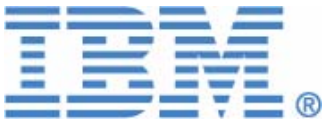
Rational. Statemate

MicroC Tutorial

Rational Statemate MicroC Tutorial



Before using the information in this manual, be sure to read the “Notices” section of the Help or the PDF file available from **Help > List of Books**.

This edition applies to IBM® Rational® Statemate® 4.6 and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Creating a Traffic Light in MicroC

This tutorial walks through the process of creating MicroC code using IBM® Rational® Statemate®. The example project is a traffic light with red, yellow, and green indicators. The design is simple so you can fully concentrate on MicroC code generation, rather than on the design problem itself.

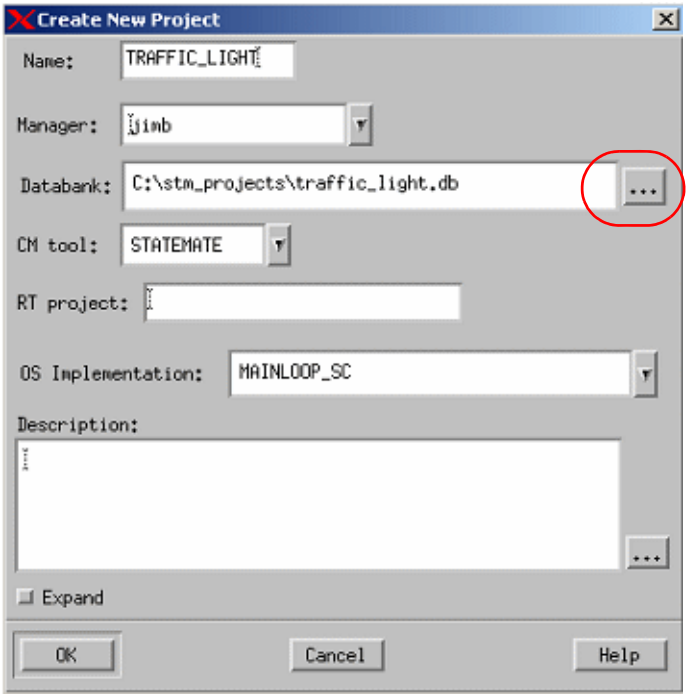
This tutorial contains the following topics:

Tutorial Setup

Prior to starting the exercises, you must set up your system and create a project within Rational Statemate following these steps:

1. Create a folder/directory called “stm_projects” on your system in which to store your project data.
2. To begin to create a project prior to starting the lab exercises, start Rational Statemate.
   a. From Windows select **Start > IBM Rational > Rational Statemate <version>.**
   b. From UNIX select `$STM_ROOT/run_stmm.bat`.
   The Rational Statemate Main windows opens.
3. Select **File > New Project**. The Create New Project window opens.

Create New Project



Clicking on the '…' opens an explorer window for browsing.

The explorer window might become minimized.

4. Enter the following data in the Create New Project window. See the figure.

   Note: Blank spaces are not allowed within the form.

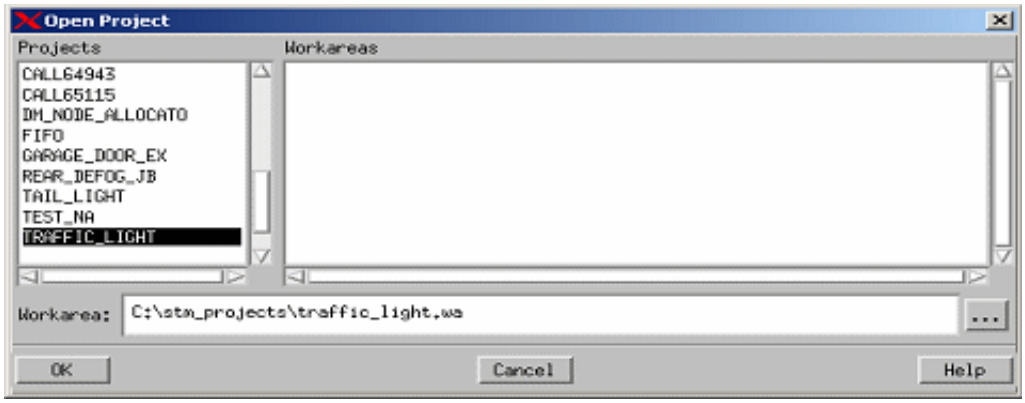
   - **Databank** - `<path>\stm_projects\traffic_light.db`
   - **Name** - `TRAFFIC_LIGHT`

- **OS Implementation** - `MAINLOOP_SC`

Note: Leave all other entries with their default value.

5. Click **OK** to create the project. The Open Project window opens.
6. Click **OK** to open the project.
7. Select the project that you just created, **TRAFFIC_LIGHT**.
8. Create a Workarea for the project by typing `<path>\stm_projects\traffic_light.wa` in the “Workarea:” field. See the figure.

Open Project Window

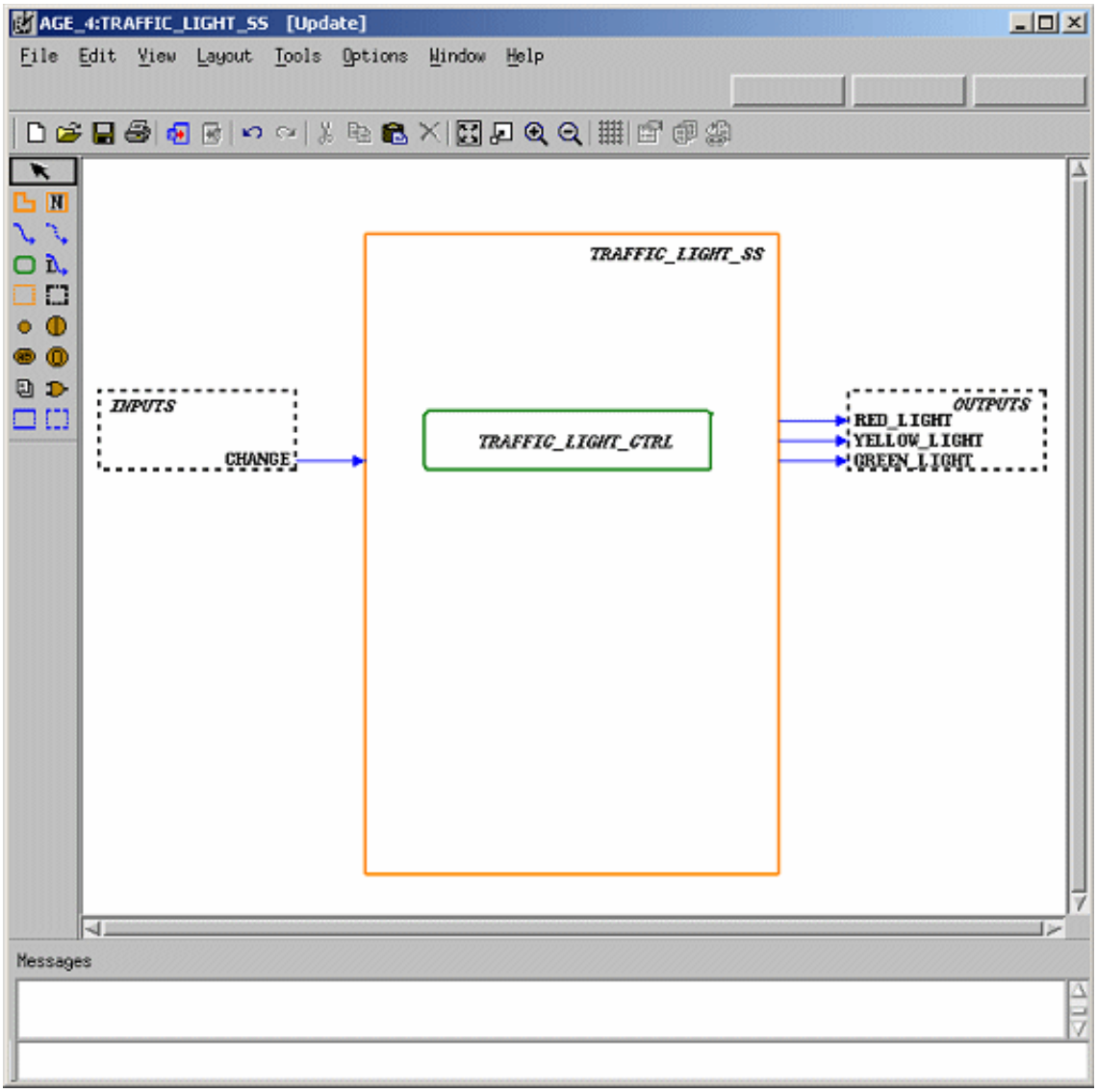


9. Click **OK**. A Question window displays asking if you want to create the Workarea.
10. Click **Yes**. The work area opens.

Exercise 1 - Constructing an Activity Chart

In this exercise, the top-level system diagram for the traffic light system is developed, as shown in the following figure.

activity Chart - TRAFFIC_LIGHT_SS



To create the activity chart:

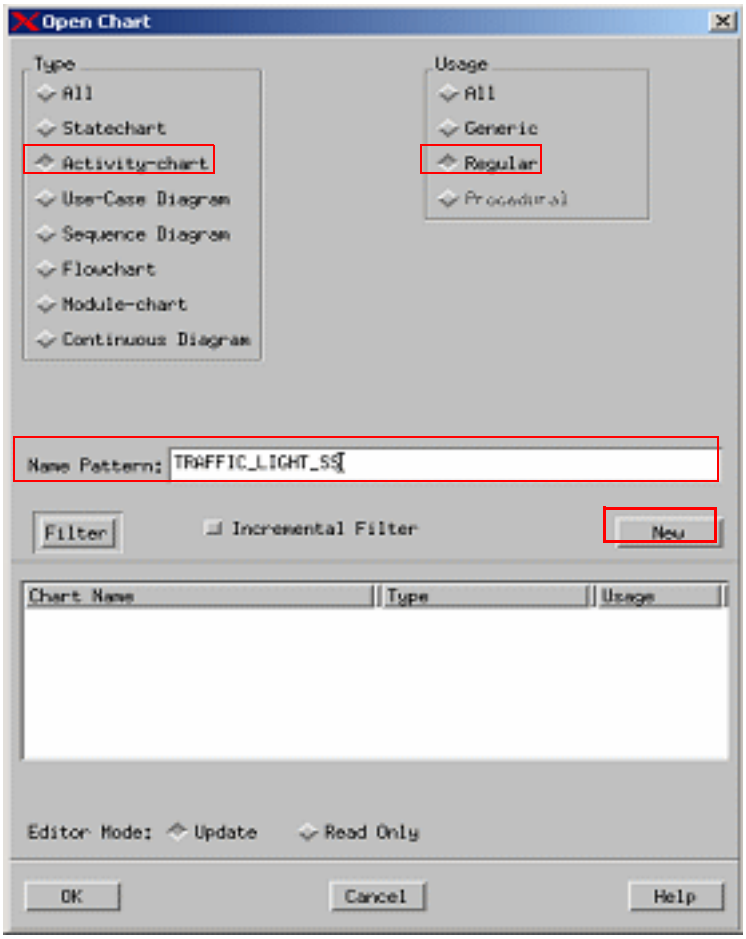
1. Click the **Graphic Editors** icon in the Rational Statemate Main window.

2. Create an activity chart called **TRAFFIC_LIGHT_SS** based on the selections shown in the figure.

Note: No spaces are allowed for chart names.

- **Type** - Activity Chart
- **Usage** - Regular
- **Name Pattern** - *TRAFFIC_LIGHT_SS*

Open/Create Chart



3. Click **New** to create and open a new diagram. See the figure. You can now edit the chart.
4. Click **Create Internal Activity**.

Draw the Internal Activity

To draw the internal activity, place the cursor at the upper-left corner of the diagram, click and drag the cursor to the lower-right corner of the activity. A ghost image shows the activity outline.

Name the Activity

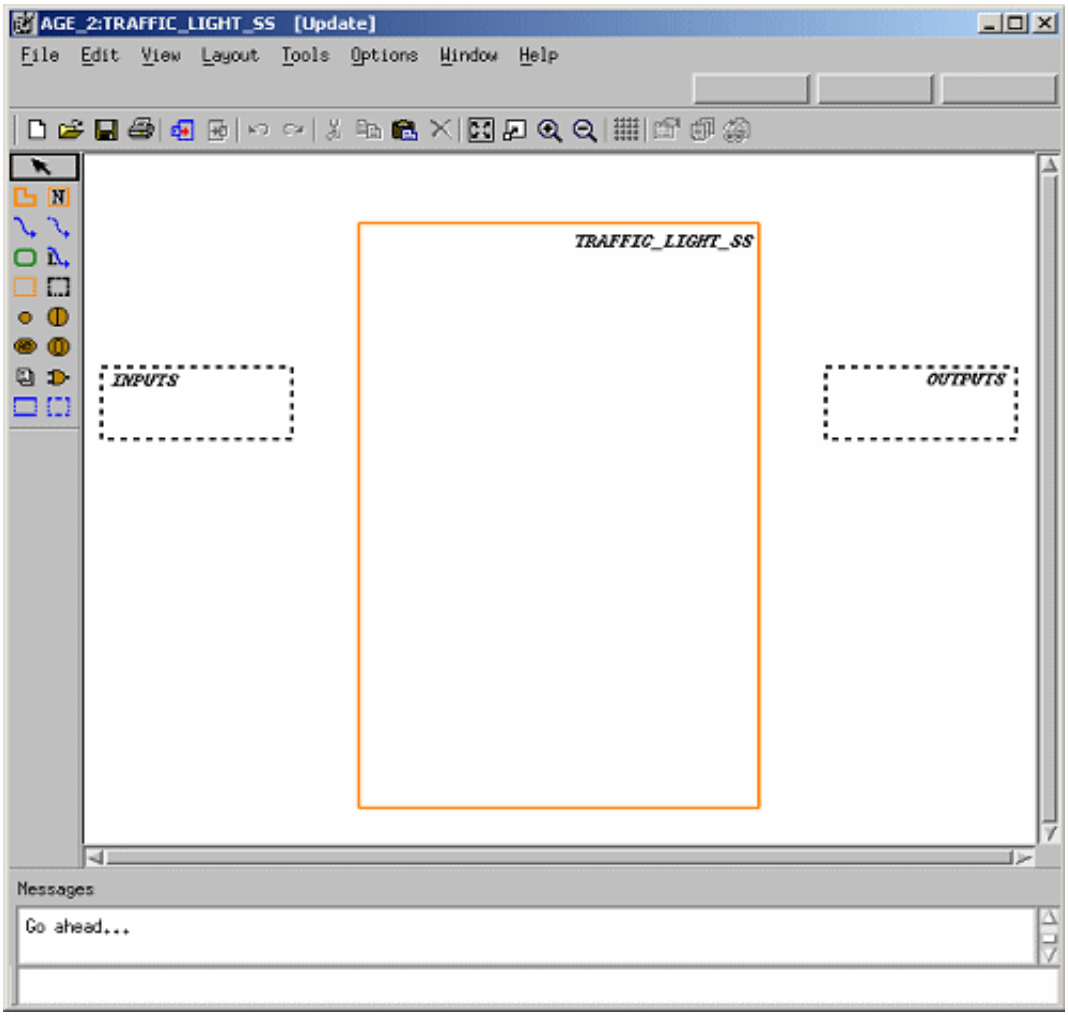
To name the internal activity you created:

1. Click **Name Existing Activity**  .
2. Click in the location where you want the Activity name or type the Activity name and then place it where you want.

   Note: Enter the names for objects using lowercase letters. Rational Statemate, where appropriate, automatically performs the conversion to uppercase letters. Do not enter words in uppercase letters. Doing so might prevent you from making further selections.

3. Type the name `traffic_light_ss`. You can select the standard pointer  and click-and-drag `traffic_light_ss` to place it.
4. Click **Create External Activity**.
5. Draw the external activities for the *TRAFFIC_LIGHT_SS*. The same drawing rules apply for drawing and naming external activities. Use the following figure as a guide.

Partial Chart Design



6. Click **Create Data-Flow**.

7. Draw a flow line from the external activity called 'INPUTS' to the internal activity called TRAFFIC_LIGHT_SS. Refer the following sections.

Drawing Flow Lines

To draw flow lines for the activities:

1. Locate the external activity called **INPUTS**. This is the source activity.
2. Click on the edge of the box to enter the tail of the arrow.
3. Locate the internal activity called **TRAFFIC_LIGHT_SS**. This is the target activity.
4. Place the cursor on the edge of the box and click to enter the arrowhead.

Note

In order to create flow lines with angles, you can click at each location an angle is wanted prior to clicking/releasing the middle mouse button.

Labeling and Placing Flow Line Labels

To label and place the flow line labels:

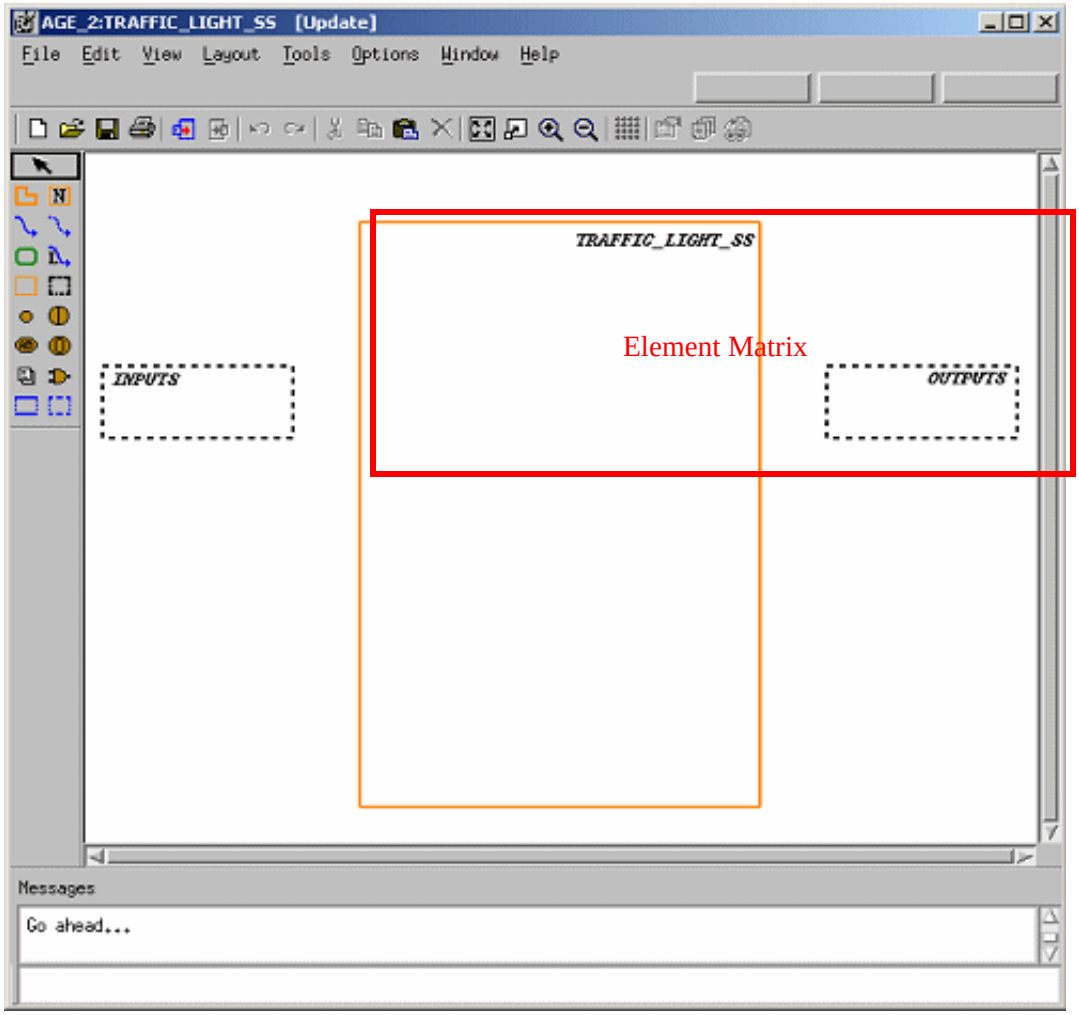
1. Click **Label Existing Flow-Line**  .
2. Click on the flow line that you want to label and type the name `change` or type the name `change` before clicking on the flow line and then place the label onto the flow line.
3. Click **Pointer**  , and move the label if you want.
4. Draw a flow line from the internal activity called **TRAFFIC_LIGHT_SS** to the external activity **OUTPUTS** and label the flow **red_light**.
5. Draw a flow line from the internal activity called **TRAFFIC_LIGHT_SS** to the external activity **OUTPUTS** and label the flow **yellow_light**.
6. Draw a flow line from the internal activity called **TRAFFIC_LIGHT_SS** to the external activity **OUTPUTS** and label the flow **green_light**.
7. Click **Create Control Activity**.
8. Draw and name the control activity, **traffic_light_ctrl** Use the same drawing techniques to draw the other activities.
9. Select **File** > **Save** from the chart and then **File** > **Exit**.

Exercise 2 - Defining Textual Elements

This exercise describes how to define the textual elements that were entered into the activity chart **TRAFFIC_LIGHT_SS**.

1. From the Chart Tab, click on the chart called **TRAFFIC_LIGHT_SS.** The element matrix opens in the right pane. See the figure.

Element Matrix for Chart TRAFFIC_LIGHT_SS



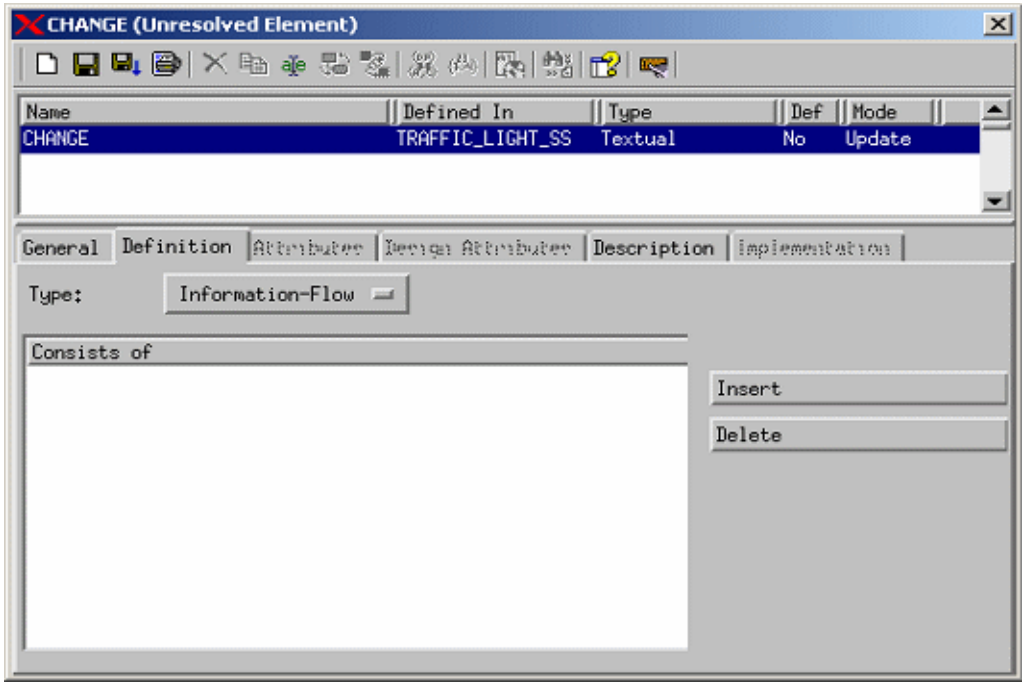
2. Select the element called **CHANGE** from within the element matrix.

3. Select **Edit** > **Properties**. This invokes the Properties window for this element. See the following figure.

Note

Double-clicking on the element name also opens the properties window for this element.

Properties - CHANGE



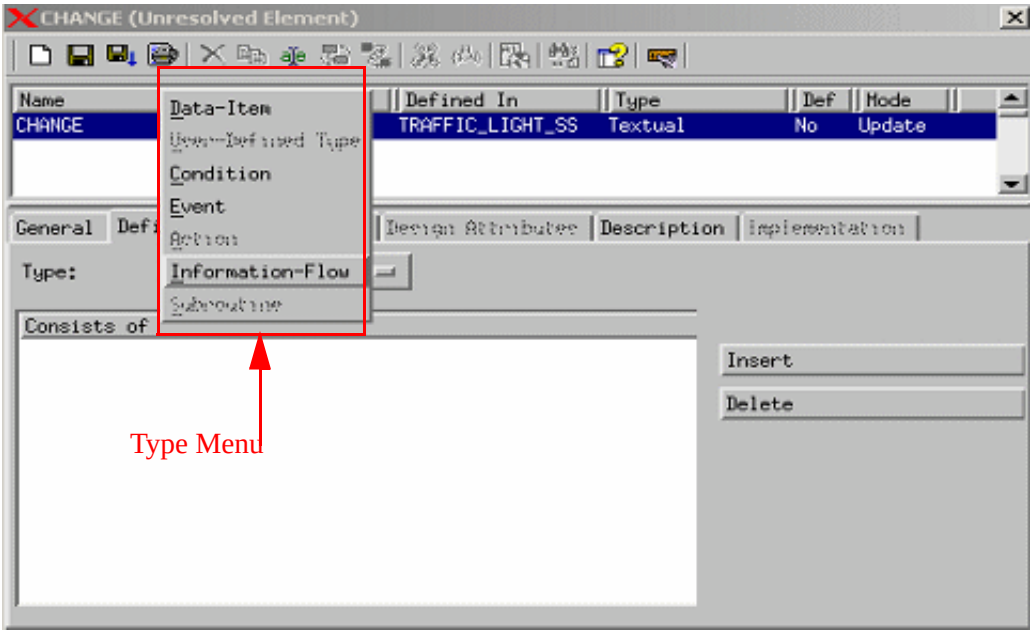
4. Define the textual element **CHANGE** using the following steps:

   a. Change the **Type:** of the element from **Information Flow** to **Event.** See the figure.

   b. Click **Save** .

   c. Exit

Unresolved Element - CHANGE



5. Set the Type**:** to **Condition** for the other three elements (RED_LIGHT, YELLOW_LIGHT, GREEN_LIGHT), and save the settings.

6. Open the external activities, INPUTS and OUTPUTS, and set the **Type:** to **Environment**.

   a. Change the **Type** from **External** to **Environment**

   b. Click **Save**

   c. Exit

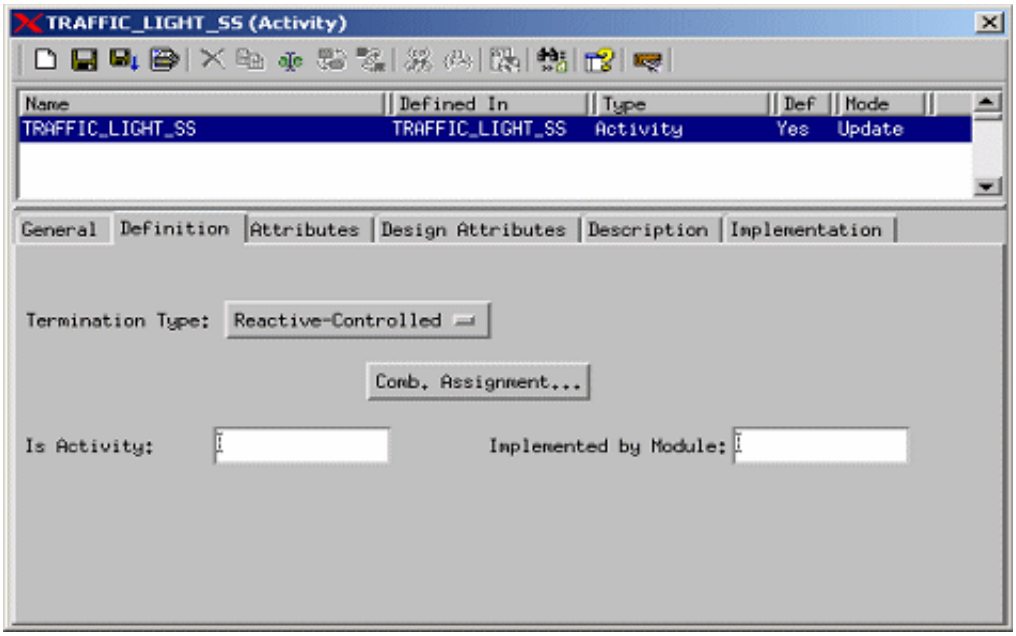
Exercise 3 - Defining Design Attributes

In this exercise describes how to add design attributes to selected elements within the model.

1. Select the activity called '**TRAFFIC_LIGHT_SS**' from within the element matrix and then select **Edit > Properties**. This invokes the Properties window for this element. See the figure.

   Note: Double-clicking on the element name also opens the properties window for this element.

TRAFFIC_LIGHT_SS Properties



2. Select the Design Attributes tab within the properties window.

   a. Change the **Value** of **Type** to **Task**.

   b. Click **Save**

   c. Exit

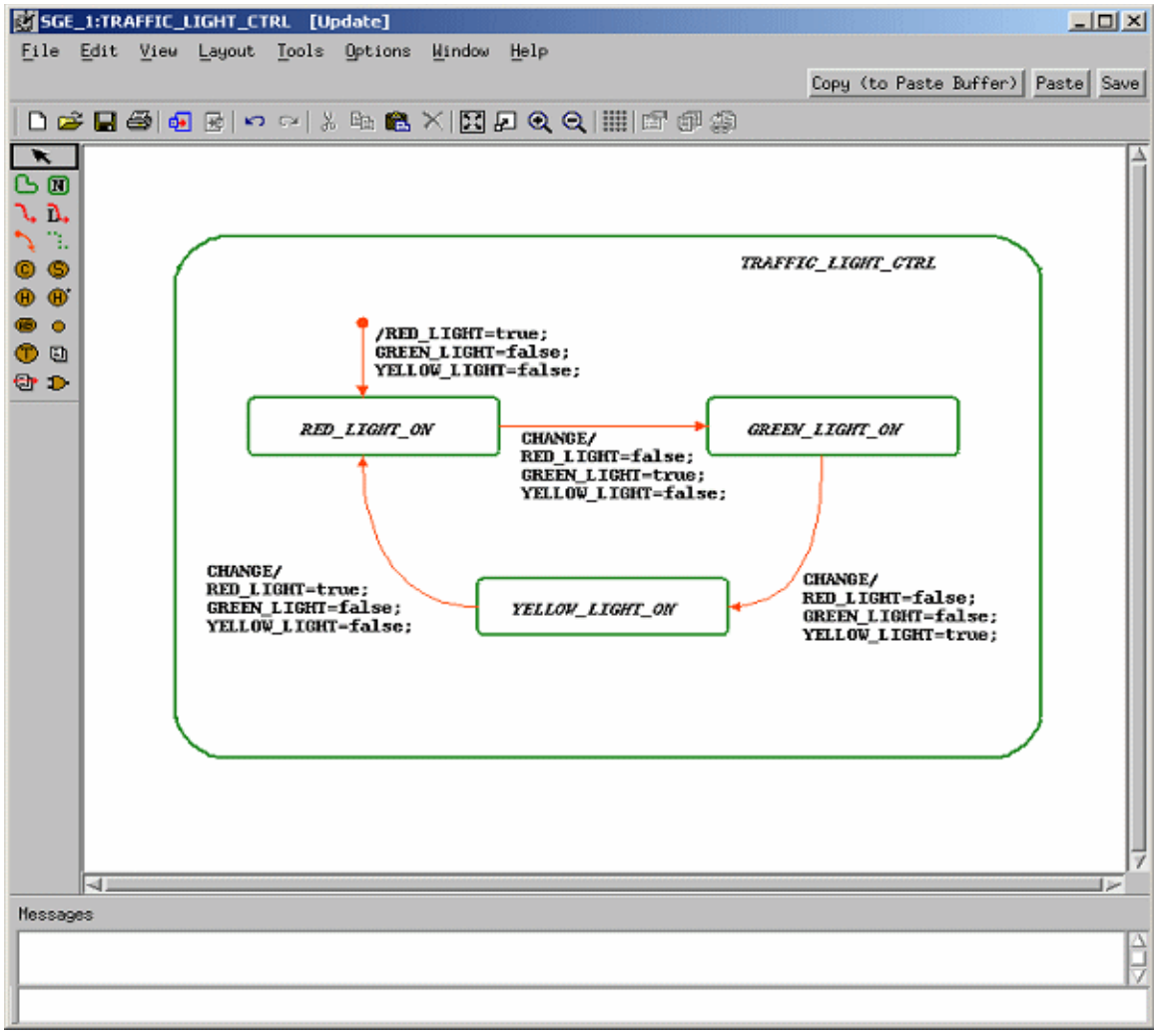
3. Open the properties window for the element called **CHANGE** and select the **Design Attributes** tab.

4. Select the Name **Its Task** and then click **Choose** to select the task for the event change.
   a. The Choose window opens. Click **Dismiss** to close the window.
   b. Click **Save**.
   c. Exit.

Exercise 4 - Constructing Statecharts

This exercise describes how to construct a statechart that demonstrates behavioral aspects of the traffic light system. See the figure.

Statechart TRAFFIC_LIGHT_CTRL



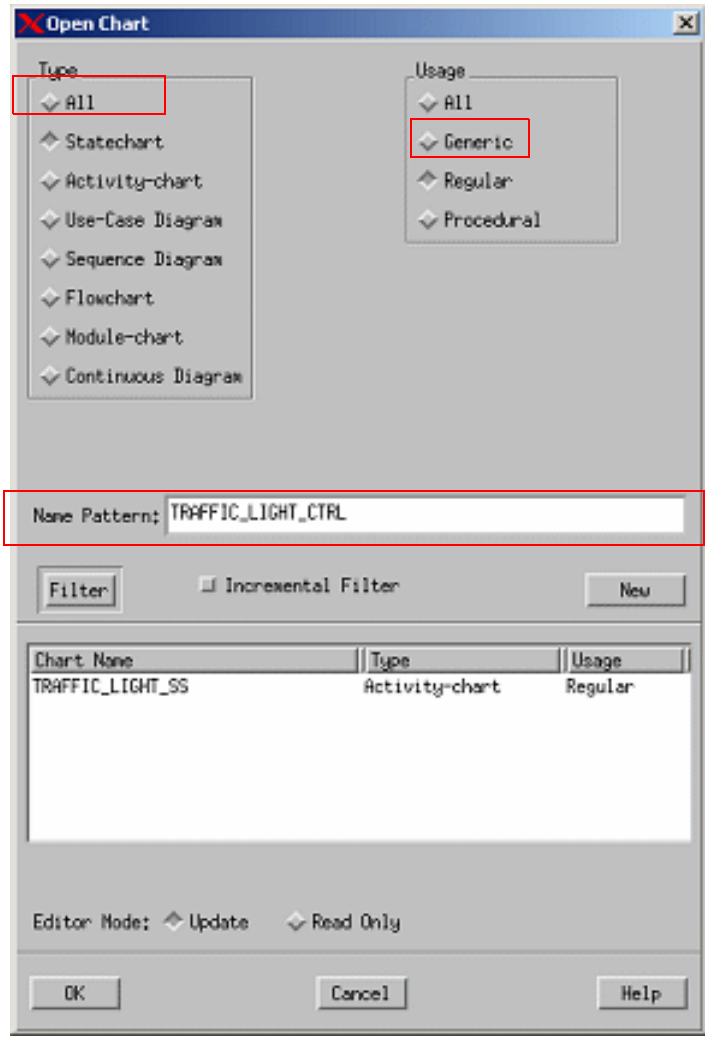
1. Click **Graphic Editor** from the main Rational Statemate window.

2. Name the statechart TRAFFIC_LIGHT_CTRL based on the selections in the figure.

   Note: No spaces are allowed for chart names.

   - **Type** - Statechart
   - **Usage** - Regular
   - **Name Pattern** - TRAFFIC_LIGHT_CTRL

Open/Create Chart



3. Select New from the Open Chart window to create and open a new diagram. The chart is now available for editing.

4. Click **Create State**.

5. Draw a state called **TRAFFIC_LIGHT_CTRL**. See the figure **Statechart TRAFFIC_LIGHT_CTRL**.

Drawing States

To draw states in the statechart, follow these steps, place the cursor in the upper-left corner of the state and click and drag to the lower-right corner of the state. A ghost image shows the state outline.

Naming States

To name the newly created states in the statechart:

1. Click the **Create State** or **Name Existing State** icon and begin typing the state name **traffic_light_ctrl.** The name of the state displays next to the cursor.
2. Click where you want the state name to reside.

   Note: Enter the names for boxes using lowercase letters. Rational Statemate, where appropriate, automatically performs the conversion to uppercase letters. DO NOT select the CAPS LOCK key before typing.

3. Using the **Statechart TRAFFIC_LIGHT_CTRL** figure as a guideline, create and name the other states within your chart.
4. Click **Create Transition**.
5. Draw a transition from the **RED_STATE_ON** state to the **GREEN_STATE_ON** state.

Drawing Transitions

To draw transitions in the statechart:

1. Locate the state called **RED_STATE_ON**. This is the source state. Click on the edge of the box to enter the tail of the arrow.
2. Locate the state called **GREEN_STATE_ON**. This is the target state. Click on the edge of the box and click to enter the arrowhead.
3. With the Create Transition icon still selected, label the transition:

   CHANGE/RED_LIGHT=false;
   GREEN_LIGHT=true;
   YELLOW_LIGHT=false;'

Labeling Transitions and Placing Labels

To label the transitions in the statechart:

1. Click on the Create Transition tool or the Label Existing Transition tool  , and type the label name.
2. The label displays next to the cursor and follows the cursor.
3. Click on the transition line you want to label to place the label.
4. Select the Create Default Transition icon.
5. Draw a Default Transition into the state **RED_STATE_ON**. Default transitions do not have a source state and cannot have triggers. Use the **Statechart TRAFFIC_LIGHT_CTRL** figure as a guide as where to draw the default transition.
6. With the Create Default Transition icon still selected, type the following label for the transition:

   '/RED_LIGHT=false;
   GREEN_LIGHT=true;
   YELLOW_LIGHT=false;'

   Place the transition label on the transition if you have not already done so.
7. Using the techniques in the above steps complete the design of the statechart based on the **Statechart TRAFFIC_LIGHT_CTRL** figure.
8. Select **File > Save** from the chart and then **File > Exit**.

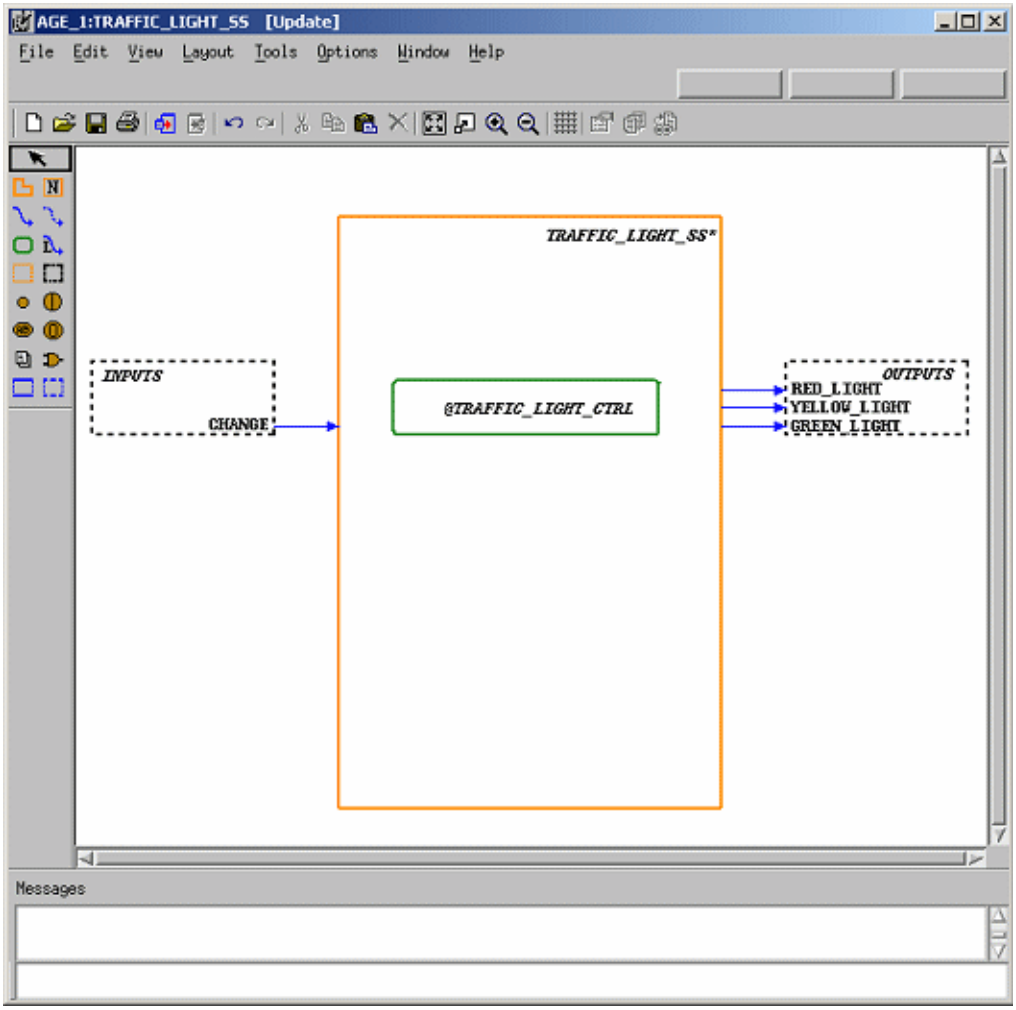
9. The statecharts that were developed in this exercise is used to describe the behavior for the activity **TRAFFIC_LIGHT_SS**. To associate the statecharts with the activity complete the following steps:

a. Open the activity chart called **TRAFFIC_LIGHT_SS**.

b. Modify the name of the control activity by placing an **@** symbol in front of the name as shown in the figure.

c. Select **File > Save** from the chart and then **File > Exit**.

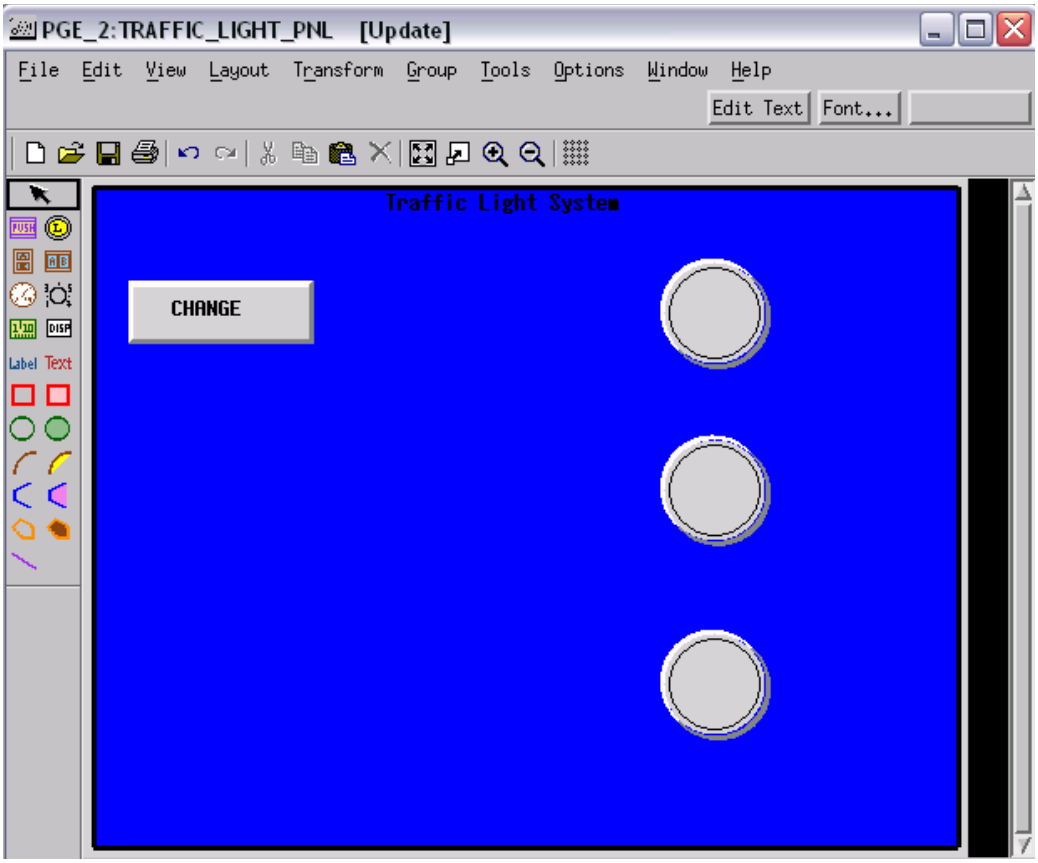
Control Activity TRAFFIC_LIGHT_CTRL



Exercise 5 - Constructing a Panel

This exercise takes you through the process of constructing the panel as shown in the following figure.

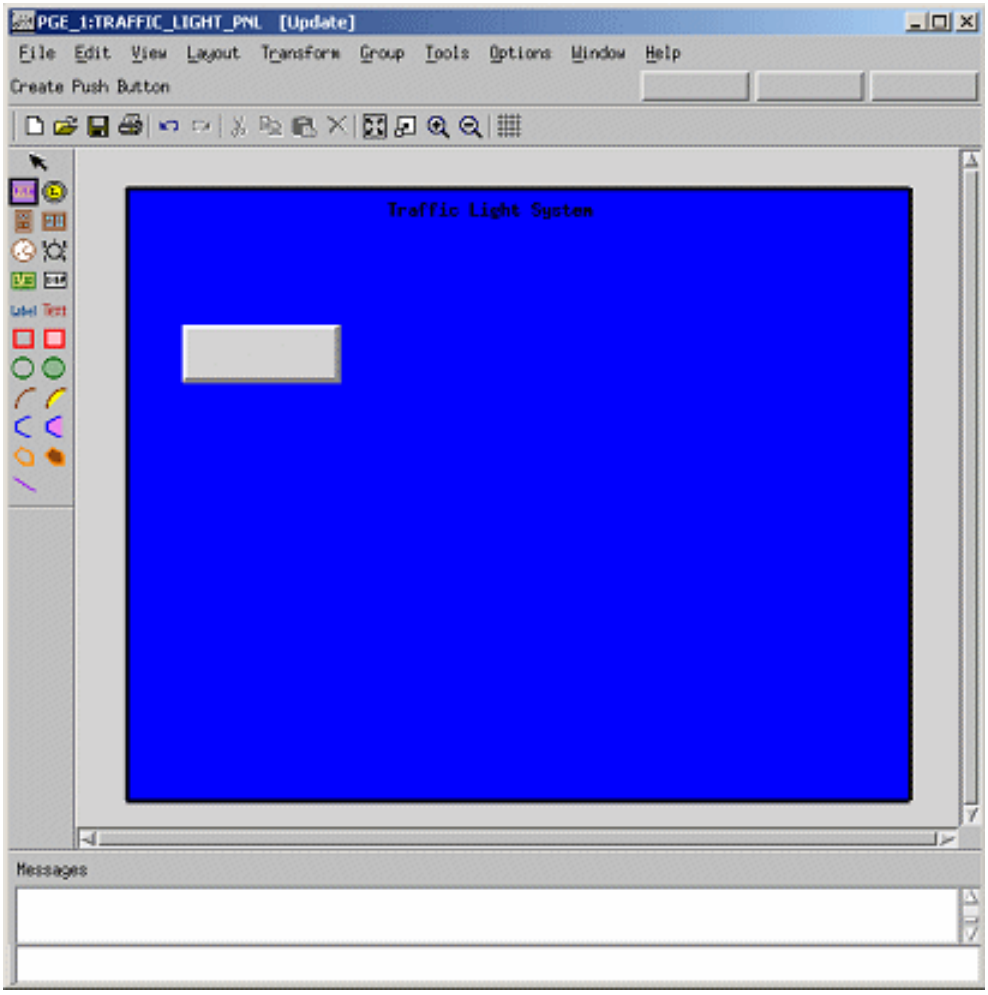
Traffic Light Panel



1. Select **Panel Editor** from the main Rational Statemate window. The Open Panel window opens.
2. Enter the name **TRAFFIC_LIGHT_PNL** for the Panel Name.
3. To begin the panel design, add a background to separate the input and output elements. Select **Create Filled Box** from the Panel Graphic Editor and stretch this to fill the panel canvas.
4. Select the new **Filled Box** and select **Tools >Properties**. The **Bindings/Properties** window opens.
5. Change the Line Width to **3**.
6. Select the **Fill Color** field and click **Choose...** The Color Viewer for PGE window opens. See the figure.
7. Select a color and click **OK**.
8. Click **Create Free Text**, and type `Traffic Light System`.
9. Add the text to the panel by clicking on the top of the panel canvas. The text follows the cursor.

10. Click **Push Button** and add this element to the panel. See the figure.

Push Button



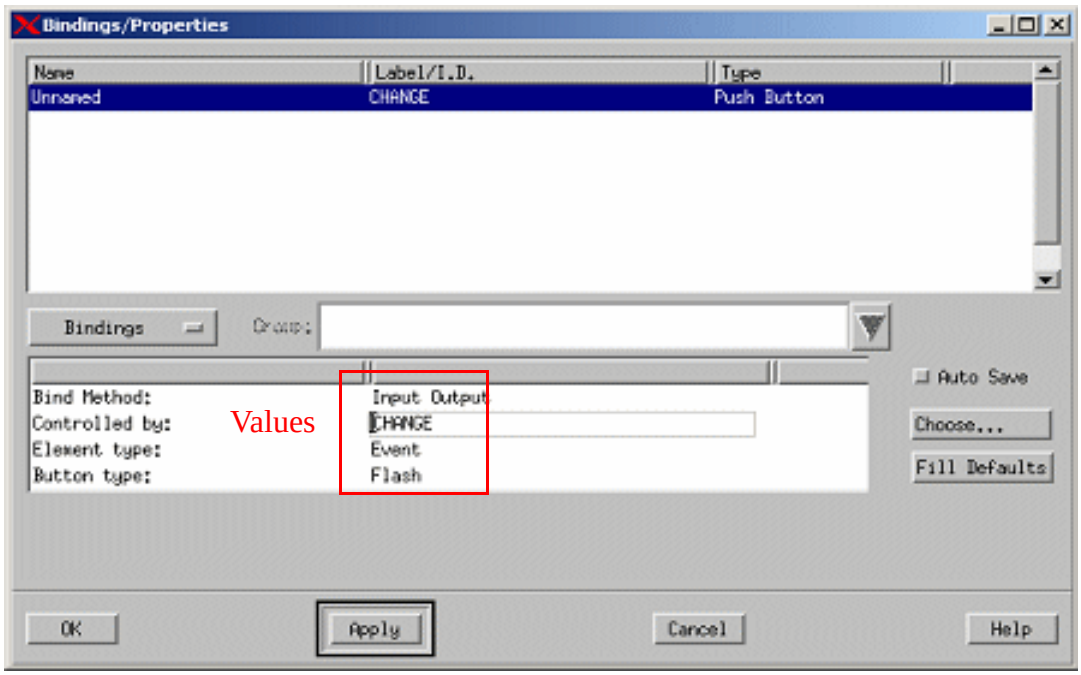
11. Click **Label Existing Interactor**, type the word `CHANGE`**.** The label name follows the cursor. Place the label over the push button object and click.

12. Select the push button object that you created and then select **Tools > Bind**. Select the following for the binding values for the object labeled **CHANGE.** See the figure.

    - **Bind Method** - Input Output
    - **Controlled by** - CHANGE
    - **Element type** - Event
    - **Button type** - Flash

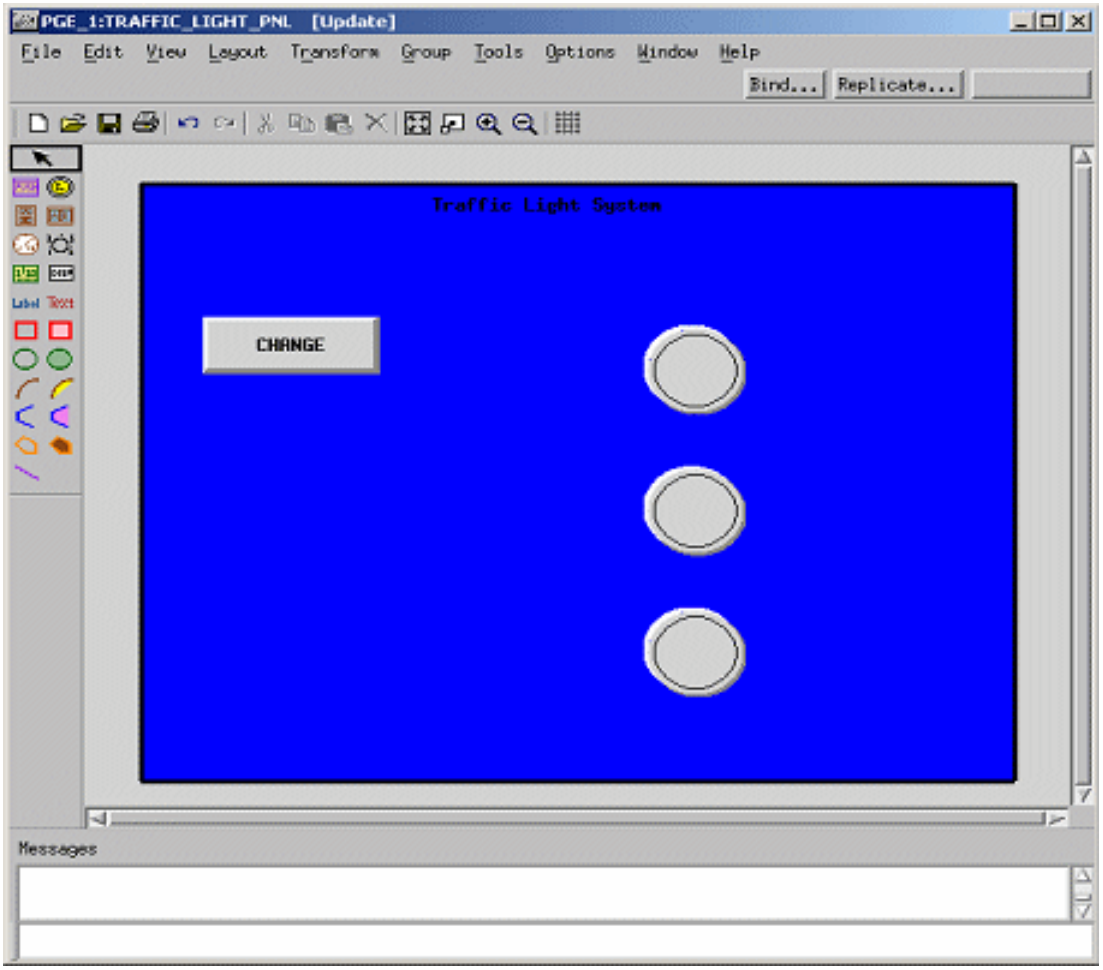
    Note: The **Controlled by:** value must be manually typed into the field. All other values are selected from the drop-down menu that opens when clicking on the value.

Push Button Bindings



13. Click **Create Lamp Interactor** and create three lamps as shown in the figure.

Lamp Interactors



14. Select the top lamp interactor and then select **Tools** > **Bind** to invoke the object's Bindings/Properties window.

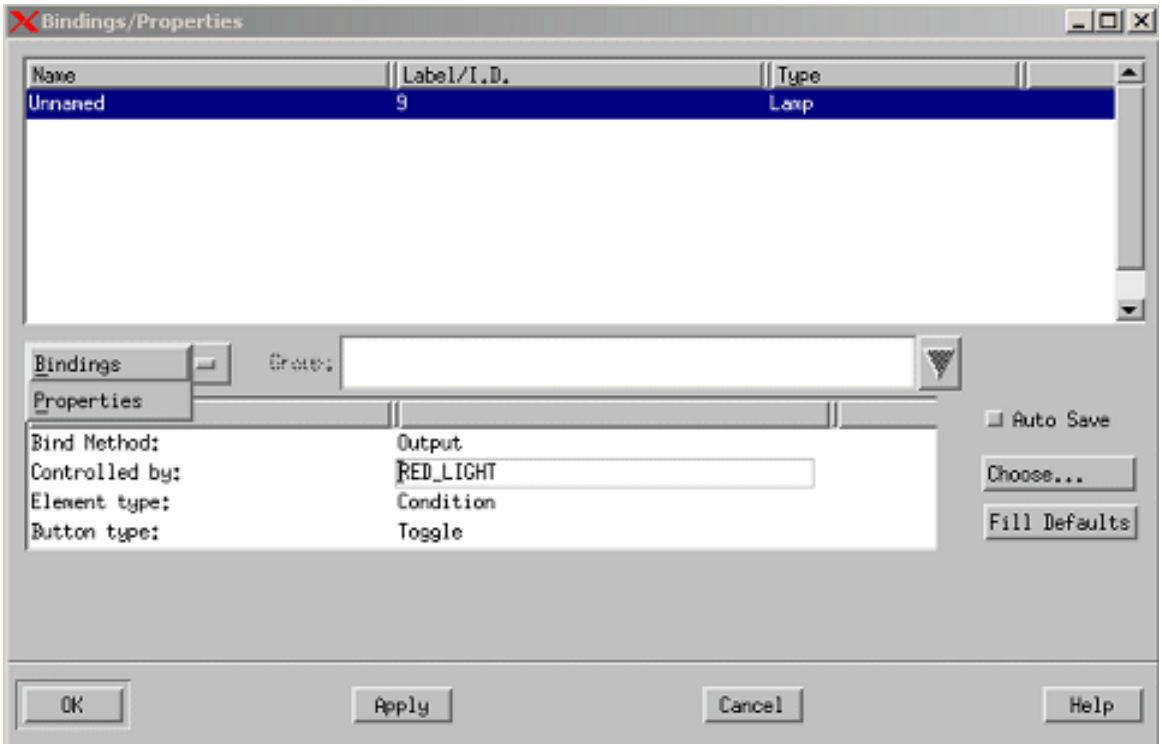
    Enter the following information as the Binding values for the object:

    - **Bind Method -** Output
    - **Controlled by** - RED_LIGHT
    - **Element type -** Condition
    - **Button type -** Toggle

15. Click **Apply**.

16. Switch to the Properties page by selecting the **Bindings/Properties** tab. See the figure.

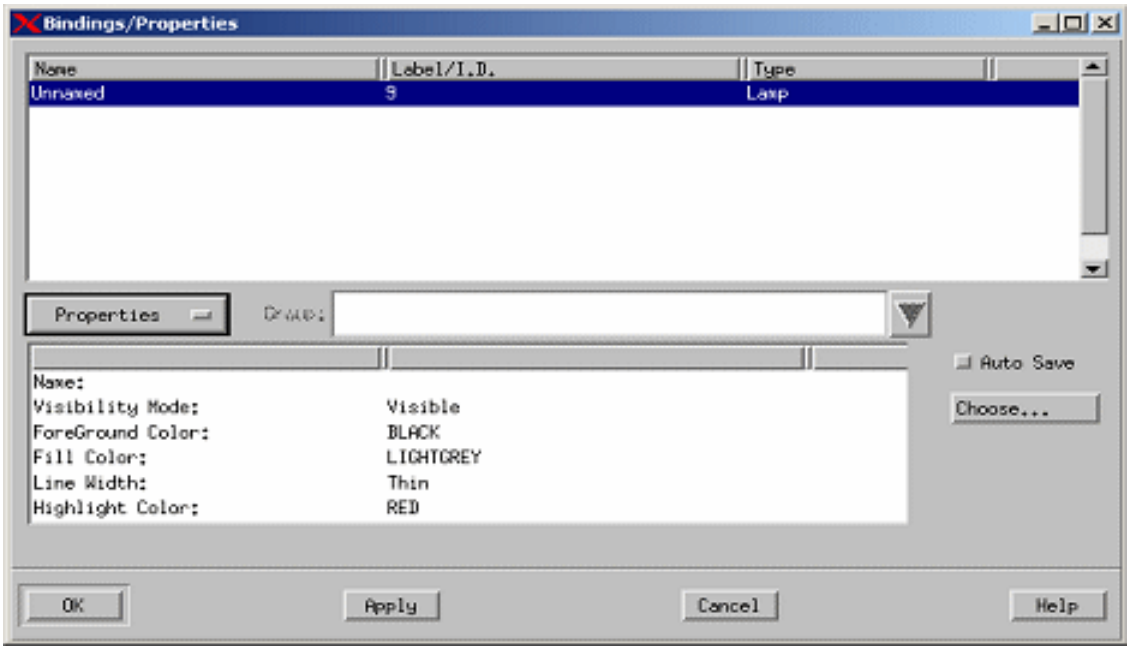
Bindings Page



17. Make sure your settings reflect those shown in the figure. The Properties values appear.

18. Click **OK**.

Properties Page



19. Select the middle lamp interactor and then select **Tools > Bind** to invoke the object's Bindings/Properties window. Enter the following for the Binding values for the object:

- **Bind Method**: Output
- **Controlled by**: YELLOW_LIGHT
- **Element type**: Condition
- **Button type**: Toggle

20. Click **Apply**.

21. Switch to the Properties page by selecting the **Bindings/Properties** tab and enter the following information for the Property values:

- **Highlight Color**: YELLOW

22. Click **OK**.

23. Select the bottom **Lamp Interactor** and then select **Tools > Bind** to invoke the object's Bindings/Properties window.

24. Enter the following for the Binding values for the object:

- **Bind Method** - Output
- **Controlled by** - GREEN_LIGHT

- **Element type** - Condition
- **Button type** - Toggle

25. Click **Apply**.
26. Switch to the **Properties** page by selecting the **Bindings/Properties** tab and enter the following for the Property values for the object:
    - **Bind Method** - Output
    - **Highlight Color** - GREEN
27. Click **OK**.
28. Select **File > Save** from the PGE and then **File > Exit**.

Exercise 6 - MicroC Code Generation

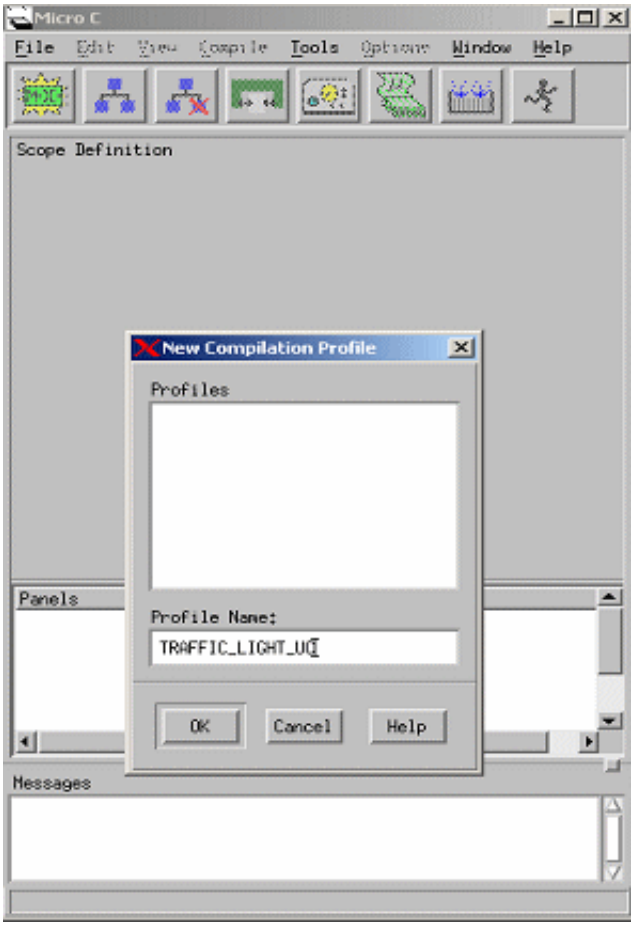
This exercise builds a MicroC Code generation profile and generates code for the traffic light system.

1. Click **Statemate MicroC Code Generation** tool from the main Rational Statemate window.

2. If no profiles currently exist, the New Compilation Profile window automatically opens. If other profiles exist, select **File > New Profile**. The New Compilation Profile window opens.

3. Type the name `TRAFFIC_LIGHT_UC` in the Profile Name: field.

MicroC Profile



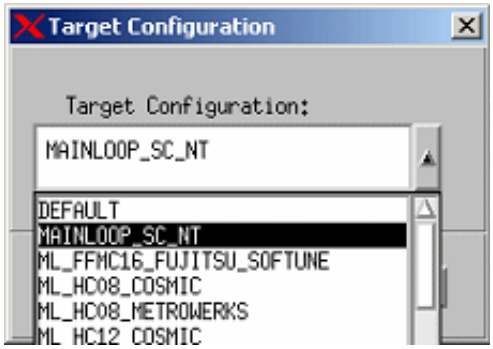
4. Click **Create New SW Module** and name the module **TRAFFICLIGHT_MOD** in the Create Module window.

5. Click on the module and then click **Add Selected Chart with Descendents to Profile**. This opens a Chart Tree View of your current design. See the figure.

Chart Tree View

6. Select the chart called **TRAFFIC_LIGHT_SS** from the Chart Tree View then click **OK**.
7. Click **Add Selected Panel to Profile** from the MicroC profile window. The Add Panels to Profile window opens.
8. Select the **TRAFFIC_LIGHT_PNL** that was created earlier in the exercise.
9. Click **OK**.
10. Select the **Options > Set Target Configuration** from the MicroC Profile window.
11. Select the **MAINLOOP_SC_NT** configuration. See the figure.
12. Click **OK**.

Set Target Configuration

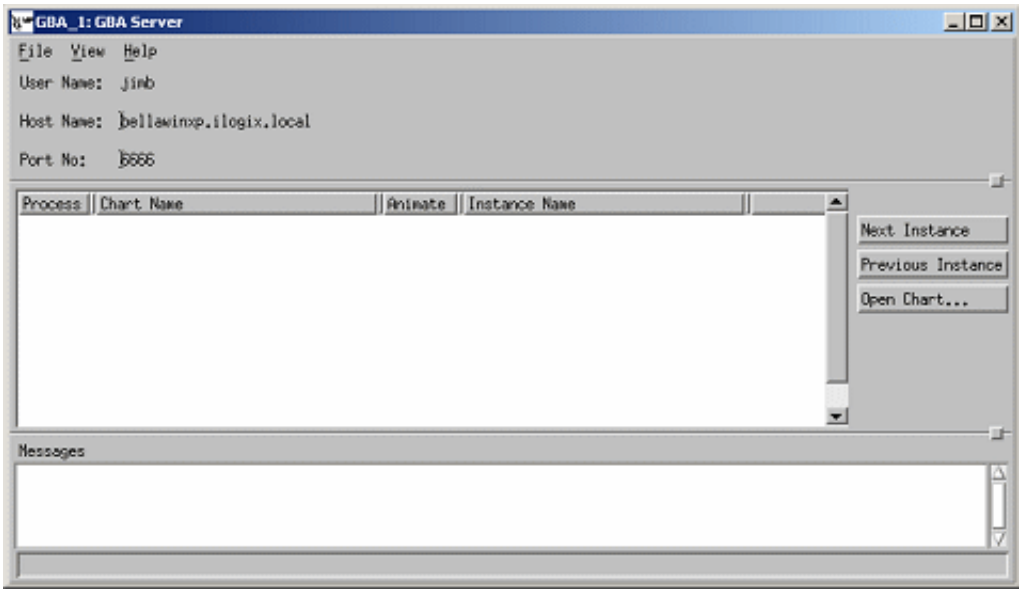


13. Select **File > Save** from the MicroC Profile window.
14. Select **Generate Code for Current Profile** in the MicroC Profile Window. This opens the Output directory for the generated code.
15. Click **OK** in the Output Directory window to accept the location for the generated code.
16. Click **Dismiss** in the generated Code window.
17. Select **Compile Generated Code** in the MicroC Profile window to compile the generated code. This opens the Selected Makefile window.
18. Select **Open** to begin compilation of the code.

    The resulting compilation output is sent to a “Command Prompt Window.” Once compilation is completed, exit out of the command prompt window.

19. The GBA Server allows you to open charts for animation. Select **Tools > Open GBA** from the MicroC Profile window. The GBA Server window opens. See the figure.

GBA Server

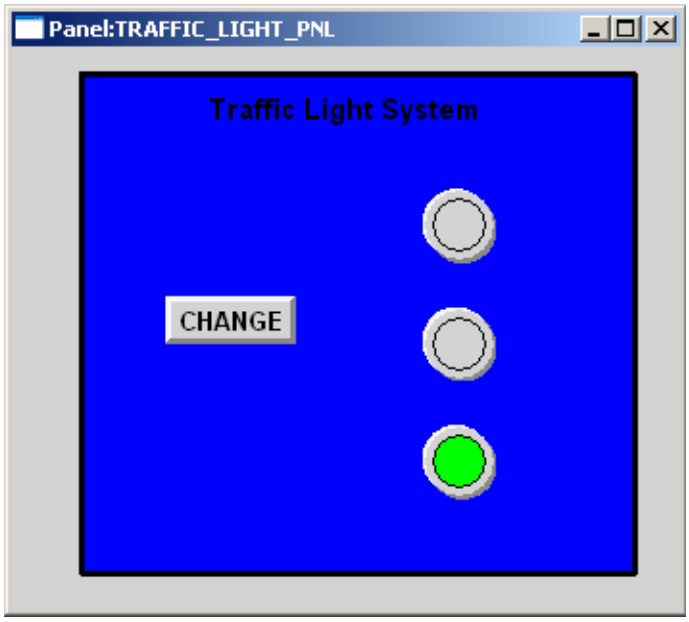


20. Click **Execute Compiled Code** from the MicroC Profile window and then select the **Open** option from the Run Command window to start the code execution.

Note: The model is now running as an executable with the panel available to provide user interaction.

21. Click **CHANGE** multiple times. Different light indicators illuminate with their respective colors. See the figure.

Panel Execution

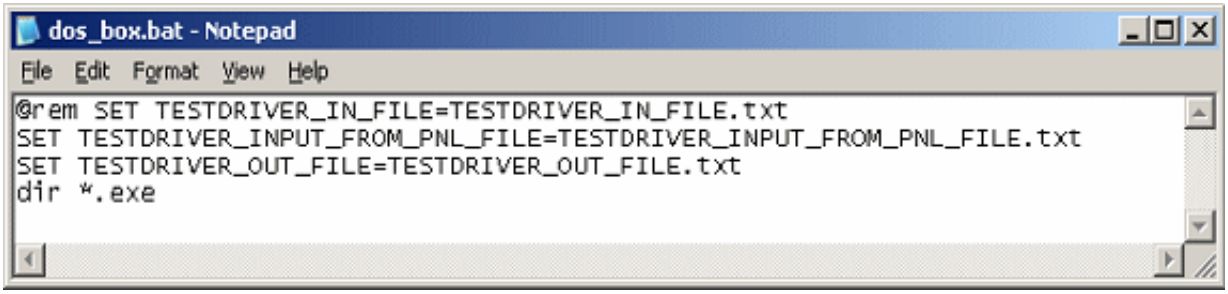


Exercise 7 - Using the Test Driver

This exercise describes how to test the dynamic behavior of the developed application non-interactively, using test vector inputs and outputs.

1. Select **Options** > **Settings** from the MicroC Profile window to open the Properties for Code Generation window.
2. Select the **Test Driver** tab and then check the **Enabled** check box.
3. Click **OK**.
4. Regenerate and recompile the model as done in the previous exercise.
5. Select **Tools** > **Open GBA** from the **MicroC Profile** window.
6. Click **Execute Compiled Code** from the MicroC Profile window and when the panel is available press the **CHANGE** push button any number of times. Within the Command Prompt you can view the Test Driver data stream output.
7. Stop the model execution.
8. Use a text editor such as WordPad (or Notepad) to create a batch file named `dos_box.bat` in the code directory. See the figure.

Dos_box.bat File



Note: Setting these environment variables redirects the standard input and output to files, making it possible to post-process the results.

9. Select **Tools** > **Open GBA** from the MicroC Profile window.
10. Click **Execute Compiled Code** from the MicroC Profile window.
11. Within the Run Command window, select the `dos_box.bat` file.

    A Command Prompt window opens as a result. See the following figure.

12. Enter the name of the executable image (in this case, `traffic_light_uc.exe`).

13. Click the **CHANGE** push button any number of times when the panel is invoked.

Note: Each time you click the **CHANGE** push button, information is written to the files `TESTDRIVER_INPUT_FROM_PNL_FILE.txt` and `TESTDRIVER_OUT_FILE.txt`. These files can be viewed once the model execution has been stopped.

The files are both of a similar format and are explained in the following tables.

Time	Variable Name	Variable Type	Variable Value	Comments
Absolute (Default) or Relative	As in Properties	See the following table.	The new value it is set to.	As shown in file.

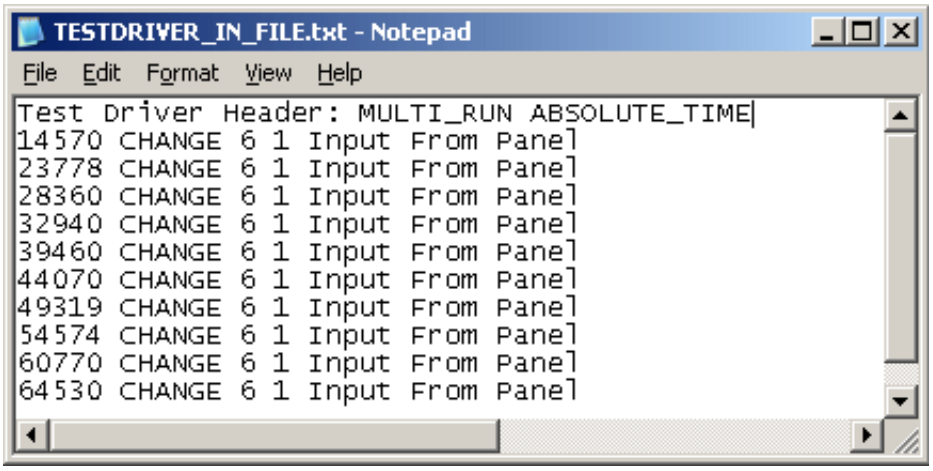
Data Type	Identifier
INTEGER_DATA_ITEM	0
REAL_DATA_ITEM	1
BIT	2
BIT_ARRAY	3
STRING_DATA_ITEM	4
CONDITION	5
EVENT	6
STATE	7
ACTIVITY	8
ENUM_DATA_ITEM	9

14. Edit the `dos_box.bat` file to enable the line that was commented out.

15. Rename the file in the code directory called `TESTDRIVER_INPUT_FROM_PNL_FILE.txt` to `TESTDRIVER_IN_FILE.txt`.

16. Open the file that was just renamed and modify the header (first line) of the file. See the figure.

Modified TestDriver In File



17. Click **Execute Compiled Code** from the MicroC Profile window.

18. Select the `dos_box.bat` file in the Run Command window. A Command Prompt window opens.

19. Enter the **name** of the executable image (in this case, `traffic_light_uc.exe`).

20. When the panel is invoked, the `TESTDRIVER_IN_FILE.txt` is responsible for providing the inputs to the system. The files `TESTDRIVER_INPUT_FROM_PNL_FILE.txt` and `TESTDRIVER_OUT_FILE.txt` is created. These files can be viewed once the model execution has been stopped.