



Quick Start Guide for Rational Rose

Version 2.2
June 16, 2008

Pathfinder Solutions LLC
33 Commercial Drive, Suite 2
Foxboro, MA 02035 USA
www.PathfinderMDA.com
888-662-7284

Table Of Contents

Model a Simple Oven	1
Task 1: Create a New Model in Rose	2
Task 2: Load Realized Domains.....	3
Task 3: Create the Application Domain.....	4
Task 4: Create Three Classes in MicrowaveCooking	5
Task 5: Create the Class Diagram	7
Task 6: Create the Door State Chart.....	9
Task 7: Create the Light State Chart.....	12
Task 8: Add Entry Actions to the State Charts	15
Transform and Build	18
Task 1: Prepare the Model for Transformation	19
Task 2: Transform Your Model	22
Task 3: Build an Executable System	28
Introduction to Spotlight	29
Task 1: Run SimpleOven with Spotlight.....	30
Task 2: Browse the Door Class	31
Task 3: Step Through Action Language	34
Task 4: Send an Event.....	35
Task 5: Animate State Transitions.....	36

Model a Simple Oven

Whether you have created hundreds of models and systems or none at all, you can follow this tutorial. Learn quickly how to use the PathMATE toolset with Rational Rose and Microsoft Visual C++ to develop an executable system. The model-building part of the tutorial includes these steps:

- Create the model.
- Load reused domains.
- Create the application domain, MicrowaveCooking.
- Create three classes in MicrowaveCooking.
- Create a class diagram.
- Create two state charts.
- Add entry actions to the state charts.

To preview the sample, click File > Open in Rational Rose and browse to *C:\pathmate\samples\SimpleOven\roseanalysis\SimpleOven.mdl*. Double-click *SimpleOven.mdl*. The Logical View for the model opens in Rational Rose.

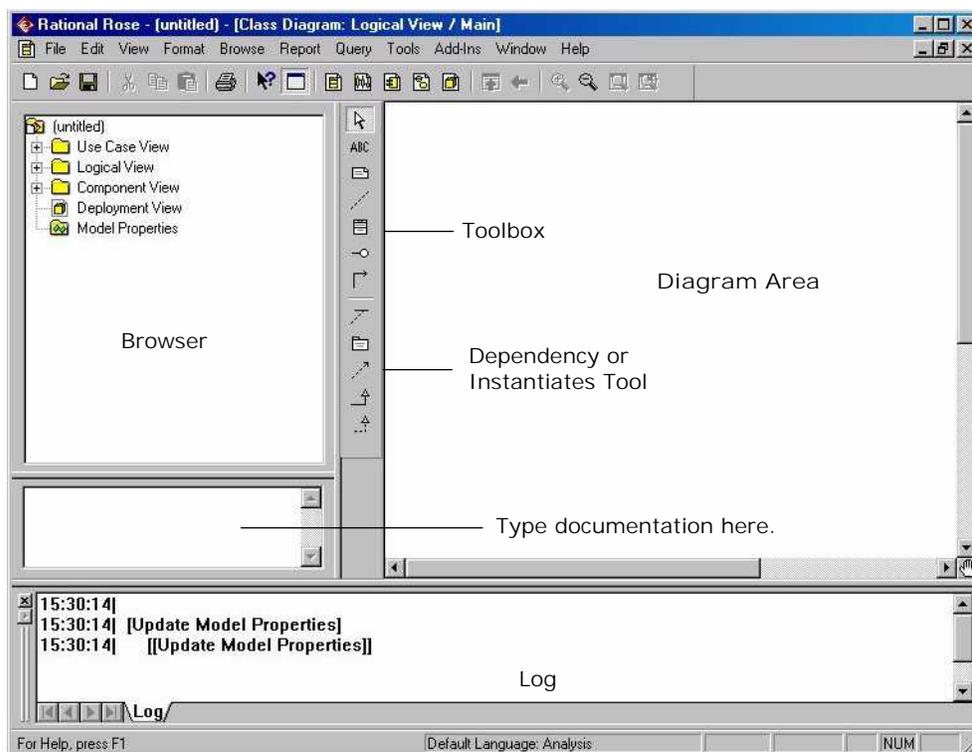
NOTE: The sample model shows you how the model should look when you are finished. As you build your own SimpleOven, the instructions occasionally ask you to use some building blocks supplied in the sample files.

Task 1: Create a New Model in Rose

To create your model, start with these steps:

1. In Windows Explorer, create this working directory for your model and all of its support files:
C:\pathmate\samples\QuickStart\rose
2. Create a subdirectory named *...\QuickStart\rose\analysis*.
3. Start Rational Rose.
4. Click Cancel in the Create New Model dialog, should it appear.

Rose opens a new model with a blank class diagram called Main contained in the Logical View. The Logical View contains your domain chart, the top level diagram in your model.



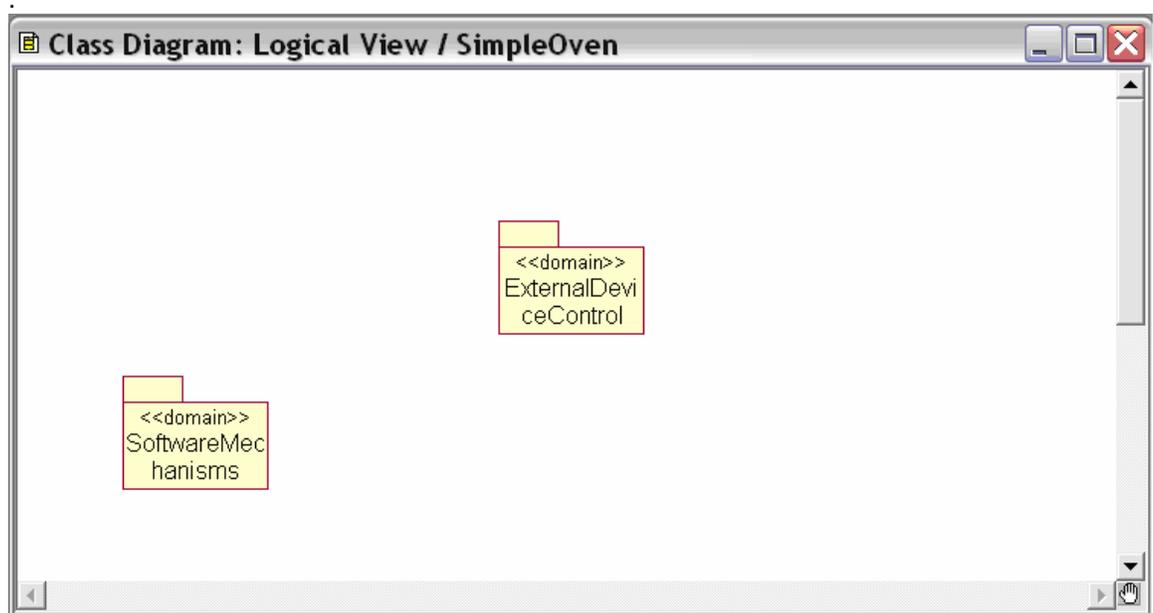
5. In the Rational Rose browser, right-click *Main* under Logical View and rename it *SimpleOven*.
6. Click File > Save As... and save the new model in *C:\pathmate\samples\QuickStart\rose\analysis*. Name the .mdl file *SimpleOven.mdl*.

Task 2: Load Realized Domains

SimpleOven reuses two realized domains to provide support services: SoftwareMechanisms and ExternalDeviceControl. Copy these domains to ...*QuickStart\roselanalysis*, and then import them into your model:

1. Locate *sw.cat* in *C:\pathmate\design\rose* and copy it to *C:\pathmate\samples\QuickStart\roselanalysis*.
2. In Rational Rose, click File > Units > Load... and select *C:\pathmate\samples\QuickStart\roselanalysis\sw.cat*. Click Open.
Software Mechanisms appears in the domain chart.
3. Drag SoftwareMechanisms down and to the left, away from the center of the domain chart. Click in the domain chart's open space to deselect the package.
4. Locate *ExternalDeviceControl.cat* in *C:\pathmate\samples\SimpleOven\roselanalysis* and copy it to *C:\pathmate\samples\QuickStart\roselanalysis*.
5. Click File > Units > Load... and select *C:\pathmate\samples\QuickStart\roselanalysis\ExternalDeviceControl.cat*. Click Open.
ExternalDeviceControl appears in the domain chart.

The figure below shows SoftwareMechanisms and ExternalDeviceControl properly placed on the domain chart

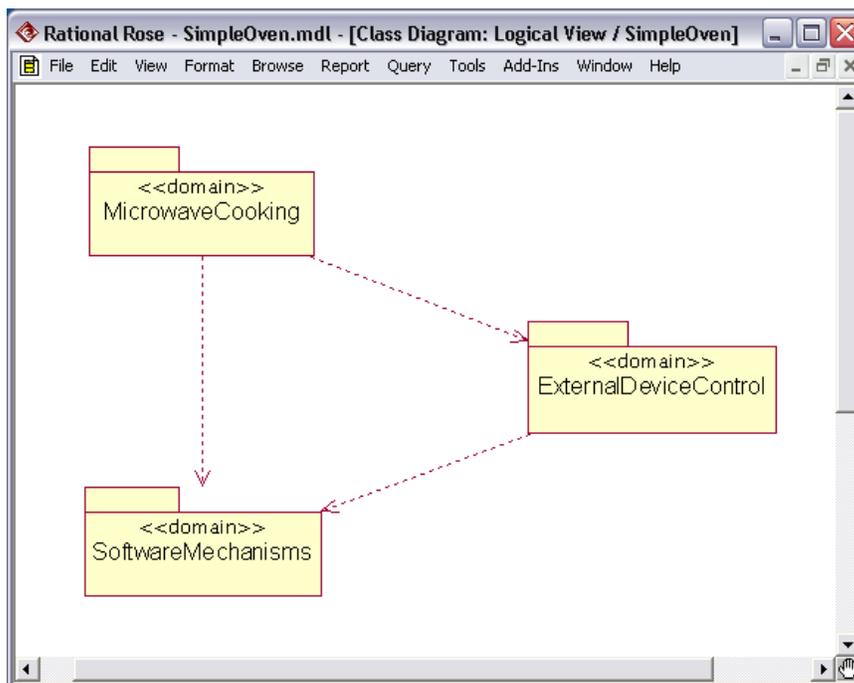


Task 3: Create the Application Domain

SimpleOven's application domain is called MicrowaveCooking. To create MicrowaveCooking:

1. Right-click Logical View in the browser and select New > Package in the pop-up menu.
<<>> *NewPackage* appears in the browser.
2. Name the new package <<domain>>*MicrowaveCooking* and press Enter. Should the PathMATE Rename dialog appear, check the box Do not show this dialog again and select OK to continue renaming.
3. Drag *MicrowaveCooking* from the browser to the domain chart.
The domain *MicrowaveCooking* appears in the domain chart.
4. Use the *Dependency or instantiates* tool in the toolbox to draw a line from MicrowaveCooking to SoftwareMechanisms.
5. In the same way, connect MicrowaveCooking to ExternalDeviceControl, and ExternalDeviceControl to SoftwareMechanisms.

The figure below shows the domain chart after you complete these steps.



Task 4: Create Three Classes in MicrowaveCooking

To create the classes in MicrowaveCooking:

1. Right-click <<domain>>*MicrowaveCooking* in the browser and select New > Class in the pop-up menu.
2. Type *Oven* in place of <<>> *NewClass* in the browser and press Enter.
3. Right-click <<domain>>*MicrowaveCooking* in the browser and select New > Class in the menu. Name the new class *Door*.
4. Right-click <<domain>>*MicrowaveCooking* a third time and select New > Class in the menu. Name the new class *Light*.

Add the attribute *dateOfManufacture* to the Oven class:

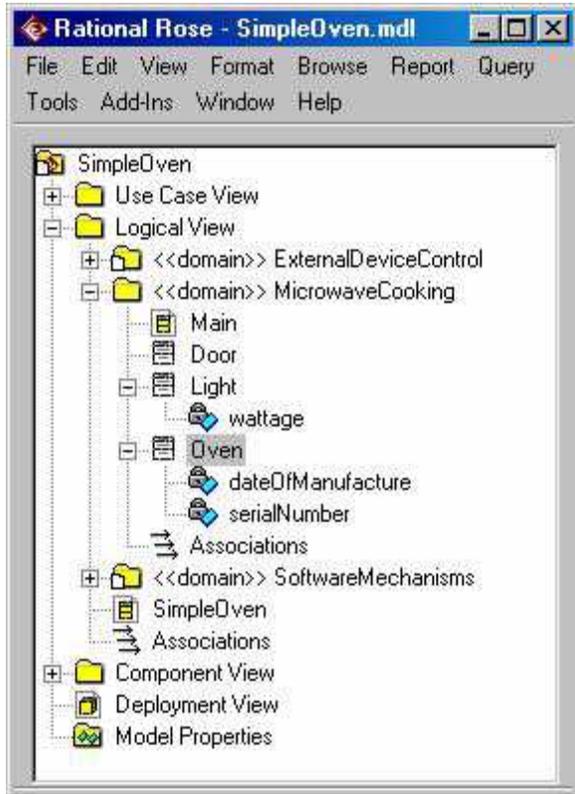
1. Right-click *Oven* in the browser and select New > Attribute in the pop-up menu.
2. Name the new attribute *dateOfManufacture* in the browser.
3. Right-click *dateOfManufacture* and select Open Specification... in the pop-up menu.
4. Select *String* in the *Type* drop-down list.
5. Click OK to close the Specification box.

Follow the same steps to add the attribute *serialNumber* to the Oven class (the data type for *serialNumber* is also *String*).

Add the attribute *wattage* to the Light class:

1. Right-click *Light* in the browser and select New > Attribute in the pop-up menu.
2. Name the new attribute *wattage* in the browser.
3. Right-click *wattage* and select Open Specification... in the pop-up menu.
4. Select *Integer* in the *Type* drop-down list.
5. Type *60* in the *Initial value* field.
6. Click OK to close the Specification box.

The figure below shows the Rational Rose browser after you create the Oven, Door and Light classes, and after you add attributes.



Task 5: Create the Class Diagram

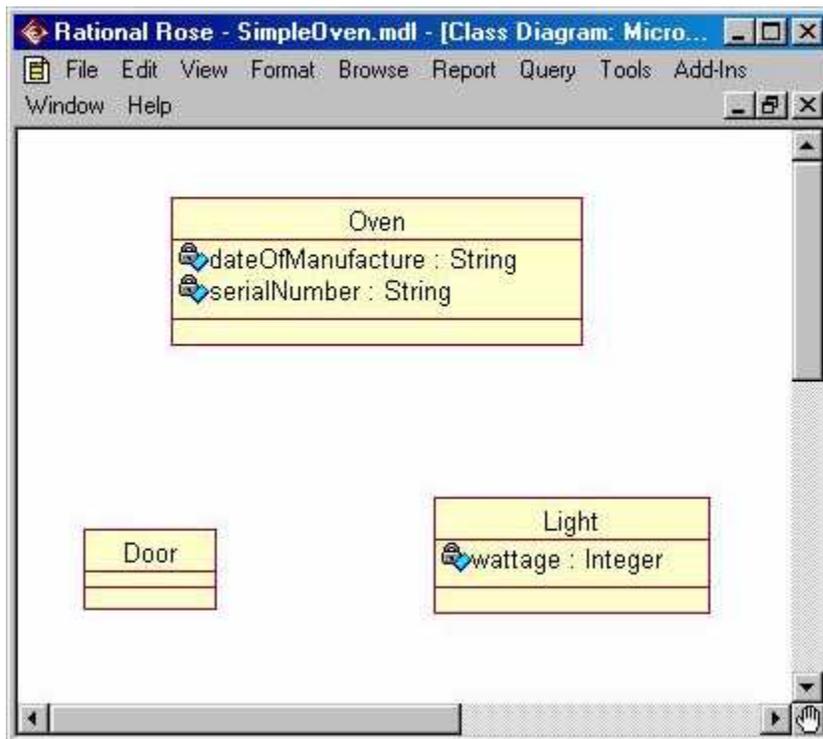
To create the class diagram in the MicrowaveCooking domain:

1. Right-click *MicrowaveCooking* in the browser and select New > Class Diagram in the pop-up menu.
2. Name the class diagram *MicrowaveCooking* in the browser.
3. Double-click the MicrowaveCooking class diagram in the MicrowaveCooking domain.

A blank diagram for MicrowaveCooking appears.

4. Select the Oven class in the browser and drag it to the class diagram.
5. Likewise select the Door class and drag it to the open space in the diagram.
6. Repeat the drag and drop procedure for the Light class.

The figure below shows the class diagram after you complete these steps.



Now associate the Oven class with the Door class:

1. Click Tools > Create > Association in the top menu bar.
2. Click *Oven* in the class diagram and draw a rectilinear line to the Door class.
3. Right-click the line and select Open Specification... from the pop-up menu.
The Association Specification box opens.
4. Enter *A1* in the *Name* field of the Specification box.

5. Enter *is_part_of* in the *Role A* field of the Specification box.
6. Click OK to close the Specification box.
7. Right-click the upper part of the association line and select Multiplicity in the pop-up menu. Select 1 in the sub-menu.

A multiplicity of 1 appears near the upper part of association A1.

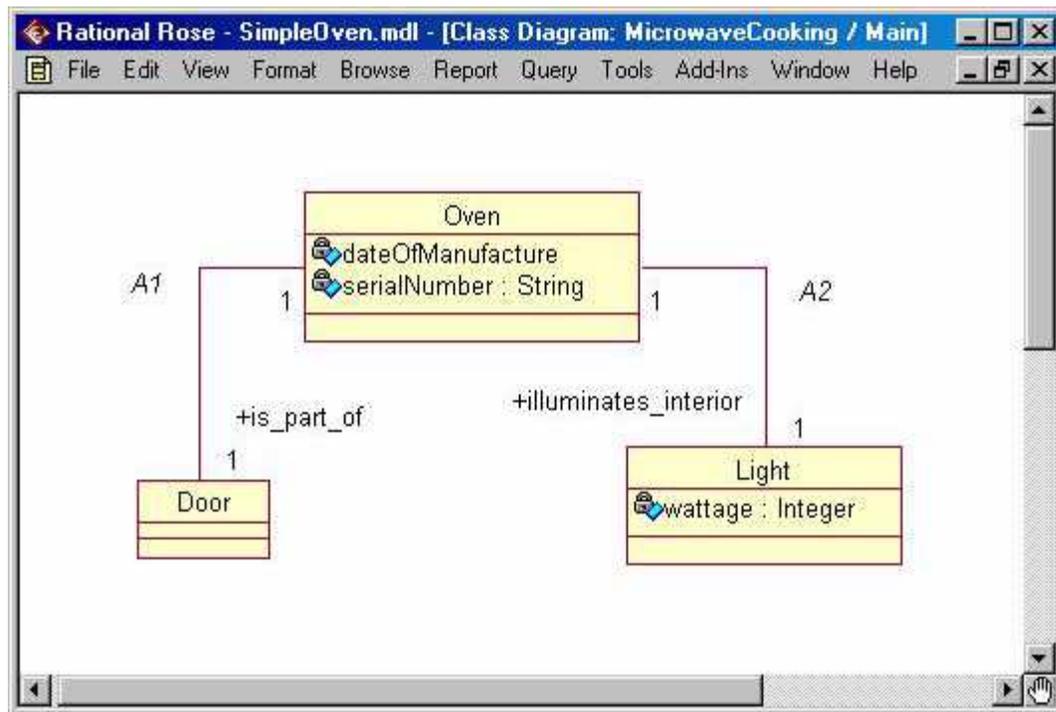
8. Right-click the lower part of the association line and select Multiplicity in the pop-up menu. Then select 1 in the sub-menu.

A multiplicity of 1 appears near the lower part of association A1.

Follow the same steps to associate the Oven class with the Light class in the diagram:

1. Draw the association line.
2. Open the Specification box for the association.
3. Type *A2* in the *Name* field of the specification box. Type *illuminates_interior* in the *Role A* field.
4. Click OK to close the Specification box.
5. Specify a multiplicity of 1 for each end of association A2.

The figure below shows the class diagram after you complete these steps.



Task 6: Create the Door State Chart

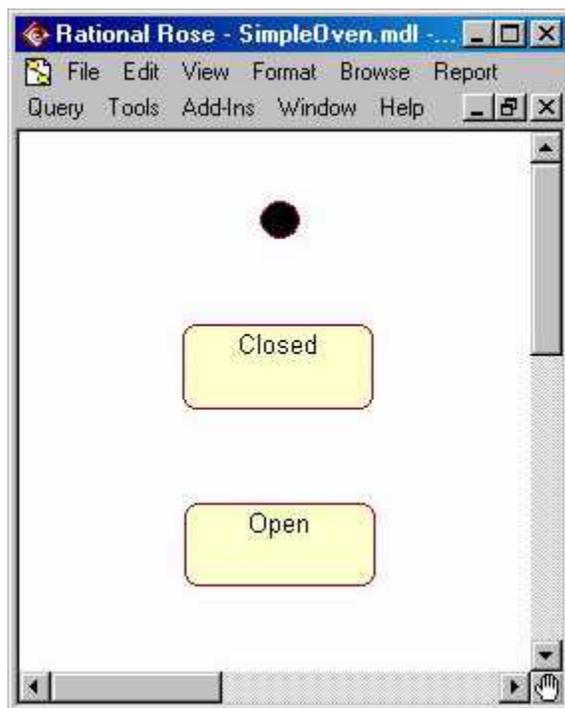
Create a state chart for the oven door to show transitions between the closed state and the open state.

1. In the Rose browser, right-click the Door class in the MicrowaveCooking domain. Select New > Statechart Diagram in the pop-up menu.
A new State/Activity Model appears under the Door class in the browser.
2. Name the new diagram in the State/Activity Model *Door* and press Enter.
3. Double-click the Door state chart in the browser to open the blank state chart.

To place the state symbols in the new state chart, follow these steps:

1. Click Tools > Create > Start State and place the Start State in the state chart.
2. Click Tools > Create > State and place the New State symbol in the diagram below the Start State symbol.
3. Type *Closed* in the state symbol and click the open space to create the new state.
4. Click Tools > Create > State in the top menu bar and place the New State symbol in the diagram below the Closed State symbol.
5. Type *Open* in the state symbol and click the open space to create the new state.

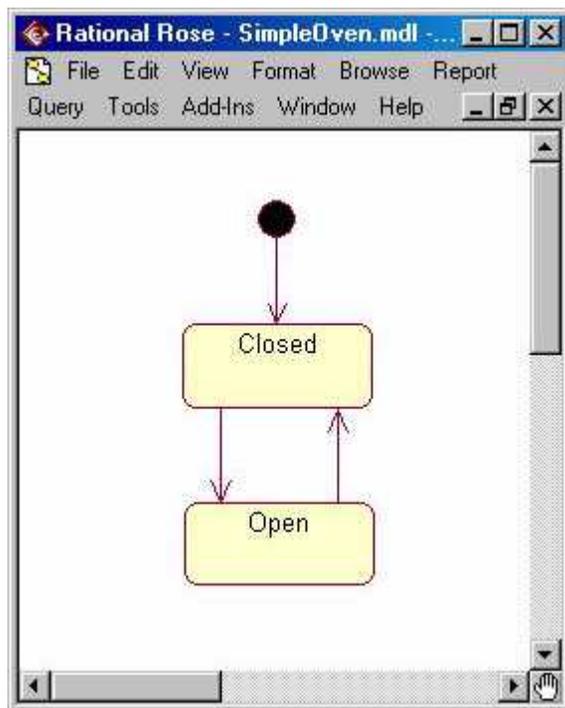
The figure below shows the Door state chart after you create the Start state, the Closed state, and the Open state.



To create the transitions:

1. Click Tools > Create > Transition in the top menu bar.
2. Click the Start State symbol and drag a transition line to *Closed*.
3. Click Tools > Create > Transition and drag a new transition line from *Closed* to *Open*.
4. With the new transition line selected, click Format > Line Style > Rectilinear in the top menu bar.
5. Drag the line to the left side of the states.
6. Click Tools > Create > Transition and draw a second transition line from *Open* to *Closed*.
7. Click Format > Line Style > Rectilinear and drag the line to the right side of the states.

The figure below shows the Door state chart after you complete these steps.



To specify the event associated with each transition:

1. Right-click the line from *Closed* to *Open* and select Open Specification... in the pop-up menu.
2. Type *IsOpen* in the *Event* field of the Specification box and click OK to close the box.

The event name *IsOpen* appears in the Door state chart.

3. Right-click the line from *Open* to *Closed* and select Open Specification... in the pop-up menu.

4. Type *IsClosed* in the *Event* field of the Specification box and click OK to close the box.

The event name *IsClosed* appears in the Door state chart.

To add the PathMATE events *IsOpen* and *IsClosed* to the Door class:

1. Open the Door state chart.
2. Right-click the transition line for *IsOpen* and select PathMATE Open Event Spec in the pop-up menu.

The Operation Specification for *IsOpen* appears. Click OK to close the box.

3. Right-click the transition line for *IsClosed* and select PathMATE Open Event Spec in the pop-up menu.

The Operation Specification for *IsClosed* appears. Click OK to close the box.

NOTE: Select the transition line, not the label, when you right-click to open the pop-up menu that contains PathMATE Open Event Spec.

Task 7: Create the Light State Chart

To create the Light state chart, repeat the same procedure you used to create the Door state chart:

1. Right-click the Light class in the browser and select New > Statechart Diagram in the pop-up menu.
2. Name the diagram *Light* in the browser.
3. Double-click *Light* in the browser to open the new state chart.

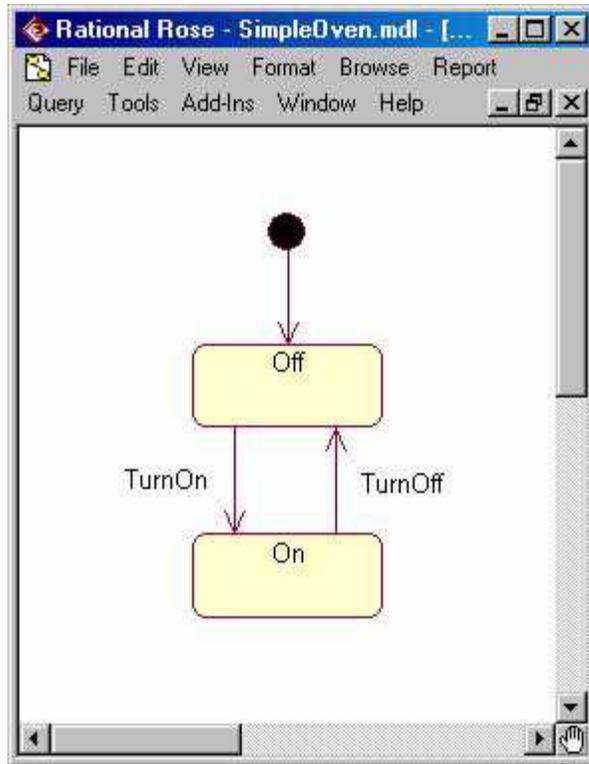
Next create the states in the diagram:

1. Click Tools > Create > Start State, and place the Start State in the state chart.
2. Click Tools > Create > State. Place the new state in the diagram and name it *Off*.
3. Click Tools > Create > State. Place the new state in the diagram and name it *On*.

Then create the transitions:

1. Click Tools > Create > Transition. Draw a line from the Start State to *Off*.
2. Click Tools > Create > Transition, and draw a line from *Off* to *On*. Click Format > Line Style > Rectilinear, and drag the line to the left side of the diagram.
3. Right-click the line from *Off* to *On* and select Open Specification... in the pop-up menu.
4. Type *TurnOn* in the *Event* field of the Specification box and click OK to close the box.
5. Click Tools > Create > Transition, and draw a line from *On* to *Off*. Click Format > Line Style > Rectilinear, and drag the line to the right side of the diagram.
6. Right-click the line from *On* to *Off* and select Open Specification... in the pop-up menu.
7. Type *TurnOff* in the *Event* field of the Specification box and click OK to close the box.

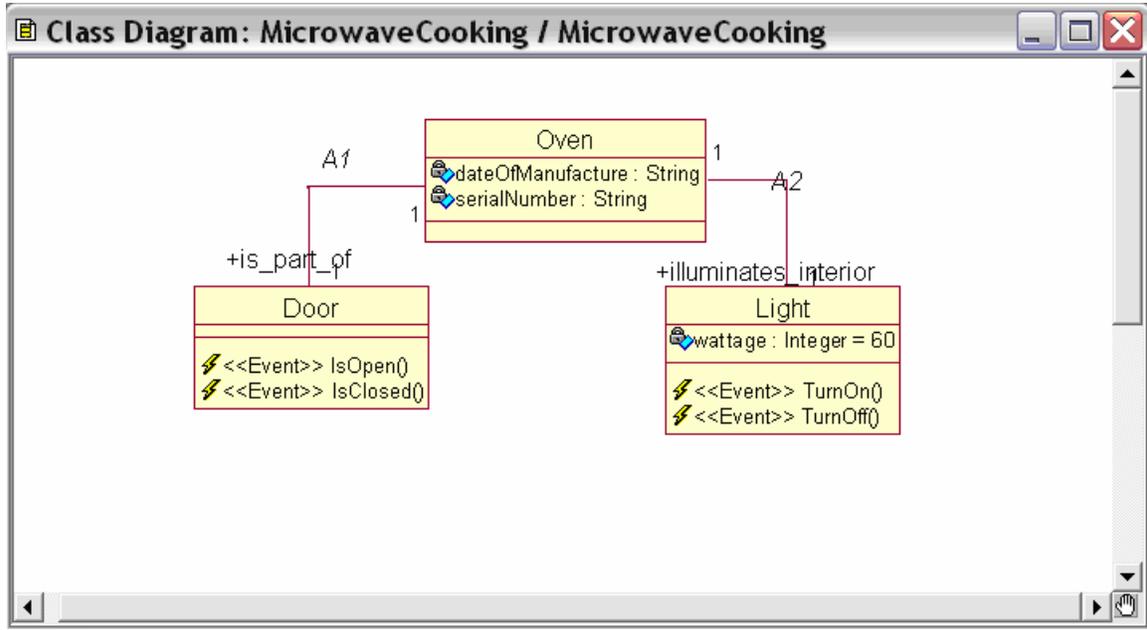
The figure below shows the Light state chart after you complete these steps.



To add the PathMATE events *TurnOn* and *TurnOff* to the Light class:

1. Open the state chart for the Light class.
2. Right-click the transition line for *TurnOn* and select PathMATE Open Event Spec in the pop-up menu.
3. The Operation Specification for *TurnOn* appears. Click OK to close the box.
4. Right-click the transition line for *TurnOff* and select PathMATE Open Event Spec in the pop-up menu.
5. The Operation Specification for *TurnOff* appears. Click OK to close the box.

The figure below shows the class diagram after you complete these steps.



Task 8: Add Entry Actions to the State Charts

Entry actions establish a relationship between the Door state and the Light state. When the door closes, the light turns off. When the door opens, the light turns on.

The procedures below require that you copy lines of text and paste them in several action language files:

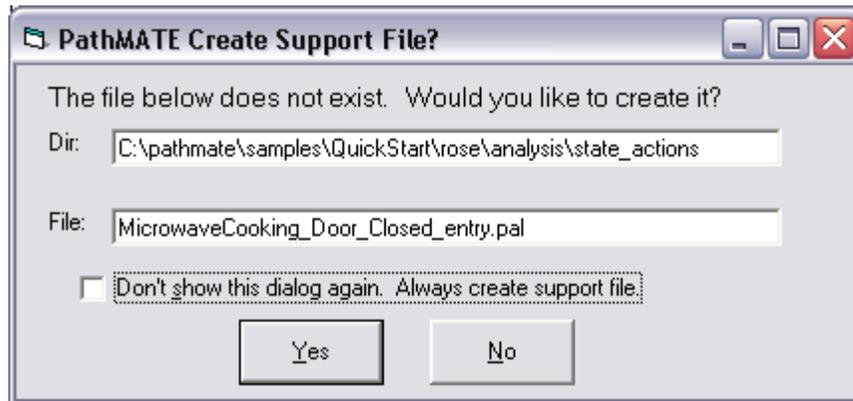
- In Adobe Acrobat, click the Select Text tool in the Acrobat toolbar. Then select the text, copy it to your clipboard, and paste it into the action language file.

Whenever you copy and paste text, make sure the line breaks and other formatting in your text editor match the appearance of the text in the *Quick Start Guide*.

To create the Entry Action for *Closed* in the Door State Chart:

1. Right-click *Closed* in the Door state chart and select PathMATE Open > Entry Action.

A confirmation dialog shows the directory and the name of the support file.



To bypass the dialog, click *Don't show this dialog again. Always create support file.*

2. Click Yes to create *MicrowaveCooking_Door_Closed_entry.pal*.

The .pal file opens in your text editor.

3. Copy the text below and paste it at the end of the open .pal file.

```
Ref<Light> interior_light = FIND this->A1->A2;
GENERATE Light:TurnOff() TO (interior_light);
```

4. Save and close the .pal file.
5. Right-click *Closed* in the Door state chart and select PathMATE Update Actions in the pop-up menu.

The entry action appears in the *Closed* state.

To create the entry action for *Open* in the Door state chart:

1. Right-click *Open* in the Door state chart and select PathMATE Open > Entry Action.

A confirmation dialog shows the directory and the name of the support file.

2. Click Yes to create *MicrowaveCooking_Door_Open_entry.pal*.

The .pal file opens in your text editor.

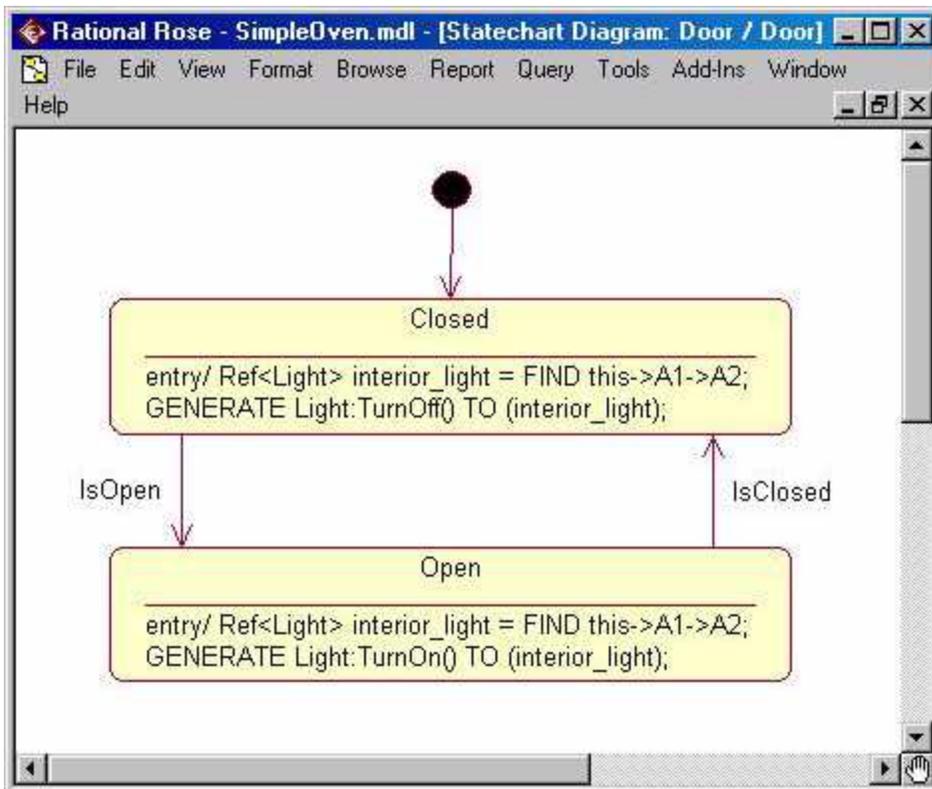
3. Copy the text below and paste it at the end of the open .pal file.

```
Ref<Light> interior_light = FIND this->A1->A2;
GENERATE Light:TurnOn() TO (interior_light);
```

4. Save and close the .pal file.
5. Right-click *Open* in the diagram and select PathMATE Update Actions in the pop-up menu.

The entry action appears in the Open state.

Readjust the lines and labels in the Door state chart. The figure below shows the Door state chart after you create the entry actions.



To create the entry action for *Off* in the Light state chart:

1. Right-click *Off* in the Light state chart and select PathMATE Open > Entry Action.

A confirmation dialog shows the directory and the name of the support file.

2. Click Yes to create *MicrowaveCooking_Light_Off_entry.pal*.

The .pal file opens in your text editor.

3. Copy the text below and paste it at the end of the open .pal file.

```
ExternalDeviceControl:DeactivateDevice(SYS_DEVICE_LIGHT);
```

4. Save and close the .pal file.

5. Right-click *Off* in the diagram and select PathMATE Update Actions in the pop-up menu.

The entry action appears in the *Off* state.

To create the entry action for *On* in the Light state chart:

1. Right-click *On* in the Light state chart and select PathMATE Open > Entry Action.

A confirmation dialog shows the directory and the name of the support file.

2. Click Yes to create *MicrowaveCooking_Light_On_entry.pal*.

The .pal file opens in your text editor.

3. Copy the text below and paste it at the end of the open .pal file.

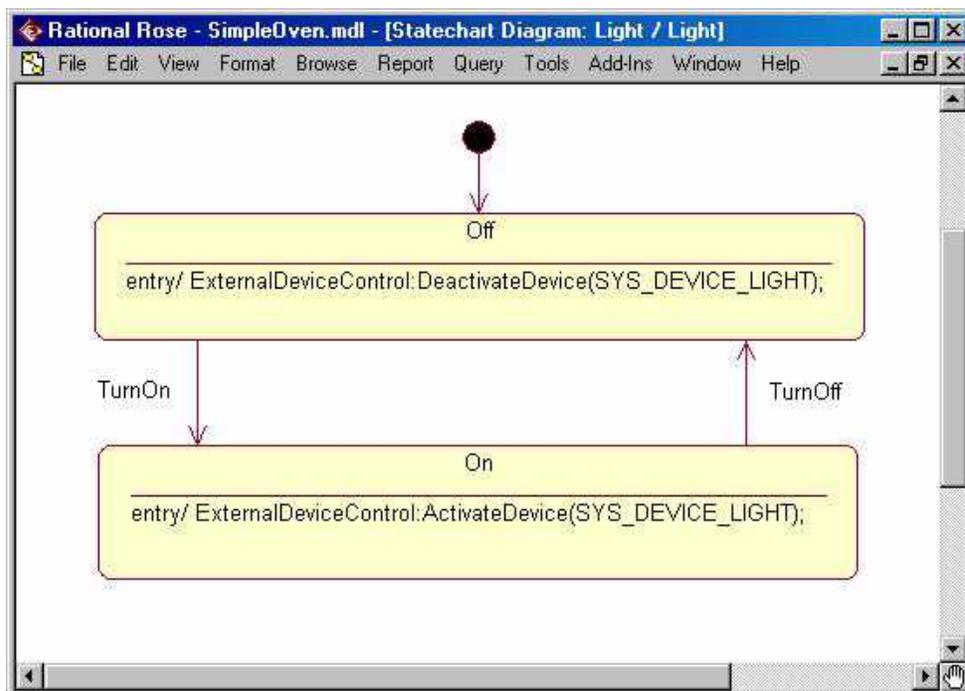
```
ExternalDeviceControl:ActivateDevice(SYS_DEVICE_LIGHT);
```

4. Save and close the .pal file.

5. Right-click *On* in the diagram and select PathMATE Update Actions in the pop-up menu.

The entry action appears in the *On* state.

Readjust the lines and labels in the Light state chart. The figure below shows the Light state chart after you create the entry actions.



Transform and Build

Now you are ready to transform your model into compilable code. This section takes you through the following tasks:

- *Prepare the Model for Transformation*
- *Transform Your Model*
- *Build an Executable System*

Task 1: Prepare the Model for Transformation

To prepare the model for transformation, complete these tasks:

- Initialize MicrowaveCooking.
- Create user defined types.
- Enable Spotlight.

Initialize MicrowaveCooking

To initialize the MicrowaveCooking domain:

1. Right-click MicrowaveCooking in the domain chart.
2. Select PathMATE Open > Domain Init in the pop-up menu.
A confirmation dialog shows the directory and the name of the support file.
3. Click Yes to create *MicrowaveCooking.pal* in ...*analysis\init*.
The .pal file opens in your text editor.
4. Copy the text below and paste it at the end of *MicrowaveCooking.pal*:

```
// Set up instances for MicrowaveCooking
Ref<Oven> mw_oven = CREATE Oven (serialNumber = "G023-4ZZ-8811",
dateOfManufacture = "2004/02/22; 10:41");
Ref<Light> interior_light = CREATE Light();
Ref<Door> door = CREATE Door();

LINK mw_oven A1 door;
LINK mw_oven A2 interior_light;

// Now just to start things off, open the door
GENERATE Door:IsOpen() TO (door);
```

NOTE: Take care when you copy the action language above to *MicrowaveCooking.pal*. The line breaks and other formatting should appear in your text editor just as they do on this page.

5. Save and close *MicrowaveCooking.pal*.

Create a User Defined Type

To create user a defined type:

1. Right-click in the open space of the domain chart, or in the open space of any model diagram.
2. Select PathMATE Open > System > Types in the pop-up menu.
A confirmation dialog shows the directory and the name of the support file.
3. Click Yes to create *SimpleOven.typ* in ...*analysis\types*.
The .typ file opens in your text editor.

- Copy the text below and paste it at the end of *SimpleOven.typ*:

```
ENUM sys_device_e
{
    SYS_DEVICE_LIGHT
}
```

- Save and close the .typ file.

Enable Spotlight

To enable Spotlight, PathMATE's model-level debugger:

- Right-click MicrowaveCooking in the domain chart and select Open Specification... in the pop-up menu.

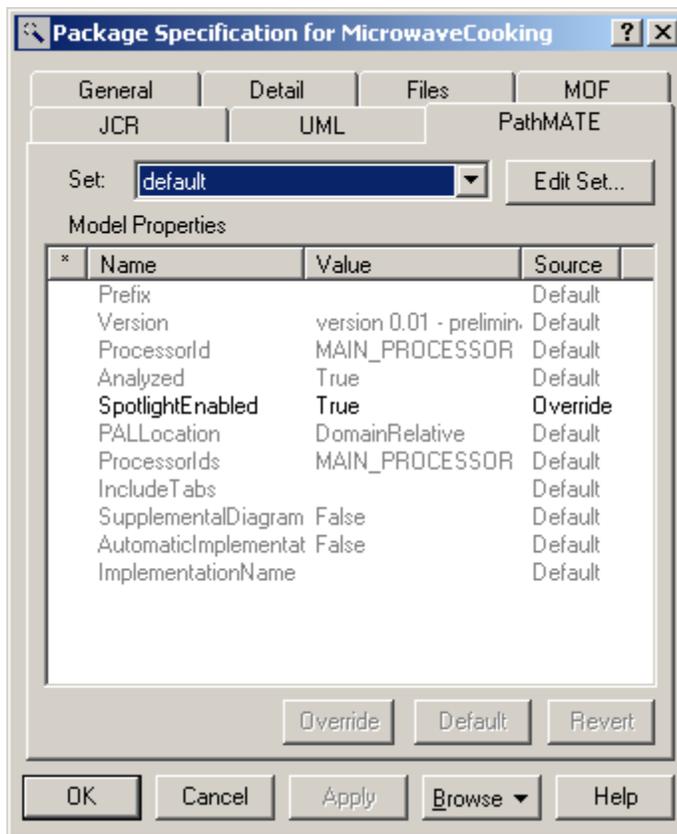
The Package Specification for MicrowaveCooking opens.

- Select the PathMATE tab in the Specification box.
- Select *SpotlightEnabled* in the *Model Properties* field.
- Select *SpotlightEnabled* a second time.

A drop-down list appears.

- Select *True* from the drop-down list. Then click outside the list to close it.

The *SpotlightEnabled* setting in Model Properties is *True*. The Source is *Override*.



- Click OK to close the Package Specification box.

Copy Files from the Sample System

Before you build SimpleOven, copy two files from the SimpleOven sample system to the QuickStart directory:

- *properties.txt* – contains markings used to set up the Visual C++ project files
- *ExternalDeviceControl_services_realized.cpp* – provides domain services to the application domain, MicrowaveCooking

The table below shows where to find each file in the SimpleOven sample system, and where to place the file in your own QuickStart directory structure. To see an overview of the entire directory structure, see *Simple Oven Directory Structure*.

File name	File Location in the SimpleOven Sample System	Copy the File to This Location in the QuickStart System
properties.txt	C:\pathmate\samples\SimpleOven\rose\project\cpp\properties.txt	C:\pathmate\samples\QuickStart\rose\project\cpp\properties.txt
ExternalDeviceControl_services_realized.cpp	C:\pathmate\samples\SimpleOven\rose\project\cpp\realized_cpp\ExternalDeviceControl_services_realized.cpp	C:\pathmate\samples\QuickStart\rose\project\cpp\realized_cpp\ExternalDeviceControl_services_realized.cpp

Create the directory *c:\pathmate\sampes\QuickStart\rose\project\cpp*. Copy *properties.txt* to the new directory. To import the realized code in *ExternalDeviceControl_services_realized.cpp*, copy both the file and the directory *realized_cpp* to *...\project\cpp*, as indicated in the table.

Task 2: Transform Your Model

To generate C++ code for the first time:

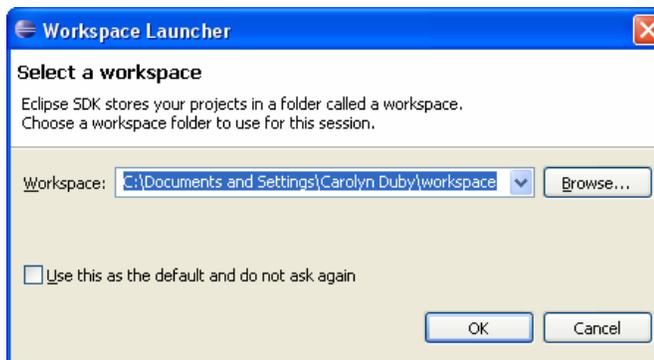
1. Click Tools > PathMATE Transformation Options in Rational Rose.

The PathMATE Extract box opens briefly as the models are exported via XML. Then the system prompts you to start the Eclipse environment.



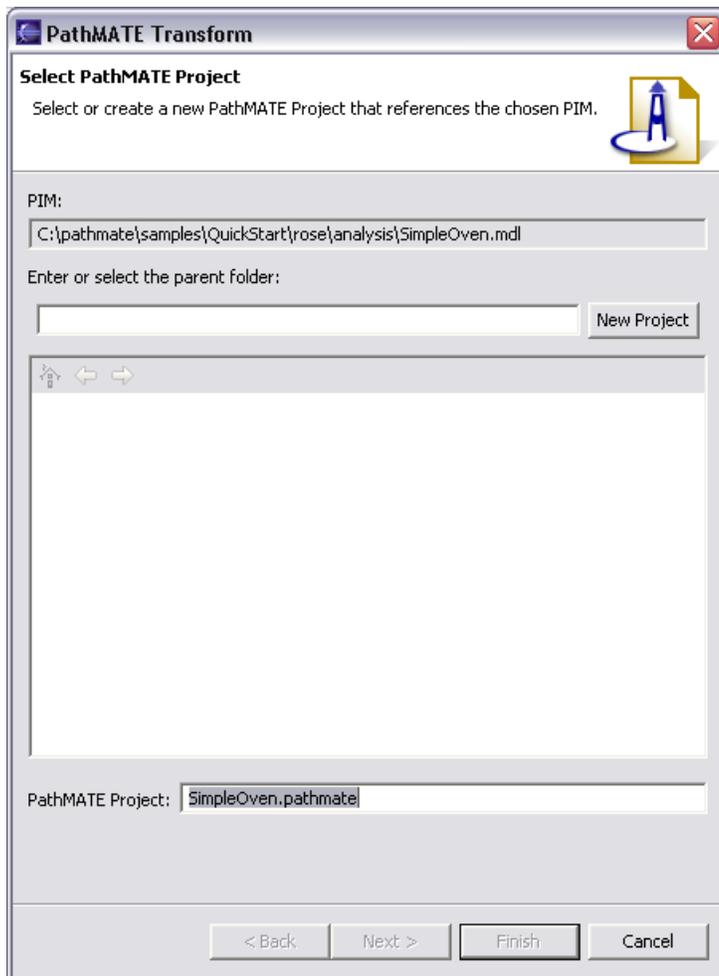
2. Click Yes.

The Eclipse environment launches and prompts you to select a workspace.

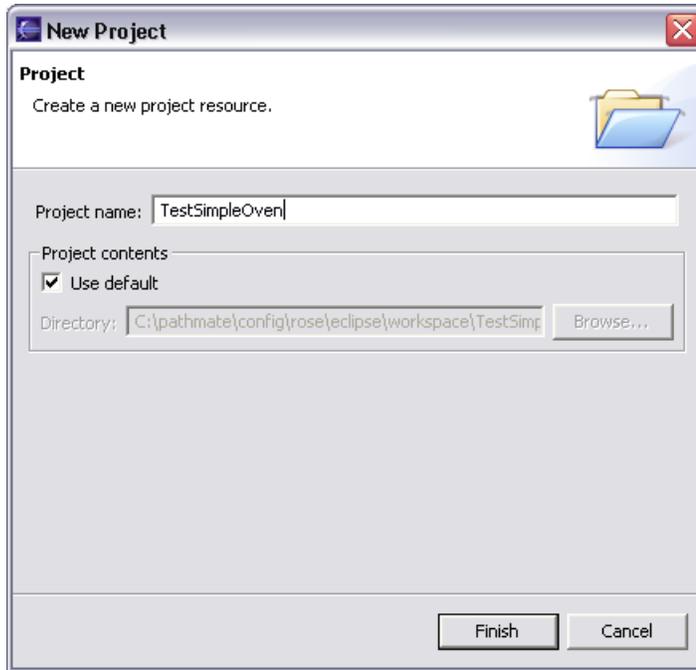


3. Click OK to select the default workspace.

Eclipse opens with the PathMATE Transform dialog active.

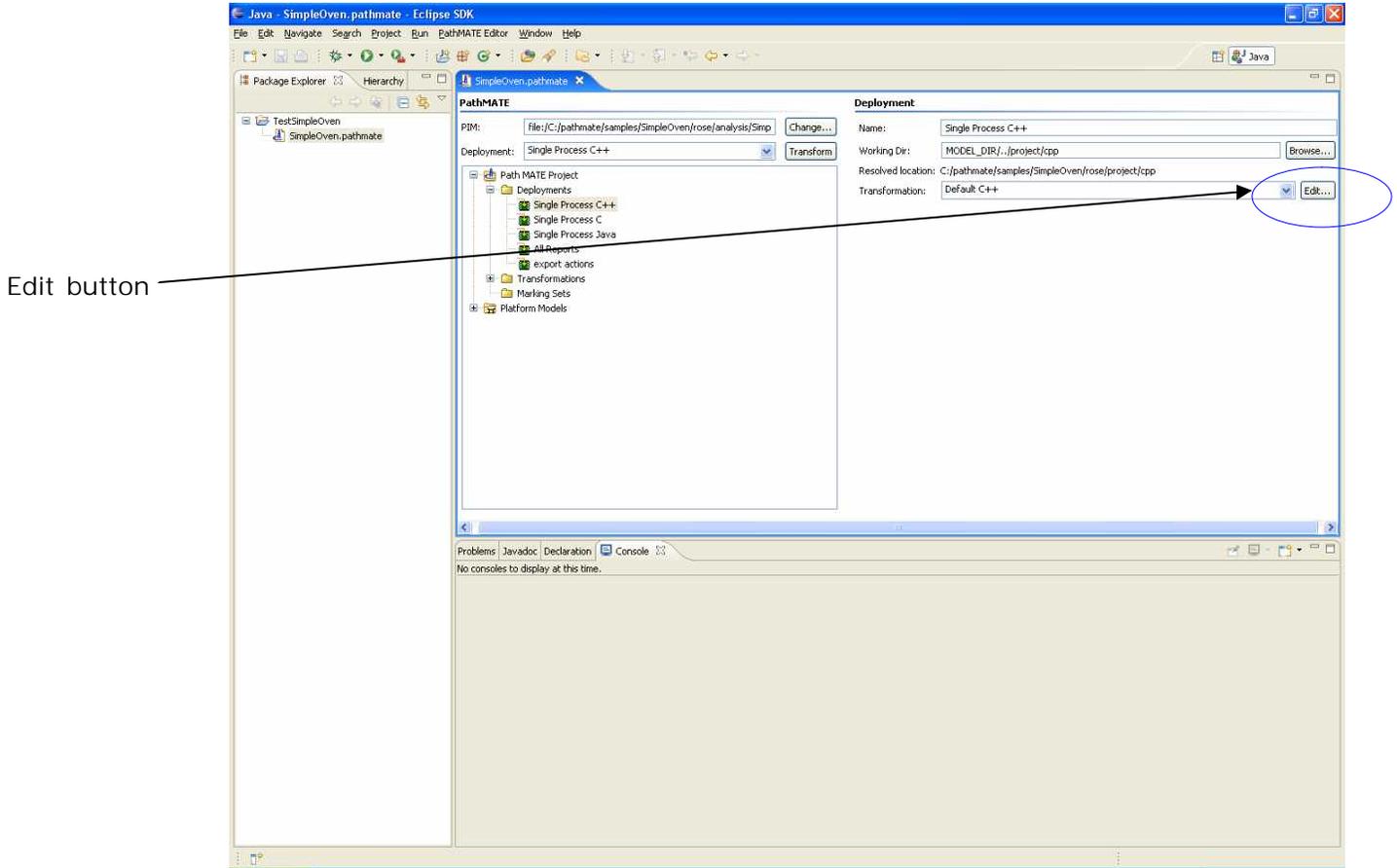


4. Click New Project in the dialog box.
The New Project dialog appears.
5. Type *TestSimpleOven* in the *Project name* field.



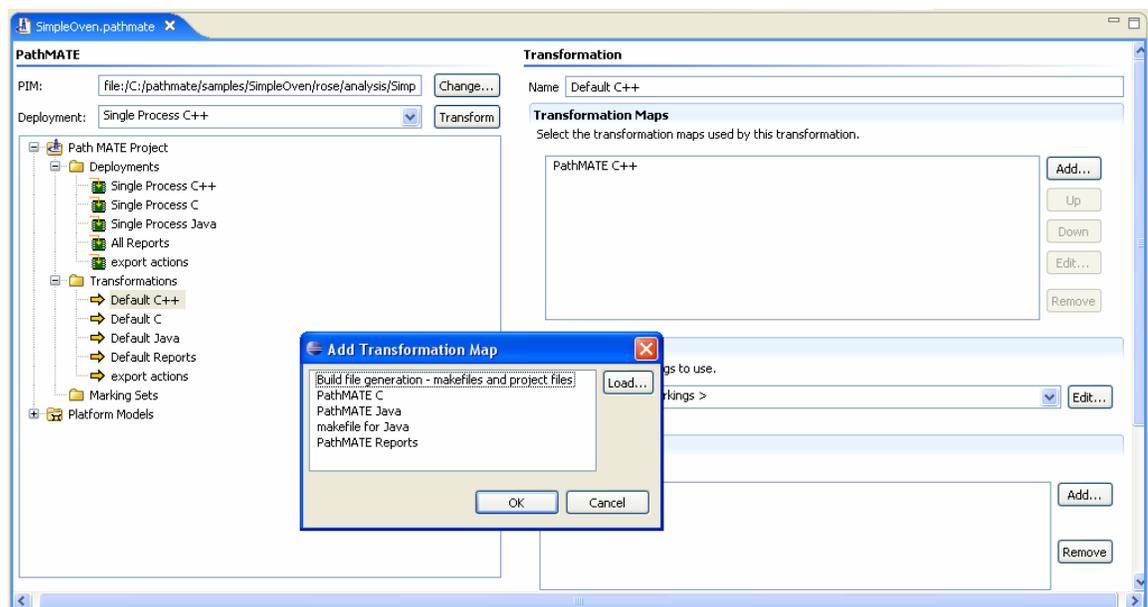
6. Click Finish in the New Project dialog.
7. Click Next in the PathMATE Transform dialog.
8. Select *Single Process C++* in the *Available Deployments* list.
9. Click Finish.

The PathMATE editor appears in Eclipse. The Single Process C++ Deployment is selected and the details of this deployment appear on the right side of the editor pane.



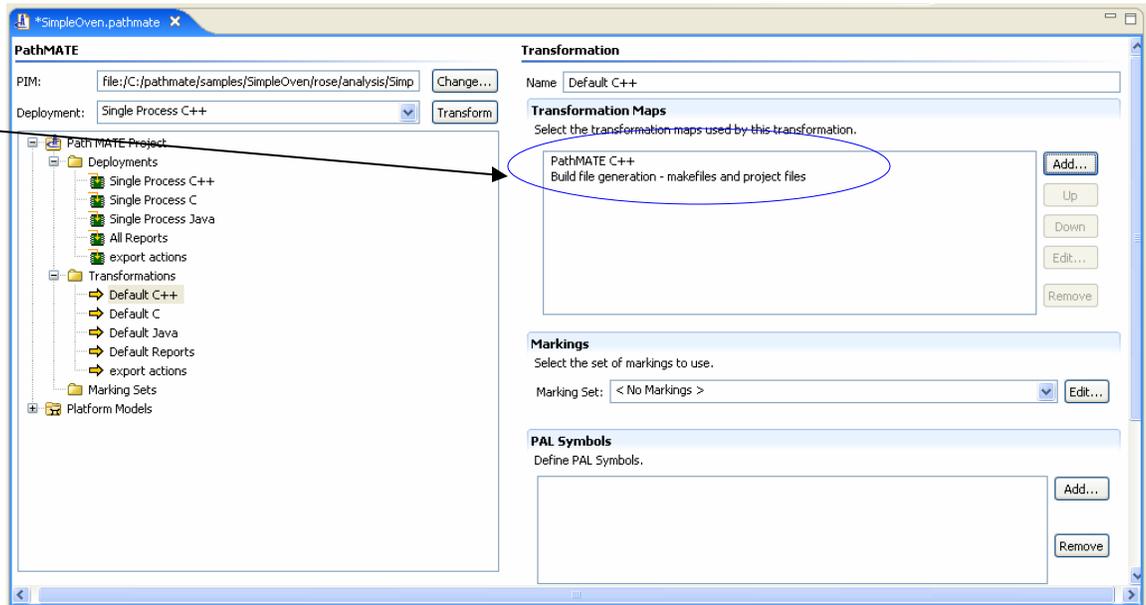
Edit button

10. Click the Edit... button next to the Transformation drop down. The Default C++ Transformation opens.
11. Click Add. The Add Transformation Map dialog appears.

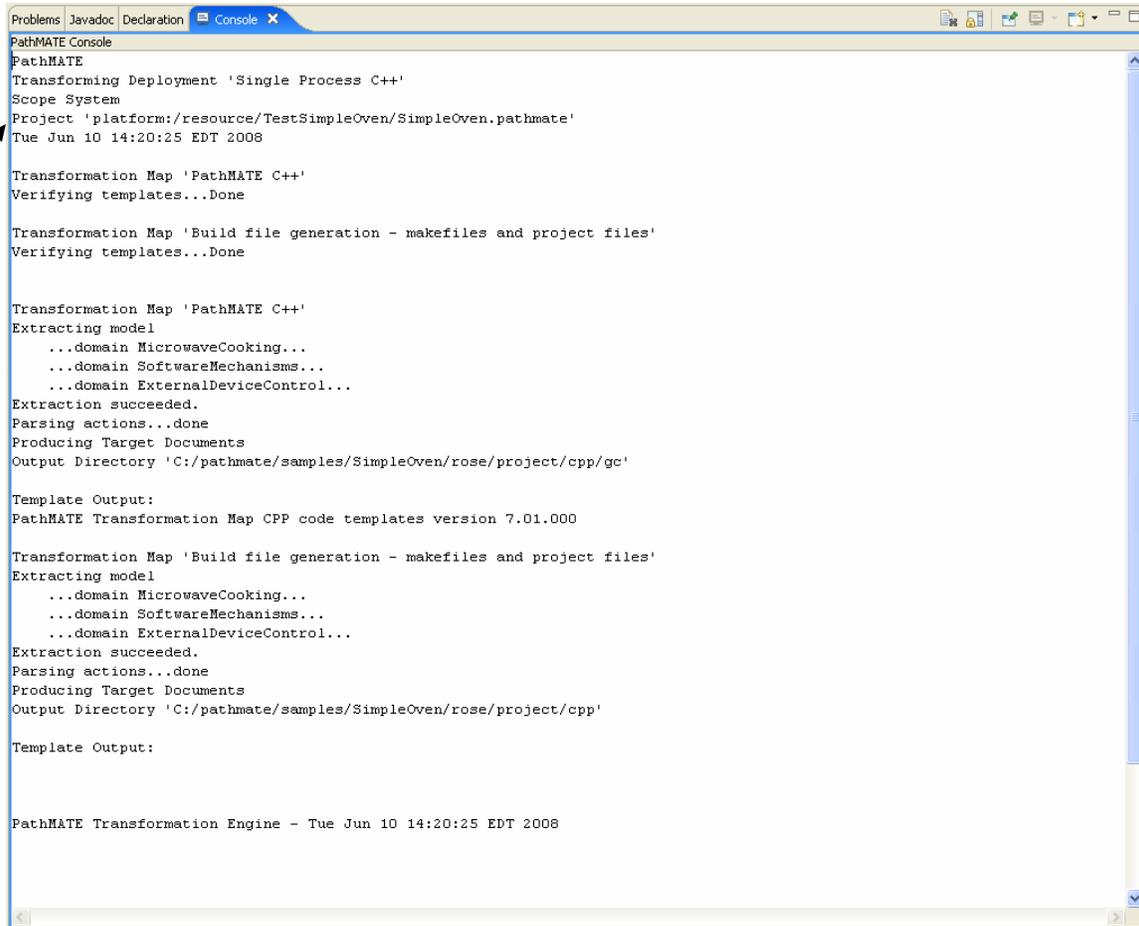


12. Select *Build file generation – makefiles and project files* from the list of Transformation Maps.
13. Click OK. The *Build file generation – makefiles and project files* appears in the Transformation Maps section. It should appear in the list below PathMATE C++. If it is above, select *Build file generation – makefiles and project files* and click the Down arrow to move it to the bottom of the Transformation Maps list.

Build...
comes last



14. Select File > Save from the main menu to save the SimpleOven.pathmate file.
15. SimpleOven is now ready to be transformed into C++ code. Return to Rose and select Tools > PathMATE Transform from the main menu. A dialog appears briefly while the model is exported. Eclipse activates and a progress dialog appears while the model is transforming. The results of the transformation appear in the Console view.



```
PathMATE Console
PathMATE
Transforming Deployment 'Single Process C++'
Scope System
Project 'platform:/resource/TestSimpleOven/SimpleOven.pathmate'
Tue Jun 10 14:20:25 EDT 2008

Transformation Map 'PathMATE C++'
Verifying templates...Done

Transformation Map 'Build file generation - makefiles and project files'
Verifying templates...Done

Transformation Map 'PathMATE C++'
Extracting model
  ...domain MicrowaveCooking...
  ...domain SoftwareMechanisms...
  ...domain ExternalDeviceControl...
Extraction succeeded.
Parsing actions...done
Producing Target Documents
Output Directory 'C:/pathmate/samples/SimpleOven/rose/project/cpp/gc'

Template Output:
PathMATE Transformation Map CPP code templates version 7.01.000

Transformation Map 'Build file generation - makefiles and project files'
Extracting model
  ...domain MicrowaveCooking...
  ...domain SoftwareMechanisms...
  ...domain ExternalDeviceControl...
Extraction succeeded.
Parsing actions...done
Producing Target Documents
Output Directory 'C:/pathmate/samples/SimpleOven/rose/project/cpp'

Template Output:

PathMATE Transformation Engine - Tue Jun 10 14:20:25 EDT 2008
```

Transformation
Results

16. Verify in the Console tab that both transformations were successful. If any problems arise they are probably due to the properties.txt file or syntax errors in action language. Correct any errors and transform again by selecting Tools > PathMATE Transform from Rose.

Task 3: Build an Executable System

To build an executable system, compile the code in Visual C++.

Build SimpleOven in Visual C++

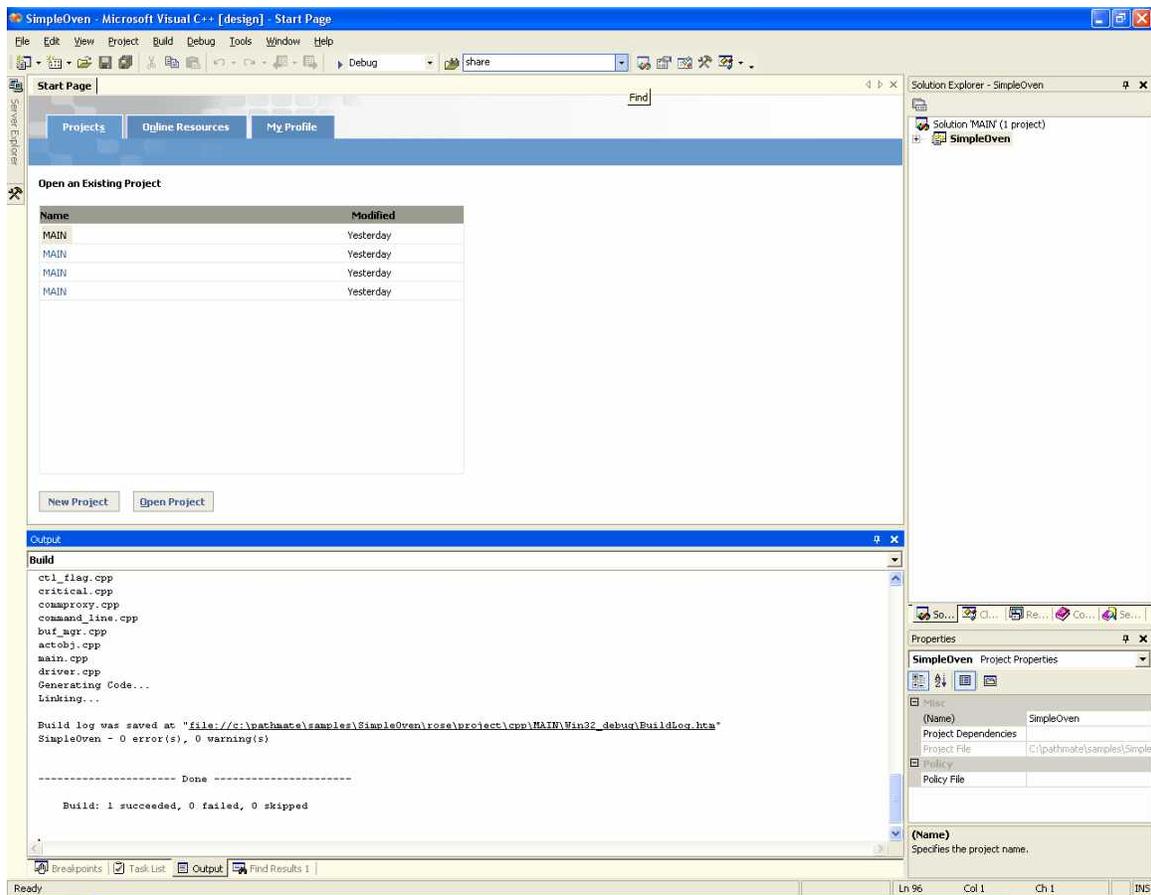
Use Visual C++ to build *SimpleOven.exe*. The QuickStart project files are compatible with versions 6.0 and 7.0 of the Microsoft Visual C++ compiler. The instructions assume you are using Visual Studio 7.0. Locate the project files for SimpleOven in `...\\QuickStart\\rose\\project\\cpp\\MAIN`. Among them is a solution file named *MAIN.sln*.

To launch the Visual C++ compiler and build *SimpleOven.exe*:

1. Double-click *MAIN.sln* in `...\\project\\cpp\\MAIN`.

Visual C++ opens.

2. Click Build > Build Solution in the top menu bar.
3. Observe the build log at the bottom of the display as Visual C++ builds the system. The figure below shows the log at the end of a successful build.



Congratulations! You now have a complete C++ executable for your system!

Introduction to Spotlight

Spotlight is PathMATE's model-level debugger. The tasks in this section introduce a few of Spotlight's capabilities:

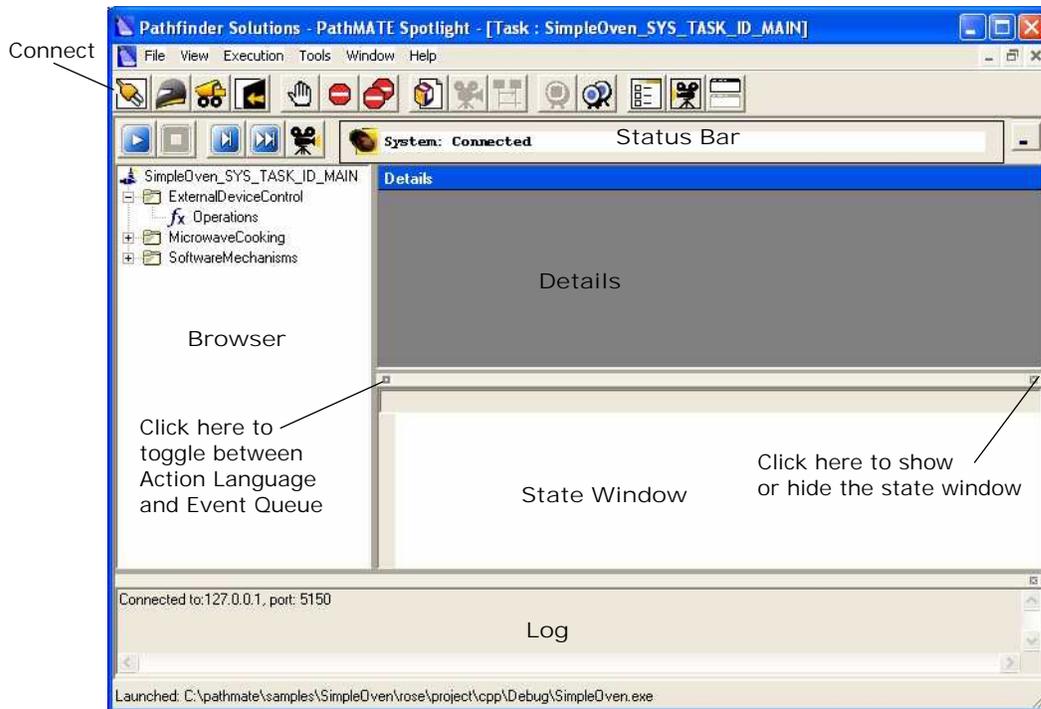
- *Run SimpleOven with Spotlight*
- *Browse the Door Class*
- *Step Through Action Language*
- *Send an Event*
- *Animate State Transitions*

Task 1: Run SimpleOven with Spotlight

After you have a successful build, launch the application and connect it to Spotlight:

1. To run the application, select Debug > Start from Visual Studio.
A command window opens to say that the application is running and waiting for a connection to Spotlight on port 5150.
2. To launch Spotlight, click Start > Pathfinder Solutions > Spotlight on the Windows desktop.
3. Click the Connect icon at the left end of the toolbar to connect Spotlight to the target application.

When Spotlight is successfully connected, the three domains in SimpleOven appear in the browser on the left, and the status bar indicates *System: Connected*.

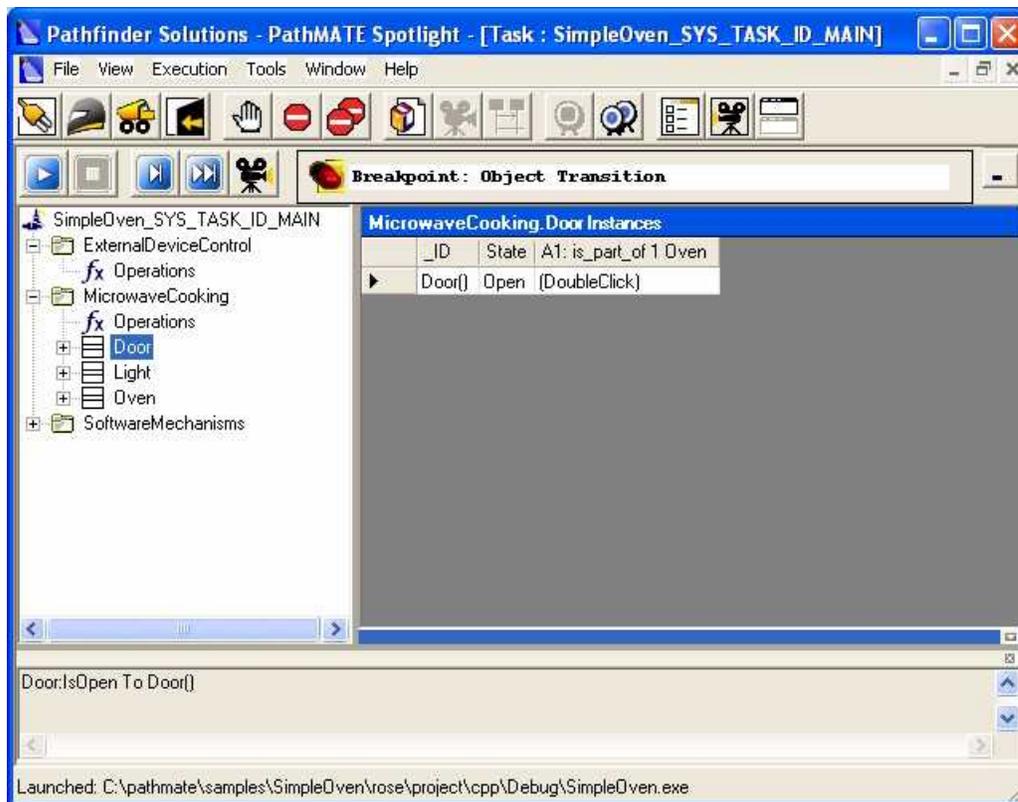


Task 2: Browse the Door Class

To browse the Door class from the Spotlight browser:

1. Click the Incident Step button  above the browser.
The status indicator changes from *Connected* to *Object Transition*.
2. Expand the MicrowaveCooking domain in the browser.
3. Select the Door class in the browser.

The details pane shows one instance of the Door class. The door object is in the Open state, and it is part of one oven (association A1).



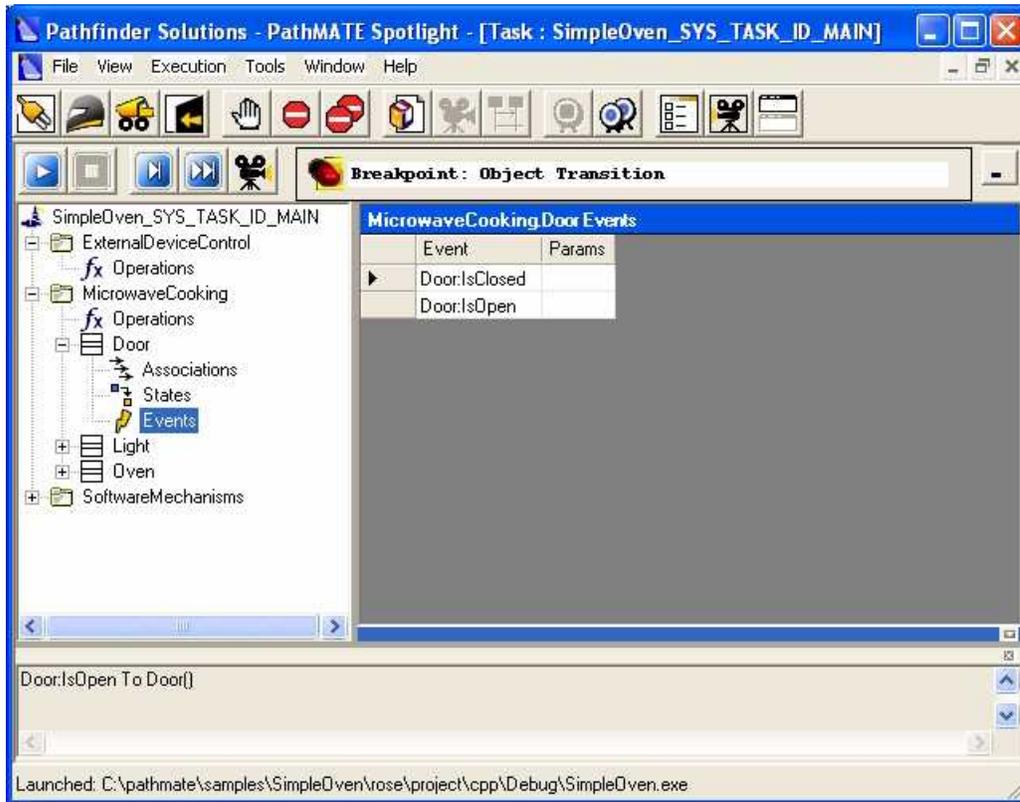
4. Expand the Door class in the browser.
5. Select *States* in the Door class.

The details pane updates to show the permitted states for the Door class.



6. Select *Events* in the browser.

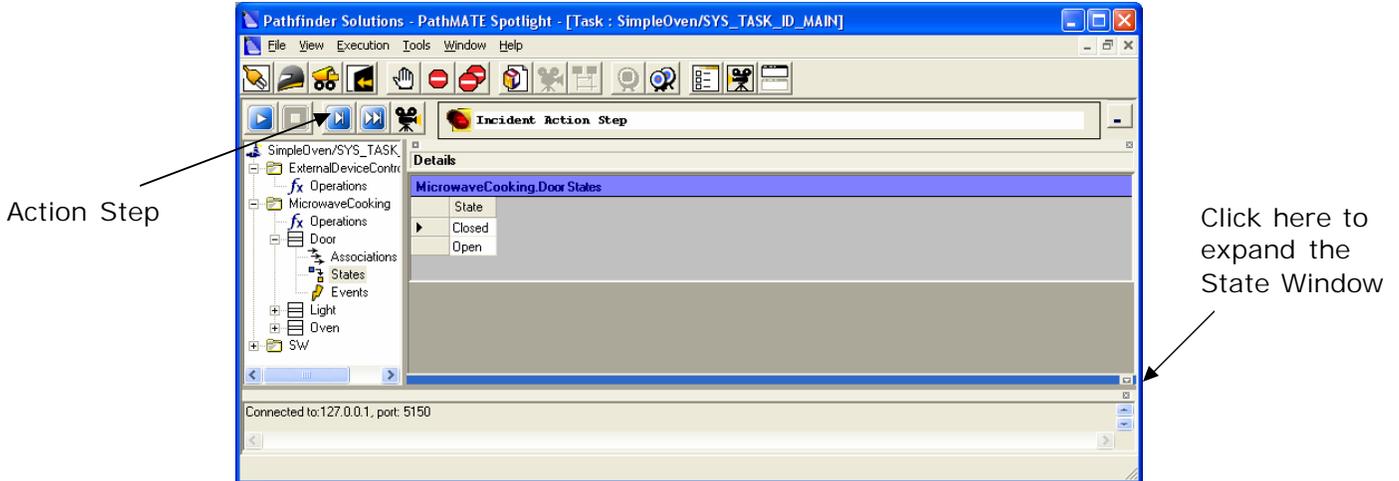
The details pane shows the events that you can send to the Door class.



Task 3: Step Through Action Language

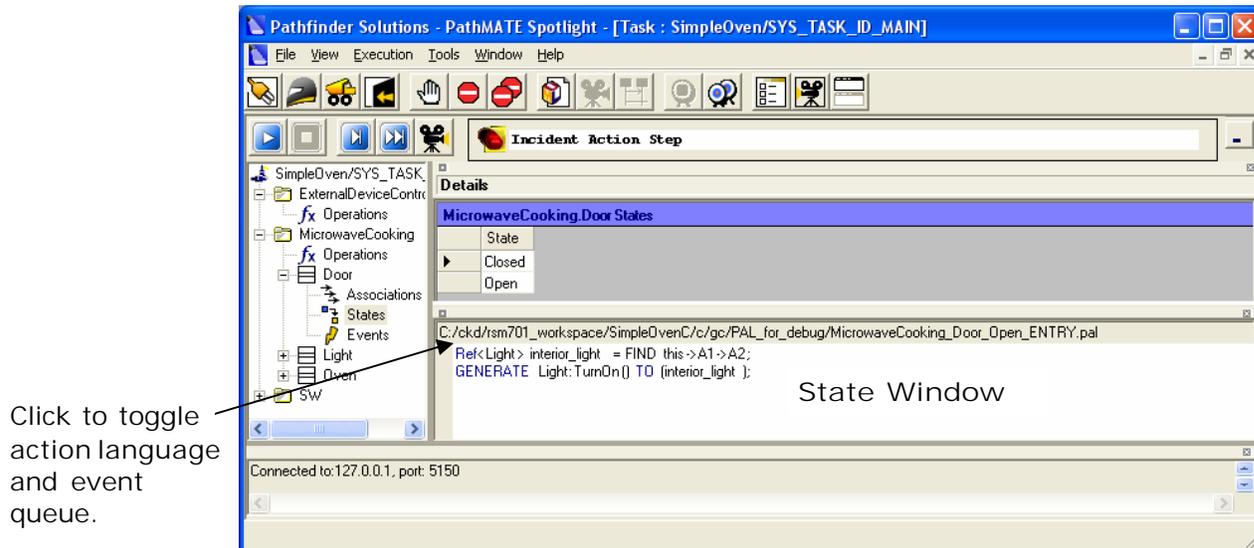
To step through the action language in the system:

1. Click the Action Step button  above the browser.
2. If the state window is not open, open it by clicking on the expand button  just above the log. The figure below shows how to open the state window.



3. Click on the toggle button  on the left hand side of the State Window to toggle between action language and the event queue. The figure below shows how to toggle the state window.

The first lines of action language appear in the state window.



4. Click Action Step again.
The active step arrow advances to the next action language statement.

- Continue to click Action Step until the active step arrow reaches the last line of action language.

Task 4: Send an Event

To send an event from Spotlight to SimpleOven:

- Select *Door* in the browser.
- Right-click *Door()* in the details pane and select Generate event for Door() in the pop-up menu.

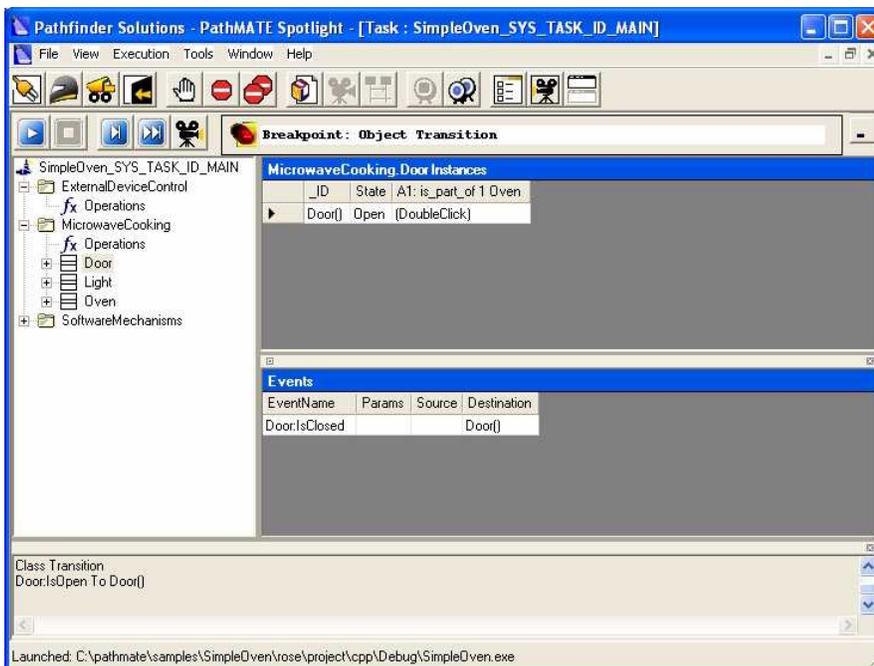
The Send Event dialog opens.

- Select *Door:IsClosed* in the drop-down list.



- Click Send.
- Press Ctrl+Q to display the event queue in the lower pane.

The event *Door:IsClosed* is queued to be sent to *Door()*. The event appears in the queue under Events in the lower pane.



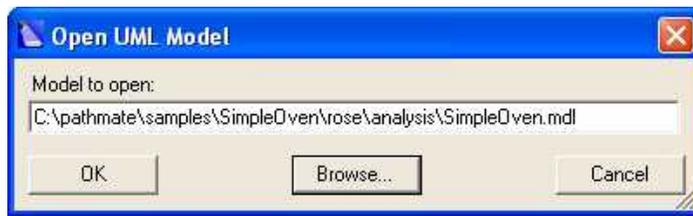
Task 5: Animate State Transitions

To animate SimpleOven's state transitions in Rational Rose:

1. Select File > Open UML Model... in the top menu bar to connect Spotlight to Rational Rose.

The Open UML Model dialog appears.

2. Click the Browse button to select the `C:\pathmate\samples\SimpleOven\rose\analysis\SimpleOven.mdl` file.



3. Click OK to close the Open UML Model dialog.

Rational Rose opens and displays the SimpleOven domain chart.

4. Select Execution > Animation On/Off... in Spotlight's top menu bar to enable state animation.

5. Click Execution > Animation Options... in Spotlight's top menu bar.

The Animation Control dialog box appears.

6. Select the Classes tab and click Select All.

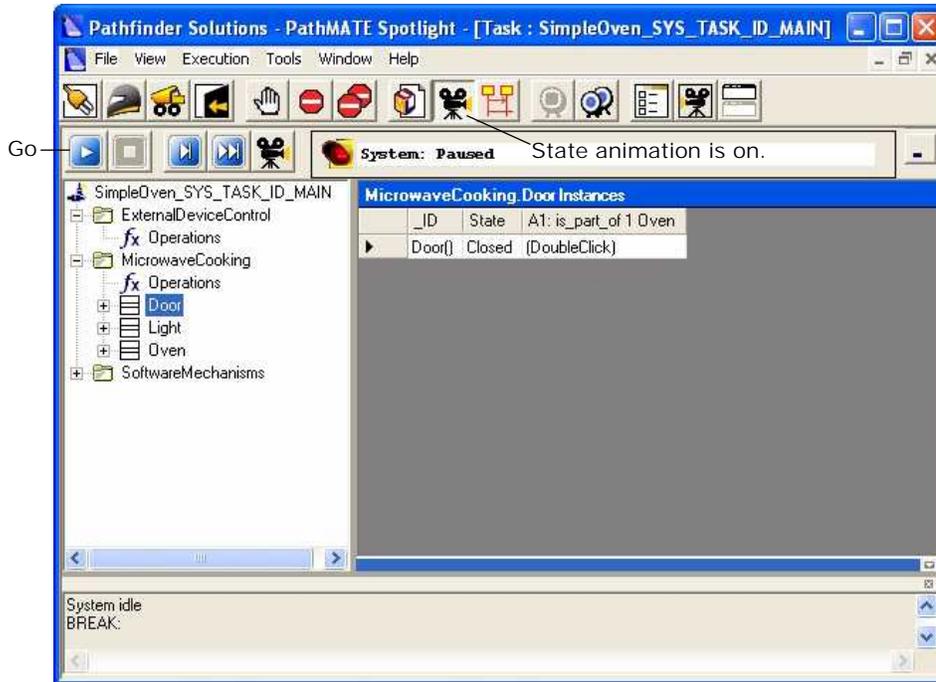
A checkmark appears in all four checkboxes in the Classes tab.

7. Click OK to close the Animation Control dialog.

8. Position Spotlight and Rational Rose so both displays are visible on your desktop.

Make sure the Rational Rose window is large enough to accommodate an entire state chart.

9. Click the Go button  above the browser in Spotlight.



Immediately after you click the Go button in Spotlight, the Door state chart opens in Rational Rose to show the state transition in the Door class. The Light state chart follows, to show the corresponding transition in the Light class.

Generate System Documentation

PathMATE transforms models into comprehensive documentation in both Rich Text and plain text formats. To generate documentation, return to Eclipse and transform:

1. Open the SimpleOven.pathmate file.
2. Select the All Reports deployment.
3. Click Transform.
4. Check the results of the transformation on the Console tab. The "Properties file properties.txt not found" warning is benign.
5. The generated reports are in the `C:\pathmate\samples\QuickStart\rose\reports` directory. See the table below for a description of the generated Rich Text Format documentation. Open the reports in Microsoft Word or any other editor that supports Rich Text Format files.

Report Title	Filename
Domain Report for SimpleOven	SimpleOven_summary.rtf
Full Analysis Report for SimpleOven	SimpleOven.rtf
Class Modeling Report for SimpleOven.MicrowaveCooking	SimpleOven_MicrowaveCooking_summary.rtf
State Modeling report for SimpleOven.MicrowaveCooking	SimpleOven_MicrowaveCooking.rtf

Congratulations!

You have documentation and reports for your QuickStart system.

You have now completed the Quick Start.