

BSOA Orchestra

BPEM Process Monitor -
Bpemdemo1 Tutorial

BULL SERVICE-ORIENTED
ARCHITECTURE (BSOA)



REFERENCE
86 A2 57ER 01

BULL SERVICE-ORIENTED ARCHITECTURE (BSOA)

BSOA Orchestra

BPEM Process Monitor - Bpemdemo1 Tutorial

BSOA Orchestra v3.0

Software

November 2006

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

REFERENCE
86 A2 57ER 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.	Bpemdmo1 BPEL Process Overview	1
1.1	Variables at the Outer "Global" Scope.....	1
1.2	Store Input Data	2
1.3	First Loop ("loop0")	3
1.4	Definition of Two Control Paths.....	3
1.5	First Parallel Path	4
1.6	Second Parallel Path	5
1.7	Summary of Bpemdmo1 BPEL Process.....	6
Chapter 2.	Starting BPEM Monitor	7
Chapter 3.	Deploying Bpemdmo1 Sample.....	9
Chapter 4.	Running the Bpemdmo1Client.....	11
Chapter 5.	Process Instance Selection with BPEM	13
5.1	Obtaining the Process Instance Selection List	13
5.2	Selecting a Process Instance.....	14
Chapter 6.	Observations for Sequential Program Loop	15
Chapter 7.	Observations for Parallel Program Loop	17
Chapter 8.	Repeating Using the Tree Display	19
Chapter 9.	Additional Tutorials	21

List of Figures

Figure 5-1.	Setup Option in the Process Console To Change Monitoring Mode.....	13
Figure 5-2.	Confirmation Screen for Changing Monitoring Mode	13
Figure 5-3.	List of BPEL Process Instances Know to the BPEL Engine	14
Figure 6-1.	Expanded Display of Statements for Sequential Program Loop	15
Figure 7-1.	Display of Two Parallel Flow Paths.....	17
Figure 8-1.	Tree Display of Parallel Program Loops	19

Preface

This tutorial will use the BPEM process monitor to observe the behavior of the `bpemdemo1` BPEL sample process. Several scenarios will be used to highlight the features and uses of the monitor while observing a running BPEL process.

The BPEM process monitor 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. For a complete description of BPEM, see [BPEM Process Monitor Overview](#).

Chapter 1. Bpemdemon1 BPEL Process Overview

This tutorial uses a BPEL process that contains several loops. While a normal BPEL process may not typically contain a "delay" loop, it is useful in this demo in order to demonstrate the real-time data maintained by the Bull BPEL engine and displayed in the monitor. In real-world scenarios, these "delay" loops would typically be "invoke" statements that request data from other web services.

Most BPEL statements within `bpemdemon1.bpel` contain a "name" attribute for easy reference. The main points of the process are as follows.



Note:

This section provides several XML code examples. To simplify the formatting of the display, the xml opening ("`<`") and closing brackets ("`>`") for each element are not shown. The complete XML file can be found at:

[\\$BPEL_HOME/ samples/bpemdemon1/bpemdemon1.bpel](#)

1.1 Variables at the Outer "Global" Scope

This variables block defines the data storage for the BPEL process using the `<variable>` element. The following five variables are defined:

```
variable name="input "  
messageType="tns:processbpemdemon1SoapRequest "  
  
variable name="output "  
messageType="tns:processbpemdemon1SoapResponse "  
  
variable name="counter0" type="xsd:int "  
  
variable name="counter1" type="xsd:int "  
  
variable name="counter2" type="xsd:int "
```

1.2 Store Input Data

The "receive" statement allows for the creation of the process instance when the client program is executed. The "assign" statement copies the integer entered on the client command line into the global "counter0" variable for easy reference later.

```
receive name="receiveInput"

    partnerLink="bpemdemon1"

    portType="tns:bpemdemon1PT"

    operation="process"

    variable="input"

    createInstance="yes"

assign name="Init_counters"

    copy

        from expression=

            "bpws:getVariableData('input',

                                   'parameters',

                                   '/value')"

        to variable="counter0"

    /copy

/assign
```

1.3 First Loop ("loop0")

Each pass through this loop delays for one second (using a wait activity), decrementing "counter0" on each pass. The exit condition for the loop is reached when "counter0" is 0. When this loop is entered, the instance will remain in this loop for the number of seconds entered on the command line of the client.

```
while name="Loop_0"

    condition="bpws:getVariableData('counter0') > 0"

    sequence

        wait for="'PT1S'"

        assign name="Update_counter"

        copy

            from expression="bpws:getVariableData('counter0') - 1"

            to variable="counter0"

        /copy

    /assign

/sequence

/while
```

1.4 Definition of Two Control Paths

At this point, the BPEL "flow" statement is used to create two paths of parallel execution. Each of two paths contains a loop similar to the one above, but using different variables.

```
flow name="ParallelPaths"
```

1.5 First Parallel Path

The first parallel path contains the same loop as before, except it uses the global "counter1" variable for the loop control.

```
sequence name="Path_1"

  assign name="Init_counter1"

    copy

      from expression=

        "bpws:getVariableData('input','parameters','/value')"

      to variable="counter1"

    /copy

  /assign

  while name="Loop1"

    condition="bpws:getVariableData('counter1') > 0"

    sequence

      wait for="'PT1S'"

      assign name="Update_counter"

      copy

        from expression=

          "bpws:getVariableData('counter1') - 1"

        to variable="counter1"

      /copy

    /assign

  /sequence

/while

/sequence
```

1.6 Second Parallel Path

The second parallel path contains the same loop, except it is embedded within a local "scope" with a local variable definition block. This local "counter2" variable is used inside this block instead of the same named variable in global scope.

```
scope name="Path_2_with_local_vars"
  variables
    variable name="counter2" type="xsd:int"
  /variables
  sequence
    assign name="Init_counter2"
    copy
      from expression=
        "bpws:getVariableData('input','parameters','/value')"
      to variable="counter2"
    /copy
  /assign
  while name="Loop2"
    condition="bpws:getVariableData('counter2') > 0"
    sequence
      wait for="'PT1S'"
      assign name="Update_counter"
      copy
        from expression=
          "bpws:getVariableData('counter2') - 1"
        to variable="counter2"
      /copy
    /assign
  /sequence
/while
/sequence
/scope
```

1.7 Summary of Bpemdemo1 BPEL Process

From the above description, it can be seen that the bpemdemo1 BPEL process performs the following when executed:

1. Input data is copied to global storage variables.
2. Delay loop named "loop0" is entered and loops for the number of seconds specified on the command line.
3. Parallel flow named "ParallelPaths" is entered.
4. Parallel path named "Path1" executes using global variables.
5. Parallel path named "Path2_with_local_vars" executes with a local variable.

This sequence can now be observed in the monitor application.

Chapter 2. Starting BPEM Monitor

To use the monitoring tool, you must first launch BPEL:

```
bsoap start
```

Then start BPEM with the following command line:

```
bsoap bpem
```

The BPEM monitor program will start and display its main window. See [BPEM Process Monitor Overview](#) for a complete description of BPEM.

This tutorial assumes an "automatic" refresh mode. This is not the default. To enable automatic refresh, select one of the automatic refresh modes in the "**Options**" menu.

Depending on your PC platform and its performance, care should be taken when selecting fast refresh rates. For this tutorial, a 5-second refresh is sufficient.

Chapter 3. Deploying Bpemdemo1 Sample

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. Running the 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. As previously discussed, 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 1 second version of the launch command (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 has successfully executed:

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

In order to have a process run long enough to allow for real-time observation, run the client again with a command line parameter of at least 50, as shown below in the 50 second version of the launch command:

```
bsoap launch -p bpemdemon1 -cc  
org.objectweb.orchestra.samples.bpemdemon1.Bpemdemon1Client 50
```

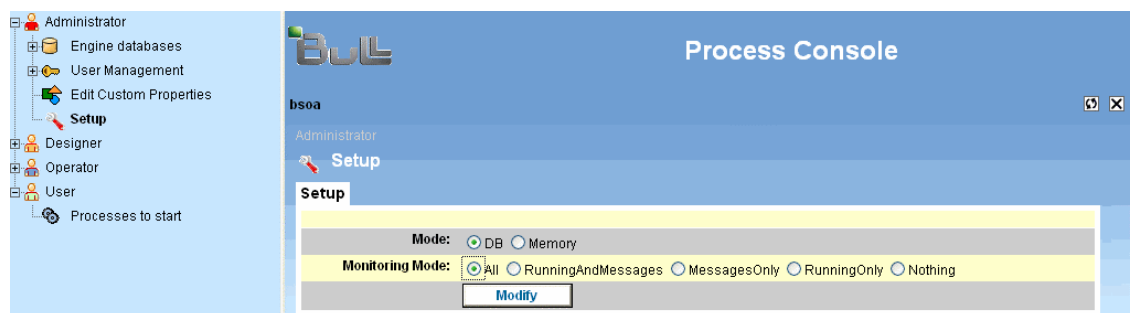

Chapter 5. Process Instance Selection with BPEM

5.1 Obtaining the Process Instance Selection List

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

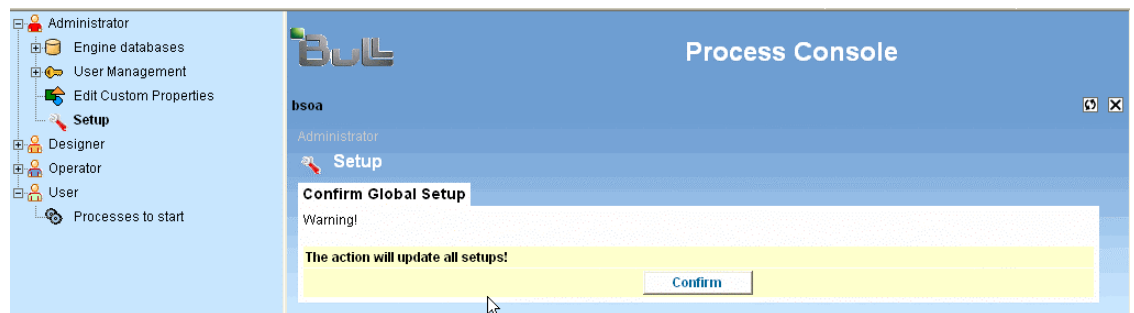
1. From the BPEL Administration Console, select **Administrator** → **Setup** → **Monitoring Mode** and change the Monitoring Mode from "Running Only" to "All", as shown in the following figure.

Figure 5-1. Setup Option in the Process Console To Change Monitoring Mode



A following screen will then display to confirm the change.

Figure 5-2. Confirmation Screen for Changing Monitoring Mode



2. Run the 1-second version of 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.

5.2 Selecting a Process Instance

Select the BPEM application window and open the drop down list in the middle of the top toolbar. This will display all the BPEL process instances known to the BPEL engine. This list is shown below.

Figure 5-3. List of BPEL Process Instances Known to the BPEL Engine



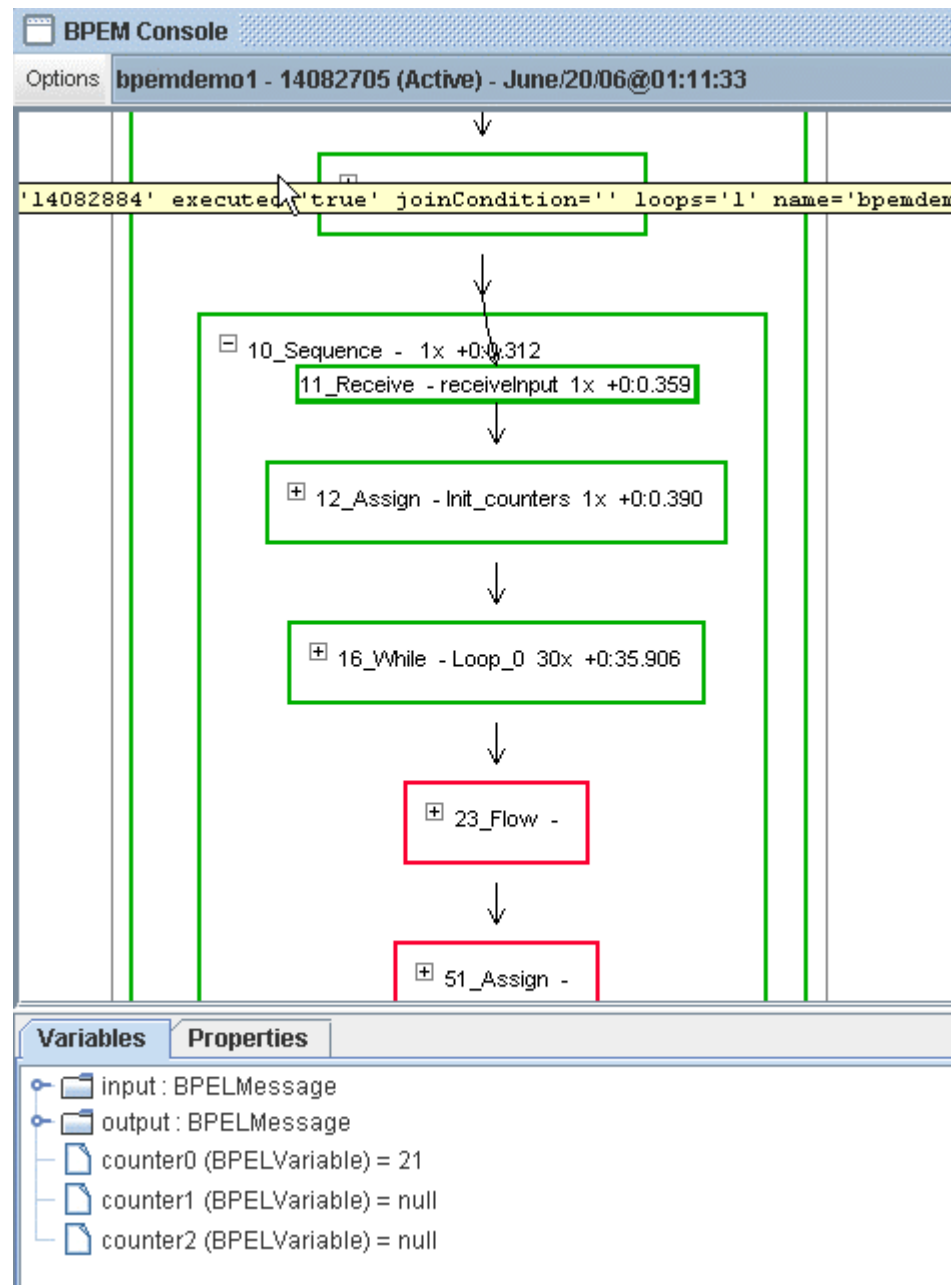
This list is dynamically refreshed. Therefore, as new BPEL-process instances are created by the BPEL engine, this list is updated.

For this part of the tutorial, now run the 50-second version of the launch command, and then scroll through the list and find the bpemdemo1 instance that was last created (observe the date/time at the end of each entry). The process should be marked as "(Active)", indicating that the instance is still executing. If the 50-second instance has terminated, simply restart it using the 50-second version of the launch command.

Chapter 6. Observations for Sequential Program Loop

Use the Options menu to select the display of statement #'s. This will help in the description of the tutorial. Expand the sequence statement (#10) and also the while statement (#16) by clicking on the "+" icons. Scroll the display as needed. The display should be similar to the following.

Figure 6-1. Expanded Display of Statements for Sequential Program Loop



Notice the following:

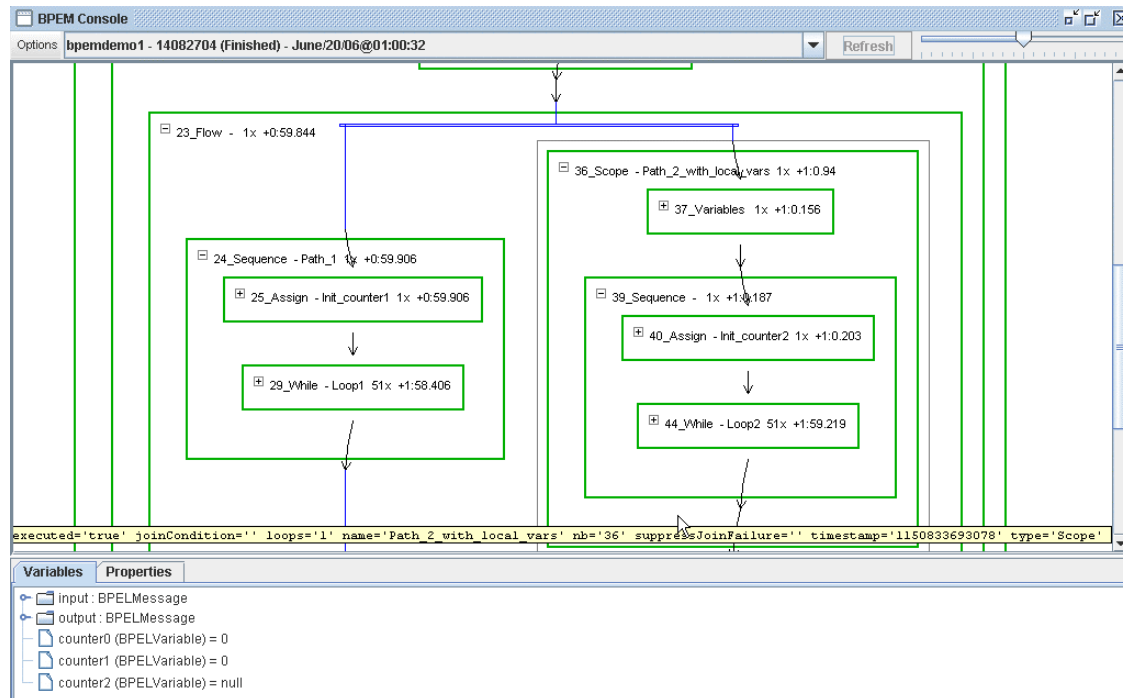
1. The execution time stamps for the statements within the "while" loop are updating as the engine executes the loop.
2. The loop counts on these same statements are also updating.
3. The "counter0" value in the bottom variables window is updating.
4. When "counter0" reaches 0, then the next statement (#22) will be executed and it will turn from red to green.

If the BPEL process instance that was started terminates, then the status in the BPEM instance list will change from "**Active**" to "**Finished**". If this occurs, you can resubmit the process and reselect the new process instance in BPEM to continue the tutorial.

Chapter 7. Observations for Parallel Program Loop

At this point it may be useful to change the refresh rate to 5 seconds depending on the performance of the computer being used. Additionally, it is possible to expand statements #23, #24, #36, and #39 and resize/scroll the display so that the two parallel flow paths can be seen as follows.

Figure 7-1. Display of Two Parallel Flow Paths

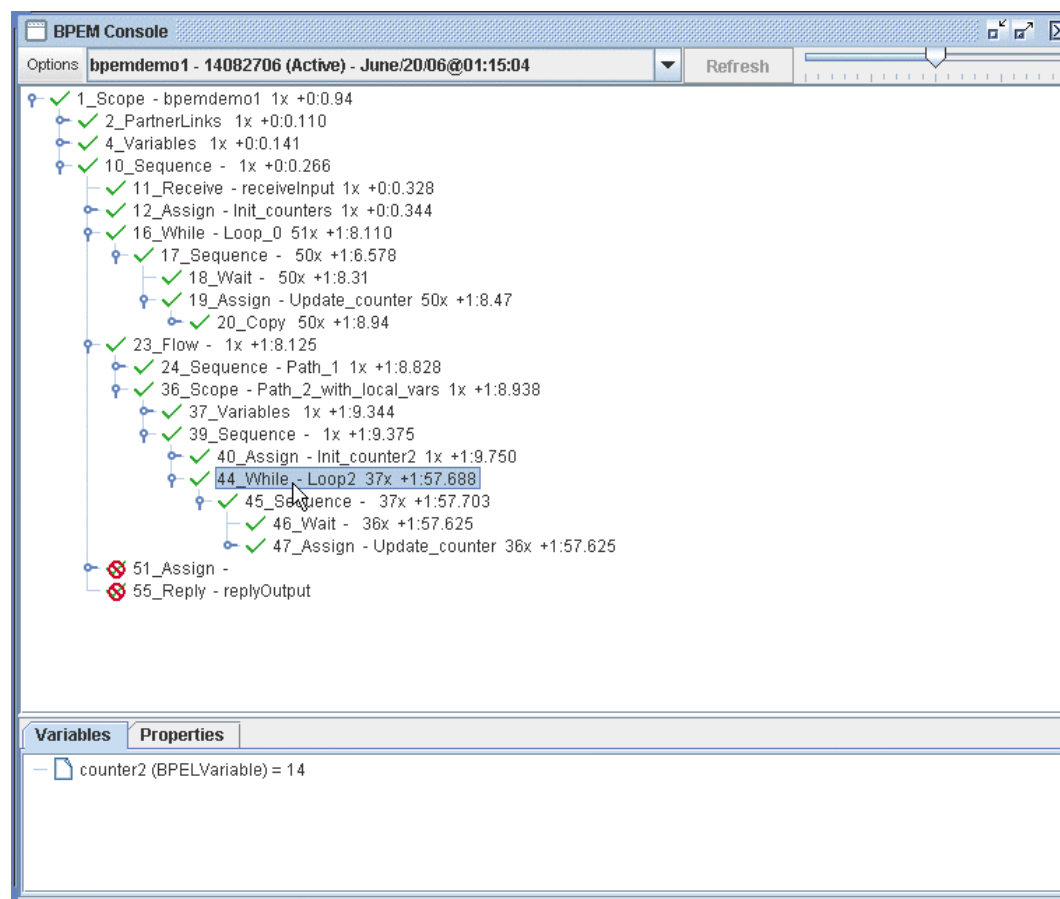


As before, you can observe the updating of the timestamps, loop counts and variable values. Remember that the right side path uses an internal scope; thus you can click on any statement within scope #36 to see the counter in the variables window for that statement.

Chapter 8. Repeating Using the Tree Display

The above scenario can be repeated using the Tree display by selecting the Tree item in the "Options" menu and restarting at Section 4, [Running the Bpemdemon1Client](#).

Figure 8-1. Tree Display of Parallel Program Loops



Chapter 9. Additional Tutorials

This overview tutorial has discussed the use of BPEM to analyze the dynamic state of a running BPEL application. The following tutorials are also available:

See the [BPEM Process Monitor Overview](#) for basic BPEM operation.

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 57ER 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 57ER 01