Chapter 6: Class Diagram

The Class Diagram

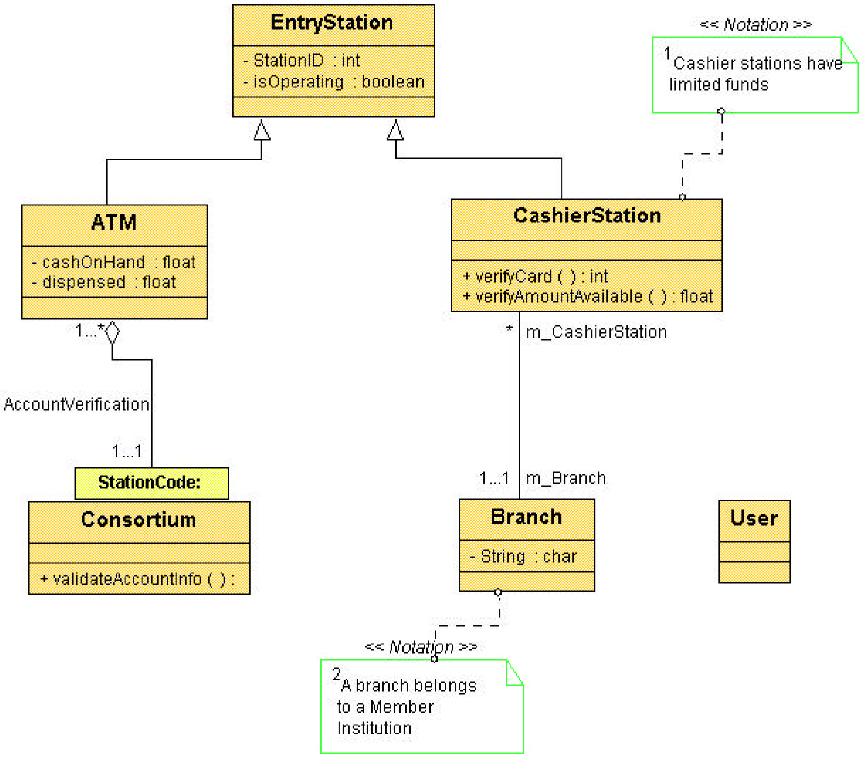
Using the Class Diagram model, you describe the static structure of the symbols in your new system. This model allows you to graphically represent symbol diagrams containing classes. Classes are arranged in hierarchies sharing common structure and behavior and are associated with other classes.

Drawing Classes

Classes define the attribute values carried by each symbol instance and the operations that each symbol performs or undergoes. When representing a class, you:

- Draw a class symbol
- Name the classes
- Enter class attributes
- Enter class operations
- Add links and associations
- Add notations

After you have completed the step-by-step procedure outlined in the tutorial, your class diagram should look similar to the following example.



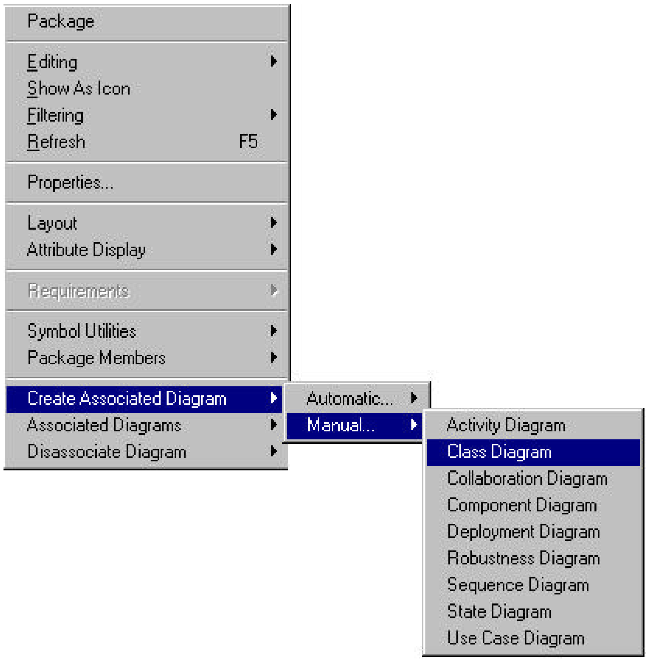
Note: The diagram shown above is for reference only. Use the instructions beginning on the next page to draw your Class diagram.

Creating a Class Diagram from the Package Diagram

The class diagram can be automatically created from the package diagram.

1. Make sure the package diagram called "CLD_1 - Banking" is the current diagram.

2. Right-click on the Member Institutions package (do not right-click on the diagram background) and the Package background menu opens.



3. Choose CREATE ASSOCIATED DIAGRAM->MANUAL->CLASS DIAGRAM. The Class Diagram Name dialog box opens. This diagram box shows the diagram type (class), the default diagram name (CLD #1 - MemberInstitutions). You can edit this name but for purposes of the tutorial we will leave it CLD #1 - Member Institutions. You can also edit the description of the diagram.

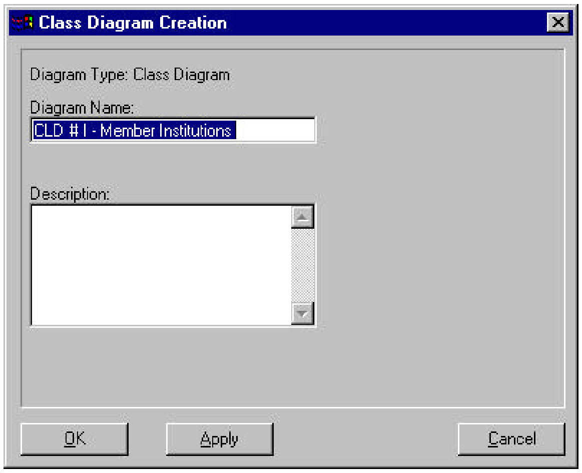

Class Diagram Creation

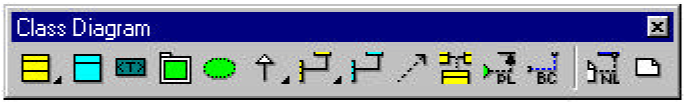
Diagram Type: Class Diagram

Diagram Name: CLD # I - Member Institutions

Description:

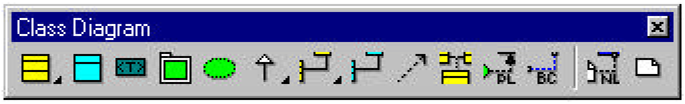
4. Click OK. The Class Diagram Name dialog box closes and Diagram Window opens with a Class Diagram labeled CLD_1 - MemberInstitutions. The Diagram Window is empty.

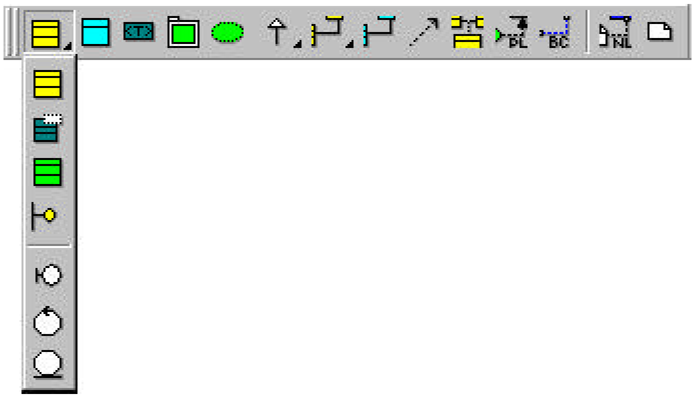
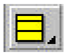
Note: The diagram you just created is now the active design model. When the Class model is created, the Diagram Window displays a palette with icon symbols used to create class diagrams.



The Class Diagram Palette

Each icon on this palette represents a notation used to create a class diagram. Some of the objects have related symbols. This is indicated by a triangle located in the lower right corner of the symbol icon. Click on the triangle and hold down the mouse button; a pull-right menu appears showing the available symbol. Slide the cursor to the symbol you want to use and the selected icon now appears on the diagram palette.



Icon	Notation	Definition
		
	Class	◆ A "type" of object ◆ Instantiated into an object ◆ Descriptor for a set of objects with the same structure, behavior and relationships

Template Class

Template is an abstract Class that is commonly used to define a collection type of object.

- contains the same information as a Class
- requires an attached Template Parameter Label
- provides the names and possible type of the parameters used by the Template

Utility Class

- A utility class is set of global variables and procedures that have been grouped in the form of a class declaration.
- It is a named collection of non-member attributes and operations scoped by the class.
- A utility class represents a type that has no instances.
- The attributes and operations of the utility become global variables and procedures.

Interface

An interface is a specifier for the externally-visible operations of a class, component, or other entities such as packages. Each interface often specifies only a limited part of the behavior of an actual class. An interface is formally equivalent to an abstract class with no attributes and no methods and only abstract operations.

Characteristics of interfaces:

- the internal structure is not specified
- do not have implementation
- lack attributes, states, or associations
- only have operations
- may have generalization relationships

Iconix Boundary Class

Actors use boundary objects in communicating with a system.

Iconix Control Class

Control objects serve as the glue between boundary objects and entity objects. These objects connect the user to the stored data.

Iconix Entity Class

Entity objects are usually objects from the domain model. These objects often map to database tables and files.

Object

- An object represents a particular instance of a class.
- You name the object, the name of the class of the object, and if applicable, the name of the package in which the object element can reside.
- If the class of the object is scoped, the scope name of the class is required, an example of this would be, Package1_ClassA.
- You can define object attributes and set the multiplicity of the object (single/multiple).

Instantiated Template Class

Template classes are models of classes. They correspond to the generic classes of Eiffel, and to the templates of C++. A template class must be instantiated before it can be used. You must instantiate it in order to obtain a real class that must in turn be instantiated to produce objects. During instantiation, actual parameters customize the real class based on the template class. Template classes facilitate the construction of universal collections typed by parameters.

Template classes are most often used in detailed design, to incorporate reusable components, for example, before instantiation the formal parameter appears in the dotted rectangle of the template class, afterwards, the actual parameter is joined to the name of class obtained by instantiation.

Package

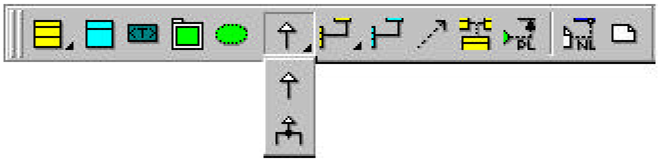
A package scopes the name of all of its members. A <<namespace>> package will implement a true name space in the generated code. Otherwise, the package is used for organization and name scoping.

The Package symbol is used in the Use Case, Class, and Component diagrams. A package is a grouping of model elements. Packages themselves may be nested within other packages. A package may contain both subordinate packages and ordinary model elements. Some packages may be Subsystems of Models. The entire system description can be thought of as a single high-level subsystem package with everything else in it.

Design Pattern

Patterns describe small, recurring entities, reusable on a day to day basis in order to resolve specific problems. These patterns do not express the general form of an application.

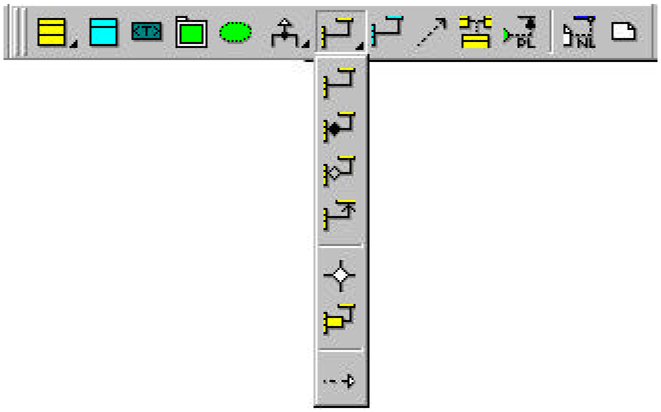
This symbol is available in the following diagrams: Use Case, Class, Collaboration, Deployment, and Component.



Generalization Link

Generalization is the taxonomic relationship between a more general element and a more specific element that is fully consistent with the first element and that adds additional information. This link is also known as a specialization or inheritance link.

Tree Generalization Link



Class Association

Drawing a line between the associated classes represents associations. Associations represent structural relationships between classes and can be named to facilitate model understanding. An association symbolizes a piece of information with a life cycle that is non-negligible in comparison to the general dynamics of object instances of the associated classes.

Composition

Aggregation

Navigable

Ternary Association

A ternary (n-ary) association is an association among 3 or more classes (a single class may appear more than once). Each instance of the association is an n-tuple of values from the respective classes. A binary association is a special case with its own notation.

Characteristics:

- Multiplicity for ternary associations can be specified but is less obvious than binary multiplicity
- The name of the association (if any) is shown near the diamond
- Role adornments can appear on each path as with a binary association
- The multiplicity on a role represents the potential number of instance tuples in the association when the other N-1 values are fixed
- Qualifiers and aggregation are not permitted
- An association class symbol may be attached to the diamond by a dashed line. This indicates a ternary association that has attributes, operations, and/or associations

Qualifier

A qualifier is an attribute or list of attributes whose values serve to partition the set of objects associated with an object across an association. The qualifiers are attributes of the association.

Characteristics:

- is shown as a small rectangle attached to the end of an association path between the final path segment and the symbol of the class that it connects to

- qualifier rectangle is part of the association path, not part of the class

- qualifier is attached to the source end of the association -- every target falls into exactly one partition

- the multiplicity attached to the target role denotes the possible cardinalities of the set of target symbols selected by the pairing of a source symbol and a qualifier value

- Common values:
  "0..1" -- a unique value may be selected, but every possible qualifier value does not necessarily select a value
  "1" -- every possible qualifier value selects a unique target symbol, therefore the domain of qualifier values must be finite
  "*" -- the qualifier value is an index that partitions the target symbols into subsets

- the qualifier attributes are drawn within the qualifier box. There may be one or more attributes shown one to a line

- qualifier attributes have the same notation as class attributes, except that initial value expressions are not meaningful

- You can have a qualifier on each end of a single association

Realize

A realization is a semantic relationship between classifiers - one classifier specifies a contract that another classifier guarantees to carry out. These relationships are used in two places: between interfaces and the classes that realize them, and between use cases and the collaboration that realize them.

Object Association

Just as associations relate classes, symbols are related by links. A link is an instantiation of an association.

Dependency Link

The dependency link is a semantic relation between the source and target elements. It indicates that when there is a change to the target element there may be a change necessary to the source element. You can label the dependency link and set the stereotype.

Association Class Link

An Association Link usually helps further define a many to many relationship.

- Association that has attributes, operations and other associations
- Link is shown as a dashed line between the association and the Class defining the attributes and operations
- There are no specific properties for the link

Pattern Member Link

Used to show the member classes or symbols participating in a pattern.

Binary Constraint

The binary constraint notation is available on all diagram palettes. A constraint is a semantic relationship among model elements that specifies conditions and propositions that must be maintained as true. Otherwise the system described by the model is invalid. Certain kinds of constraints (such as association "or" constraint) are predefined in UML, others can be user defined. A constraint represents semantic information attached to a model element, not just a view of it.

A binary constraint allows a constraint to be defined between any symbols on the diagram. The binary constraint allows the constraint to be defined on the link rather than in a note symbol. If there is a need for a single constraint or three or more way constraint, then a note symbol is used to

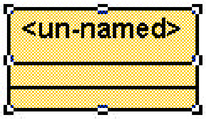
		explain the constraint and the note symbol is linked to the constrained symbols using a note link.
NL	Note Link	The note link notation is available on all diagram palettes. Use this link to connect the note to the symbol
□	Note Pad	The note pad notation is available on all diagram palettes. The note pad can be used to record information for a symbol or link in a diagram. This information is not included in generated code but is for information only. Each note pad can contain unlimited text, and be numbered. You can also define a stereotype, and enter a noted element.

The EntryStation Class

The first class you will draw in the current diagram is the "EntryStation" class.

1. From the Class Diagram palette select the Class symbol icon .

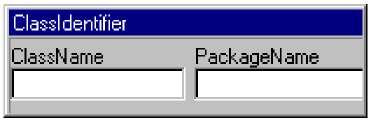
2. Position the cursor in the top middle portion of the drawing area and click. The class symbol is placed on the design area. When you first place the symbol, it is drawn at its default size. The symbol expands or shrinks to fit any inserted text.



Entering Data in a Symbol

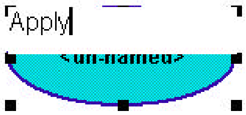
There are three methods that can be used to name the symbol. The first method is outlined below:

1. Once you place the class symbol you can begin typing immediately without opening a dialog box or a pop-up editor. When you do this the ClassIdentifier pop-up editor opens with the cursor in the ClassName text box.



2. Type the name "EntryStation" and press Enter. The Class Identifier pop-up editor closes and the <unnamed> text is replaced with the label "EntryStation". This is the fastest, most efficient method of entering data and can be used on any symbol.

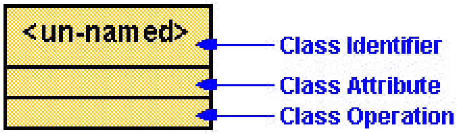
Note: Depending on the symbol, a text box may open instead of a pop-up editor. Enter the text and click anywhere outside the text box.



We will demonstrate the other method for naming the class symbols later.

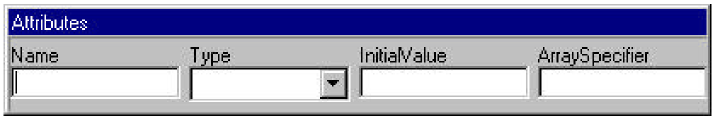
To Represent the Attributes of the EntryStation Class

Now that you have drawn the EntryStation symbol and named it, you need to describe its attributes. The attributes for this class are Station id, and interface. There are no operations for this class.



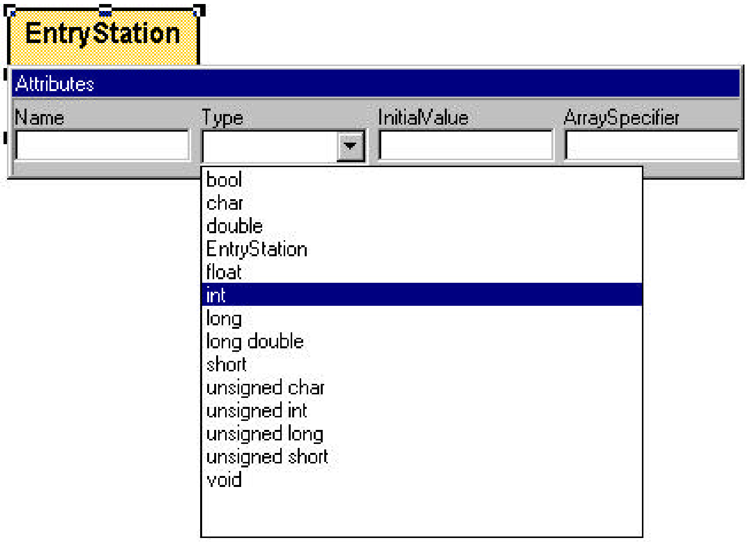
1. Click in the Attributes section (center section) of the "EntryStation" class symbol to select the class symbol.

2. Click once again in the Attributes section and an Attributes pop-up editor opens.



3. Type “StationID” in the Name text field.

4. Tab to the Type text field and click the drop-down arrow. A drop-down menu opens.

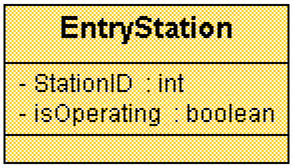


5. Select "int" from the drop down list. The drop-down list closes and "int" appears in the Type text box.

6. Press the Enter key. An empty pop-up editor is displayed.

7. Type “isOperating” in the Name text field.

8. Type “boolean” in the Type text field and click anywhere on the drawing area to close the pop-up editor. The attributes appear in the class symbol.



This method of entering attributes is called direct entry.

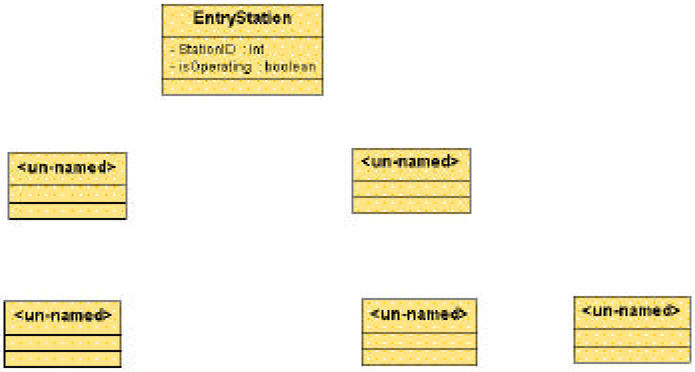
The Remaining Classes

The next classes you will draw are the ATM, Consortium, CashierStation, Branch and User classes. We will use the other two methods to name these classes.

To represent the Classes:

1. Double-click on the Class icon to place multiple symbols.

2. Using the cursor, click your drawing area five times to place the other five class symbols as shown in the following diagram.



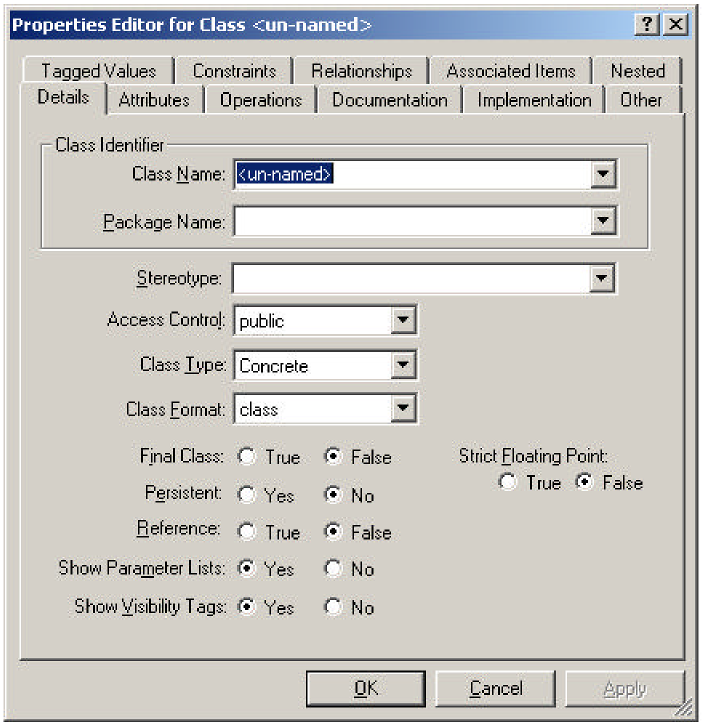
Note: All of the symbols can be rearranged in the drawing area if required.

3. Deselect the class symbol icon by clicking the cursor icon located by the Class Diagram palette or pressing ESC.

Using the Properties Editors

Using the dialog box is the second method for entering attributes. However, you will probably use the direct entry method most of the time.

1. Double-click in the top section of the class symbol that will become the "ATM" class. The Properties Editor for Class dialog box opens.



Class Name: <un-named>

Package Name:

Stereotype:

Access Control: public

Class Type: Concrete

Class Format: class

Final Class: ○ True ● False

Strict Floating Point: ○ True ● False

Persistent: ○ Yes ● No

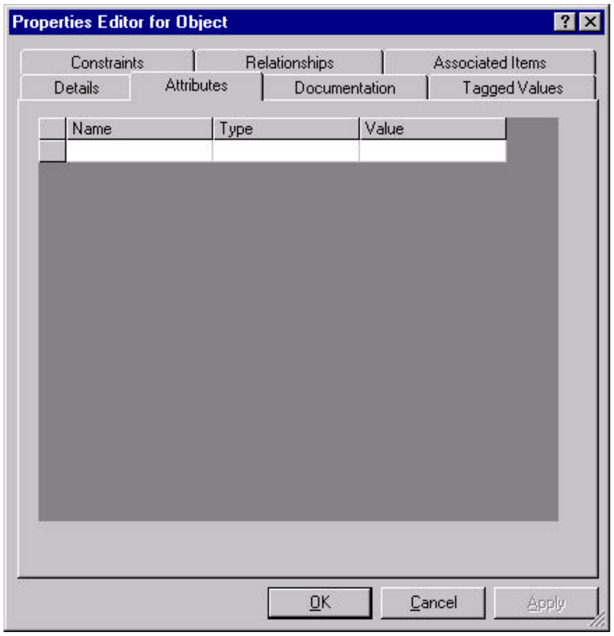
Reference: ○ True ● False

Show Parameter Lists: ● Yes ○ No

Show Visibility Tags: ● Yes ○ No

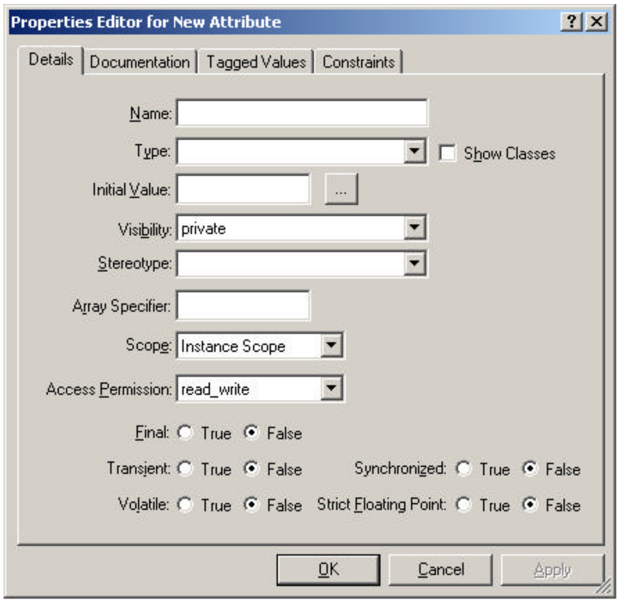
2. Enter the name “ATM” in the ClassName text field.

3. Click the Attributes tab and the Properties Editor for Class dialog box opens.



Name	Type	Value

4. Click New... and the Details portion of the properties editor opens.



Name:

Type: □ Show Classes

Initial Value:

Visibility: private

Stereotype:

Array Specifier:

Scope: Instance Scope

Access Permission: read_write

Final: ○ True ● False

Transient: ○ True ● False Synchronized: ○ True ● False

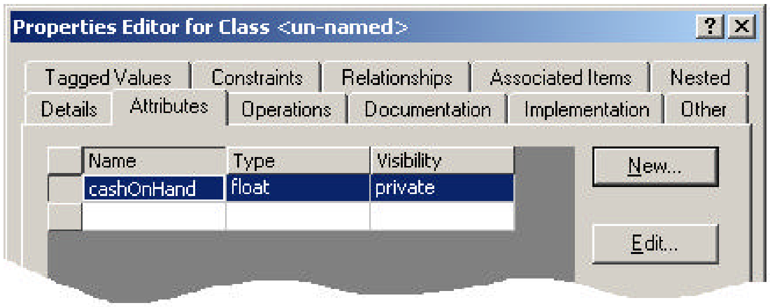
Volatile: ○ True ● False Strict Floating Point: ○ True ● False

5. Enter “cashOnHand” in the Name text field.

6. Enter “float” in the Type text field

7. The default setting for Visibility is "internal" which appears as a minus sign in front of the attribute in the class symbol. We will not change this default setting for this tutorial.

8. Click OK. The Details properties dialog box closes and the information you entered appears in the Attributes dialog box.



Name	Type	Visibility
cashOnHand	float	private

Note: Do not close the Properties Editor dialog box. We will make one more entry.

Continue Entering Attributes

1. Click New... once again in the Attributes section. Another New Attributes dialog box opens.

2. Enter “dispensed” in the Name text field.

3. Click the drop-down arrow located to the right of the Type text box and a drop down menu opens. Select “float” from the list and the drop down menu closes.

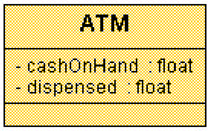
4. Leave the default setting of "internal" for Visibility.

5. Click OK. The New Attributes properties dialog box closes and the information you entered appears in the Attributes dialog box.

6. Click the Documentation Tab in the same properties editor dialog box and the related dialog box opens.

7. Enter the text "Describes the actions of the ATM cash dispensing machine" in the Description text box.

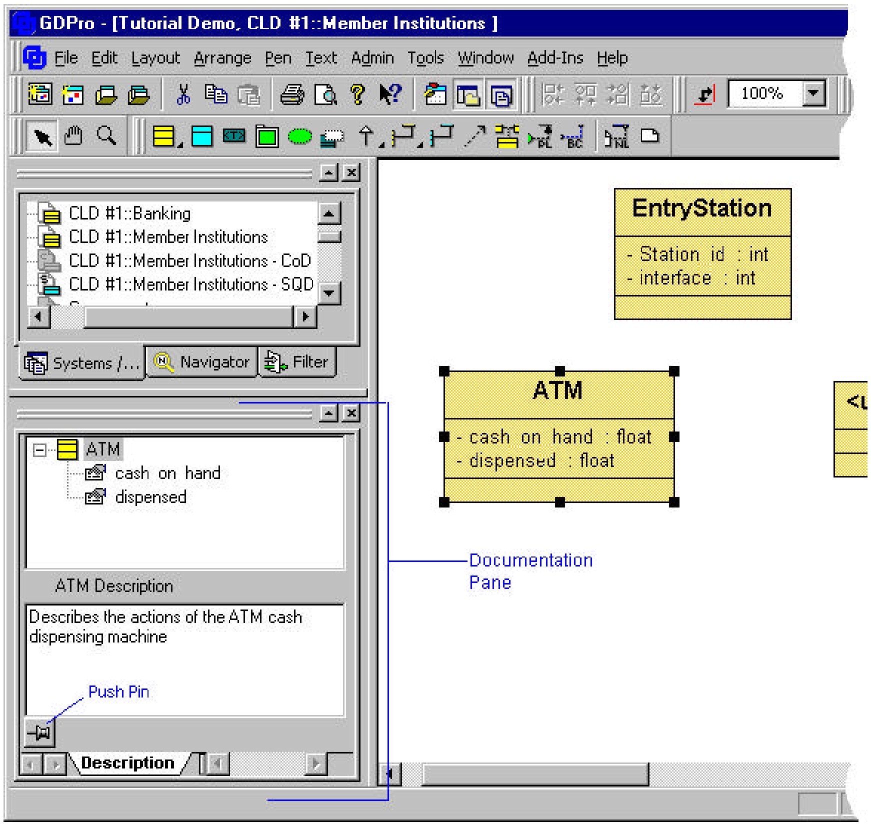
8. Click OK to close the Properties Editor dialog box. The attributes appear in the ATM class symbol. However note that the description you entered does not appear in the class symbol itself. It does appear in the Documentation Pane located in the bottom left corner of the GD*Pro* window.



Note: Notice that the size of the class symbol adjusts to fit the text entry.

Using the Documentation Pane

As you create diagrams, the symbol you are working on appears in the Documentation Pane. This pane allows you to view the operations and attributes of the symbol. You can also view and/or modify the description of the model element.


GDPro - [Tutorial Demo, CLD #1::Member Institutions]
File Edit Layout Arrange Pen Text Admin Tools Window Add-Ins Help
100%
CLD #1::Banking
CLD #1::Member Institutions
CLD #1::Member Institutions - CoD
CLD #1::Member Institutions - SQD
Systems /...
Navigator
Filter
ATM
cash on hand
dispensed
Push Pin
Description
EntryStation
- Station id : int
- interface : int
ATM
- cash on hand : float
- dispensed : float
<L
Documentation Pane

ATM Description

Describes the actions of the ATM cash dispensing machine

The description you entered for the ATM class symbol appears in the lower portion of the Documentation Pane. You can edit or add additional information in this text box.

1. Enter the text "These are stand-alone ATMs." in the Documentation Pane Description text box.

2. Double-click on the ATM class symbol to open the Properties Editor for Class ATM dialog box. Click the Documentation Tab. The text that you entered in the Documentation Pane also appears in the Properties Editor dialog box.

3. Click OK to close the Properties Editor dialog box.

All attributes and operations that you add to the class symbol appear in the upper portion of the Documentation Pane. The attributes are represented by an attribute symbol . The operations are represented by .

1. Click on the attribute "cash on hand" in the Documentation Pane to select it. The description portion of the pane is now labeled "cash on hand".

2. Enter the text "maintains a balance of available cash".

3. Double click the ATM symbol to open the Properties Editor. Click the Attributes Tab and then select the "cash on hand" attribute in the list box.

4. Click the Edit button and the Attribute Editors for cash on hand dialog box opens.

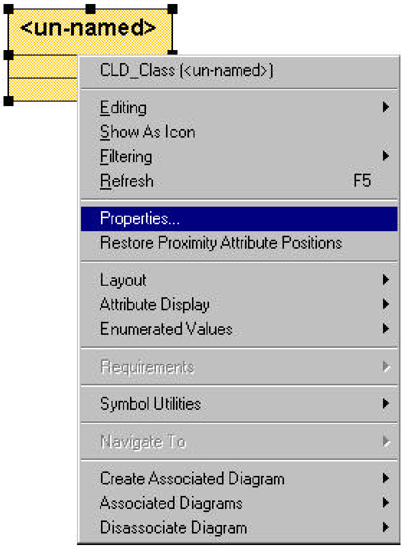
5. Click the Documentation Tab and the description you entered in the Documentation Pane appears in this description text box.

6. Click OK twice, once to close the New Properties Editor dialog box, and once to close the main dialog box.

Using the Background Menu to Enter Data

You can use the background menus as a third method of entering data in a symbol.

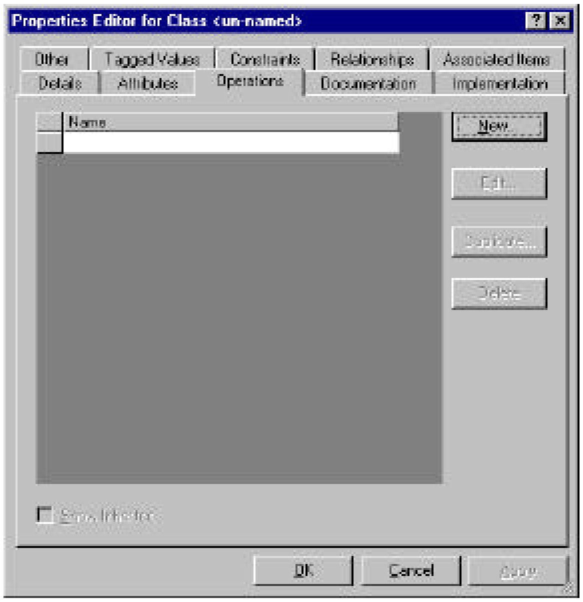
1. Right-click the class symbol that will become the "Consortium" class. The Class background menu is displayed.



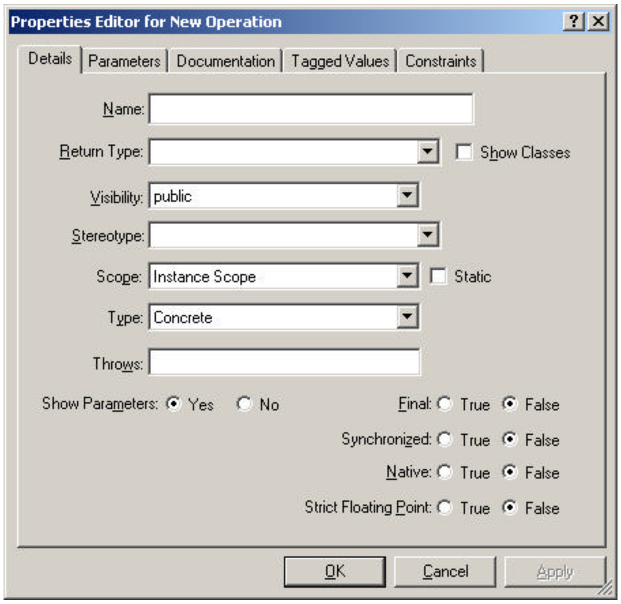
2. Choose Properties from the menu. The Properties Editor for Class dialog box opens.

3. Enter the name “Consortium” in the ClassName text field.

4. Click the Operations tab and the Name portion of the dialog box opens.



5. Click New... and the Details portion of the Properties Editor dialog box opens.



Properties Editor for New Operation

Details | Parameters | Documentation | Tagged Values | Constraints

Name:

Return Type: □ Show Classes

Visibility: public

Stereotype:

Scope: Instance Scope □ Static

Type: Concrete

Throws:

Show Parameters: ● Yes ○ No

Final: ○ True ● False

Synchronized: ○ True ● False

Native: ○ True ● False

Strict Floating Point: ○ True ● False

OK Cancel Apply

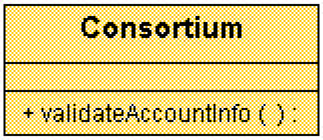
6. Enter the text "validateAccountInfo" in the Name text box. There is no Return Type for this operation.

7. The default setting for Visibility is "public" which appears as a plus sign in front of the operation in the class symbol. We will not change this default setting for this tutorial.

8. Click OK. The Details dialog box closes and the text appears in the Name dialog box.

9. Click OK in the Operations dialog box. The dialog box closes and text is entered into the "Consortium" class symbol.

Note: There are no attributes for the "Consortium" class symbol**.**



To Complete the Remaining Class Symbols

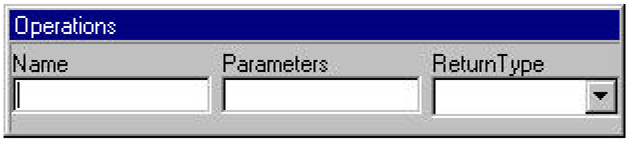
We will use the “direct entry” method to complete the remaining Class symbols but you may use any of the methods you prefer.

1. Position the cursor over the text <unnamed> in the Class symbol that will become the "CashierStation" class and click once to select the symbol. Click once again and the Class Identifier pop-up editor opens.

2. Enter the name "CashierStation" and click anywhere outside the pop-up editor. The pop-up editor closes and the name “CashierStation” is entered in the top section of the Class symbol.

Enter Operations Attributes

1. Click in the operations section (bottom section) of the "CashierStation" class symbol. A ClassOperations pop up window opens.

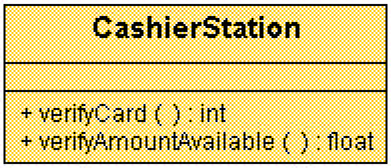


2. Enter “verifyCard” in the Name text field.

3. Tab to the ReturnType text field and enter “int” in the text field. Press Enter and a new Operations pop-up editor opens.

4. Enter “verifyAmountAvailable” in the Name text field.

5. Enter “float” in the ReturnType text field and click anywhere outside the pop-up editor. The pop-up editor closes and the operation appears in the bottom section of the class symbol.

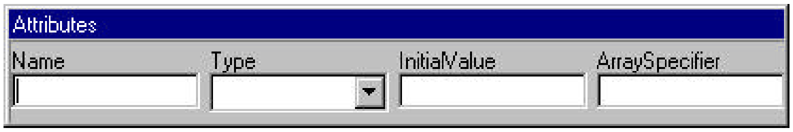


Enter Name for the Branch and User Classes

1. Position the cursor over the text <unnamed> in the Class symbol that will become the "Branch" class and click once to select the symbol. Click again and a pop-up editor opens.

2. Enter the name “Branch” and press Enter. The name “Branch” is entered in the top (name) section of the Class symbol.

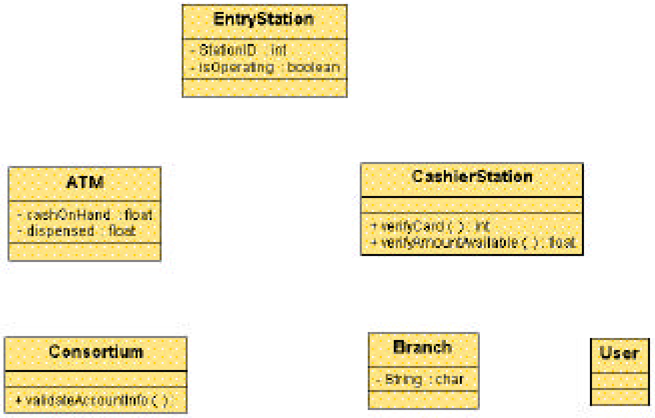
3. Click in the attributes section (center section) of the Branch class symbol. An Attributes pop-up editor is displayed.



4. Enter “String” in the Name text field.

5. Enter “char” in the Type text field and click anywhere outside the pop-up editor. The pop-up editor closes and text is entered into the Branch class symbol. There are no operations for this class symbol.

6. Following steps 1 through 2, enter the class name "User" for the last unlabeled class symbol. This class has no attributes or operations.



Depicting a Qualified Association

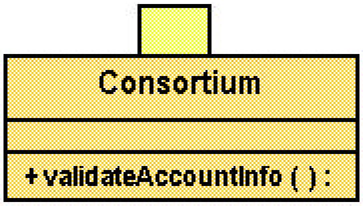
A qualified association relates two symbol classes and a qualifier. The qualifier is a special attribute that reduces the effective multiplicity of an association. You depict a qualified association as a small box on the end of the association line near the class it qualifies.

The qualified association you are about to depict is between the "ATM" class and the Consortium class.

To depict a qualified association:

1. From the Class Diagram palette select the Qualifier icon . Note that this icon is located on the pull-down menu located under the Class Association Icon

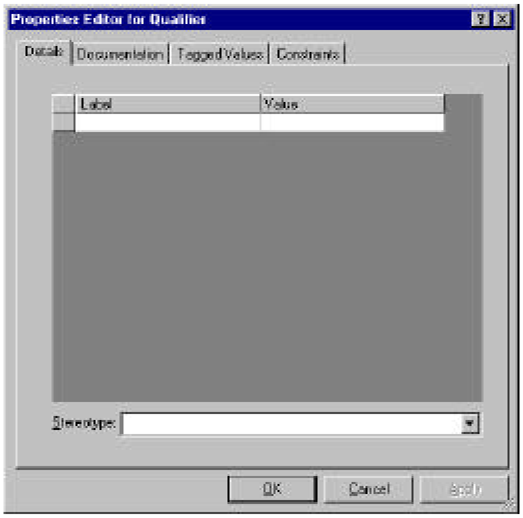
2. Position the cursor near the top edge of the "Consortium" class symbol and click. The qualifier symbol appears and is attached to the "Consortium" class symbol.



Note: You know the qualifier symbol is attached to the class symbol because the "handles" around the qualifier symbol are blue once you place it on the diagram. If there is no relationship, the handles are black

Naming a Qualified Association

1. Double-click the Qualifier. The Properties Editor for Qualifier dialog box opens.



2. Double-click in the Label portion of the text box. Enter “Stationed” in the text box and click OK. The dialog box closes and the text is entered in the Qualifier box attached to the Consortium class symbol.

Identify Associations

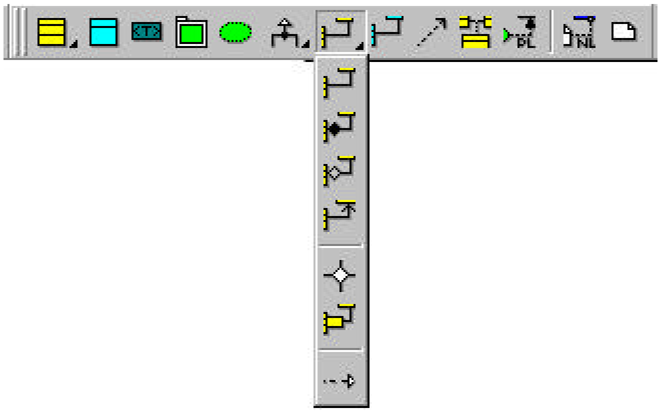
An association describes a group of links with common structure and common semantics. All the links in an association connect symbols from the same classes.

GD*Pro* not only allows for associations but also multiplicity of associations. Multiplicity specifies how many instances of one class may relate to a single instance of an associated class. Multiplicity constrains the number of related symbols.

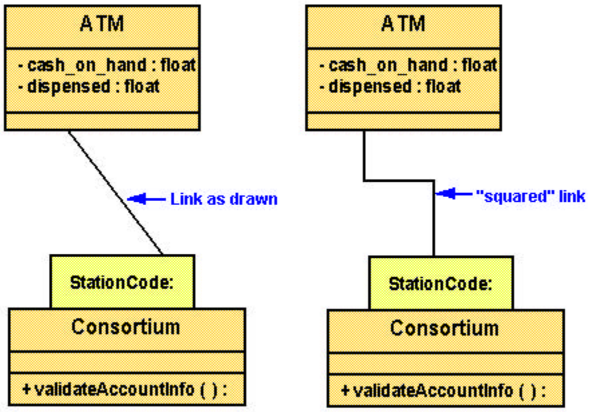
Role names are also supported and provide a way of traversing associations from an symbol at one end, without explicitly mentioning the association. Role names are often used for associations between two symbols of the same class.

Class Associations

1. Select the Class Assocation-Aggregation from the Class palette pull-down menu

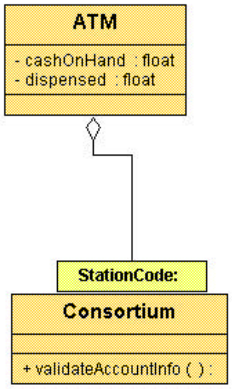


Key: Because LAYOUT->ORTHOGONAL LINKS is selected from the menu the links automatically adjust to right angles when completed.

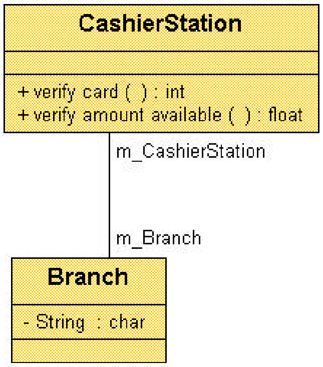


Note: If you did not select Orthogonal Links command before drawing the link you can square the link after it is drawn as well. Select the link to be squared and choose LAYOUT->CHANGE LINE STYLE->ORTHOGONAL LINK MODE from the menu. The selected link adjusts to right angles.

2. Click inside the bottom of the "ATM" class symbol. Drag the cursor down inside the top portion of the Station Code Association Qualifier symbol and click once again. A valid link is drawn between the two classes.



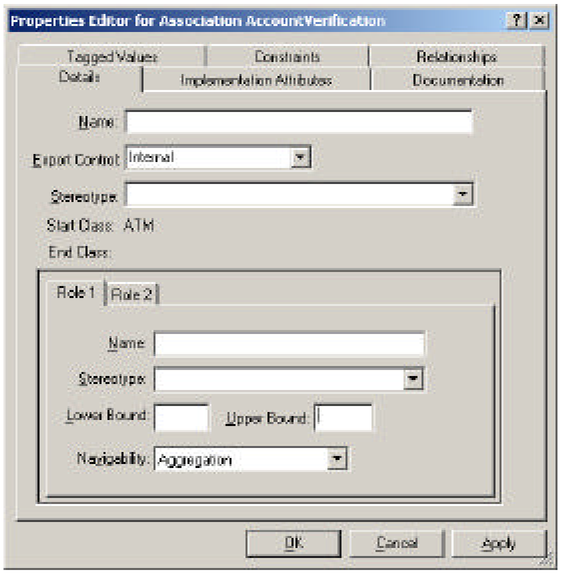
3. Select the AssociationLink from the draw-down menu and draw an association link from the CashierStation class to the Branch class.



Note: You can also change the Association by using the Enumerated Values command

Labeling the Class Association

1. To label the class association between the ATM and Consortium classes, double-click the association link. The Properties Editor dialog box opens.



Properties Editor for Association AccountVerification

Name:

Export Control: Internal

Stereotype:

Start Class: ATM

End Class:

Name:

Stereotype:

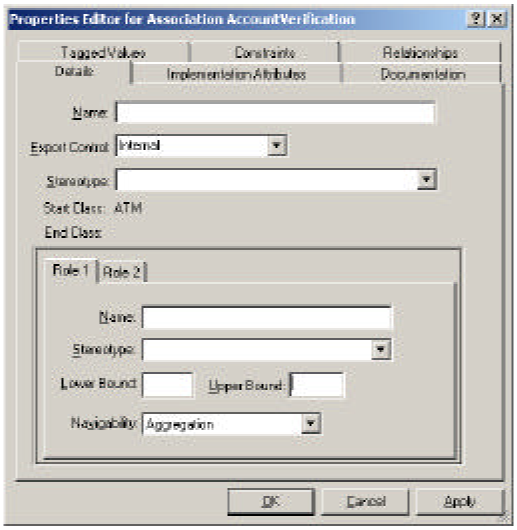
Lower Bound: Upper Bound:

Navigability: Aggregation

2. Enter the name "AccountVerification" in the Name text box.

3. Leave the Properties Editor dialog box open to set the Role Multiplicity for this association.

Role Multiplicity for the First Association



Properties Editor for Association AccountVerification

Name:

Export Control: Internal

Stereotype:

Start Class: ATM

End Class:

Name:

Stereotype:

Lower Bound: Upper Bound:

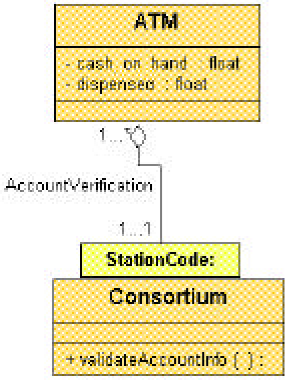
Navigability: Aggregation

1. Under the Role 1 tab, enter a "1" in the LowerBound text box.

2. Enter an "*" in the UpperBound text box.

3. Under the Role 2 tab enter a "1" in both the LowerBound and UpperBround text boxes.

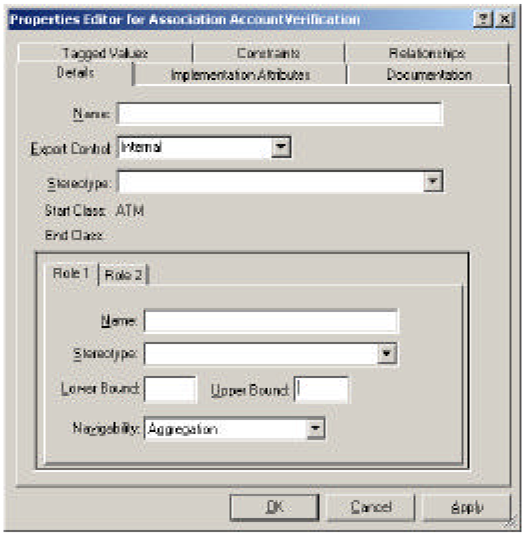
4. Click OK, the Association dialog box closes and the association symbol is labeled.



Note: You can reposition the AccountVerification label by placing the cursor on the label and while holding down the mouse button, move the label. Release the mouse button when the label is in place.

Setting the Role Multiplicity for the Second Association

1. Double-click on the Association symbol between the "CashierStation" class and the "Branch" class symbol. The Properties Editor for Association dialog box opens.

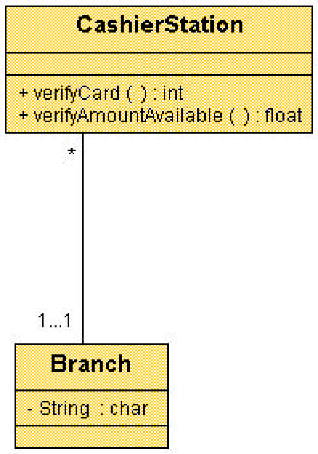


Name:
Export Control: Internal
Stereotype:
Start Class: ATM
End Class:
Name:
Stereotype:
Lower Bound:
Upper Bound:
Navigability: Aggregation

2. Under the Role 1 tab, enter an "*" in the UpperBound text box.

3. Under the Role 2 tab, enter "1" in the LowerBound and UpperBound text boxes.

4. Click OK. The Properties Editor dialog box closes and the ClassAssociation link is labeled.



Depicting Generalization and Inheritance

Generalization is the relationship between a class and one or more refined versions of this class. The class being refined is called the superclass and each refined version is called a subclass. The attributes and operations of the superclass can be exhibited in its subclasses; as a result, each subclass is said to inherit the features of the superclass. Organize classes by using inheritance to share common structure.

When representing a generalization, you draw the generalization links, attaching them to the desired subclasses.

The generalizations you are about to depict are between the EntryStation class (superclass) and the "ATM" and "CashierStation" classes (subclasses).

Draw Generalization Links

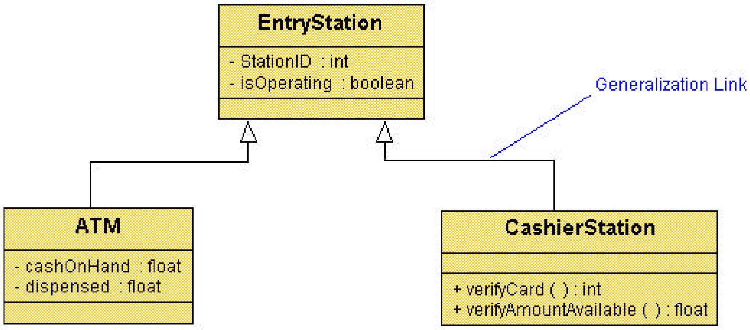
1. From the Class Diagram palette, double-click the Generalization Link symbol icon .

2. Click in the ATM class symbol and drag the cursor up to the lower portion of the EntryStation class symbol and click again. A generalization link snaps in place between the two class symbols.

3. Click in the CashierStation class symbol and drag the cursor up to the EntryStation class symbol and click again. The second generalization link symbol is drawn.

Note: Because the Orthogonal Links command is selected from the Layout menu, the generalization links will be squared automatically.

4. Deselect the Generalization Link symbol icon by clicking the cursor icon located by the Class Diagram palette or press the ESC key.

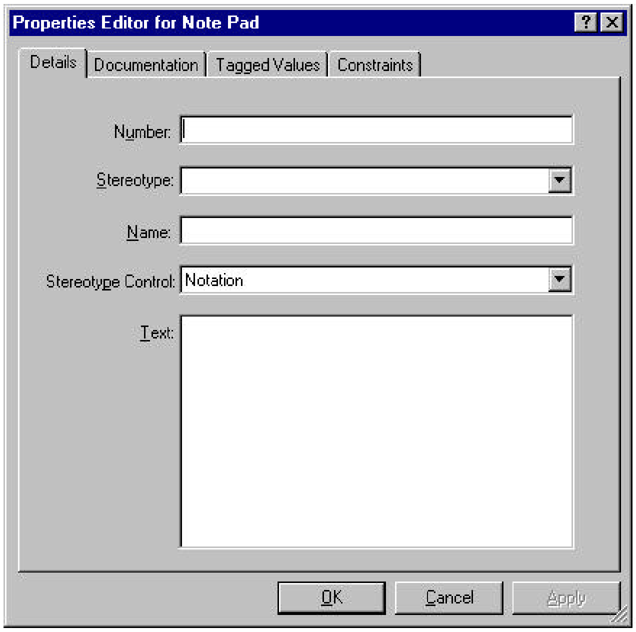


Adding Notes to the Diagram

1. Click the Note Pad icon 🗋 in the Class Diagram Palette.

2. Place the cursor to the right and just above the CashierStation class symbol. Click once and a note pad symbol is placed on the diagram.

3. Double-click the note symbol and the Properties Editor for Note Pad dialog box opens.



Number:

Stereotype:

Name:

Stereotype Control: Notation

Text:

4. Enter the number "1" in the Number text box.

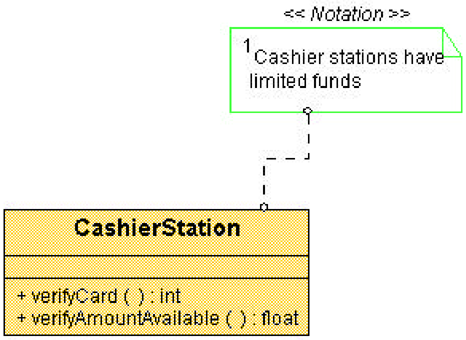
5. We will leave the Stereotype setting as "Notation" for this tutorial.

6. Click inside the Text box and enter the following text: "Cashier stations have" and press the Enter key. The cursor advances to the next line in the text box.

7. Enter the text "limited funds" and click OK. The Properties Editor dialog box closes and the Note pad is populated..

8. Click the NoteLink icon in the Class Diagram palette.

9. Click once inside the NotePad symbol, drag the cursor to inside the Cashier Station symbol and click again. The link snaps in place between the NotePad symbol and the class symbol.

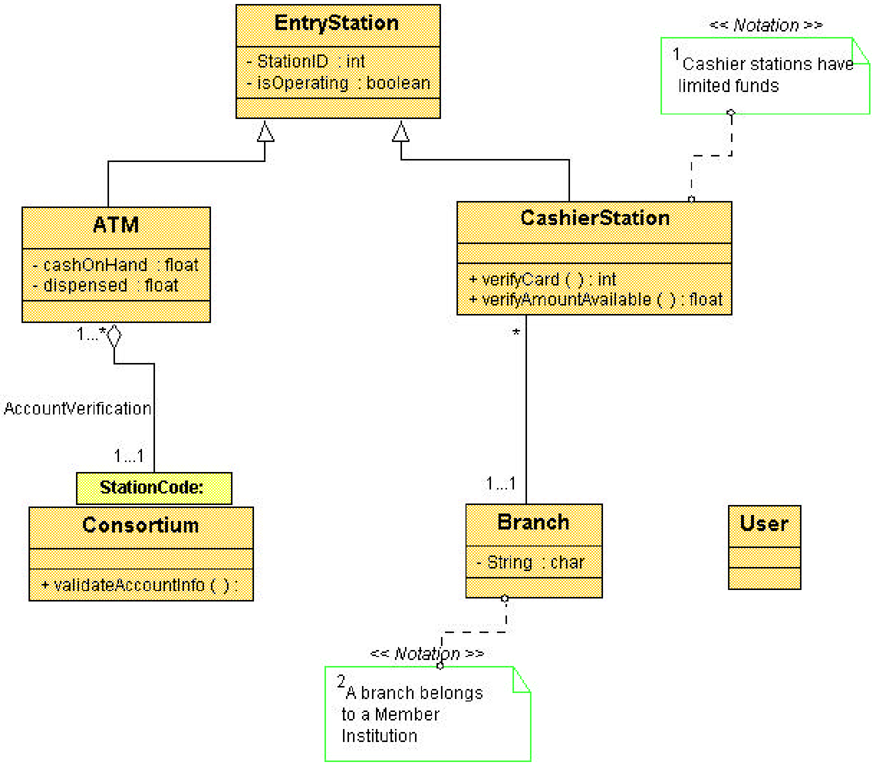

<< Notation >>
1 Cashier stations have limited funds
CashierStation
+ verifyCard ( ) : int
+ verifyAmountAvailable ( ) : float

10. Follow the procedure outlined in steps 1 - 9 to place and label the second notation. Position this notation symbol below the Branch class symbol.

11. Number this notation as "2" and enter the following text: "A branch belongs to a Member Institution"

The Completed Class Diagram

Your completed class diagram should resemble the following graphic:


EntryStation
- StationID : int
- isOperating : boolean
<< Notation >>
1 Cashier stations have limited funds
ATM
- cashOnHand : float
- dispensed : float
CashierStation
+ verifyCard ( ) : int
+ verifyAmountAvailable ( ) : float
1...*
*
AccountVerification
1...1
StationCode:
Consortium
+ validateAccountInfo ( ) :
1...1
Branch
- String : char
User
<< Notation >>
2 A branch belongs to a Member Institution