams and model elements. Contents of a package can be displayed on a special type of the Class Diagram that is synchronized with the package contents, i.e. all the classifiers directly owned by this package automatically appear on the package diagram. Each package contains the single package diagram that is created automatically and cannot be added explicitly.

  The root package of a project (Model) is usually referenced as the default package. The package diagram of this package is called the default diagram. This diagram is created and opened just after the modeling project creation.
- **Diagram**. Each modeling project contains a set of diagrams that are graphical representations of parts of the model. Diagrams contain graphical elements (nodes connected by paths) that represent model elements.

  Each diagram belongs to a certain diagram type (for example, UML 2.0 Class Diagram). The diagram type defines the typical contents of the diagram (the kind of elements that are usually placed on this diagram) and the notation used to represent the model elements. For example, a Class in a UML 2.0 project can be added to the Class Diagram and to the Composite Structure Diagram and will have different representations there. Each diagram has a specific Palette and context menu that allows creation of the model elements specific to this diagram type.

Diagrams exist within the context of a project. You will have to create or open a project before creating a new diagram. The set of available diagram types depends on the type of project. In UML 2.0 project you have a set of standard UML diagrams defined in the UML2.0 specification. Along with the design diagrams that are explicitly created by the user, ER/Studio SA models have so-called "Package" diagrams. These diagrams have the Class Diagram type, but they will be generated automatically for each package and show its contents.

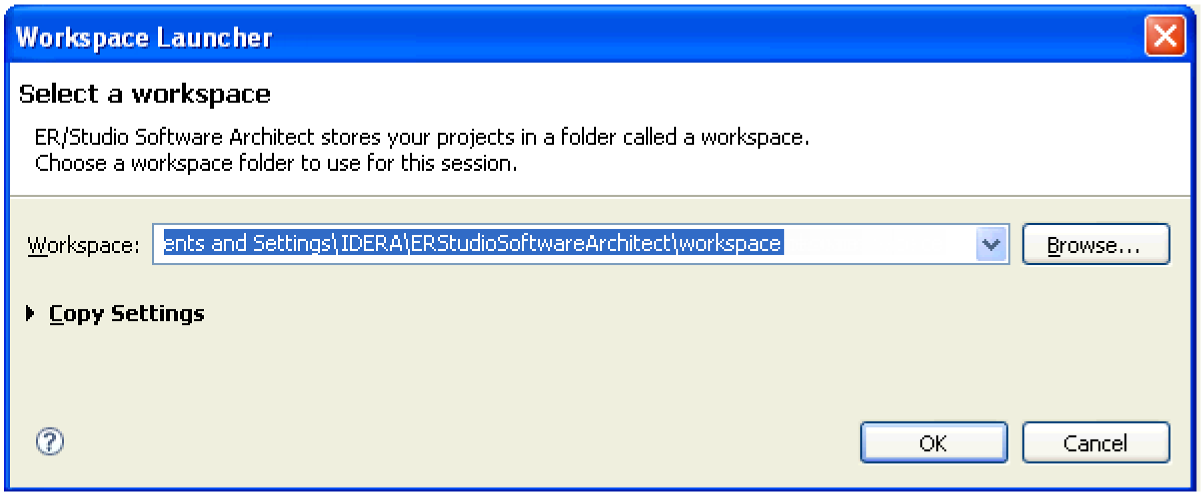
- **Model Elements**. Each model in a modeling project is a set of entities that are instances of metaclasses of the metamodel chosen for the project. These instances are the Model Elements.

  Each model element has a set of properties and notations defined for its metaclass. For example, when you create a UML 2.0 project, every element created in this project instantiates a metaclass from the UML 2.0 metamodel. That is, each actor on a use case diagram in a UML 2.0 project is an instance of use cases/Actor, and each component is an instance of components/Component.

  The model elements have graphical notation and can be explicitly placed on diagrams, are nodes and links.

Beginning to Model with ER/Studio SA

1. From the **Start** > **Programs** menu, choose **ERStudio Software Architect** > **ERStudio Software Architect** and the application opens.
2. When you first open the application, the Workspace Launcher dialog appears. The workspace (named “workspace) is placed in the default directory. You can browse to a new location if you want to change this.


Workspace Launcher

Select a workspace

ER/Studio Software Architect stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: ents and Settings\IDERA\ERStudioSoftwareArchitect\workspace

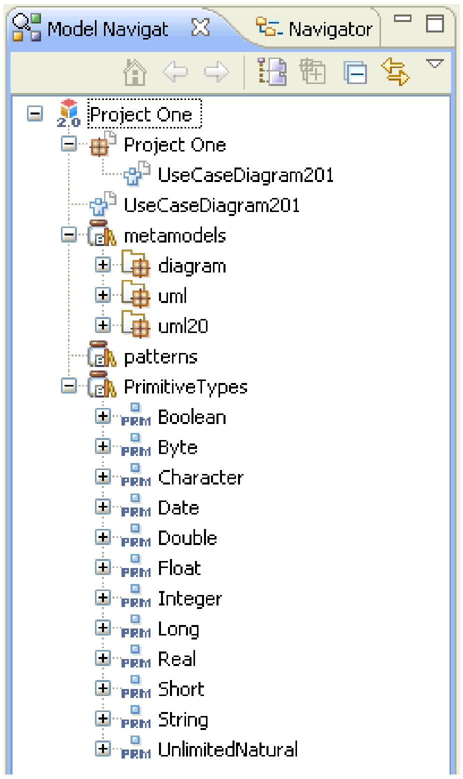
3. Click **OK** and a new workspace opens.
4. To create a new project, click **File** > **New** > **UML 2.0 Project** and the New Project dialog appears.

Note: You can also click Alt + Shift + N.

5. Enter the project name “Project One” under Project name. Leave “use default location” selected and leave “Add project to working sets” unselected.

6. Click **Next** and the Modeling Settings dialog appears. Deselect "use default" and select **Use Case** from the drop-down list. The default diagram name "UseCaseDiagram201" appears. Leave this diagram name as is.
7. When you click **Finish**, a new project and diagram are created and appear in the Model Navigator.

The items are displayed in the hierarchal mode in the tree, as shown in the following image.



```
Project One
  Project One
    UseCaseDiagram201
  UseCaseDiagram201
  metamodels
    diagram
    uml
    uml20
  patterns
  PrimitiveTypes
    Boolean
    Byte
    Character
    Date
    Double
    Float
    Integer
    Long
    Real
    Short
    String
    UnlimitedNatural
```

Session 3: Creating a Use Case Diagram

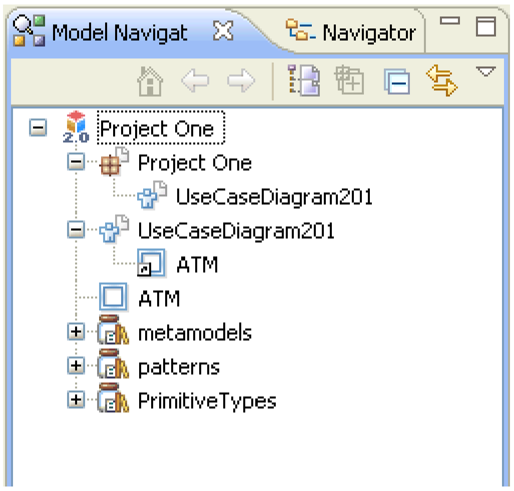
We will create elements for this use case diagram using the palette in the Diagram View.

Placing the Diagram Elements

You can build a use case diagram by placing elements in the Diagram View directly from the palette. The diagram we are going to create shows an example of actors and use cases for an ATM system.

1. In the Use Case palette, click the Subject icon (□).

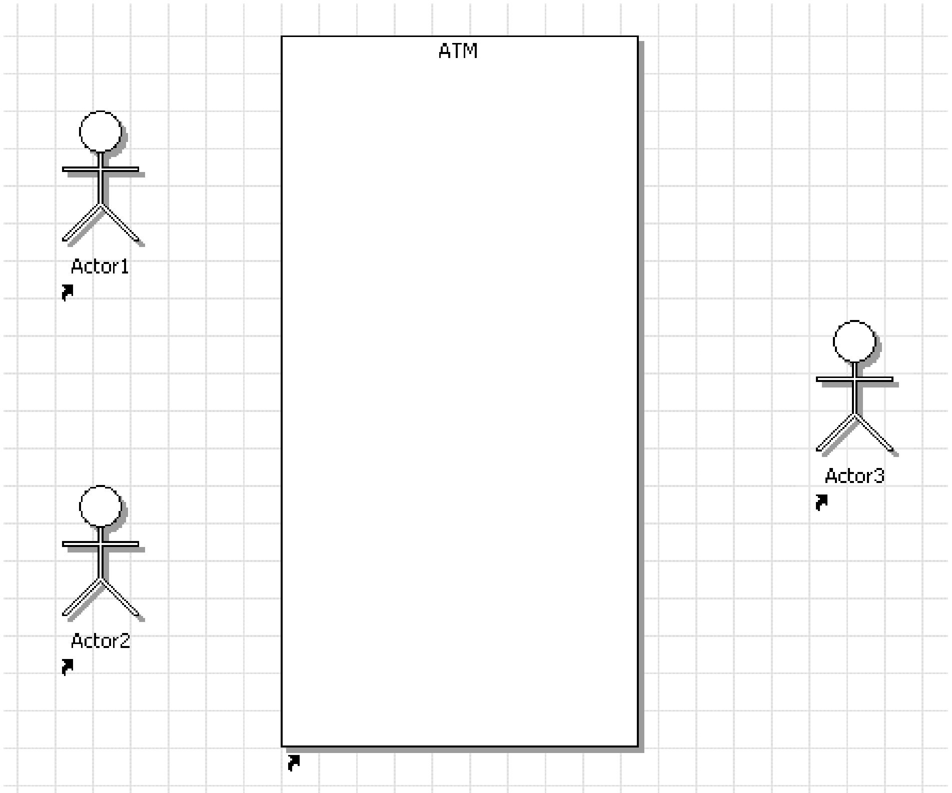
    2. Then click in the center of the Diagram View and the subject is placed in the diagram with the label "Subject1" highlighted. Rename it "ATM" and the label is centered at the top of the subject. The Model Navigator now has detail in its hierarchy similar to the following image.



```
Project One
    Project One
        UseCaseDiagram201
    UseCaseDiagram201
        ATM
    ATM
    metamodels
    patterns
    PrimitiveTypes
```

3. Click the Actor element ( ) on the Use Case palette and place three actors in the Diagram View as shown in the following graphic.

Note: You can also place multiple elements in a diagram. Holding down the CTRL key, click the Palette icon for the element you want to create (the icon stays highlighted). Continue to hold down the CTRL key and click in the diagram to place the desired number of elements. Press **ESC** to exit the multi-drop mode.



4. Next, label the actors by right-clicking on an actor element, selecting **Rename** from the context menu, entering your label, and then pressing **Enter**. Each actor is named as follows:

Actor 1 = Customer

Actor 2 = Administrator

Actor 3 = Bank

The Use Cases are placed within the subject to show tasks or activities.

5. Click Use Case ( ) and place five elements inside the ATM subject.
6. Select an element and then click once again to label each element as follows:

UseCase1 = Withdraw

UseCase2 = Transfer Funds

UseCase3 = Deposit Money

UseCase4 = Register ATM at Bank

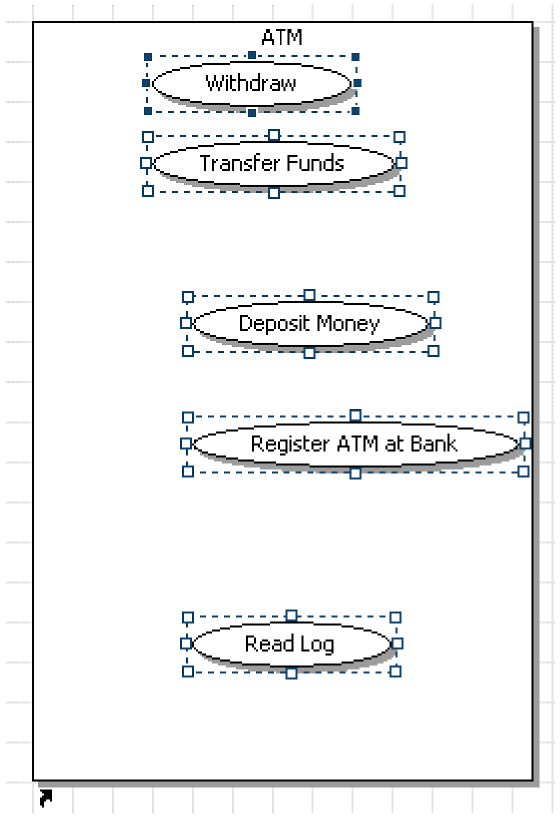
UseCase5 = Read Log

Aligning in the Diagram View

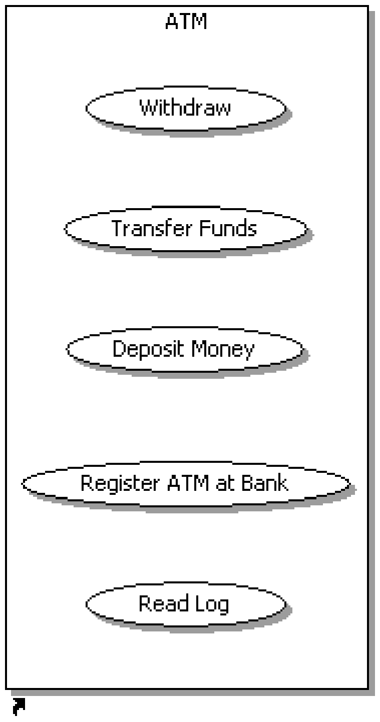
There is a quick and simple way to align your placed objects. In this example we want to make sure our top set of use cases are aligned and distributed evenly.

1. Select the five use cases you placed in the subject by holding down the left mouse button and drawing a box around the tasks. Notice that all five elements are now marked with handles.

    2. Place the mouse cursor on the task labeled "Withdraw" and **Shift+Click** because you want to align all the other elements with it. That element is now highlighted with black box handles instead of hollow boxes, as shown in the following image.



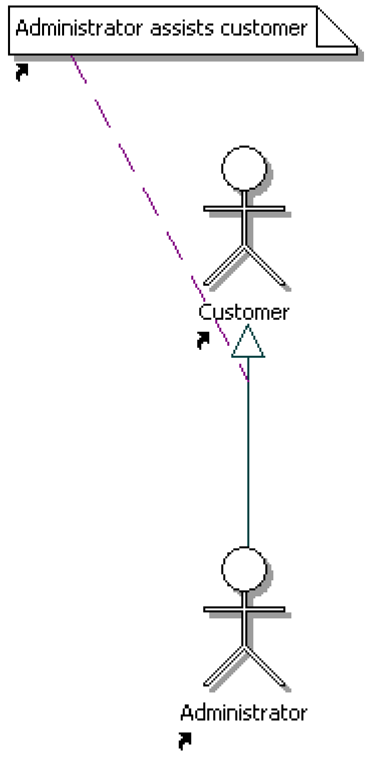
3. On the toolbar, click the drop-down arrow next to the align icon and select the Align Center icon ( ). The five elements are now aligned, as shown in the following image.



Create Generalization Link

One relationship between the Actors is the Generalization Link. This is useful in defining overlapping roles between actors. We are going to create a link from the Administrator actor to the Customer actor.

1. Click the Generalization Link ( ) in the Use Case palette.
2. Click the Administrator actor and drag the cursor to the Customer actor. A generalization link is created.
3. You can create a note for this link to display additional information. Right-click the link and select **New > Insert > Linked Note**. A note connected to the link is created.
4. Select the note once and then click once again. Enter the follow text: "Administrator assists customer" and then click outside the note box.



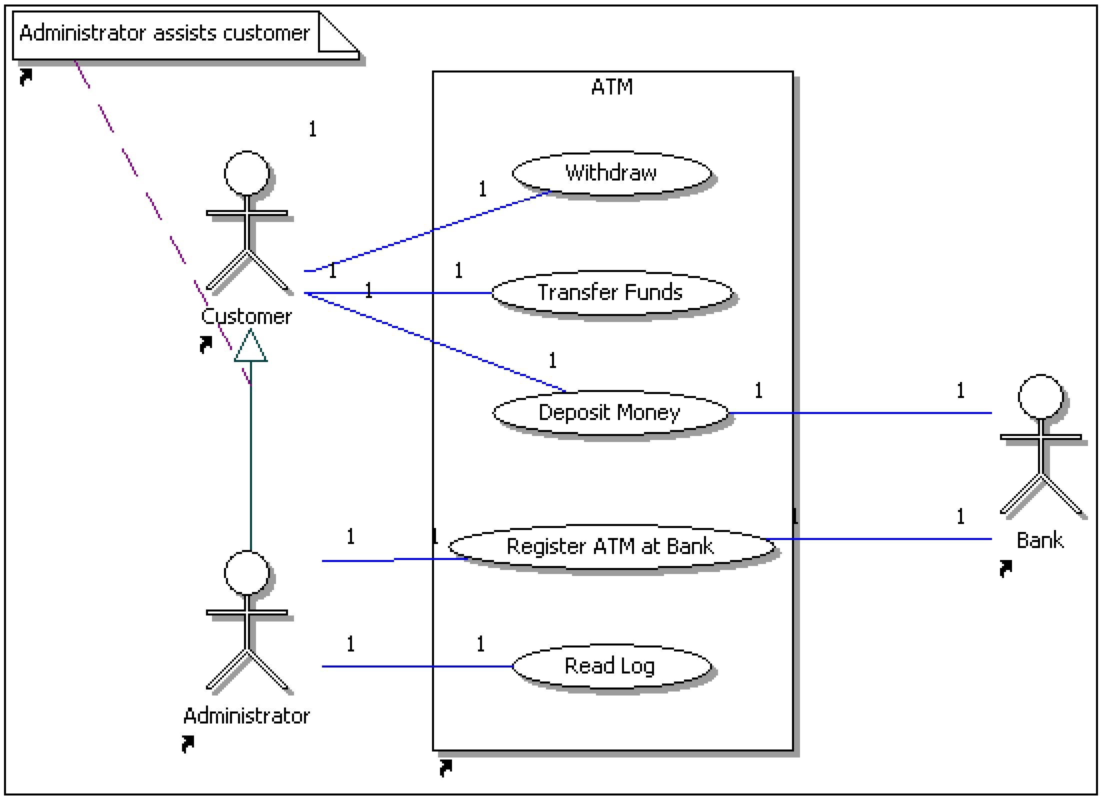
Creating Association Links

You will now connect the actors to the use cases using an Association Link. A direct association links an element and a requirement and synchronizes both artifacts so you can manage them as a single, conceptual item.

1. While holding down the Ctrl key, click the Association Link ( ) in the Use Case Palette.

2. While still holding down the Ctrl key, create association links between the following:

- Customer to Withdraw
- Customer to Transfer Funds
- Customer to Deposit Money
- Administrator to Register ATM at Bank
- Administrator to Read Log
- Bank to Deposit Money
- Bank to Register ATM at Bank



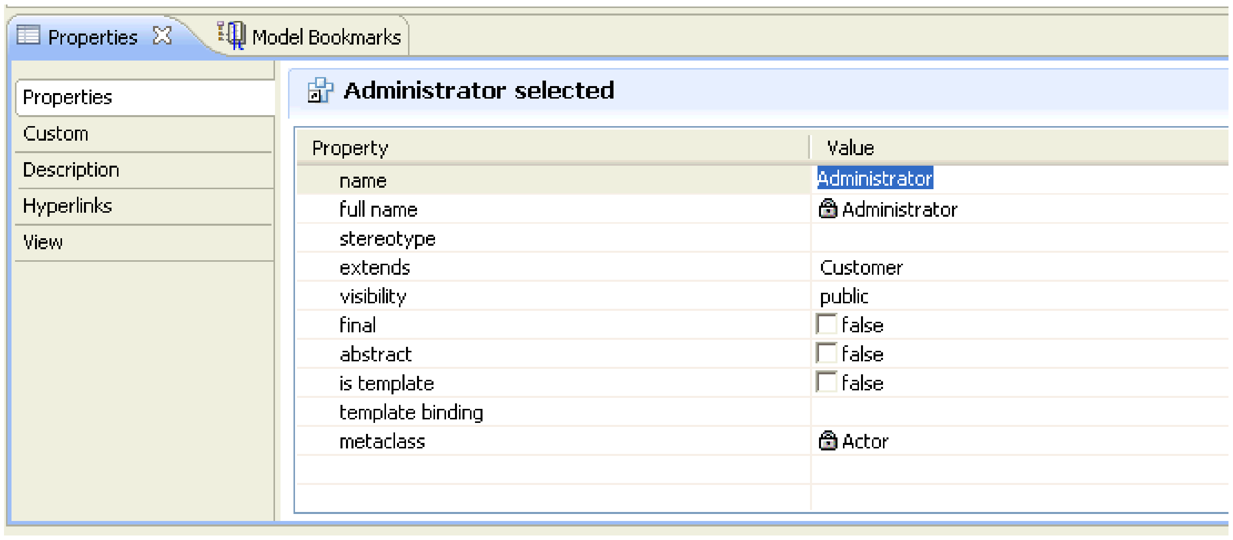
Changing the Properties of an Element

When you double-click on an element, the Property View for that element opens. You can use the Property View to:

- Change the name or properties of an element.
- Enter a description and notes for the element, if applicable.
- View any links or usages that have been created to the element.
- Change the appearance of the object.

For purposes of this diagram, we are going to rename the Administrator actor and add a description.

1. Right-click the Administrator actor in the Diagram View and select **Properties** to open the Property View.

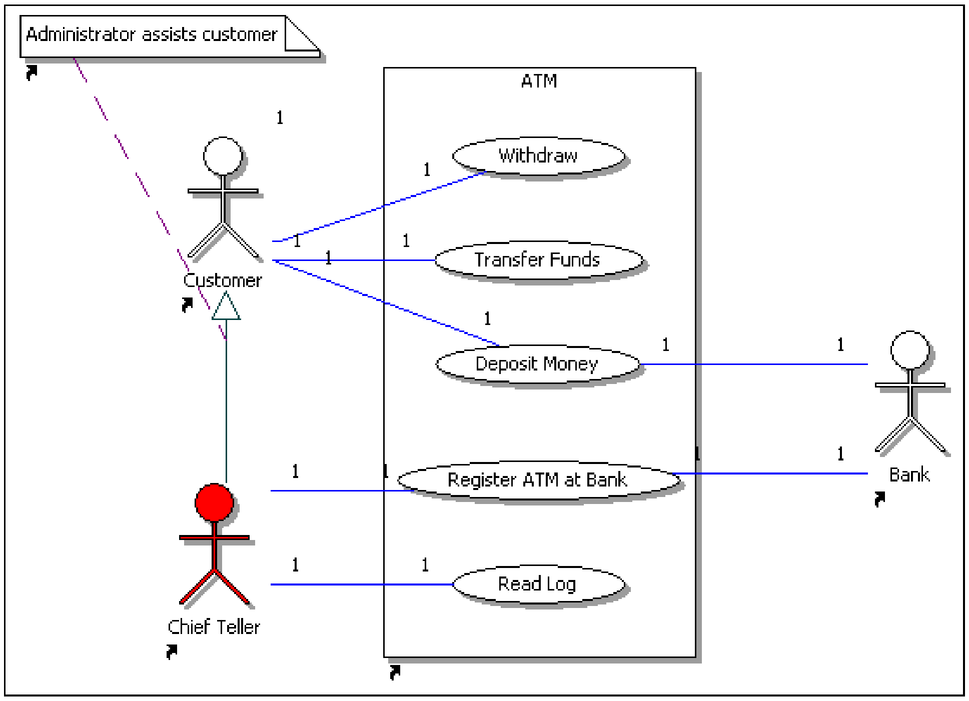


Administrator selected

Property	Value
name	Administrator
full name	🔒 Administrator
stereotype	
extends	Customer
visibility	public
final	☐ false
abstract	☐ false
is template	☐ false
template binding	
metaclass	🔒 Actor

2. On the Properties tab, change the name from “Administrator” to “Chief Teller”.
3. Click the View tab in the Properties tab, and then click background color.
4. Click Browse (▭) and the color dialog appears.
5. Click red, and then click **OK**. The Chief Teller actor is now red.

This completes the process of creating the use case diagram. Your diagram should resemble the following image.



Session 4: Create a Pattern from Existing Elements

Patterns provide you with powerful reuse facilities. Instead of starting from scratch for each design problem, you can use predefined patterns supplied with ER/Studio Software Architect.

Patterns are pluggable extensions for ER/Studio Software Architect enabling you to:

- Create new and frequently used elements
- Modify existing elements

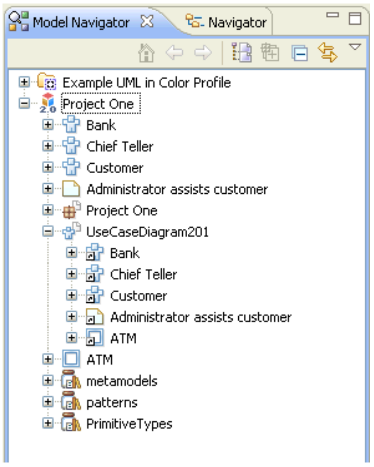
Each pattern describes a set of model elements, relations between them, and constraints applied to those elements. Patterns are represented by special modeling projects covering all the aspects of patterns. Patterns are independent of any programming or markup language. You can use them to create or modify any type of element. Concrete patterns are designed to work with elements of a specific type. The Pattern Registry is used to manage patterns.

Pattern instances appear as a result of recognition of the existing model or creating new instances (along with model elements playing pattern roles) in the model. Pattern instances contain information about the pattern name and the role of each participant. They are shown in the Pattern Explorer view and under the Patterns node in the Model Navigator.

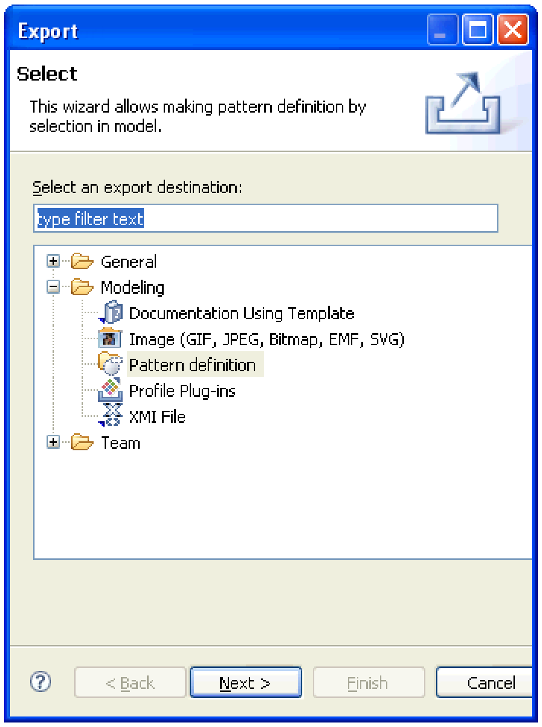
When applied to a diagram, such patterns create their entities and are presented on the diagram itself, with the links to the created entities. Such patterns enable further modification by means of adding new participants (new pattern parts). All patterns that appear in the Pattern Explorer are represented in the project model in the form of entities with metaclass "pattern". Visually, pattern instances will be displayed as ovals (like collaboration occurrences). Pattern entities will have children links to pattern participants and this allows map links on diagrams from pattern instances to pattern participants. Actions on pattern instances in the model will be the same as in the pattern explorer.

This portion of the Evaluation Guide walks you through the steps to create a pattern from existing elements using ER/Studio SA. We are going to create a pattern from the project we created in the last step (Project One).

1. Expand the "Project One" project in the Model Navigator



2. If the diagram is not open in the Diagram View, double-click "UseCaseDiagram201" in the Model Navigator tree.
3. Next, select the Subject element labeled "ATM" which will be transformed to a pattern definition.
4. Right-click in this element and select **Export**.
5. In the Export dialog select **Modeling > Pattern Definition**, and then click **Next**.


Export

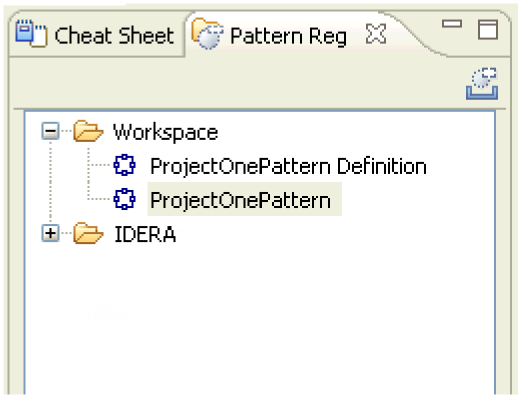
Select

This wizard allows making pattern definition by selection in model.

Select an export destination:

type filter text

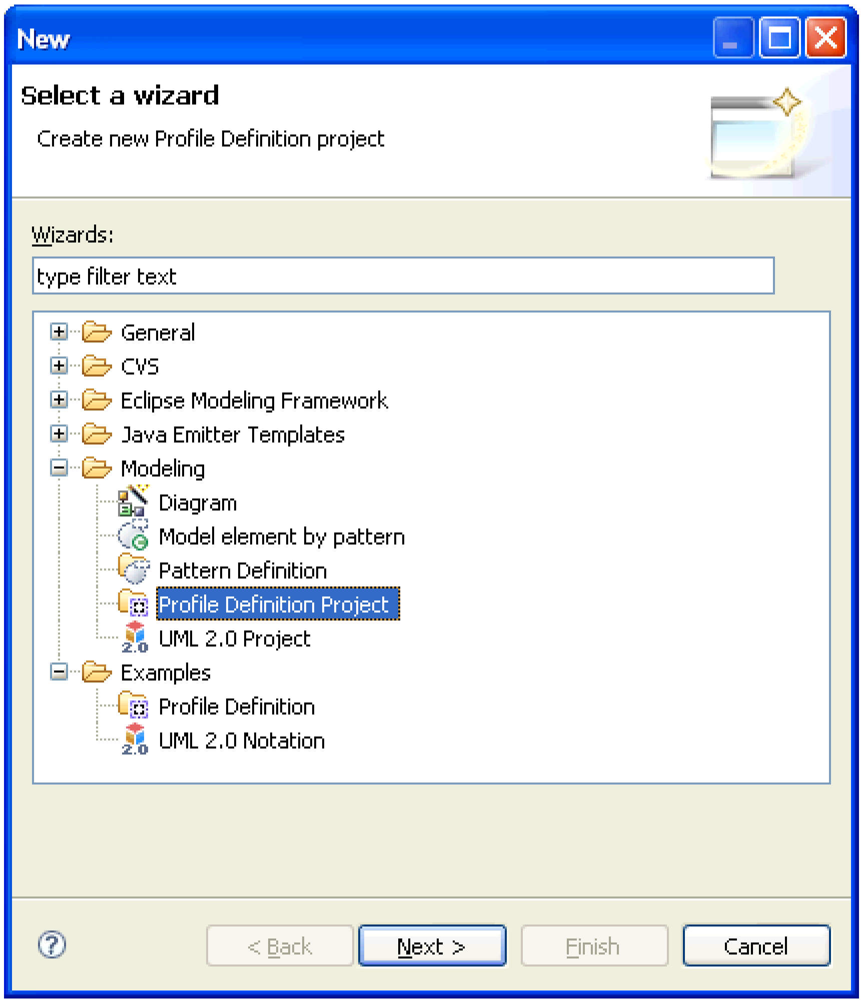
6. In the Create Pattern from Elements dialog that opens, enter the pattern name “ProjectOnePattern” and then click **Finish**. The new "ProjectOnePattern” appears in the "Workspace" folder of the Pattern Registry. This Pattern Registry can appear either on the right side of the Diagram View or below the Diagram view. Now you can run recognition using this pattern or generate new pattern instances.



Session 5: Define the UML Profile

The next three sessions will walk you through the steps of defining, deploying and applying a UML profile. We are going to define a sample UML profile for Web Services modeling. Web Service is modeled as a UML Component having the additional property "namespace". This property is mapped to the "targetNamespace" attribute during WSDL file generation. Web Service ports are modeled as UML Ports owned by Web Service Component. Each Web service port has two additional properties "style" and "transport". Style can have only two values: "rpc" and "document". Transport URI is modeled as a String property. Profile extensions will be applied to Component UML diagrams.

1. Create a Profile Definition Project by selecting **File > New > Other**. Expand the Modeling node, choose Profile Definition Project, and then click **Next**.



New

Select a wizard

Create new Profile Definition project

Wizards:

type filter text

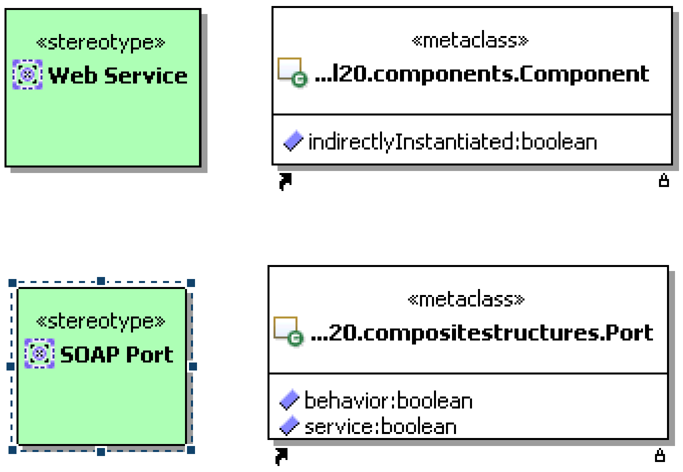
```
+ General
+ CVS
+ Eclipse Modeling Framework
+ Java Emitter Templates
- Modeling
    Diagram
    Model element by pattern
    Pattern Definition
    Profile Definition Project
    UML 2.0 Project
- Examples
    Profile Definition
    UML 2.0 Notation
```

2. In the New Project dialog, enter the project name "WSDL profile," and then click **Finish**. The WSDL profile project containing an empty WSDL profile diagram is created. Also note that the Palette now has a Profile Definition section as well.

3. Click the “stereotype” icon (). This icon is located in the Profile Definition portion of the palette.
4. Click in the diagram’s background and a stereotype rectangle is created. Name the stereotype “Web Service,” and then press **Enter**.
5. Create another stereotype icon and name it “SOAP Port.”

Next we are going to add the metamodel elements “Component” and “Port” to the diagram.

1. Right-click in the diagram’s background, and then select **New > Shortcut** on the context menu.
2. In the Shortcuts dialog, expand **metamodel > UML20 > components**, select “Component,” and then click **Add**.
3. In the same dialog, expand **metamodels > UML20 > compositestructures**, select “Port,” and then click **Add**.
4. Click **OK** and the “component” and “Port” classes are added to the diagram.



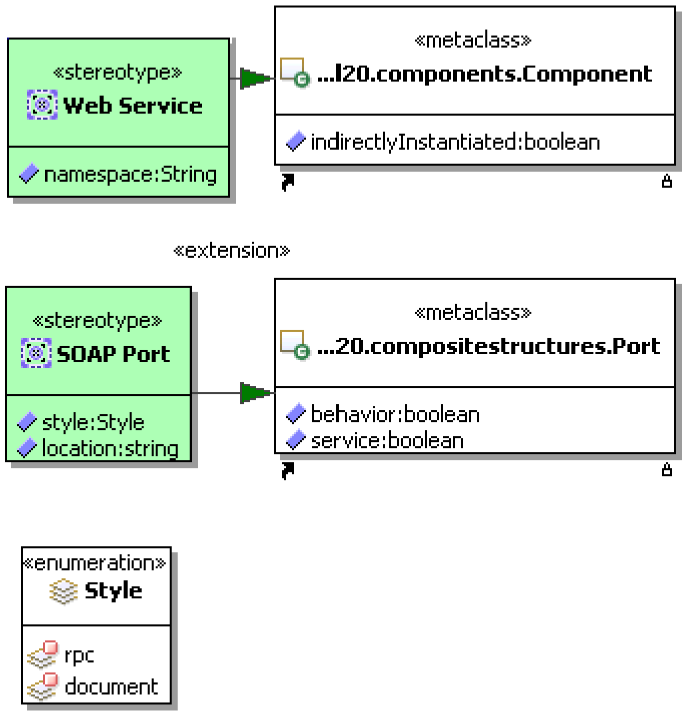
We will continue the process by creating extension links, a namespace, and enumerations.

1. To create extension links, click the Extension icon ( ) in the Profile Definition portion of the palette and draw a link from “Web Service” to “component”. Also draw a link from “SOAP Port” to “Port”. Notice that the links are labeled “<extension>”.
2. To create the Web Service “namespace” property, right-click the “Web Service” stereotype and select **New > Attribute** on the context menu.
3. Enter “namespace:String” and press **Enter**. The “namespace” property of type “String” is created. Notice that if you enter incorrect syntax in the label the text turns red and remains so until the error is corrected.
4. Once again, right-click “Web Service” and select **Properties** from the context menu.
5. Click **Profile Definition** in the Properties tab and click “viewmap”.

6. Click browse ( ) to open the Specify Node Viewmap. Uncheck the "use default viewmap" option.
7. Select "color" and click "choose color". Select Yellow from the color dialog and click **OK**. Click **OK** once again to close the Specify Node Viewmap dialog.
8. Click "Enumeration" ( ) in the Class Diagram Elements palette and click the diagram's background.
9. Label the enumeration "Style" and press **Enter**.
10. Right-click the Enumeration element and select **New > Insert > Enumeration Literal**. Label it "rpc".
11. Create another Enumeration Literal and label it "document".

Next we will specify SOAP Port properties, create a contribution element, and contribution links.

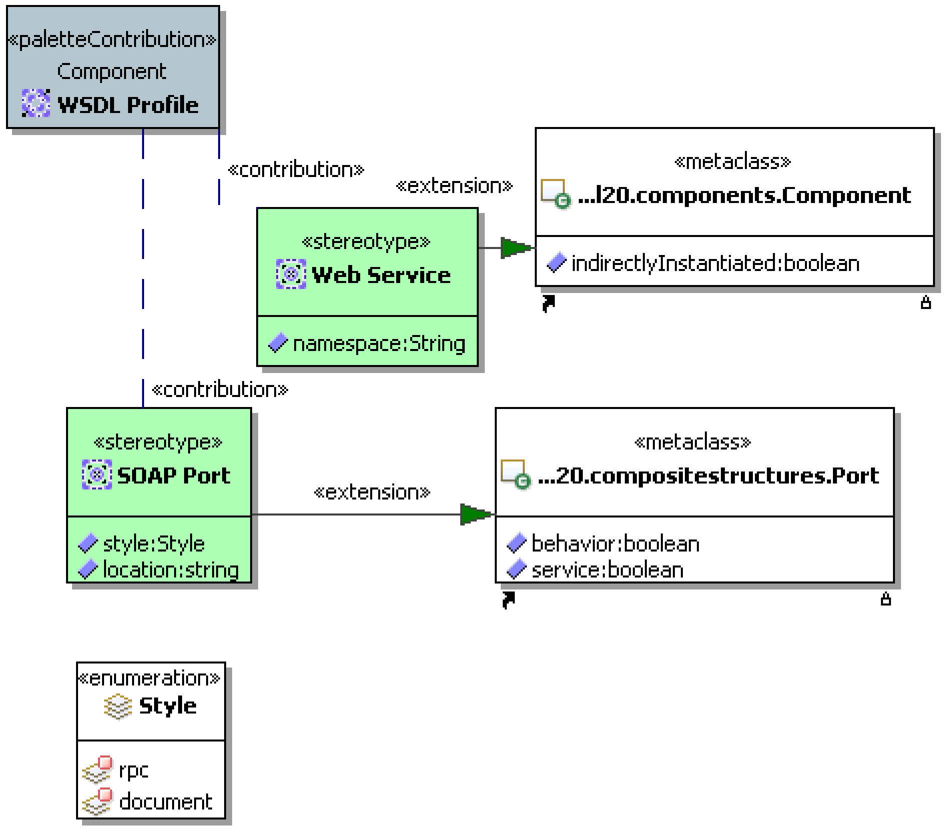
1. Right-click the SOAP Port stereotype and select **New > Attribute**. Enter "style:Style" and press **Enter**.
2. Create another attribute and name it "location:String".



3. Click ìPalette Contributionî () in the Profile Definition portion of the Palette and click in the diagram view. Label it ìWSDL Profileî

4. Open the properties of the "WSDL Profile" and select "diagrams' in the Profile Definition view. Click Edit (). In the Select Diagrams dialog, select "Component" and click OK.

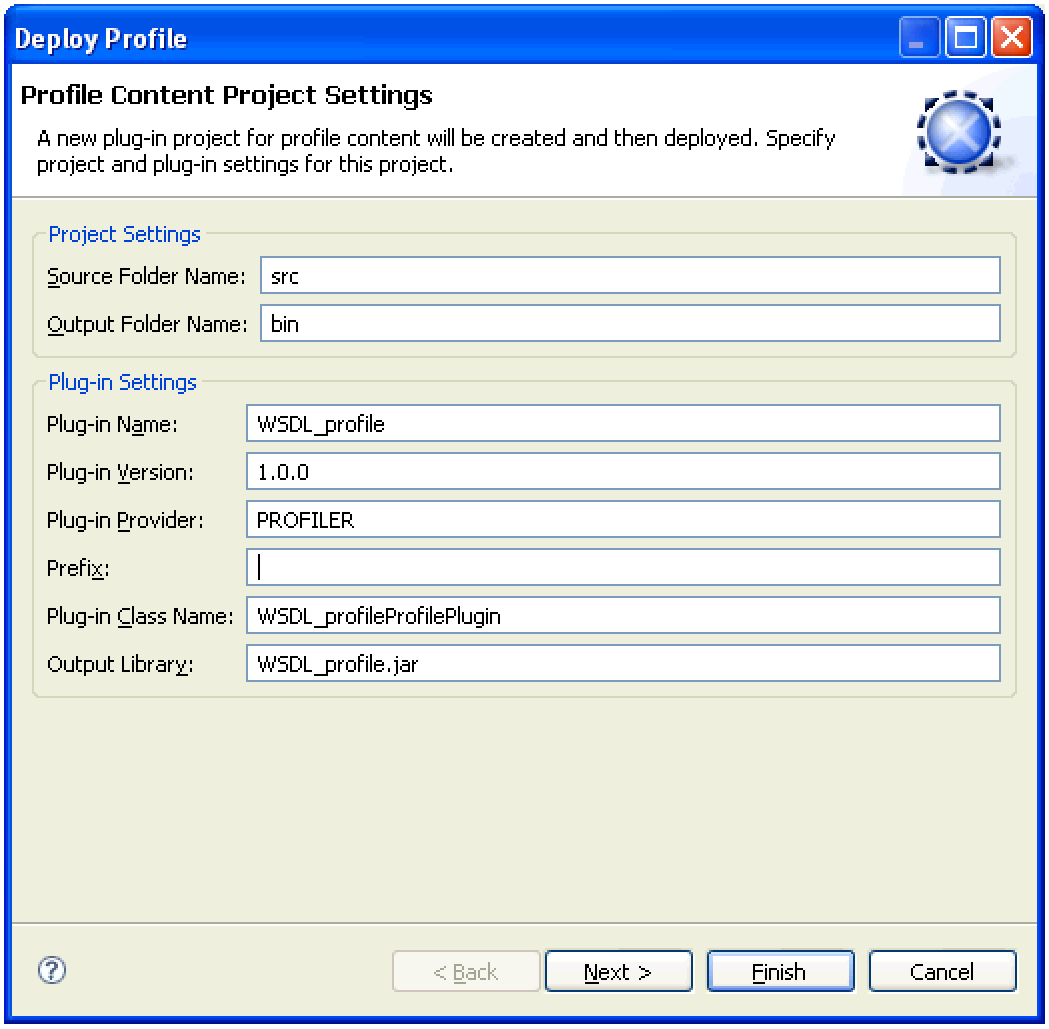
5. Click the contribution link () and draw a link from the "WSDL Profile" to both the "Web Service" and SOAP Port" stereotype elements.



Session 6: Deploy the UML Profile

Once the UML Profile has been created it then needs to be deployed.

1. Select "WSDL profile" in the Model Navigator
2. From the main menu select **Model** > **Profile** > **Deploy Profile** and the Deploy Profile dialog appears.



Profile Content Project Settings

A new plug-in project for profile content will be created and then deployed. Specify project and plug-in settings for this project.

Project Settings

Source Folder Name:	src
Output Folder Name:	bin

Plug-in Settings

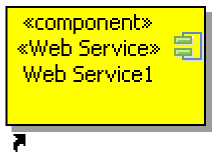
Plug-in Name:	WSDL_profile
Plug-in Version:	1.0.0
Plug-in Provider:	PROFILER
Prefix:	
Plug-in Class Name:	WSDL_profileProfilePlugin
Output Library:	WSDL_profile.jar

3. Leave all the default settings and click **Finish**. The deployment process consists of generating the corresponding Eclipse plugin that implements WSDL profile functionality. This may take a few moments and a progress dialog appears.
4. At the end of the process, a dialog appears asking you to restart. Click **Yes**.

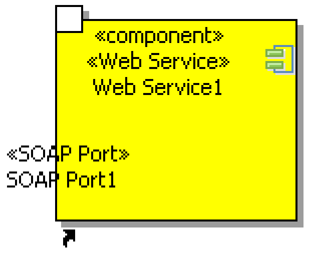
Session 7: Apply the UML Profile

The final session applies the sample UML Profile for Web Services modeling. Web Service is modeled as an UML Component having an additional property "namespace". This property will be mapped into "targetNamespace" attribute while WSDL file generation. SOAP Ports are modeled as UML Ports owned by Web Service Component. SOAP Port has two additional properties "style" and "location". Style can have only two values: "rpc" and "document". Location URI is modeled as String property. Profile extension will be applicable to Component UML diagrams.

1. The first step is to create a new UML project. Select **File** > **New** > **UML 2.0 Project** and in the New Project dialog, enter the project name "My Web Service".
2. Click **Next** and leave the default settings in the Modeling Settings dialog.
3. Click **Next** once again and in the Profile dialog select "WSDL profile" in the Available profiles list box and click **Finish**.

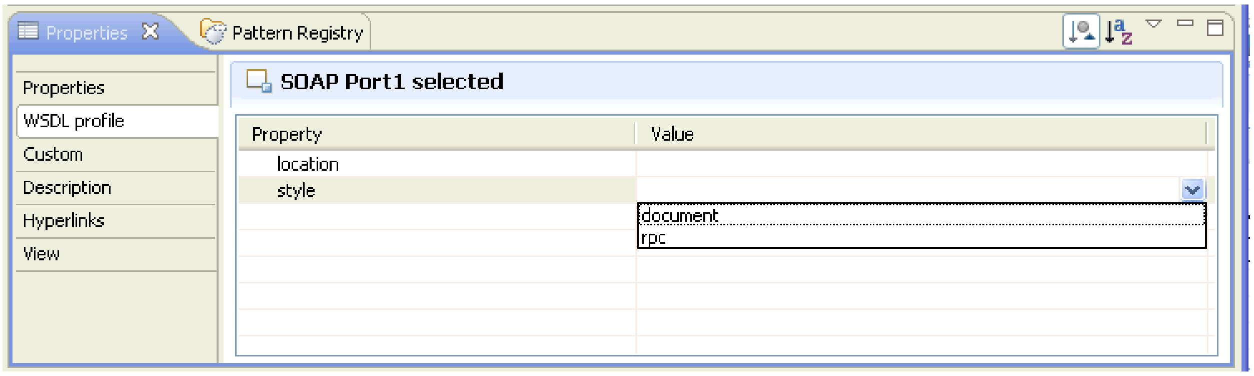


4. Right-click "My Web Service" project in the Model Navigator and select **New Diagram** > **Component**. The component diagram is created and the diagram palette has an additional category labeled "WSDL Profile".



5. In the WSDL Profile section of the palette, click "Web Service" ( ) and click the diagram view background. The web service component is created. Leave it labeled as is.
6. Right-click the component and select "Properties" from the context menu. Notice that in the Properties tab an additional "WSDL profile" category with a "namespace" property appears.
7. Right-click "Web Service1" component in the diagram and select **New** > **WSDL profile** > **SOAP Port**. "SOAP Port1" is created on the component boundary.
8. If you open the port properties (double-click on the white box located on the border of the component), notice that the SOAP port properties contain an additional WSL profile category with location and style properties.

9. Also, if you click the drop-down arrow in the style, "document" and "rpc" are listed in the Value column.



SOAP Port1 selected

Property	Value
location	
style	

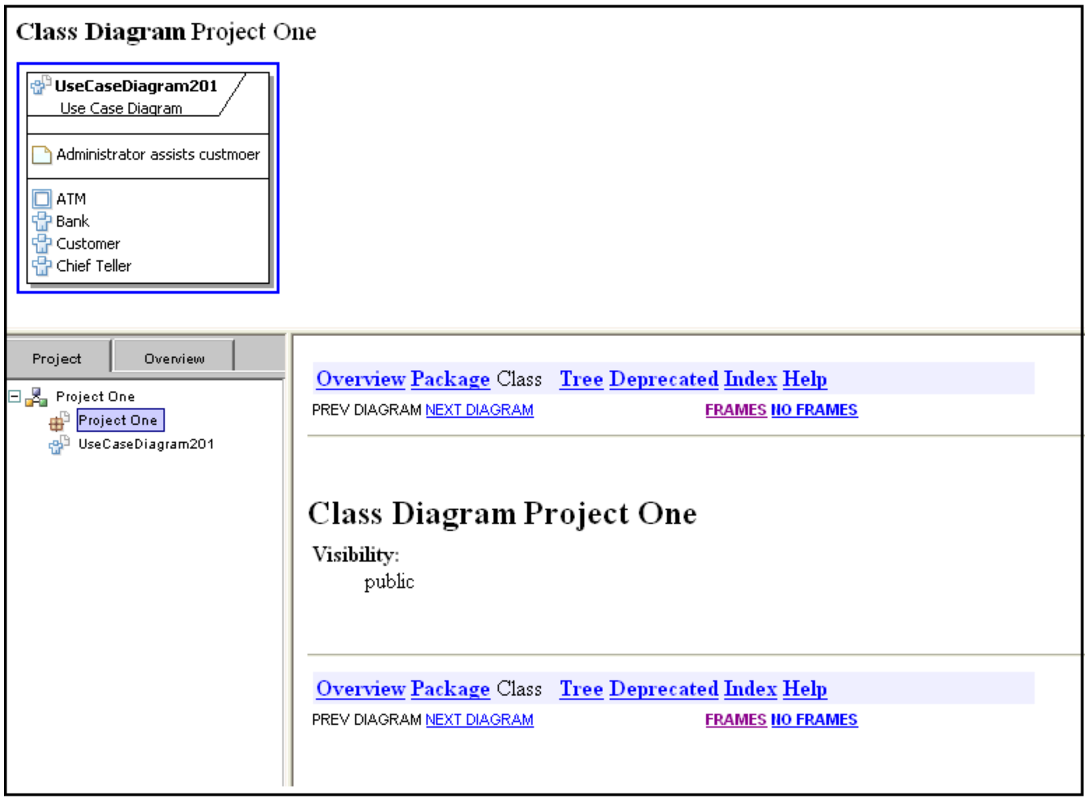
document
rpc

Session 8: Generating HTML Documentation

In this session we will generate project documentation in HTML format, using the default template, supplied with the product.

1. Open the “UseCaseDiagram201” that you created at the beginning of this Evaluation Guide.
2. Select Model > Documentation > Generate HTML on the main menu.
3. In the **Generate HTML Documentation** dialog that opens, leave the default setting in the output folder.
4. In the Scope option, click the drop-down arrow and select “Project One”.
5. In the Options section, leave the default settings.
6. Click **Finish** and a progress dialog appears. The generated document opens in a separate window.

   The first document that appears shows the class diagram for the selected project and section showing the Project hierarchy and the different elements of the package:



Class Diagram Project One

UseCaseDiagram201
Use Case Diagram

Administrator assists custmoer

ATM
Bank
Customer
Chief Teller

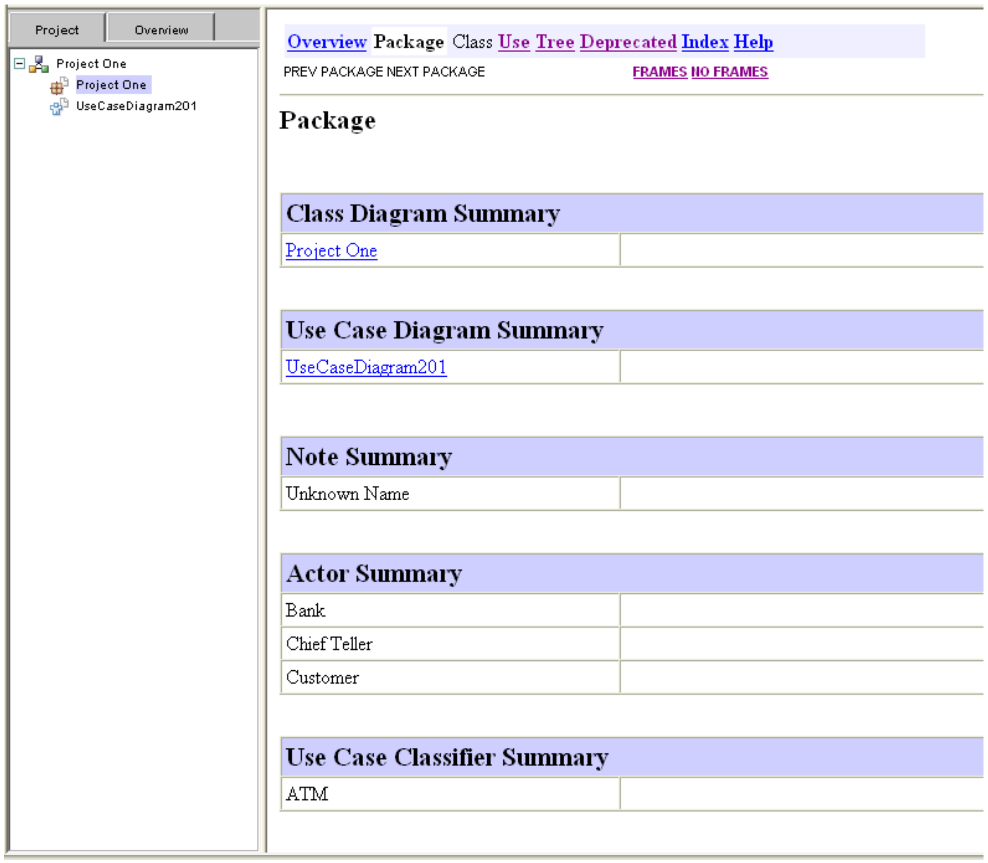
Project
Overview
Project One
Project One
UseCaseDiagram201
Overview Package Class Tree Deprecated Index Help
PREV DIAGRAM NEXT DIAGRAM
FRAMES NO FRAMES

Class Diagram Project One

Visibility:
public

Overview Package Class Tree Deprecated Index Help
PREV DIAGRAM NEXT DIAGRAM
FRAMES NO FRAMES

7. Click **Package** to view the summary of all elements in the package:



Project
Overview
Project One
Project One
UseCaseDiagram201
Overview Package Class Use Tree Deprecated Index Help
PREV PACKAGE NEXT PACKAGE
FRAMES NO FRAMES

Package

Class Diagram Summary	
Project One	

Use Case Diagram Summary	
UseCaseDiagram201	

Note Summary	
Unknown Name	

Actor Summary	
Bank	
Chief Teller	
Customer	

Use Case Classifier Summary	
ATM	

Setting Preferences for the Documentation Generation

1. On the main menu, choose **Window** > **Preferences** > **Generate Documentation**.
2. Expand this node and then expand the "Generate HTML" node and enter the options for HTML documentation:
   - o classes and members to be included in the documentation
   - o tags to be included in the documentation
   - o Javadoc Options
3. Under the HTML Output category, choose the option of processing line breaks.
4. Under the RTF Output category, set up text formatting options.

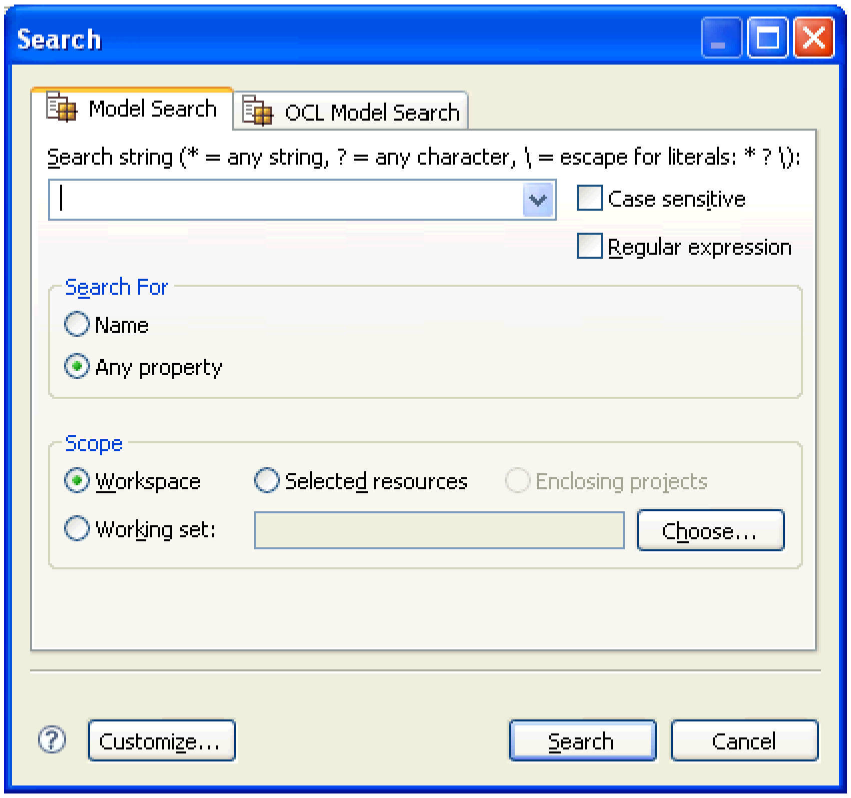
Session 9: Using ER/Studio SA Tools

As you work on your diagrams, ER/Studio SA offers many tools that make it easier to find elements, navigate, and customize your diagrams.

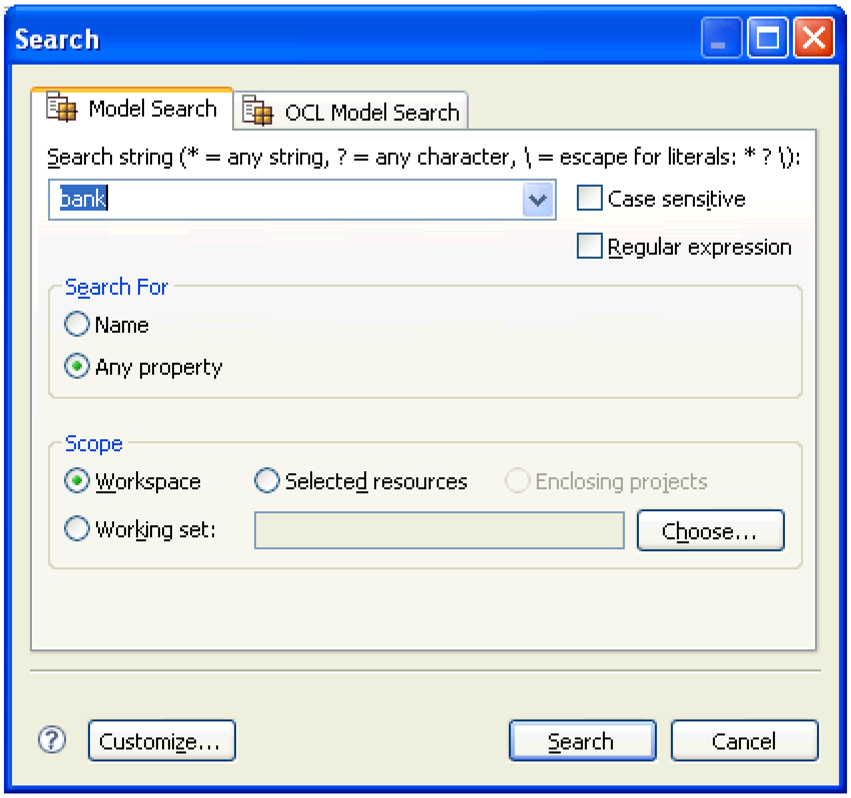
Search

The Search function searches the entire workspace and then provides a list of the objects that were found. It searches the data itself, not the element in the diagram.

To perform a search, select **Search** > **Model** on the main menu and the Search dialog appears:



You can search a name or any property in the model. The scope of your search can include the entire workspace, or selected resources or working sets. The search can be limited to specific types of objects and you can match case and whole words. In the example below we are going to search any element with the search string of "bank".

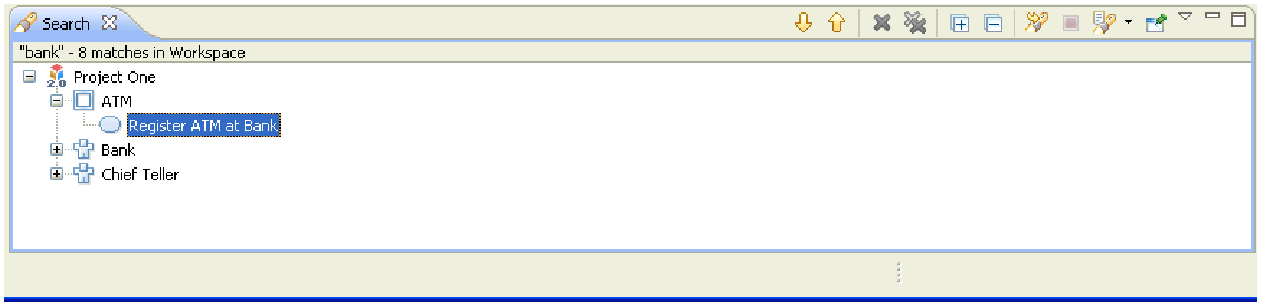


Search string (* = any string, ? = any character, \ = escape for literals: * ? \): bank
Case sensitive [ ]
Regular expression [ ]

Search For: Name [✓ Any property]

Scope: [✓ Workspace] Selected resources Enclosing projects Working set: Choose...

After you click Search, a Search tab view appears with all the elements that matched the search. From this view, there are several toolbar buttons available which will assist in locating the element in active diagram, all diagrams, or the model view. There are also toolbar buttons to modify the search and filter out the results.



"bank" - 8 matches in Workspace

- Project One
  - ATM
    - Register ATM at Bank
  - Bank
  - Chief Teller

Diagram Commands

There are four commands available under **Diagram** on the main menu that you can use to customize your diagrams:

- **Grid**. When working in the Diagram View you have the option to display grid lines in the view.

  You also can 'snap objects to grid' so that when you place an object in the Diagram View, it automatically snaps to the nearest grid line. When an element is moved, it also snaps to the nearest grid.

  1. With a diagram open in the Diagram View, select **Diagram** > **Grid** from the main menu.
  2. Select **Diagram** > **Snap to Geometry** to activate this feature. This feature is active even if the grid lines are not displayed.

- **Layout**. Select **Diagram > Layout** and you have access to six different layout options for your diagram: all, all for printing, all and optimize size, selected elements, all except selected elements, and inner substructure.
- **Align**. Select **Diagram > Align** to access the different ways to align elements in your diagram: top, left, right, bottom, center horizontally, and center vertically.
- **Match**. When working with elements you can match elements in either width or height.
  1. Select elements in the diagram that you want to match in either width or height**.**
  2. Shift+Click the element you want to be matched
  3. When you select **Diagram > Match** all the selected elements will confirm to the selected element.

Zooming

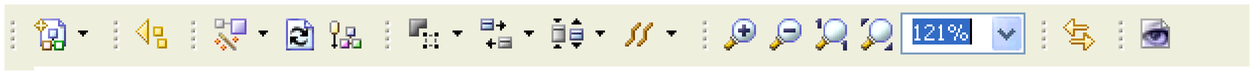
From the main menu select **Diagram** to gain access to the commands to adjust the zoom level of any diagram. These commands are also available on the Diagram View toolbar. You have several different options available:

- **Fit to window**. Zoom level is reduced in the Diagram View so that the entire diagram fits in the visible screen area.
- **Actual Size**.   Returns the diagram to the original size when created.
- **Zoom with preset percentages**. You can enlarge or reduce a diagram using a preset percentage.
- **Zoom In and Zoom Out**.   Each time your click the command the view increases or decreases by a set percentage.

Toolbars

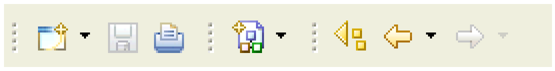
Layout and Alignment

Use this toolbar to align, evenly distribute, evenly size, stack, or group the elements in a diagram.



Formatting Toolbar

Use this toolbar to: create new projects, diagrams, etc.; save projects or diagrams; print diagrams; create a new diagram; open the parent diagram; or select elements in an opened diagram.



Overview Window

Click Show Overview ( ) on the Layout and Alignment toolbar. This is a 'thumbnail' view of your model and is useful when working with large diagrams that appear on more than one screen. You have the ability to pan the entire model. As you move the blue shaded box in the Overview window, the portion in the highlighted area appears in the Diagram View. Also, if you click an object in the Overview pane, that object appears in the Diagram View.

For large diagrams, the Overview window is a much more powerful navigation tool than using a combination of ER/Studio SA scroll bars and zoom features.

Summary

This completes the ER/Studio SA Evaluation Guide. For more information please refer to the online help in the application.

Additional Evaluation Resources

IDERA provides a variety of resources to help support your evaluation and selection of a Data Modeling product for your organization.

Web site

Visit our Web site for current product and company information, educational materials and supporting information at www.idera.com.

Electronic Documentation

Detailed reference documentation is available on the ER/Studio SA Evaluation CD or online at www.idera.com/support/productdocuments.

Email Support

You can contact ER/Studio SA support engineers, consultants and engineers directly by sending inquiries to: support@embarcadero.com or log a case at: https://www.embarcadero.com/support.