

BSOA Orchestra

BPEM Process Monitor Overview

BULL SERVICE-ORIENTED
ARCHITECTURE (BSOA)



REFERENCE
86 A2 56ER 01

BULL SERVICE-ORIENTED ARCHITECTURE (BSOA)

BSOA Orchestra

BPEM Process Monitor Overview

BSOA Orchestra v3.0

Software

November 2006

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE
86 A2 56ER 01

The following copyright notice protects this book under Copyright laws which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright © Bull SAS 2006

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

Trademarks and Acknowledgements

We acknowledge the right of proprietors of trademarks mentioned in this book.

Intel® and Itanium® are registered trademarks of Intel Corporation.

Windows® and Microsoft® software are registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark in the United States of America and other countries licensed exclusively through the Open Group.

Linux® is a registered trademark of Linus Torvalds.

The information in this document is subject to change without notice. Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.

Table of Contents

Chapter 1.	Introduction to the BPEM Monitor.....	1
Chapter 2.	Starting the BPEM Monitor.....	3
2.1	Using the Monitoring Tool	3
2.2	Starting BPEM from the Command Line.....	3
2.3	Starting BPEM from a Remote Computer	3
Chapter 3.	Deploying Bpemdemon1 Sample.....	5
Chapter 4.	Short Test Run of Bpemdemon1 Client.....	7
Chapter 5.	BPEM Initial Display and Menu.....	9
Chapter 6.	Process Instance Selection with BPEM	11
6.1	Obtaining the Process Instance Selection List	11
6.2	Selecting the Process Instance	12
Chapter 7.	Basic Graphic View Features.....	13
Chapter 8.	Basic Tree View Features	19
Chapter 9.	Variables Window	21
Chapter 10.	Properties Window	25
Chapter 11.	Label Options	27
11.1	Display Statement #'s.....	27
11.2	Display Loop Counts	28
Chapter 12.	Time Format Options.....	29
Chapter 13.	Refresh Options	31
Chapter 14.	Additional Tutorials	33

List of Figures

Figure 5-1. Initial BPEM Window.....	9
Figure 5-2. BPEM Options Menu	10
Figure 6-1. Setup Option in the Process Console To Change Monitoring Mode.....	11
Figure 6-2. Process Instance Selection List on BPEM Toolbar	12
Figure 7-1. Graphic View of the bpmdemo1 Process.....	13
Figure 7-2. Example Tooltip Display Showing Attributes of a BPEL Statement	14
Figure 7-3. Display of Subordinate Child Nodes in BPEM Graphic View	15
Figure 7-4. Display of Forking for Parallel Execution Paths.....	16
Figure 7-5. "Zoom" Capability of Graphic Display Window	17
Figure 8-1. Basic Tree View.....	19
Figure 8-2. Tree View Showing Icon for Unexecuted Statements	20
Figure 9-1. Variables Window	21
Figure 9-2. Expanded Variable Block Statement.....	22
Figure 9-3. Global Variables Block.....	23
Figure 10-1. Properties Window.....	25
Figure 11-1. Sample Graphic Display with Statement #s	27
Figure 11-2. Example Loop Count Display	28
Figure 12-1. "Time Format: Full" from the Options Menu	29
Figure 12-2. "Time Format: short" from the Options Menu	30
Figure 13-1. "Refresh" Portion of the Options Menu	31

Preface

The **B**usiness **P**rocess **E**xecution **M**onitor (BPEM) is a standalone tool that allows for both dynamic, real-time monitoring of a running BPEL process, as well as analysis of final-state information for a BPEL process that has terminated.

Chapter 1. Introduction to the BPEM Monitor

The BPEM monitor is a tool that provides assistance with developing, debugging, and deploying BPEL applications. While the BPEL engine itself executes BPEL processes, it is often necessary to gain information about a BPEL process that has terminated or a BPEL process that is in execution. This is the role of BPEM. The monitor provides both static and dynamic views of the state of BPEL applications that are executed by the Bull BPEL engine.

The BPEM monitor is useful in the following circumstances:

- A BPEL application has terminated and the final results need to be examined.
- A BPEL application has given incorrect results and the path of execution and final variable values need to be examined.
- A BPEL application seems to be "stalled" and its current state needs to be examined.
- A BPEL application is being developed and assistance is needed in debugging it.
- A BPEL application is performing slowly and timing information is required.

The BPEM monitor accesses state information in the BPEL engine. The state information represents a specified BPEL process instance. This data is then displayed in meaningful ways to assist the user in understanding the BPEL process state.

Chapter 2. Starting the BPEM Monitor

2.1 Using the Monitoring Tool

- Launch the BPEL engine with the following command line:
`bsoap start`
- Or use the Windows® Start menu (Start\Programs\BSOA\BSOA Start).

2.2 Starting BPEM from the Command Line

- Start BPEM with the following command line:
`bsoap bpem`
- Or use the Windows® Start menu (Start\Programs\BSOA\BPEL Monitoring Tool\BPEM).

2.3 Starting BPEM from a Remote Computer

From a web browser, enter the following URL:

```
http://"hostname":"port"/bpem/BPEM.jnlp
```

The above URL IS CASE SENSITIVE. The "hostname" part should be the computer running Orchestra, and the "port" part is the port #, usually 9000.

The BPEM monitor program will start and display its main window.

Chapter 3. Deploying Bpemdemo1 Sample

A sample BPEL application will be used for this tutorial.

If the "bpemdemo1" sample has not been deployed, then it must be done now.

The following command line will deploy the bpemdemo1 sample:

```
bsoap deploy -p bpemdemo1 -samples
```

This makes the bpemdemo1 model available for execution. It can be executed multiple times without redeployment.

Chapter 4. Short Test Run of Bpemdemon1Client

An instance of the bpemdemon1 process can now be placed into execution by running a client that connects to it. This client program is provided. A single command line integer is passed that represents the number of seconds to delay in the loops within the process. For a quick run of the process, enter the following command line (all on one line):

```
bsoap launch -p bpemdemon1 -cc  
org.objectweb.orchestra.samples.bpemdemon1.Bpemdemon1Client 1
```

This will produce the following output indicating that the process instance was executed successfully:

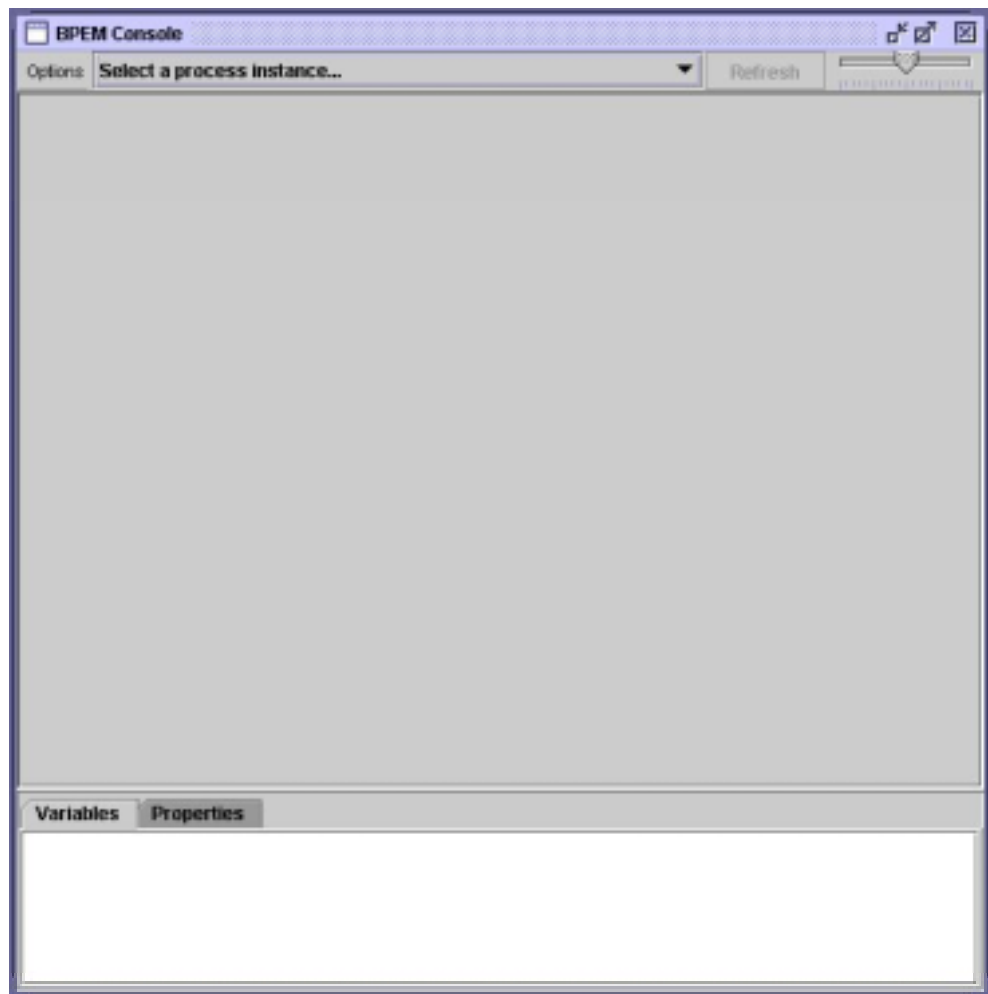
```
[java] ClientContainer.info : Starting client...  
[java] Loops will delay for: 1 seconds each...  
[java] Loops delayed for: 1 seconds each.
```

This tutorial will now view the final state of the BPEL process that was just executed. To use BPEM to monitor the dynamic state of a running BPEL process, see the [BPEM Process Monitor - Bpemdemon1 Tutorial](#).

Chapter 5. BPEM Initial Display and Menu

Select the BPEM window that was started previously. The initial BPEM displays as follows.

Figure 5-1. Initial BPEM Window



The BPEM display is divided into the following areas:

BPEM toolbar: This horizontal area just below the title bar contains the "Options" menu, Process instance selection drop down list, Manual refresh button, and zoom control for graphic sizing.

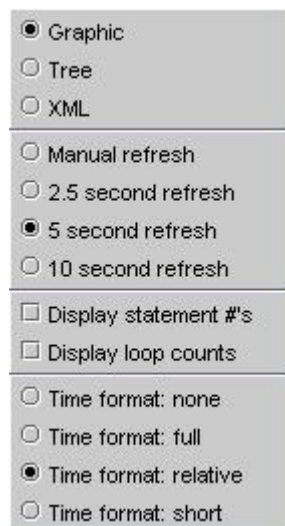
Main display area: The central largest display area is used to display a representation of the selected BPEL process.

Variables/Properties window: This bottom tab panel displays variables and properties in table format.

The ratio of space used by the top and bottom windows can be changed by positioning the mouse pointer over the horizontal divider and moving it up or down as desired.

BPEM Options menu: The Options menu provides items to control various characteristics of the program. The full Options menu is shown left. The use and effect of each menu item will be described later in this tutorial.

Figure 5-2. BPEM Options Menu



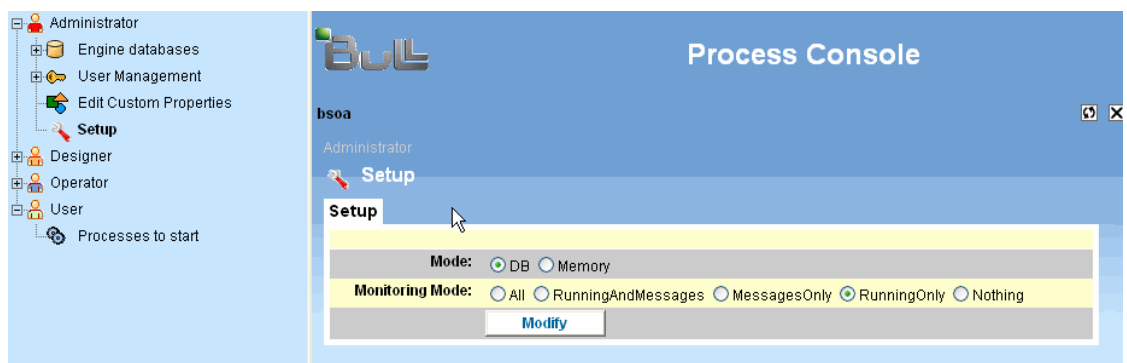
Chapter 6. Process Instance Selection with BPEM

6.1 Obtaining the Process Instance Selection List

To obtain the Process Instance Selection list, the following steps are required.

1. From the BPEL Administration Console, select **Administrator** → **Setup** → **Monitoring Mode** and change the Monitoring Mode from "Running Only" to "All", as show in the following figure.

Figure 6-1. Setup Option in the Process Console To Change Monitoring Mode



2. Run the launch command several times in order to see more than one finished process instance when using the "BPEM Select an instance..." button described next.

6.2 Selecting the Process Instance

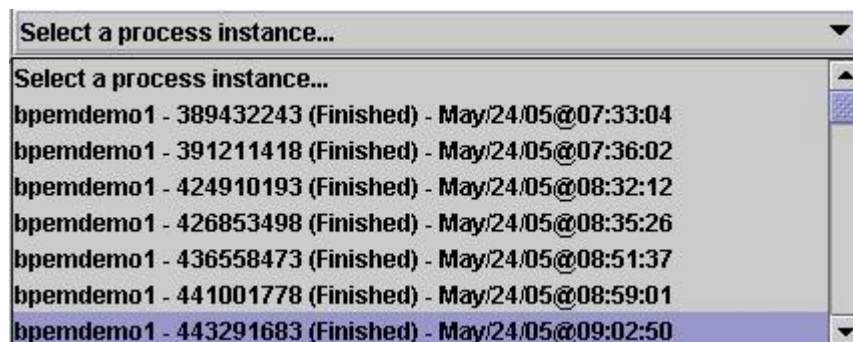
In the middle of the BPEM toolbar is the Process Instance Selection list. Clicking on it will cause the list to open. The list contains a single entry for every BPEL process instance currently known to the Bull BPEL engine. This list is updated whenever a refresh occurs, either manual or automatic; thus as the BPEL engine creates new BPEL process instances, this list can be updated.

Each line has four parts.

- **Process model name:** This is the name of the model (bpel file) used to create this instance.
- **Instance ID:** This is the ID assigned to the instance by the BPEL engine.
- **Process State:** This is the current state of the process instance.
- **Process creation time:** This is the time the instance was created in the BPEL engine.

An example is shown below.

Figure 6-2. Process Instance Selection List on BPEM Toolbar



For this part of the tutorial, scroll through the list and find the bpemdemo1 instance that was last created (observe the date/time at the end of each entry).

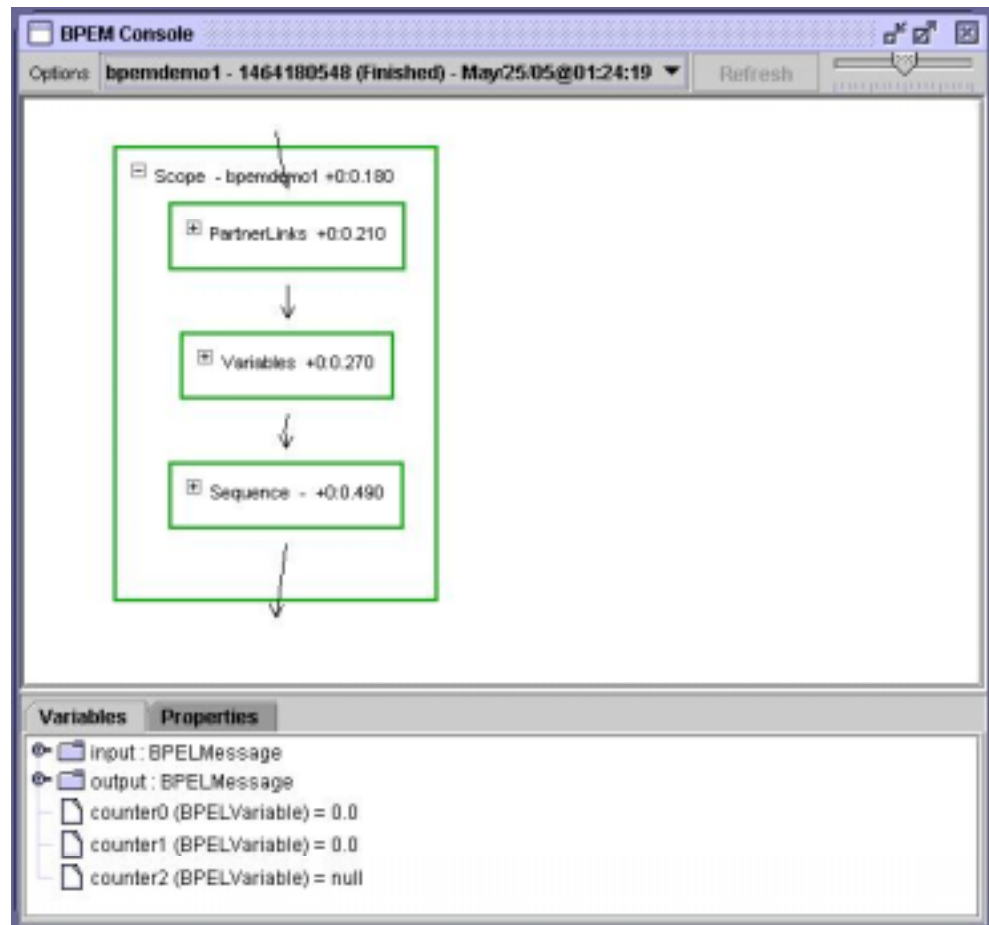
If the bpemdemo1 instance cannot be found, then it is likely that the **"Manual Refresh"** mode is enabled: click the **"Refresh"** button to force updates.

The process should be marked as **"(Finished)"**, indicating that the instance is no longer executing. However, the BPEL engine can keep the instance data available for analysis by BPEM, even though the process is no longer executing.

Chapter 7. Basic Graphic View Features

The initial display should be similar to the following.

Figure 7-1. Graphic View of the bpmdemo1 Process

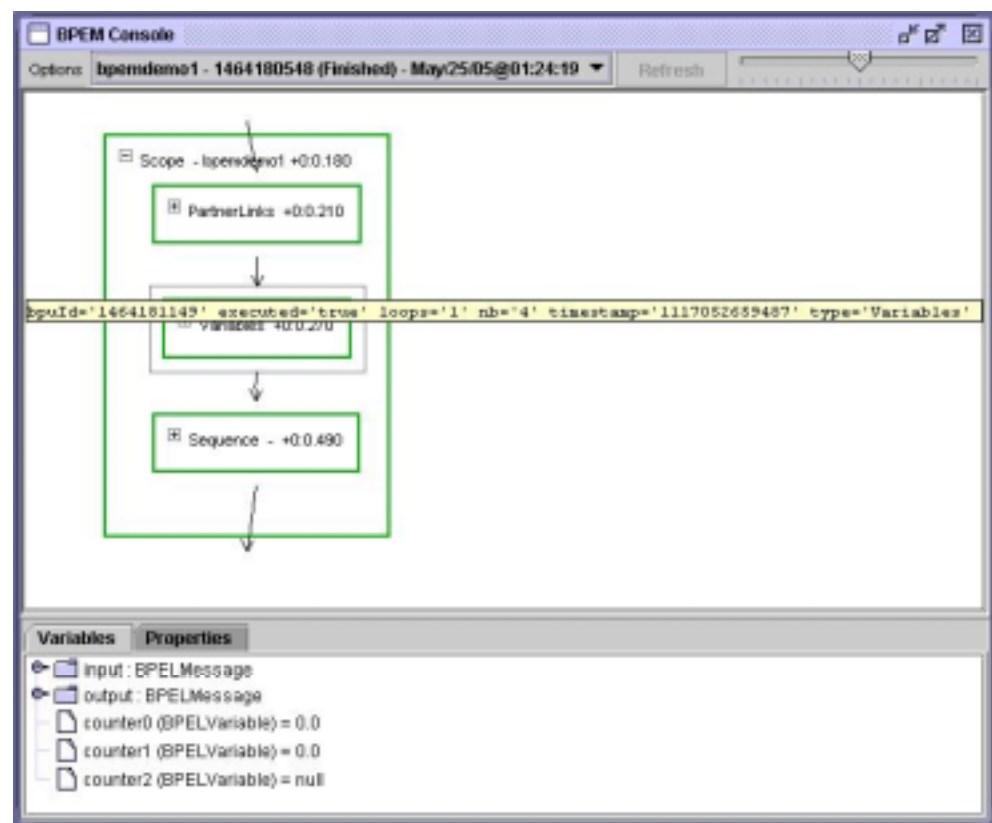


The graphic view is a visual representation of the bpmdemo1 process. Any BPEL source file is in structured xml, which has implicit parent/child relationships that can be displayed as "node within container" style graphics.

The initial display above shows the top-level scope of the process, which contains PartnerLinks, Variables, and a Sequence. These correspond directly to the same elements in the BPEL file for the process. Each statement is represented by either a red or green box. Green is used for statements that have been executed, and red for statements that have not been executed. The connecting arrows represent the sequential flow of execution. The ability to expand and contract each node is a key to reducing the complexity of the display, while still allowing for the detailed analysis of the area of interest.

Positioning the mouse over any displayed node, such as the "Variables" node, does two things: the node is highlighted by the presence of a black surrounding rectangle, and a tooltip will be displayed. The tooltip will have the xml attributes for the corresponding BPEL statement, as well as some additional attributes used by the BPEL engine. The tooltip is shown below.

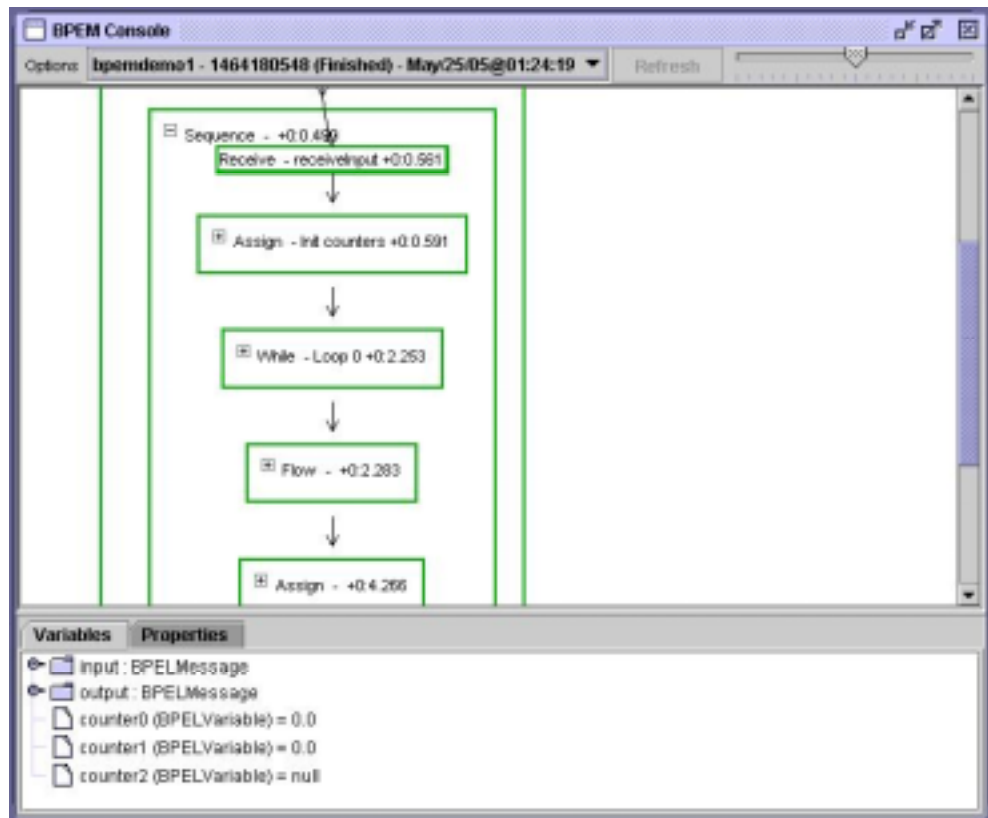
Figure 7-2. Example Tooltip Display Showing Attributes of a BPEL Statement



In some cases, it is desirable to position the mouse so that no tooltip is displayed. Moving the mouse within the graphic window to an area that is not within a node will cancel any tooltip. If the display is extremely complicated, positioning the mouse over the whitespace on the leftmost side of the graphic area will do this.

Nodes containing a "+" contain subordinate child nodes. This means that the corresponding BPEL xml has additional child tags within the statement. Left clicking on a "+" will cause the child nodes to be displayed and the "+" is changed to a "-" to allow it to be retracted. Left clicking on the "+" in the "Sequence" node and scrolling slightly, will display the following.

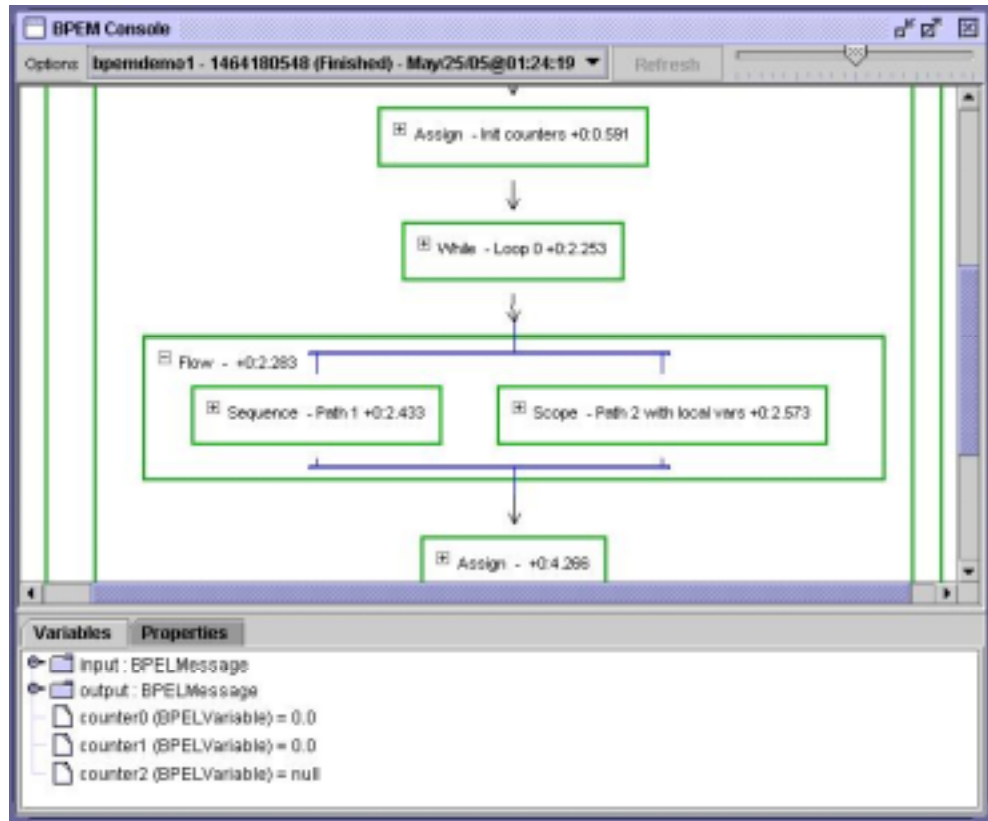
Figure 7-3. Display of Subordinate Child Nodes in BPEM Graphic View



Notice here that the node for the While "loop0" can be seen followed by the "flow." Left clicking the "+" on the flow expands it. This also demonstrates another feature of the graphic display.

As seen below, a horizontal blue line is used to represent the initial forking for parallel execution paths, and the final joining to end parallel execution paths

Figure 7-4. Display of Forking for Parallel Execution Paths

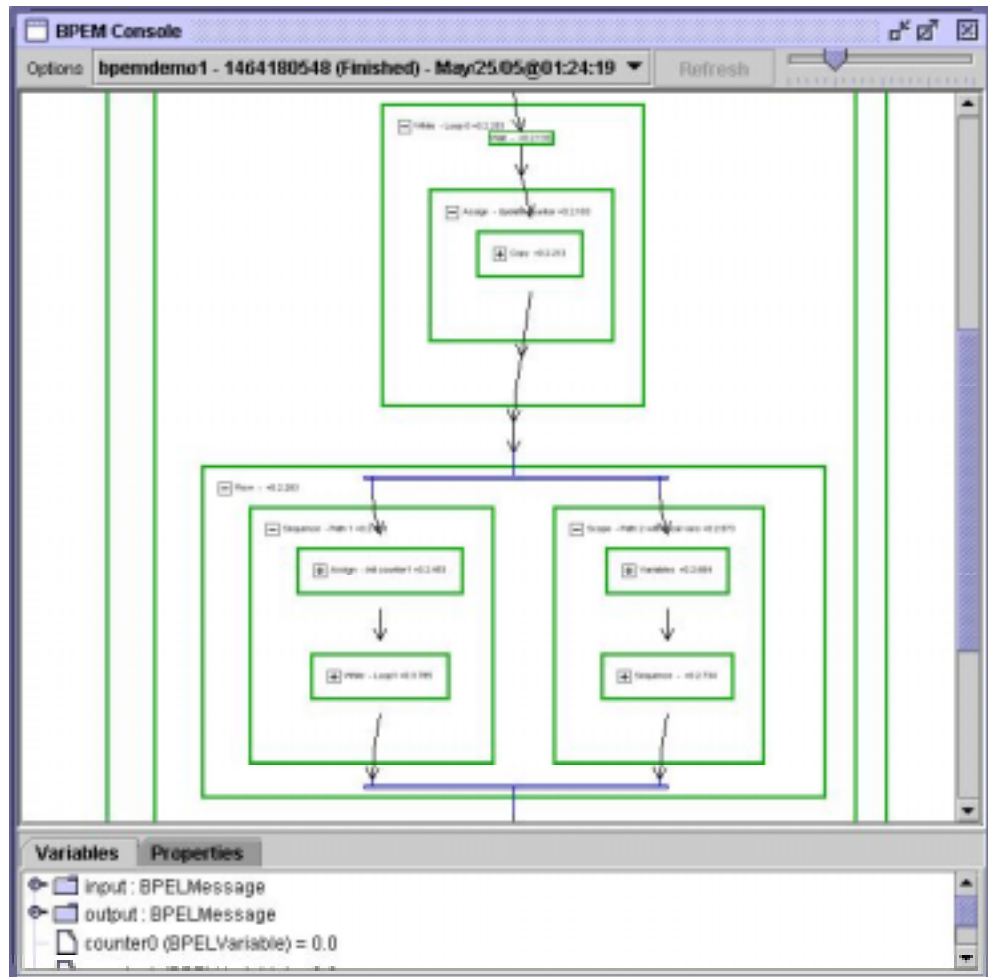


At this point, try expanding/contracting several nodes and scroll around the display to become familiar with it.

A final feature of the graphic window is the **"zoom"** capability. The graphic zoom slider is at the right end of the BPPEM toolbar. Moving the slider to the left causes the graphic to shrink, allowing more to be displayed in the same space. Moving the slider to the right causes magnification of the nodes.

The following is an example with the slider at about 25% original size.

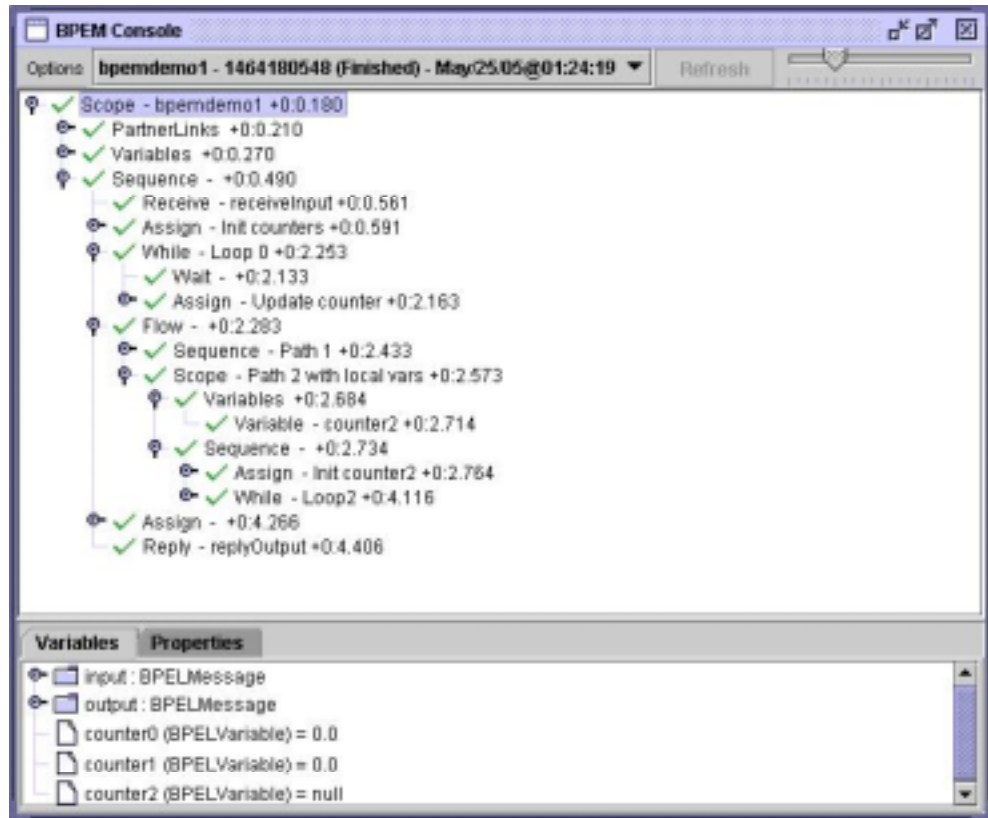
Figure 7-5. “Zoom” Capability of Graphic Display Window



Chapter 8. Basic Tree View Features

Switch to the Tree view by selecting "Tree" in the Options menu. Expanding a few nodes of the tree will result in a display similar to the following.

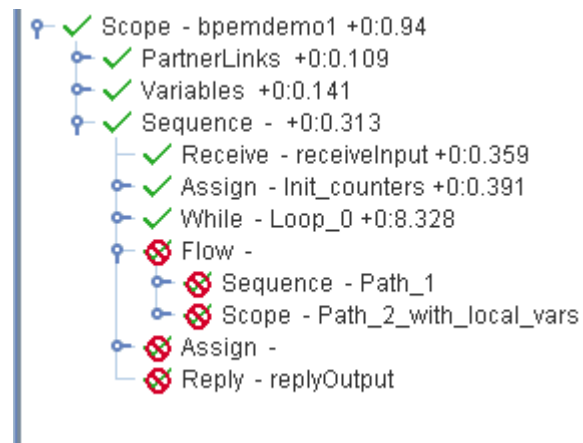
Figure 8-1. Basic Tree View



Again, because a BPEL process is described by an xml file, it makes sense to display it as parent/child nodes in a Tree control. Each node in the tree corresponds to a BPEL statement. The green checkmark icon is used for statements that have been executed by the BPEL engine. For statements that have not been executed, the checkmark will be overlaid with a red circle/line.

The following figure provides an example that shows the red circle overlaying the green check marks.

Figure 8-2. Tree View Showing Icon for Unexecuted Statements



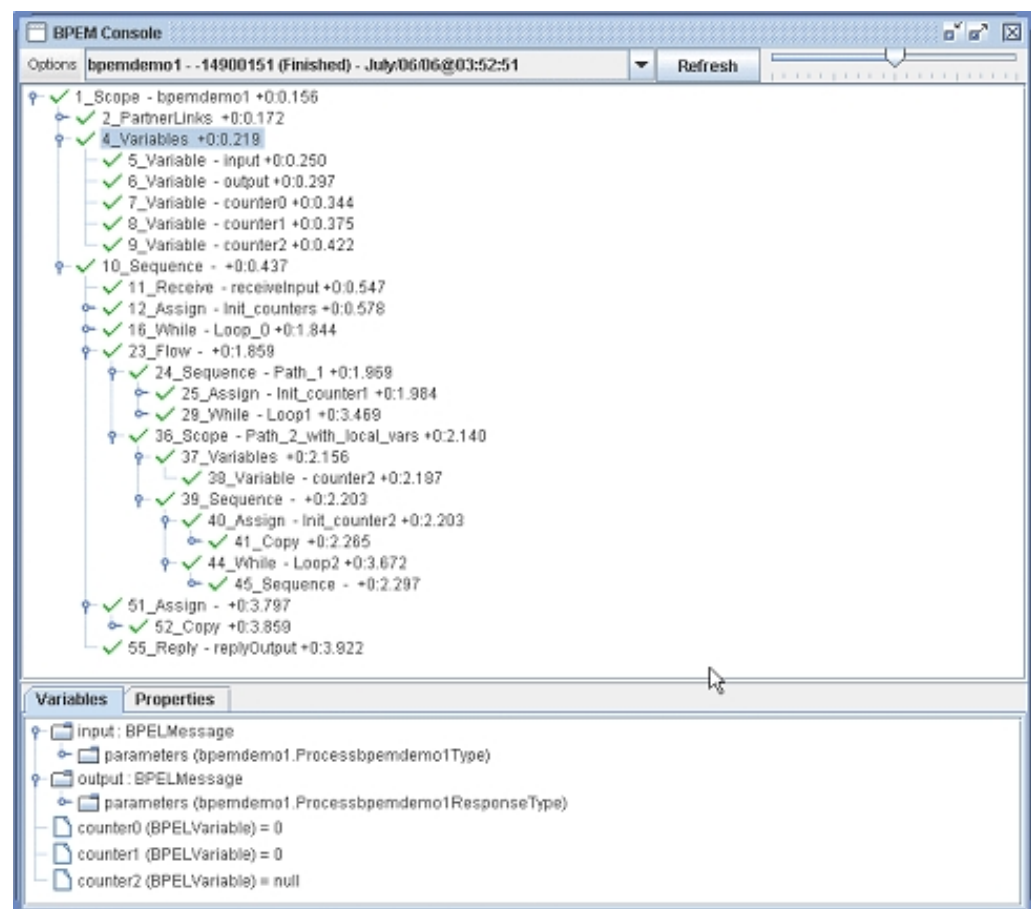
When the mouse pointer is positioned over a node, a tooltip is displayed showing all the attributes for that BPEL statement, as well as additional attributes created by the BPEL engine.

Chapter 9. Variables Window

The "Variables" window allows the inspection of all variables used within the BPEL process. All variables declared within a given "Scope" are displayed.

The default display will show the variables for the outermost "global" scope. For the purposes of clarity in the tutorial, at this time, continue displaying the Tree view as is shown above, and also use the "Options" menu and select "Display statement #'s" (this option will be described later). Each line in the tree is now prefixed with a statement number. This will assist in further discussion. The display will now look as follows.

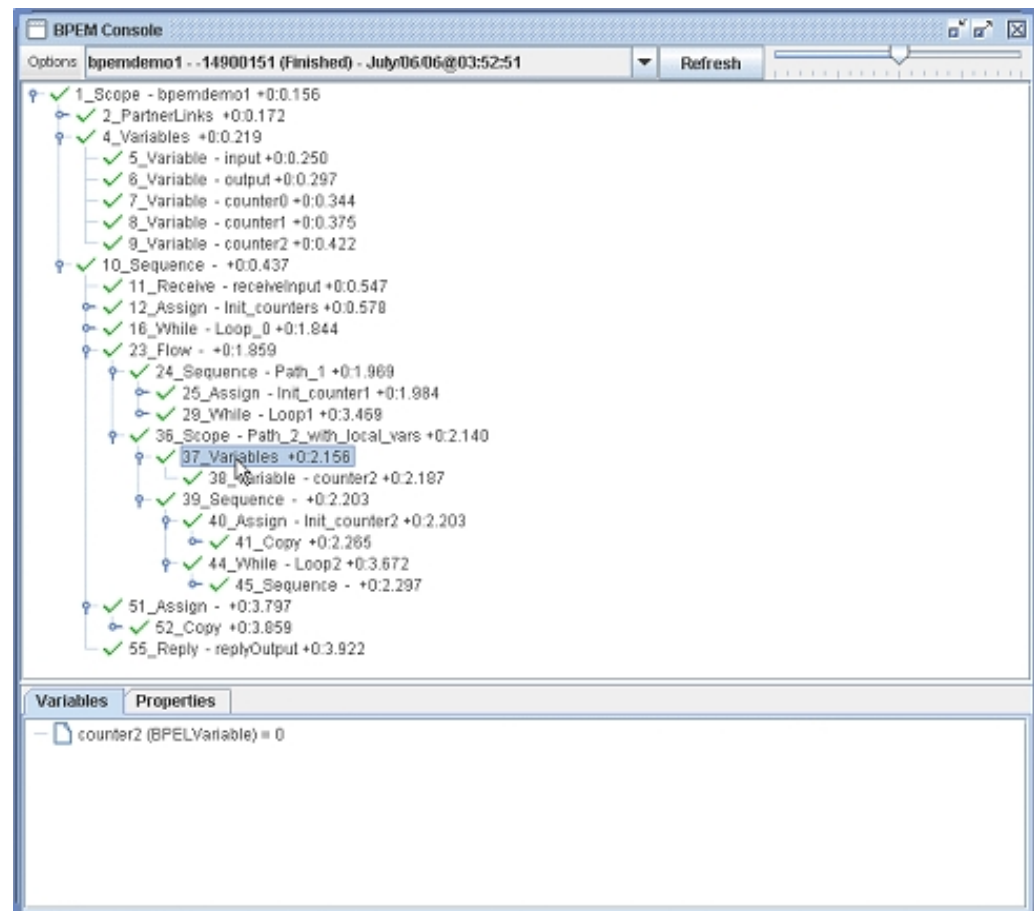
Figure 9-1. Variables Window



Notice that this process contains the global scope (statement #1) with a variable block definition at statement #4. There is also a nested scope at statement #36 with a variable block at statement #37.

Expand the variables block at statement # 4. These are the variables displayed in the bottom variables window. Selecting any statement that is NOT within the nested scope at statement #36 will display this block of global variables. Now select any statement within the nested scope. For example, selecting statement #37 results in the following display.

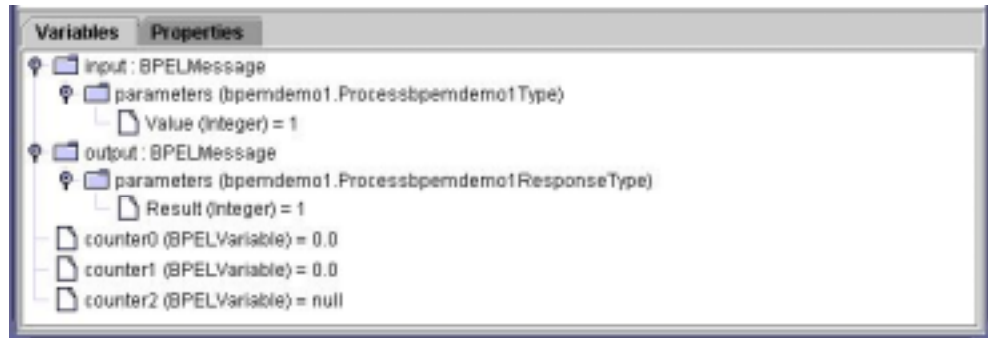
Figure 9-2. Expanded Variable Block Statement



Notice that the "**Variables**" panel now displays the variable block from the nested scope. Click on several statements between the two scopes to get a better feel for how this works.

Now select a statement in the global scope to see the global variables block. Expand all variables in the **"Variables"** panel, which should now look like the following.

Figure 9-3. Global Variables Block



There are two main types of variables. "Simple" variables like ints and strings are displayed on a single line as expected. Complex nested variable structures, as are often used in soap messages, are displayed level-by-level until final, simple variable elements are found. Notice that the input.parameters.value of "1" above is the command-line parameter that was used to start this instance of the bpemdemo1 process. Also note that the global "counter2" is uninitialized because it is not used in this process; it is hidden by the nested scope.

The Variables window works the same regardless of the main display (graphic or tree). Switch back to the graphic display by selecting the "Graphic" item in the Options menu and select various nodes. Notice that the variables window always displays the variable block for the scope containing the selected statement.

Chapter 10. Properties Window

The tabs at the top of the bottom panel can be used to switch between the Variables and Properties windows. Selecting the Properties tab will display the properties for the currently selected BPEL statement. This is useful for getting detailed information about a specific statement. For example, clicking on the "While Loop0" statement will display its properties, which allows examination of the exit condition and other attributes of the statement. This is shown below.

Figure 10-1. Properties Window

Variables Properties	
Name	Value
bpuid	1464181140
condition	bpws.getVariableData('counter0') > 0
executed	true
joinCondition	
loops	2
name	Loop 0
nb	18
suppressJoinFailure	
timestamp	1117052661470
type	While

Chapter 11. Label Options

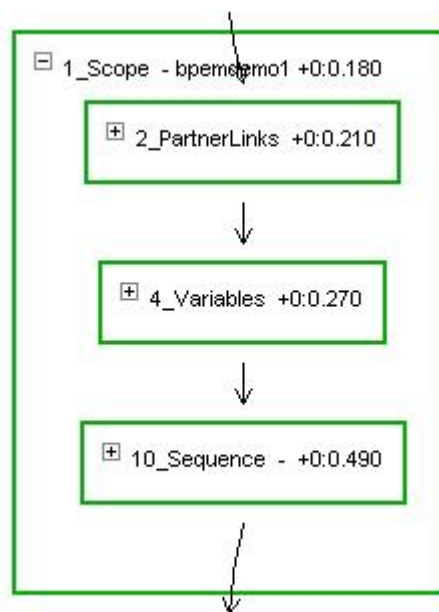
The text displayed for each node can be changed with menu options. The same text is displayed for labels in both the graphic and text views. For this part of the tutorial, switch to the graphic view.

11.1 Display Statement #'s

The use of the "Display statement #'s" menu item has already helped to clarify points of the tutorial. When this option is set, the statement text is prefixed with the statement # that is assigned by the BPEL engine. This value is NOT the BPEL source code line #. Rather, it corresponds to the xml tag parsing of the BPEL, and can be thought of as representing the tag "index" of the statement.

The following is a sample of the graphic display with statement #'s displayed.

Figure 11-1. Sample Graphic Display with Statement #s

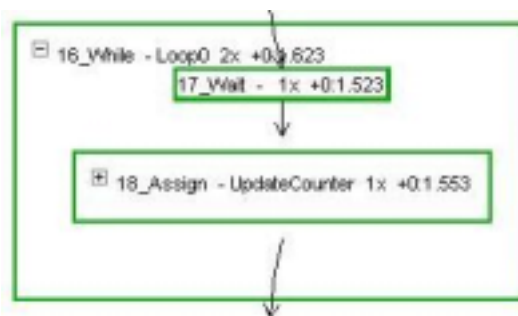


11.2 Display Loop Counts

The Display Loop Counts choice in the "Options" menu causes the statement text to contain an integer representing the number of times the statement was executed by the BPEL engine. This is most useful for statements that are contained in loops.

An example of the loop count display is shown below. Note that the "While" statement #16 was executed two times and the statements within the loop were only executed once. This is because the While statement must be executed again to test the exit condition of the loop.

Figure 11-2. Example Loop Count Display



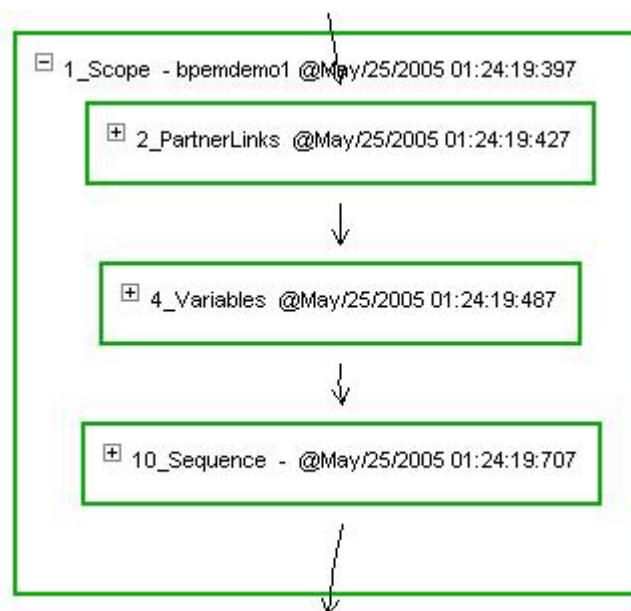
Chapter 12. Time Format Options

If a statement has been executed by the BPEL engine, then 1) its graphic display is green, and 2) the node label ends with a value representing the timestamp that the BPEL engine initiated execution of the statement. There are several formats in which this value can be displayed, selectable in the "**Options**" menu:

"Time Format: relative": This is the default time display format. This is shown above. In this format, the value displayed on each statement is the time elapsed since the beginning of the process. The format used is hh:mm:ss.millis. In the above display, statement #3 was executed 0.490 seconds after the BPEL process started.

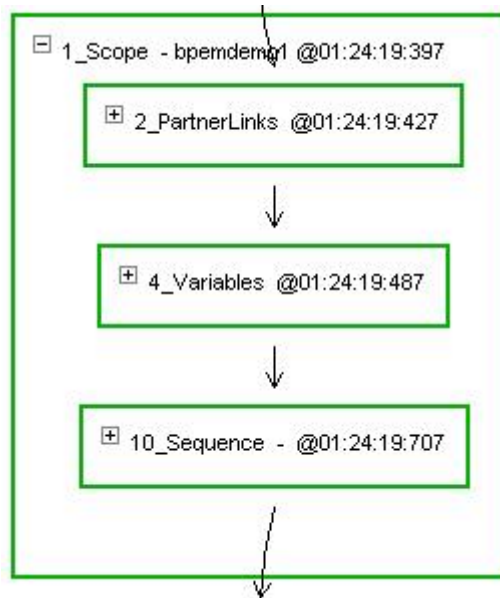
"Time Format: full": This option causes the full, time value to be displayed as shown below.

Figure 12-1. "Time Format: Full" from the Options Menu



"Time Format: short" - This options causes just the time of day to be displayed as shown below.

Figure 12-2. "Time Format: short" from the Options Menu



"Time format" none - This option causes the execution time to not be displayed on the node labels.

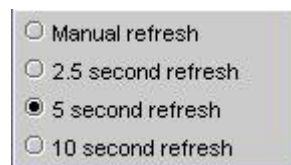
Chapter 13. Refresh Options

The BPEM process can periodically request updated information from the BPEL engine. The rate of this refresh can be controlled with the "**Refresh**" item in the Options menu. All display items in the program are refreshed as follows:

- Process instance list: This list is updated on every refresh cycle. This means that as BPEL processes are created, it is not necessary to restart the monitor for them to appear in the list. However, to prevent confusing operation, when the BPEM process list is expanded/dropped, it is not updated on refresh cycles.
- If the process being displayed is not "Active," then there is no need to update any of the displays once the initial images are loaded.
- If the process being displayed is "Active," then each refresh cycle will cause all display elements to be updated. All labels/text are updated for execution time and loop counts, and colors are updated accordingly. In addition, all variable and property displays are updated as needed.

The refresh part of the Options menu is shown:

Figure 13-1. "Refresh" Portion of the Options Menu



Manual refresh: If this option is selected, then the "**Refresh**" button on the BPEM toolbar is enabled and can be pressed at any time to force a refresh. Manual refresh is the default.

Automatic refresh rates: These options enable automatic refresh at the specified intervals.

Chapter 14. Additional Tutorials

This overview tutorial has discussed the use of BPEM to analyze the final state of a terminated BPEL application. The following tutorials demonstrate the use of BPEM to analyze a running BPEL application.

See the [BPEM Process Monitor - Bpemdemo1 Tutorial](#) to observe multiple paths of execution and dynamically changing variable values.

See the [BPEM Process Monitor - MarketPlace Tutorial](#) to observe complex variable values as well as a BPEL application that is waiting for additional user input.

Technical publication remarks form

Title: BSOA Orchestra

Reference No.: 86 A2 56ER 01

Date: November 2006

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.

If you require a written reply, please include your complete mailing address below.

NAME: _____ Date: _____

COMPANY: _____

ADDRESS: _____

Please give this technical publication remarks form to your BULL representative or mail to:

Bull - Documentation Dept.
1 Rue de Provence
BP 208
38432 ECHIROLLES CEDEX
FRANCE
info@frec.bull.fr

Technical publications ordering form

To order additional publications, please fill in a copy of this form and send it via mail to:

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

Phone: +33 (0) 2 41 73 72 66
FAX: +33 (0) 2 41 73 70 66
E-Mail: srv.Dupilcopy@bull.net

CEDOC Reference #	Designation	Qty
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
__ __ __ __ __ __ __ [__]		
[__] : The latest revision will be provided if no revision number is given.		

NAME: _____ Date: _____

COMPANY: _____

ADDRESS: _____

PHONE: _____ FAX: _____

E-MAIL: _____

For Bull Subsidiaries:

Identification: _____

For Bull Affiliated Customers:

Customer Code: _____

For Bull Internal Customers:

Budgetary Section: _____

For Others: Please ask your Bull representative.

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

REFERENCE No.
86 A2 56ER 01