Chapter 9: State Diagram

The State Diagram

What is a State Diagram

- Provides a very detailed picture of how a specific symbols changes states.
- A state refers to the value associated with a specific attribute of an object and to any actions or side effects that occur when the attribute's value changes

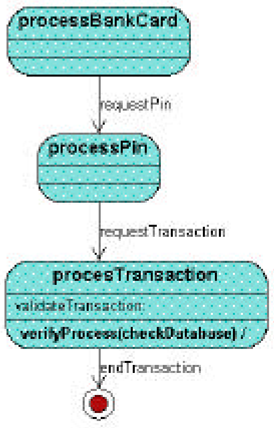
When to use a State Diagram

- Used when you are working on real-time process control applications or systems that involve concurrent processing
- When you want to show the behavior of a class over several use cases

Creating a State Diagram

A state diagram shows the sequences of states that an object or an interaction goes through during its life in response to received stimuli, together with its responses and actions.

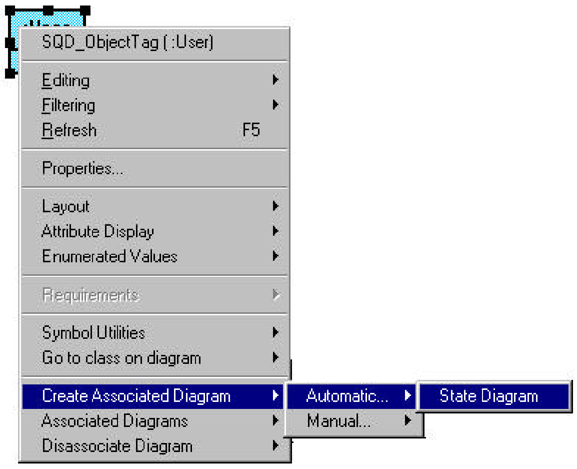
In this tutorial, you will create the State Diagram using GD*Pro's* automatic model generation feature. When you have completed your State Diagram portion of the tutorial your model should be similar to the following example.



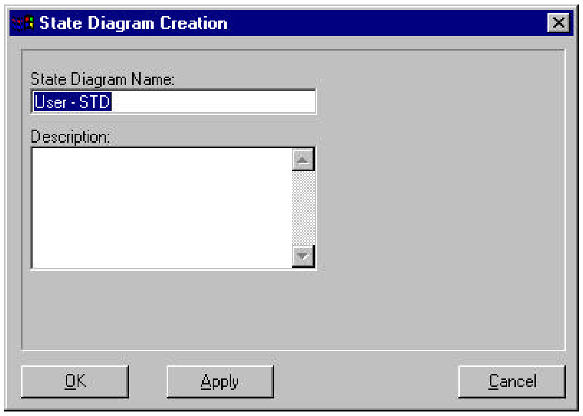
Note: The diagram shown above is for reference only. Use the instructions beginning on the next page to draw your State diagram.

To Draw the State Diagrams Automatically

1. Right-click the object tag labeled "User" in the sequence diagram. The Object Tag background menu opens.



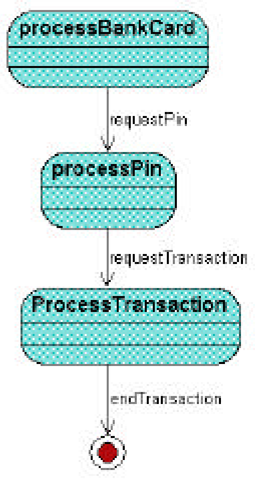
2. Choose CREATE ASSOCIATED DIAGRAM->AUTOMATIC->STATE DIAGRAM. The State Diagram Name dialog box opens. The default name includes the object tag name (User) and the type of diagram (STD). You can edit this name but for purposes of the tutorial we will leave it User - STD. You can also edit the Description of the diagram.



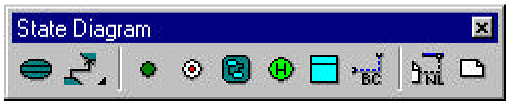
State Diagram Name:
User - STD

Description:

3. Click . The State Diagram Name dialog box closes and a Diagram Window opens with a State Diagram labeled "User". If the diagram does not resemble the following diagram, go back to the sequence diagram and make sure all the message links are connected to the object activation bars.



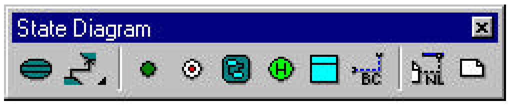
Note: The diagram you just created is now the active design model. When the State model is created, the Diagram Window displays a palette with icon symbols used to create state diagrams.



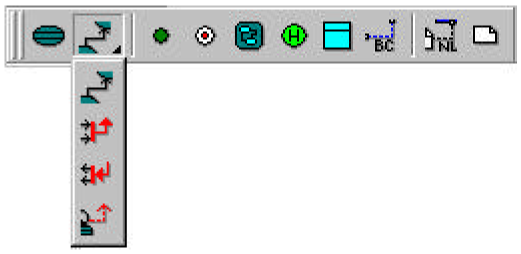
Note: When the State Diagram diagram is created, the class and sequence diagrams remain open as well.

The State Diagram Palette

Each icon on this palette represents a notation used to create a state diagram.



Icon	**Notation**	**Definition**
	State	◆ Each object is in a particular state at a given point in time. States are represented as rounded rectangles with an identifying name. ◆ An object is not always in the same state at a given time, and an object cannot be in an unknown or undefined state. ◆ A state is the image of an instantaneous combination of the values contained in the object's attributes, and the presence or the absence of links from the given object to other objects.



	State Transition	A transition is a relationship between two states. This transition indicates that an object in the first state is entered the second state and perform certain actions when a specified event occurs if specified conditions are satisfied. The trigger to "fire" a transition is the occurrence of the event labeling the transition. The event can have parameters, which are available within actions specified on the transition or within actions initiated in the sub-sequent state. Events are processed one at a time.
	Transition Merging	The transition merging symbol merges concurrent transitions to a single target state. Both source transitions must be fired to perform the merge.
	Transition Branching	The state transition branching symbol splits a state transition to multiple target states. It splits the control into concurrent threads.
	Send Event	Transitions "inside" of one object can send events to other objects. The event is sent as part of the action fired by the transition. A send event to an object can trigger a transition.
●	Initial State	A pseudo state to establish the start of the event into an actual state.
	Final State	The final state symbol represents the completion of the activity.
	Composite State	The composite state is also know as a "Superstate". It is a state that contains substates that compose the superstate. The Composite State has a name property.
	History State	The history state symbol is contained in the Composite State region. If transition to this state is activated, the object resumes the state it last had.
	Object	The objects for which states and transitions are modeled can themselves be depicted on the State Diagram. An object encapsulates (contains) the states and transitions modeled for that object.
	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.

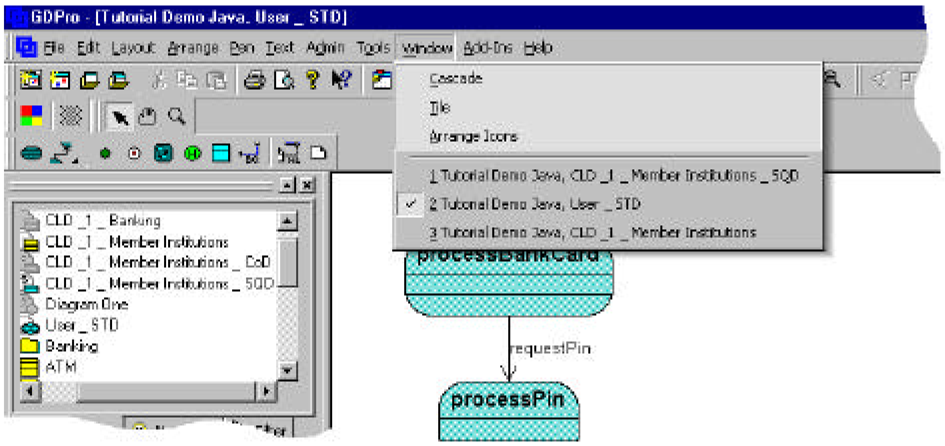
Note Link	The note link notation is available on all diagram palettes. The note pad can be used to record information for an object or link in a diagram. This information is not included in generated code but is for information only. Each note pad can contain unlimited text, can be numbered, a stereotype defined, and a noted element entered.
Note Pad	The note pad notation is available on all diagram palettes. The note pad can be used to record information for an object 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.

Return to the Sequence Diagram

It is easy to go from diagram to diagram within an open system.

Method One

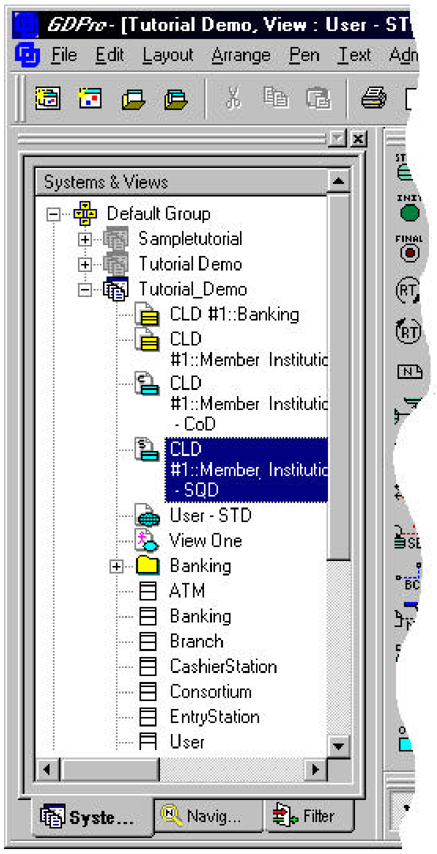
1. To return to the sequence diagram choose WINDOW from the main menu. The lower portion of the Window commands list box displays the open diagrams. A check mark indicates the current diagram.


GDPro - [Tutorial Demo Java. User _ STD]
File Edit Layout Arrange Pen Text Admin Tools Window Add-Ins Help
Cascade
Tile
Arrange Icons
1 Tutorial Demo Java, CLD _1 _ Member Institutions _ SQD
✓ 2 Tutorial Demo Java, User _ STD
3 Tutorial Demo Java, CLD _1 _ Member Institutions
CLD _1 _ Banking
CLD _1 _ Member Institutions
CLD _1 _ Member Institutions _ CoD
CLD _1 _ Member Institutions _ SQD
Diagram One
User _ STD
Banking
ATM
processBankCard
requestPin
processPin

2. Choose **Tutorial Demo, diagram: CLD_1_Member Institutions - SQD** from the list. The sequence diagram opens.

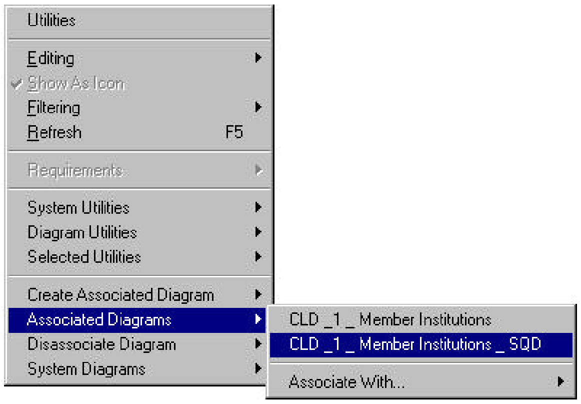
Method Two

You can double-click the name of the diagram (CLD_1_Member Institutions - SQD) you want to open in the System Hierarchy Window.



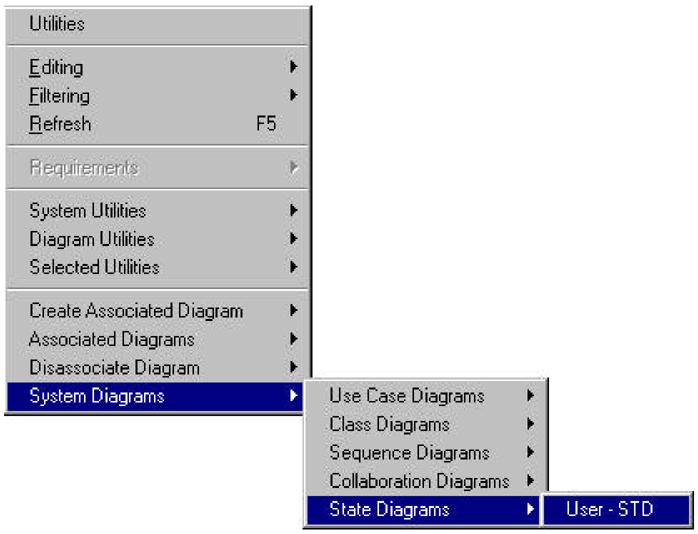
Method Three

1. To return to the Sequence Diagram, right-click the drawing area of the State Diagram. The Utilities background menu is displayed.



2. Choose ASSOCIATED DIAGRAMS->CLD_1_MEMBER INSTITUTIONS - SQD. The Sequence Diagram is displayed.

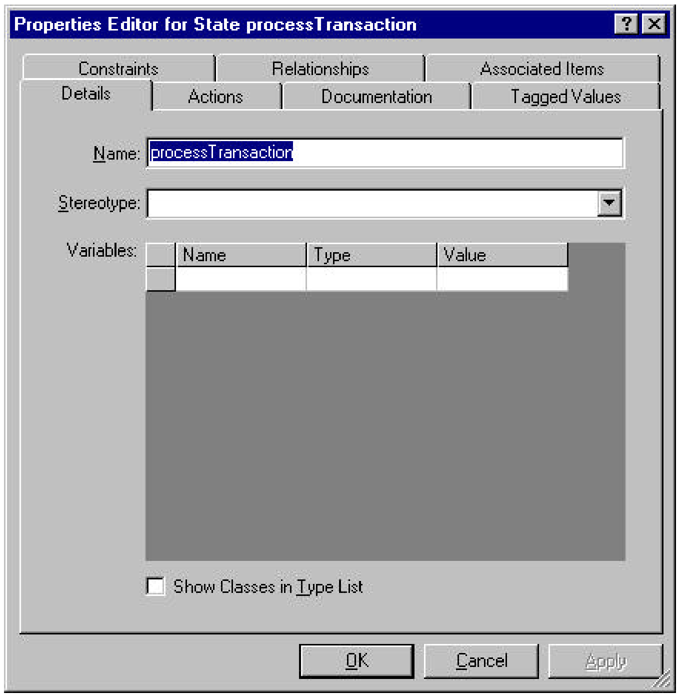
3. To return to the State diagram, right-click in the drawing area of the Sequence Diagram. The Utilities background menu is displayed.



4. Choose SYSTEM DIAGRAMS->STATE DIAGRAMS->USER - STD. The State Diagram opens.

Enter Variables

1. Double-click the "processTransaction" State symbol. The State Identifier Properties dialog box opens.



Name: processTransaction

Stereotype:

2. Double-click the text box located below Name in the Variables list box. Enter "validateTransaction" in the activated text box.

3. Leave the Properties Editor dialog box open so we can enter the action for this state.

Enter Actions

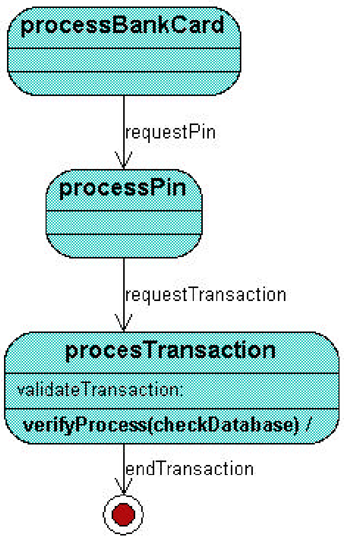
1. Click the Action tab and click the New button located on the right side of the dialog box.

2. Enter "verifyProcess" in the Name text box.

3. Double-click in the Arguments list box and enter "checkDatabase" in the activated text box. Click OK and the Detail dialog box closes.

4. Click OK once again and the Properties Editor dialog box closes.

   Your completed diagram should resemble the following illustration.


processBankCard
requestPin
processPin
requestTransaction
procesTransaction
validateTransaction:
verifyProcess(checkDatabase) /
endTransaction