BSOA Orchestra

Quickstart Tutorial - Create a New BPEL File

BULL SERVICE-ORIENTED ARCHITECTURE (BSOA)

REFERENCE
86 A2 17ET 01

BULL SERVICE-ORIENTED ARCHITECTURE (BSOA)

BSOA Orchestra

Quickstart Tutorial - Create a New BPEL File

BSOA Orchestra v3.0

Software

November 2006

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

REFERENCE
86 A2 17ET 01

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.

Table of Contents

List of Figures

Preface

This tutorial is a guide for using the Zenflow plug-in to create a new BPEL file. (The information in this document applies to the Eclipse plug-in version of Zenflow.)

Chapter 1. Tutorial Overview

This tutorial will guide the user in the use of the Zenflow plug-in to build a new BPEL file from scratch. The existing "weather" sample will be rebuilt from the ground up using its existing WSDL files.

The new process will take three input parameters. The first two parameters are strings and represent the latitude and longitude of a location within the United States. The third parameter is the date for which the weather forecast for that location is desired. The US National Weather Service web server will be called by an "invoke" statement to obtain the weather forecast. This is returned in an xml document that is displayed on the console.

This tutorial demonstrates the following concepts:

1. Creating a new BPEL process
2. Importing a WSDL files
3. Defining partner links
4. Defining variables
5. Building a sequence of statements
6. Setting statement properties
7. Using a third party web service
8. Converting data types in BPEL
9. Deploying and running BEPL process

The user should examine this target web service at:

http://www.weather.gov/xml

Chapter 2. Prerequisites

This tutorial assumes that the user has the Zenflow plug-in installed in an appropriate version of Eclipse, and is familiar with basic Eclipse concepts.

Zenflow currently does not provide assistance in generating/editing WSDL files. These must be obtained from target web service providers, or generated manually. In either case, it is important that the Zenflow user understands the relationships between the WSDL files and the namespaces they define.

The user should have a clear understanding of the two types of WSDL files used in a BPEL scenario. The first is the WSDL files that represent the externally referenced "target" web services with which this BPEL process is going to interact. The second is the WSDL file that represents the interface of the newly defined BPEL process itself.

Chapter 3. Examining the WSDL Files

This section describes the various WSDL files that will be used in this tutorial.

3.1 WSDL Files for the External Web Service

WSDL files for "standard" web services can usually be obtained if the URL for the web service is known. In this case, append "?wsdl" to the web service URL to get the WSDL file for that web service. Thus, for the US national weather service, using the information from the above web site, the WSDL file for that web service can be obtained from the following URL:

http://www.weather.gov/forecasts/xml/SOAP_server/ndfdXMLserver.php?wsdl

This WSDL file can also be viewed in the Orchestra weather samples directory at:
"Orchestra installation location"\BPEL\samples\weather\nws\ndfdXML.wsdl

Important elements from this WSDL file are:

targetNamespace="http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl"
This namespace must be referenced in the WSDL file for the BPEL process in order to gain visibility to message types and operations defined by the target web service.

porttype name="ndfdXMLPortType"
This name is needed to build the corresponding "partnerlinktype" file.

A corresponding "partnerLinkType" file must be created. For this tutorial, this example file can be seen at:
"Orchestra installation location"\BPEL\samples\weather\nws\partnerlinktype.wsdl

The content of this file is as follows:

```
<definitions
   targetNamespace=
      "http://orchestra.objectweb.org/samples/weather"
   xmlns:p9807=
      "http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl"
   xmlns:plnk=
      "http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns="http://schemas.xmlsoap.org/wsdl/">

  <plnk:partnerLinkType name="NWSService">
    <plnk:role name="service">
      <plnk:portType name="p9807:ndfdXMLPortType" />
    </plnk:role>
  </plnk:partnerLinkType>
</definitions>
```

Several important observations can be made about this file:

1. The targetNamespace="http://orchestra.objectweb.org/samples/weather" is the namespace that will be used in the BPEL file that is generated.
2. The "http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl" namespace is used to reference elements in the target WSDL file.
3. The "partnerLinkType" name of "NWSService" is a user-define name that will be referenced in the BPEL process.
4. The "portType" p9807:ndfdXMLPortType is a reference to the port type definition in the target WSDL file.

3.2 WSDL Files for the BPEL Process

The WSDL files for the BPEL process that will be written are available to review at:

"Orchestra installation location"\BPEL\samples\weather\weather.wsdl

and

weatherPartnerLink.wsdl

These files should be examined to understand the namespaces, messages, and operations used. It is useful, but not required, for these WSDL files to be written prior to writing the BPEL process.

Chapter 4. Creating New BPEL File

This section will discuss how to create a new BPEL file using the WSDL files supplied with the "weather" sample.

4.1 Create a New Project

To create a new project:

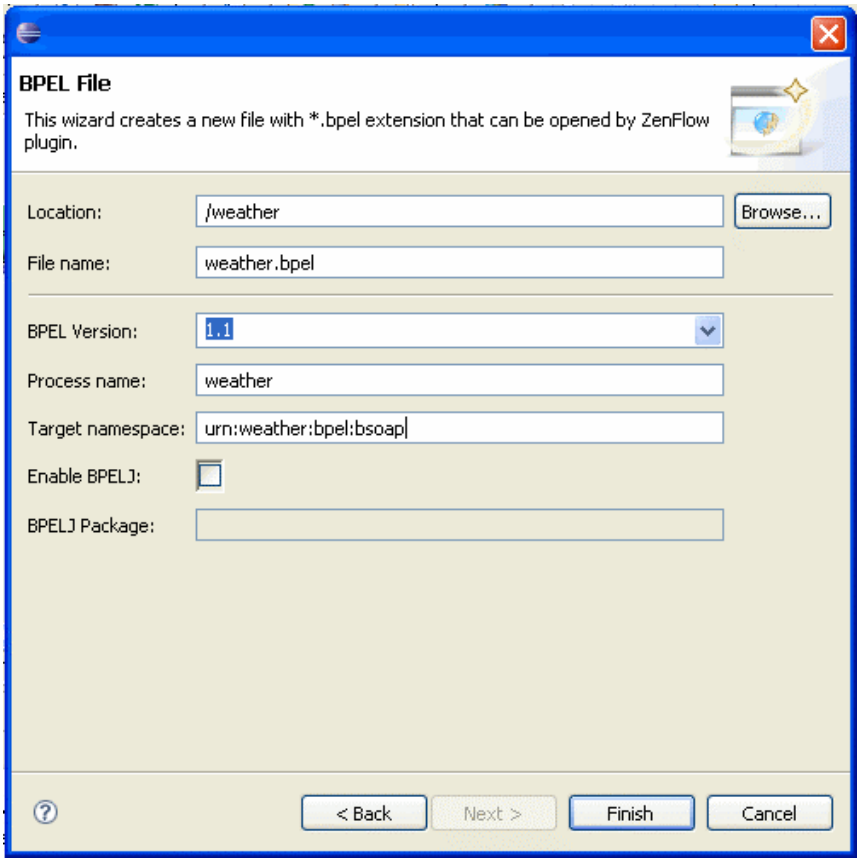
1. Start Eclipse and select a workspace.
2. Then use the menus "File -> New -> Project" and General->Project to create a new "Simple" project.
3. Set the new project's name to "weather" and click on Finish.

4.2 Create New BPEL File

To create a new BPEL file:

1. Right click on the "weather" project and select "New -> Other -> Zenflow -> BPEL file".
2. Then select the "next" button to display the BPEL wizard.
3. Set the filename to "weather.bpel", and the process name to "weather".
4. Then set the target Namespace to:
   http://orchestra.objectweb.org/samples/weather,
   which must match the target namespace used in the weather.wsdl file.
   The wizard should look like the following example:

Figure 4-1. Wizard To Create a New BPEL File



Press the "finish" button to close this dialog. This creates a "weather.bpel" file, then opens a blank graphic designer window for this file.

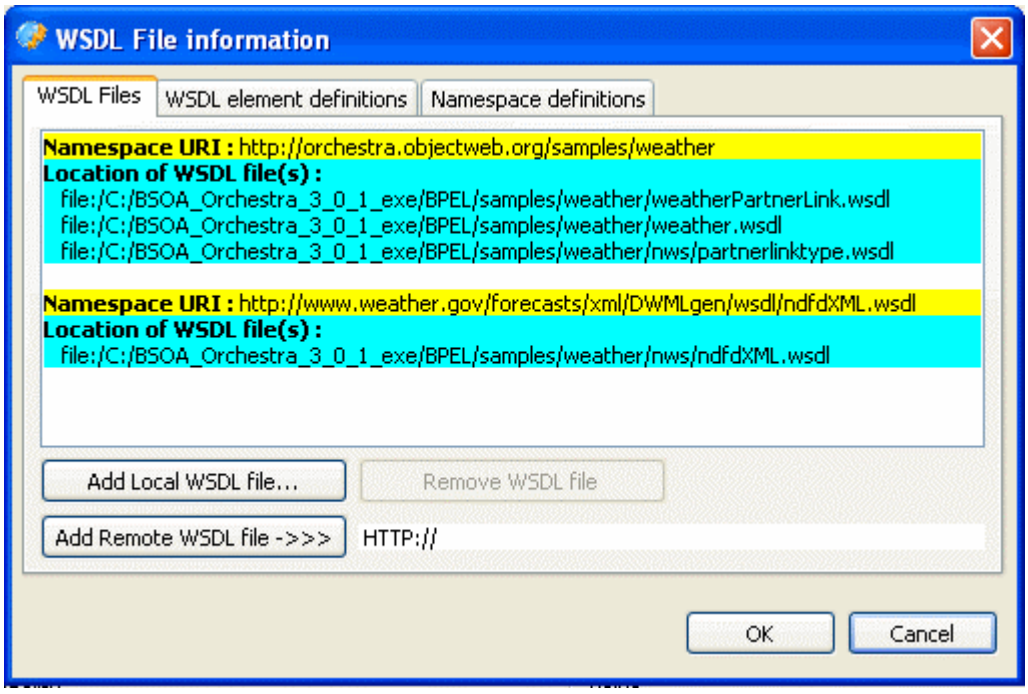
4.3 Import WSDL Files

Now use the WSDL dialog to import the previously discussed WSDL files. Use the main menu "Zenflow -> Link WSDL files..." to access this dialog. If the menu is disabled, left-click on the blue background of the graphic window to enable it. Then use the "Add local WSDL file" button to add each of the following WSDL files:

```
"Orchestra installation location"\BPEL\samples\weather\weather.wsdl
"Orchestra installation location"\BPEL\samples\weather\weatherPartnerLink.wsdl
"Orchestra installation location"\BPEL\samples\weather\nws\ndfdXML.wsdl
"Orchestra installation location"\BPEL\samples\weather\nws\partnerlinktype.wsdl
```

The WSDL dialog should now appear as follows.

Figure 4-2. WSDL File Information



Now close the dialog by pressing the "OK" button. If the cancel button is pressed, then none of the WSDL file definitions will be available to Zenflow.

Notice at this point if eclipse is terminated and restarted, the location of the WSDL files is remembered due to the WSDL cache. This cache is maintained in a ".workDirectory" immediately under the eclipse workspace location. If the eclipse workspace is cleared or changed, then it is necessary to repeat the above steps to repopulate the WSDL cache.

4.4 Create Partner Links

Now the partner links can be created using the main menu "Zenflow -> Partner links..." to open the partner links editor dialog.

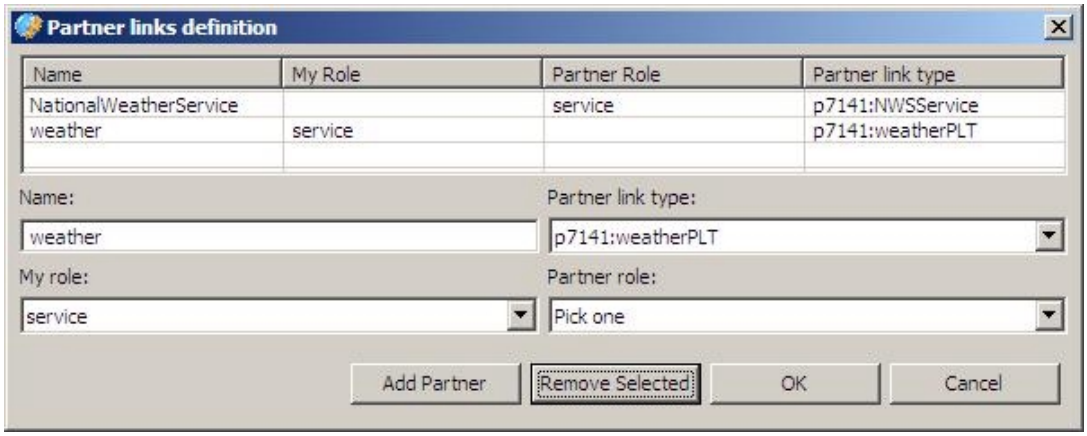
First create a partner link for the target web service. This corresponds to the US National Weather Service web service. Set the name field to "NationalWeatherService" and select the partner link type ending with "NWSService" in the combo box. Notice that the prefix assigned to this type is dynamically generated by Zenflow. Select a partner role of "service" to indicate the relationship of this partner link.

Be sure to press the "Add Partner" button to add this new definition.

Next create the partner link for the web service defined by the BPEL process itself. Set the name to "weather" and select the partner link type ending with "weatherPLT" in the combo box. This time, select a "My Role" of "service" to indicate that the BPEL process is providing this service.

Be sure to press the "Add Partner" button to add this new definition. The dialog should now appear as follows. The dialog box can be resized as needed to view the list contents.

Figure 4-3. Partner Links Definition Dialog



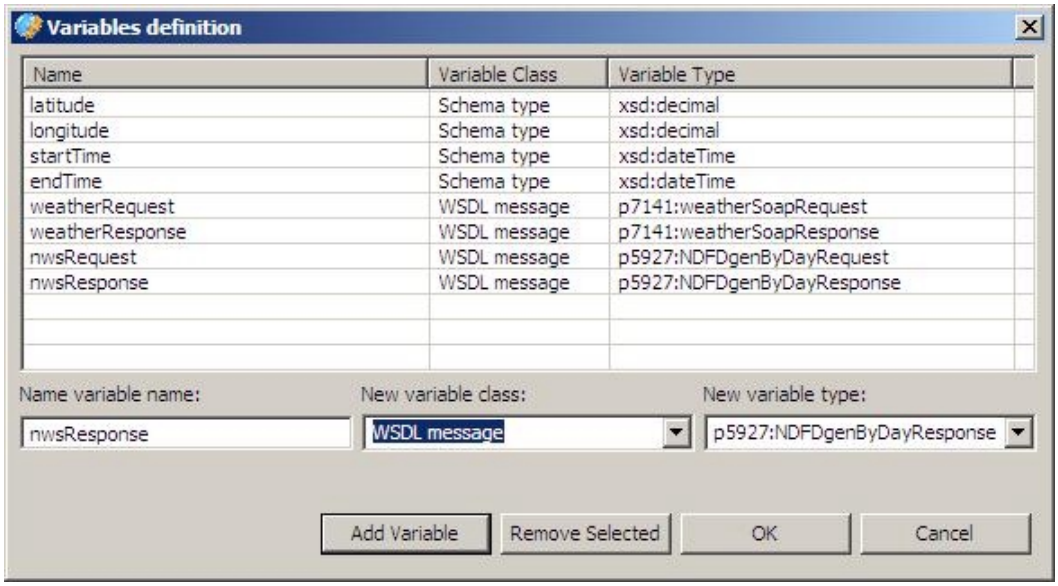
Close the dialog with the "OK" button. If the cancel button is used, then any definitions that have been added are discarded.

4.5 Create Variables

Now global variables that will be needed are defined. To access the "Variables definition" dialog, use the main menu "Zenflow -> Variables...". This dialog can be resized as needed to assist in proper display of all variable definitions.

Enter variable data from the image below to build the necessary variables. After each row of data is entered, press the "Add Variable" button to add that definition. If a mistake is made, the variable can be selected in the list and deleted using the "Remove Selected" button. The order of these variable definitions is not important. The final display should look like the following.

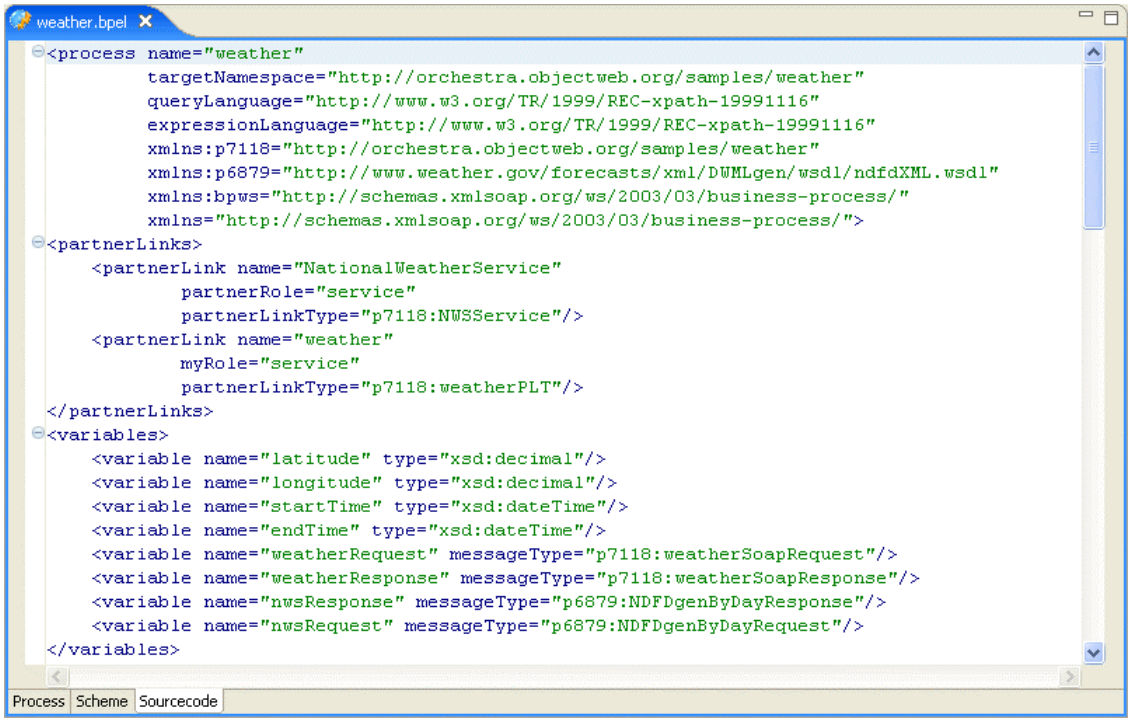
Figure 4-4. Variables Definition Dialog



Close this dialog by pressing the "OK" button.

This is a good point to save this BPEL file and examine the generated source code. Use the standard Eclipse toolbar or menu to save the current file. Then switch to the source view by selecting the "Sourcecode" tab at the bottom of the main designer window. The following window is displayed.

Figure 4-5. View of Generated Source Code for BPEL File



Notice the generated xml for the partner links and variables, as well as the xmlns references. The prefixes displayed will be different than those above because they are generated automatically when the namespace is referenced.

To continue the tutorial, switch back to the graphic view by selecting the "Process" tab at the bottom of this display window.

4.6 Create Receive Activity

Now a "receive" activity will be created. This activity represents a service provided by the BPEL process being created.

First, a top level "Sequence" activity is necessary. To insert this, right click on the blue background of the main graphic window and select the popup menu "Insert -> Sequence".

Now left click on the new "Sequence" node, to select it. Then right click on it and select the popup menu "Insert -> Receive". This will cause a new receive activity to be added inside the sequence.

The properties of the receive activity must now be set. This is done in the standard eclipse "Properties" window. If it is not open, open it by either using the main menu "Window -> Show View -> Properties", or by opening the Zenflow perspective by using the main menu "Windows -> Open perspective -> Other -> Zenflow".

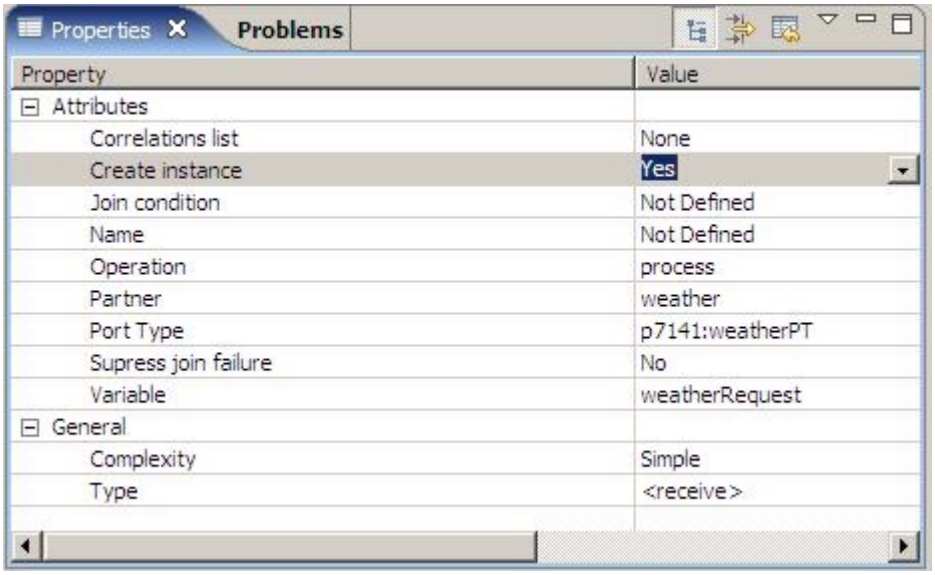
The properties window will display the properties for the currently selected activity. Select the receive activity that was just inserted to display its properties. In the properties window, set the "Partner" to "weather" by left clicking on the row and then select the edit control at the end of the line. Using similar edit controls, set the "portType" to "weatherPT" and then the "operation" to "process". Finally, set the input "variable" to be "weatherRequest". Note that setting the operation makes particular values available for the variable in its edit control.

Zenflow populates the above combo box editors with the values that are valid for the current context.

Finally be sure to set the "Create Instance" property to "Yes" to indicate that a new instance of this process should be created when a client tries to send a message to this receive activity.

The properties for the new receive activity should appear as follows.

Figure 4-6. Display of Properties for an New Receive Activity



4.7 Create First Assign Activities

Now an "assign" activity will be created. Left click on the "Sequence" node, to select it. Then right click on it and select the popup menu "Insert -> Assign". This will cause a new assign activity to be added at the end of the sequence.

Use the properties window for the new assign activity and set the activity name by typing "Convert Input Variables". Then open the "assign definition" dialog by clicking on the "..." icon at the end of the "Assign definition". This dialog box can be resized as needed.

This dialog displays the current copy statements in the list at the top, followed by groups of controls that allow new copy statements to be created. In the "From type" and "To type" combo boxes, the desired format for a copy statement can be selected. This results in the other combo boxes being populated accordingly.

Three copy statements will be created.

- The **first copy statement** is to simply copy one of the input variables to the output variable. This can be useful when debugging the process. To create this copy statement in the "Assign definition dialog", select the first "From type" in the combo box. This specifies that we want a copy statement using a format that allows specification of a variable, part, and query. Now select weatherRequest in the variable combo box. Then select "location" in the part combo box, and directly enter "latitude" in the query text box.

  In the "To Definition", select the first type. This specifies that we want a copy statement using a format that allows specification of a variable, part, and query. Now select "weatherResponse" in the variable combo box, then select “result” in the part combo box, and directly enter "result" in the query text box.

  Press the "Add definitions" button to add this copy statement definition to the list.

- The **second copy statement** will copy and convert the first input variable, which is a string, to one of the global variables created earlier. Select the "From Type" format that allows an "expression" to be entered. Then in the "expression" text box, directly enter the following:

```
number(bpws:getVariableData('weatherRequest', 'location', '/latitude'))
```

  In the "To definition", the first type should already be selected, thus just select "latitude" in the variable combo box.

  Press the "Add definition" button to add this copy statement to the list.

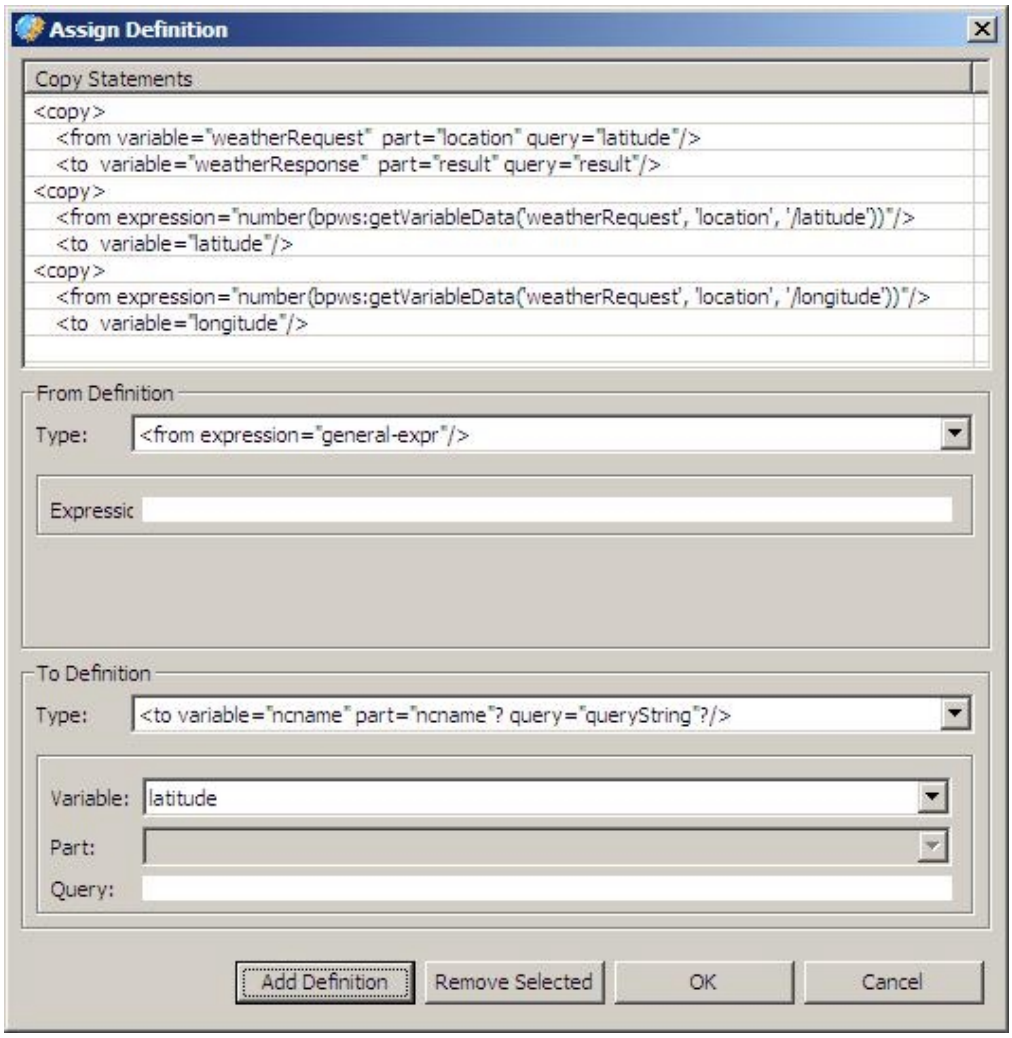
- The **third copy statement** is the same as the second, except it copies the longitude from the input message to the corresponding local variable. The "expression" to use for it is:

```
number(bpws:getVariableData('weatherRequest', 'location', '/longitude'))
```

  Again be sure to press the "Add definition" button to add new definitions. If a mistake is made, the copy statement can be selected in the list. This causes its data to be filled into the edit controls. The mistake can be corrected, and press the "Add Definition" button again. Pressing this button with a single copy statement selected will display a message box to ask if the copy statement is to be replaced or a new one added.

  The "Assign definitions" dialog should look like the following.

Figure 4-7. Assign Definitions Dialog



Be sure to close the dialog using the "OK" button. If the "Cancel" button is used, then all edits made are discarded.

Note:

The copy statements above used XPath expressions. For assistance with writing XPath expressions, the following tutorials may be helpful:

http://www.w3schools.com/xpath/

http://www.zvon.org/xxl/XPathTutorial/General/examples.html

The “www.w3schools.com/xpath/”on-line reference above contains a list of XPath 2.0 functions, which are not currently supported by Orchestra.

The following XPath on-line specification may be helpful with the version 1.0 functions:

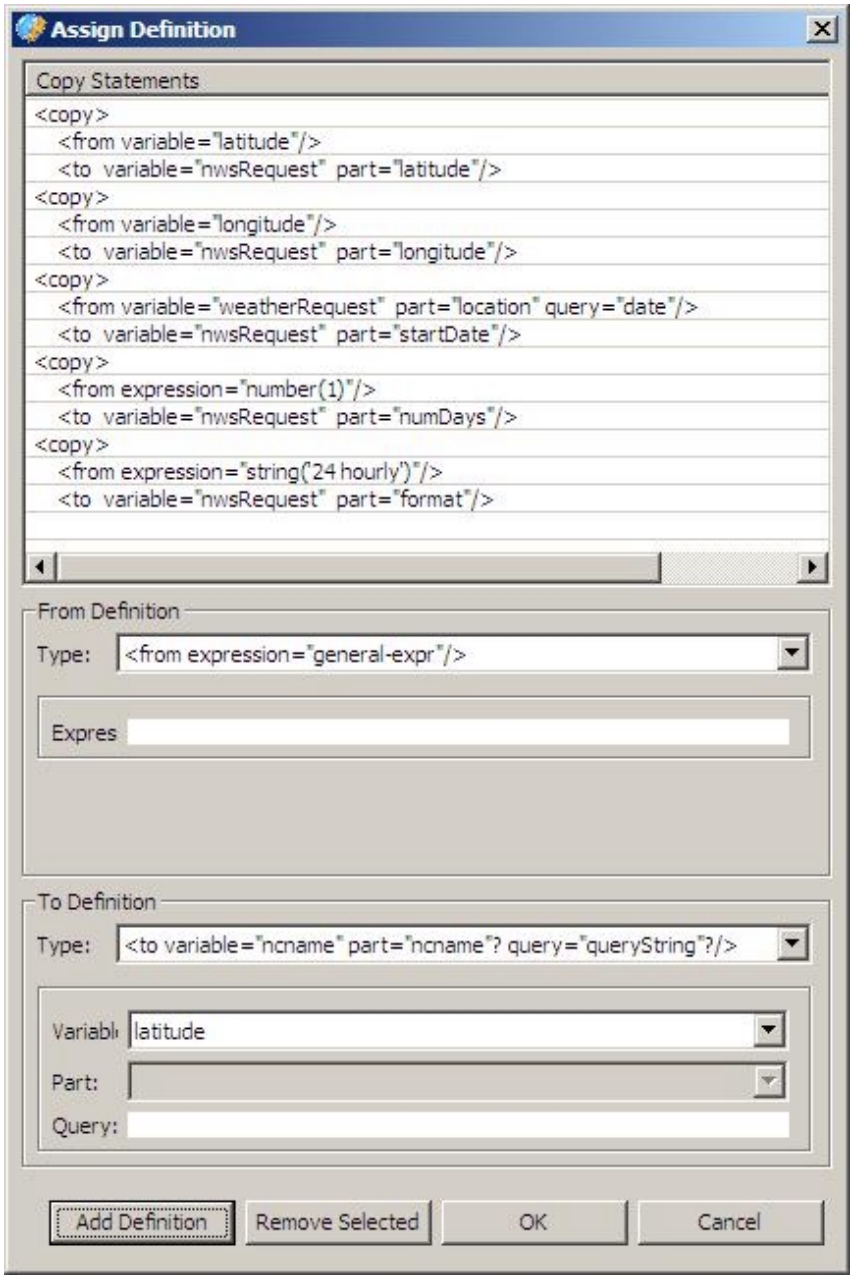
http://www.w3.org/TR/1999/REC-xpath-19991116

4.8 Create Second Assign Activity

A second assign activity will be created to initialize the "nwsRequest" message variable. This could be done in the same assign activity as above, but is done separately here to clarify the tutorial steps.

Insert another "assign" activity in the same way as above. Set the activity name to "Build NWS Request". Use the activity definition dialog to create the five necessary copy statements as shown below.

Figure 4-8. Display of Copy Statements in the Assign Definition Dialog



It is expected that the user will examine the input and message requirements for the target web service to know what data is needed in the nwsRequest variable.

Again, close this dialog using the "OK" button. This is a good point to save the file again, and possibly switch to the "Sourcecode" window to see the generated xml code.

Switch back to the main graphic view to continue the tutorial.

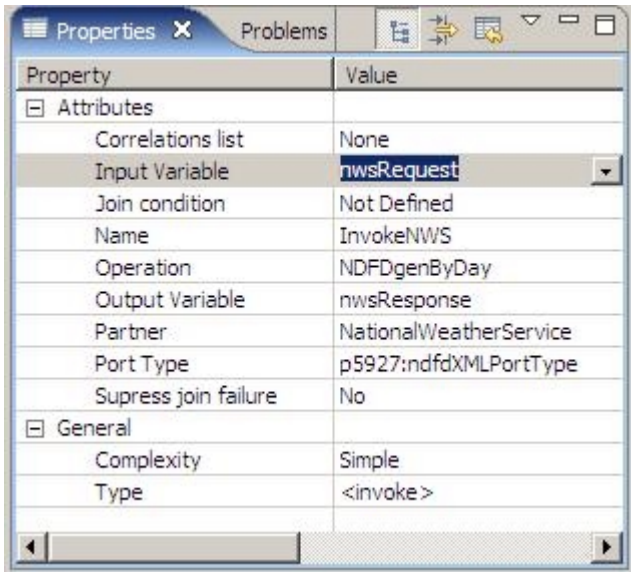
4.9 Create Invoke Activity To Call the NWS Web Service

At this point, all the fields of the nwsRequest variable have been initialized and it is now possible to call the NWS web service to get the desired weather forecast.

Insert a new "Invoke" activity by using the same popup menu as was previously used. Look at the properties of the invoke and set the "Name" property to "InvokeNWS".

Now set the "Partner", "PortType", "Operation", "InputVariable", and "OutputVariable". These must be set in this specific order because each setting enables values for the next item. The invoke properties should now look like the following.

Figure 4-9. Display of Invoke Properties To Call the NWS Web Service



4.10 Create Assign Activity To Copy Data To Output Variables

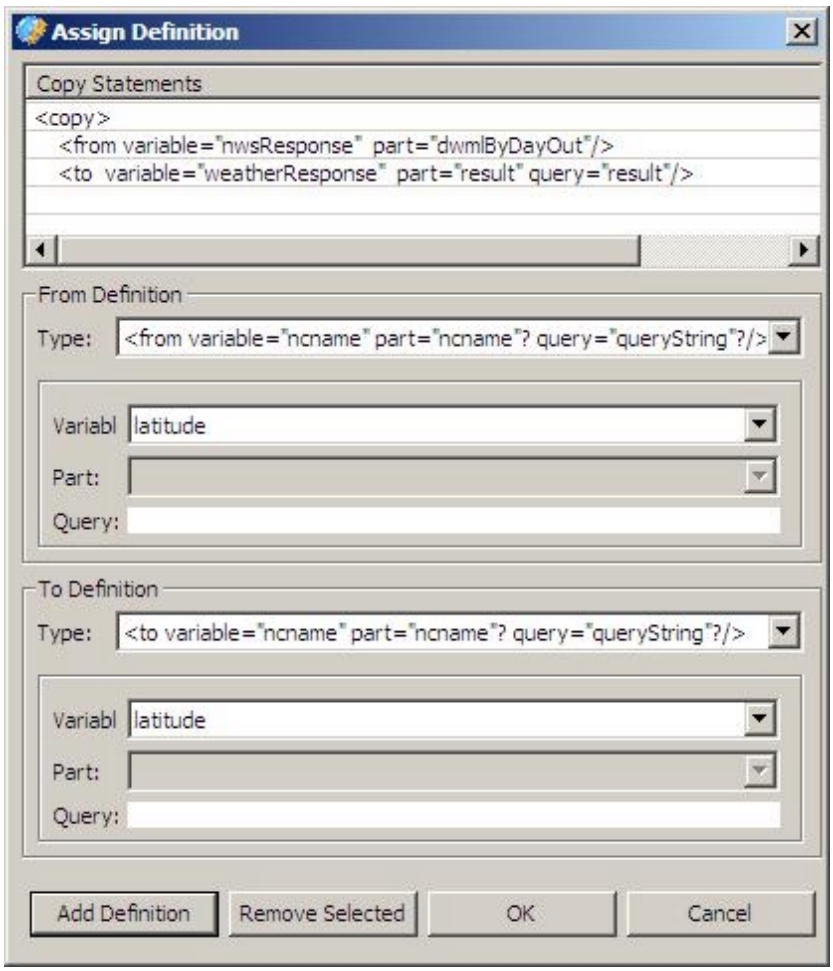
After the "Invoke" is called, the weather forecast is in the nwsResponse message. Now that data will be copied to the weatherResponse message for use in the final reply activity.

Insert a new "Assign" activity by using the same popup menu as was previously used. Select the new assign activity and display its properties. Open the assign definition dialog by clicking on the "..." icon at the end of the "Assign definition" row.

Create a new copy statement that copies the "dwmlByDayOut" part of the nwsResponse variable to the "result" part, "result" query, of the weatherResponse variable.

Be sure to press the "Add Definition" button to add the copy statement. The assign definition dialog should appear as follows.

Figure 4-10. Assign Definitions Dialog To Add the Copy Statement



Be sure to close the dialog by pressing the "OK" button so that the changes will be applied.

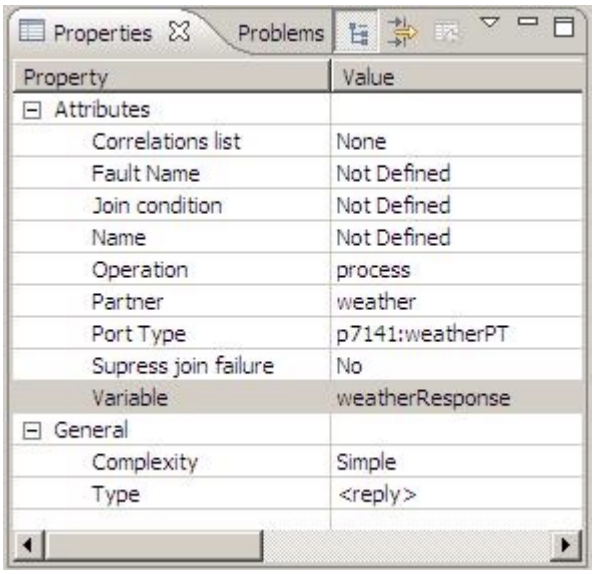
4.11 Create Reply Activity To Return Output Data

The previous assign statement put the weather forecast into the weatherResponse message. Now the message is going to be returned to the original caller of this process. A new "Reply" activity will be built that corresponds to the initial "receive" activity of this BPEL process.

Insert a new "Reply" activity by using the same popup menu as was previously used. Select the new reply activity and display its properties. The following properties must be set in this specific order: partner=weather, portType=prefix:weatherPT, operation=process, variable=weatherResponse.

The reply properties should now look as follows.

Figure 4-11. Reply Properties Dialog



The graphic display for the total process is shown below.

Figure 4-12. Graphic Display of the Total Process

Chapter 5. Deploying and Running the "weather" BPEL Process

This section will describe how to run the BPEL process that has been generated in this tutorial. This assumes that the user has the BSOA Orchestra engine running (use “bsoap start”, if it is not running), and the computer has Internet connectivity to www.weather.gov in order to actually call the US weather service web server.

5.1 Copy Files To Correct Location

In this tutorial, the "weather" sample has effectively been regenerated. This new BPEL file can now be placed into the "Orchestra installation location"\BPEL\samples\weather directory to allow easy deployment and execution using the command line utilities provided with Orchestra.

First it is recommended that the existing "weather.bpel" file delivered as part of Orchestra be either copied or renamed. Then copy or move the new version to this location. The new version built in the tutorial is located in the
"eclipse workspace directory"\weather\weather.bpel.

5.2 Clean Any Existing Deployment

Each time a new version of the process is defined, it is necessary to ensure that any previous deployment of the process is removed. This can be done with the following Orchestra command line:

```
bsoap clean -p weather -samples
```

Note:

The Deploy action described in Section 5.3, “Deploy the New “weather” Process,” does a “clean” operation as part of its processing.

5.3 Deploy the New "weather" Process

The new weather process can now be deployed into the BPEL engine. This causes the WSDL and BPEL files to be scanned and corresponding code is generated and compiled.

From the "`Orchestra installation location`"`\BPEL\samples\weather` directory, use the following command line:

```
bsoap deploy -p weather -samples nws/ndfdXML.wsdl
```

During this step, there is a significant amount of output in the Orchestra command window about the results of the deployment. This should be examined carefully to determine if the deployment is successful. Errors at this point are likely typographical errors.

A comparison of the newly generated weather.bpel file can be made with the saved copy to determine if this is the case. The differences in generated namespace prefixes can be ignored.

5.4 Run the “weather” process

Now it is time to run the new BPEL process. To do so, it is necessary to have a client program that accesses the first "Receive" statement by sending the appropriate message. Fortunately, because this tutorial duplicates an existing sample application, the client provided for the weather sample can be used. The source code for this client can be seen at:

`"Orchestra installation location"\BPEL\samples\weather\org\objectweb\orchestra\samples\weather\WeatherClient.java`

This program takes two input parameters, and builds the input message needed by the weather.bpel process. The input data is put into a "LocationType" variable, along with the current date. Then the BPEL process is invoked, which in turn sends the data along to the National Weather Service web service. The resulting xml document, which contains the weather forecast for the selected location, is displayed.

The command to launch the weather sample client is: (using the latitude/longitude for 1600 Pennsylvania Ave NW Washington DC 20502, see http://rpc.geocoder.us)

bsoap launch -p weather -cc org.objectweb.orchestra.samples.weather.WeatherClient 38.99 -77.99

The following is an example of the expected output:

>bsoap launch -p weather -cc org.objectweb.orchestra.samples.weather.WeatherClient 38.99 -77.99

ClientContainer.info : Starting client...

Running AdminProcessClient...

choice = launch

AdminProcessClient launch processName = weather, className = org.objectweb.orchestra.samples.weather.WeatherClient, arguments = 38.99, -77.99,

[initProperties] Starting InitPropertiesTask...

[null] Ignoring:

[null] ———

[null] DEFINE.BAT

[echo] bpel_home = C:\BSOA_Orchestra_3_0_1_exe\BPEL

```
[initProperties] Finishing InitPropertiesTask...

[launchProcess] Starting LaunchProcessTask...

    [java] ClientContainer.info : Starting client...

    [java] Latitude: 38.99, Longitude: -77.99

    [java] Result = <?xml version='1.0' ?>

    [java] <dwml version='1.0' xmlns:xsd="http://www.w3.org/2001/XMLSchema"

    [java]                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    [java]
xsi:noNamespaceSchemaLocation="http://www.nws.noaa.gov/forecasts/xml/DWMLgen
/schema/DWML.xsd">

    [java]    <head>

    [java]      <product concise-name="dwmlByDay" operational-mode="developmental">

    [java]         <title>NOAAs National Weather Service Forecast by 24 Hour
period</title>

    [java]         <field>meteorological</field>

    [java]         <category>forecast</category>

    [java]         <creation-date refresh-frequency='PT1H'>2006-11-
01T19:41:13Z</creation-date>

    [java]      </product>

    [java]      <source>

    [java]         <more-information>http://www.nws.noaa.gov/forecasts/xml/</more-
information>

    [java]         <production-center>Meteorological Development Laboratory <sub-
center>Product Generation Branch</sub-center></production-center>

    [java]         <disclaimer>http://www.nws.noaa.gov/disclaimer.html</disclaimer>

    [java]         <credit>http://www.weather.gov/</credit>

    [java]         <credit-logo>http://www.weather.gov/images/xml_logo.gif</credit-
logo>

    [java]         <feedback>http://www.weather.gov/survey/nws-
survey.php?code=xmlsoap</feedback>
```

```
    [java]     </source>
    [java]   </head>
    [java]   <data>
    [java]     <location>
    [java]       <location-key>point1</location-key>
    [java]       <point latitude="38.99" longitude="-77.99" />
    [java]     </location>
    [java]     <time-layout time-coordinate="local" summarization="24hourly">
    [java]       <layout-key>k-p24h-n1-1</layout-key>
    [java]         <start-valid-time>2006-11-01T06:00:00-05:00</start-valid-time>
    [java]           <end-valid-time>2006-11-02T06:00:00-05:00</end-valid-time>
    [java]     </time-layout>
    [java]     <time-layout time-coordinate="local" summarization="12hourly">
    [java]       <layout-key>k-p12h-n2-2</layout-key>
    [java]         <start-valid-time>2006-11-01T06:00:00-05:00</start-valid-time>
    [java]           <end-valid-time>2006-11-01T18:00:00-05:00</end-valid-time>
    [java]         <start-valid-time>2006-11-01T18:00:00-05:00</start-valid-time>
    [java]           <end-valid-time>2006-11-02T06:00:00-05:00</end-valid-time>
    [java]     </time-layout>
    [java]     <parameters applicable-location="point1">
    [java]       <temperature type='maximum' units="Fahrenheit" time-layout="k-p24h-n1-
1">
    [java]         <name>Daily Maximum Temperature</name>
    [java]         <value>65</value>
    [java]       </temperature>
    [java]       <temperature type='minimum' units="Fahrenheit" time-layout="k-p24h-n1-
1">
```

```
    [java]            <name>Daily Minimum Temperature</name>
    [java]            <value>44</value>
    [java]          </temperature>
    [java]          <probability-of-precipitation type='12 hour' units="percent" time-layout="k-
p12h-n2-2">
    [java]            <name>12 Hourly Probability of Precipitation</name>
    [java]            <value>10</value>
    [java]            <value>67</value>
    [java]          </probability-of-precipitation>
    [java]          <weather time-layout="k-p24h-n1-1">
    [java]            <name>Weather Type, Coverage, and Intensity</name>
    [java]            <weather-conditions weather-summary="Rain Likely">
    [java]              <value coverage="likely" intensity="light" weather-type="rain"
qualifier="none">
    [java]              </value>
    [java]            </weather-conditions>
    [java]          </weather>
    [java]          <conditions-icon type="forecast-NWS" time-layout="k-p24h-n1-1">
    [java]            <name>Conditions Icons</name>
    [java]              <icon-
link>http://www.nws.noaa.gov/weather/images/fcicons/ra70.jpg</icon-link>
    [java]          </conditions-icon>
    [java]        </parameters>
    [java]    </data>
    [java] </dwml>

[launchProcess] Finishing LaunchProcessTask...
Finishing AdminProcessClient...
```

The two input parameters must be a latitude and longitude within the United States. To see some of the acceptable values, run the client with no arguments using the following command line:

```
bsoap launch -p weather -cc
org.objectweb.orchestra.samples.weather.WeatherClient
```

This will display the following:

```
    [java] >>>>> ERROR: 2 decimal args expected.
    [java] >>>>> The first must be a latitude within the US. Approximate range is 24.00 to
50.00
    [java] >>>>> The second must be a longtitude within the US. Approximate range is -
72.00 to -125.00
    [java] >>>>> Here are some example locations within the US.
    [java] >>>>>     City              Latitude  Longitude
    [java] >>>>>     Washington, DC.     38.90     -77.04
    [java] >>>>>     New York, NY.       40.76     -73.92
    [java] >>>>>     Los Angelas, CA.    34.10     -118.33
    [java] >>>>>     San Francisco, CA.  37.74     -122.42
    [java] >>>>>     Phoenix, AZ.        33.62     -112.12
```

Using input values from the above list, it is now possible to get the weather for a US location by using the following command line:

```
bsoap launch -p weather -cc
org.objectweb.orchestra.samples.weather.WeatherClient 33.62 -112.12
```

The output is in the form of an xml document. The max/min temperatures as well as a summary forecast can be seen in the output.

This concludes this part of the tutorial. A subsequent tutorial will extend this example for ease-of-use.

Technical publication remarks form

Title:	BSOA Orchestra

Reference No.:	86 A2 17ET 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 17ET 01