

# BSOA Orchestra

Zenflow Weather2 Tutorial

BULL SERVICE-ORIENTED  
ARCHITECTURE (BSOA)



REFERENCE  
86 A2 50ET 00



# BULL SERVICE-ORIENTED ARCHITECTURE (BSOA)

## BSOA Orchestra

### Zenflow Weather2 Tutorial

**BSOA Orchestra v3.0**

#### **Software**

November 2006

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

REFERENCE  
86 A2 50ET 00

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.	Tutorial Overview.....	1
Chapter 2.	Prerequisites .....	3
Chapter 3.	Weather2 Process Flow .....	5
Chapter 4.	Load “weather2” BPEL Process Into Zenflow .....	7
4.1	Create a New Project in the Workspace.....	7
4.2	Import the weather2 Sample .....	7
4.3	Open the Weather2 File for Display .....	8
4.4	Import the Associated WSDL File .....	8
Chapter 5.	Examine Weather2 Steps in Zenflow.....	11
5.1	Step 1 – BPEL Process Receives Input From Client Application.....	11
5.2	Step 2 – BPEL Process Passes US Address to the Geocoder Web Service .....	13
5.3	Step 3 – Geocoder Returns Latitude and Longitude.....	15
5.4	Step 4 – BPEL Process Passes Latitude, Longitude, and Date to NWS .....	15
5.5	Step 5 – NWS Returns Weather Forecast .....	17
5.6	Step 6 – BPEL Process Returns Weather Forecast to Client Application .....	17
Chapter 6.	Deploying and Running the “weather2” BPEL Process.....	19
6.1	Clean Any Existing Deployment.....	19
6.2	Deploy the New “weather2” Process .....	19
6.3	Run the “weather2” Process .....	20

---

## List of Figures

Figure 3-1. Weather2 Data Flow.....	5
Figure 4-1. "weather2.bpel" Node and outline displays .....	8
Figure 4-2. WSDL File Information Dialog Box.....	9
Figure 5-1. "Receive" Activity Properties for Weather2 .....	11
Figure 5-2. Partner Links Definition Dialog .....	12
Figure 5-3. Variables Definition Dialog.....	13
Figure 5-4. Assign Definition Dialog .....	13
Figure 5-5. "Invoke" Activity Properties Dialog.....	14
Figure 5-6. Output Variable in Variables Definition Dialog.....	14
Figure 5-7. Assign Statement Following the "Invoke" Activity .....	15
Figure 5-8. "nwsRequest" Variable in Variables Definition Dialog .....	16
Figure 5-9. Assign Definition Dialog Showing Final "Assign" Activity .....	17
Figure 5-10. Properties Dialog for Final "Reply" Activity.....	18

---

## Preface

This tutorial is oriented to the user that has some experience with Zenflow, BPEL, and WSDL. It is intended to demonstrate how Zenflow represents various aspects of BPEL and WSDL.





---

## Chapter 1. Tutorial Overview

This tutorial will use Zenflow to examine the “weather2” sample. This sample is an extension of the “weather” sample. The “weather2” sample is closer to a real world BPEL application than the “weather” sample. It uses data from two independent Web services in order to achieve the desired result.

Two external Web services are used in this tutorial. The Geocoder Web service returns the latitude and longitude for a US street address. The second Web service returns the weather forecast for a given US location identified by latitude and longitude. The user should examine these target Web services at:

<http://geocoder.us/>

<http://www.weather.gov/xml>



---

## Chapter 2. Prerequisites

This tutorial assumes the following:

- the Zenflow plug-in is installed in Eclipse 3.2 or later
- the user is familiar with basic Eclipse concepts
- the user has some working knowledge of Zenflow, BPEL, and WSDL

This tutorial will examine the weather2 Orchestra sample. No changes will be made to that sample.



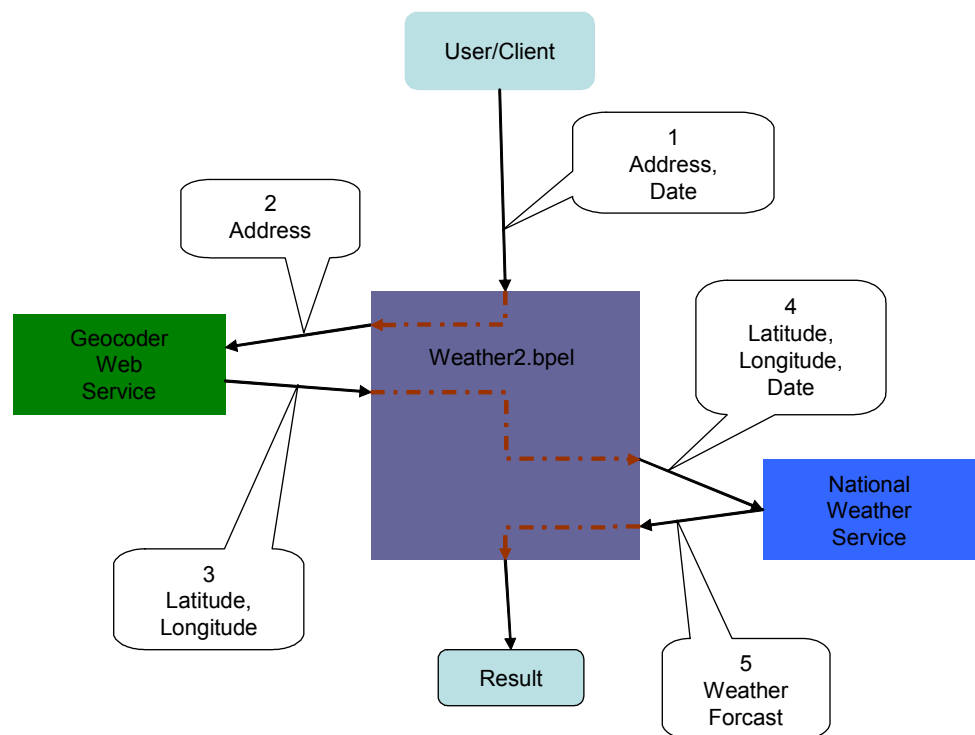
## Chapter 3. Weather2 Process Flow

The “weather2” sample makes use of two external Web services:

1. **Geocoder** – This Web service takes a US address as an input parameter and returns the latitude and longitude for that address.
2. **National Weather Service** – This Web service takes a US location in the form of latitude and longitude as parameters and returns the weather forecast for that location.

By “Orchestrating” these two Web services together, the “weather2” sample is able to return a weather forecast for a given US address. This is illustrated in the following diagram.

Figure 3-1. Weather2 Data Flow



The process flows as follows:

3. The java client application sends a US address and the current date to the weather2 BPEL process.
4. The BPEL process invokes the Geocoder Web service, passing the US address as a parameter.
5. The Geocoder Web service returns the latitude and longitude for the specified address.
6. The BPEL process extracts the latitude and longitude from the Geocoder response and invokes the National Weather Service. The current date from the java client is also used as a parameter to the NWS.
7. The NWS returns the weather forecast for the specified location.
8. The forecast is extracted from the NWS response and is returned to the java client.

---

## Chapter 4. Load “weather2” BPEL Process Into Zenflow

The Zenflow plug-in works on BPEL files that are part of a project in a workspace. Perform the steps described in this chapter to create a new Eclipse project and import the weather2 BPEL file into that project.

### 4.1 Create a New Project in the Workspace

1. Start Eclipse and select a workspace.
2. Then use the following menu sequence "File -> New -> Project" and select General-> Project to create a new "Simple" project. The project name is user defined; "weather2" is used here as an example.

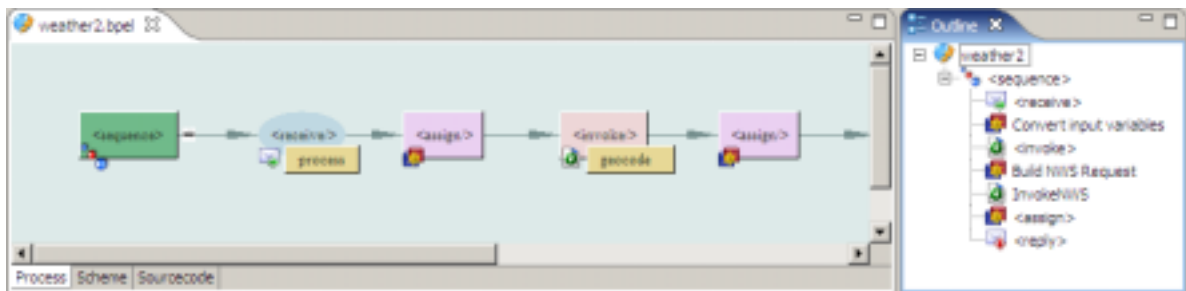
### 4.2 Import the weather2 Sample

1. Right click on the "weather2" project folder in the package explorer view and select the "Import" menu item.
2. Then select the option for importing from the "File System".
3. Use the Browse button to go to the "weather2" samples folder that is part of the Orchestra installation. This will be in a location similar to:  
*"Orchestra installation location"\BPEL\samples\weather2*
4. Expand this directory in the "File System Import" dialog and select the check box next to the "weather2.bpel" file. DO NOT import the WSDL file using this dialog.

## 4.3 Open the Weather2 File for Display

In the package explorer, expand the "weather2" project node. Then double click on the "weather2.bpel" node to open it in the main display window. The display should look similar to the following.

Figure 4-1. "weather2.bpel" Node and outline displays



Notice the problems identified in the "Problems" window. These have occurred because the Zenflow syntactic analyzer cannot find definitions for all the data types used in the BPEL file. The definitions are not available because the WSDL file for the process has not been loaded yet.

Go to the next step to load the WSDL file for the process.

## 4.4 Import the Associated WSDL File

Once the Zenflow plug-in has been referenced by opening a BPEL file, a new "Zenflow" main menu is added to the Eclipse menu bar.

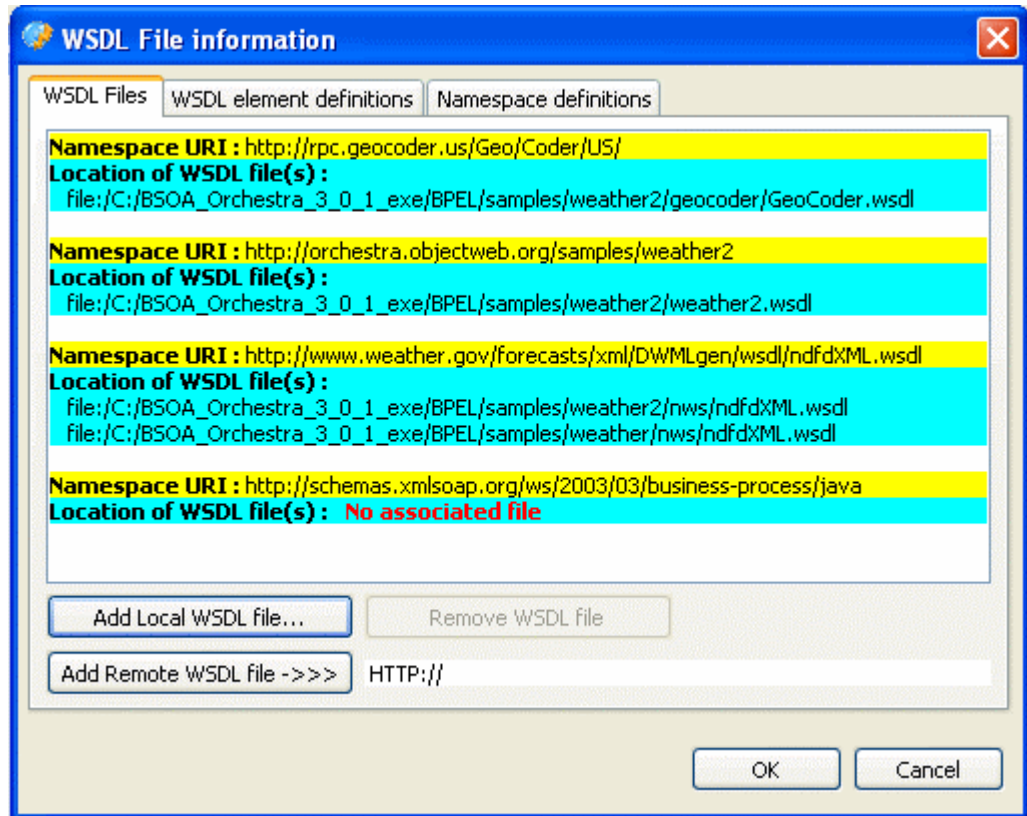
Now use the WSDL dialog to import the necessary WSDL files. Use the main menu "Zenflow -> Link WSDL files..." to access this dialog. If the menu is disabled, left click on the blue background of the graphic window to enable it. Then use the "Add local WSDL file" button to add each of the following WSDL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

["Orchestra installation location"\BPEL\samples\weather2\weather2.wsdl](#)  
["Orchestra installation location"\BPEL\samples\weather2\nws\ndfdXML.wsdl](#)  
["Orchestra installation location"\BPEL\samples\weather2\geocoder\GeoCoder.wsdl](#)



The WSDL dialog should now appear as follows.

Figure 4-2. WSDL File Information Dialog Box



Notice that there is now a file associated with each namespace. A missing definition of the BPELJ namespace is OK at this point. Press the "OK" button to close the dialog.



**Note:**

A current program bug prevents the "Problems" window from being refreshed at this time. Closing and opening the project containing the BPEL file, and then double clicking on the BPEL file in the package explorer view will refresh the Problems view.



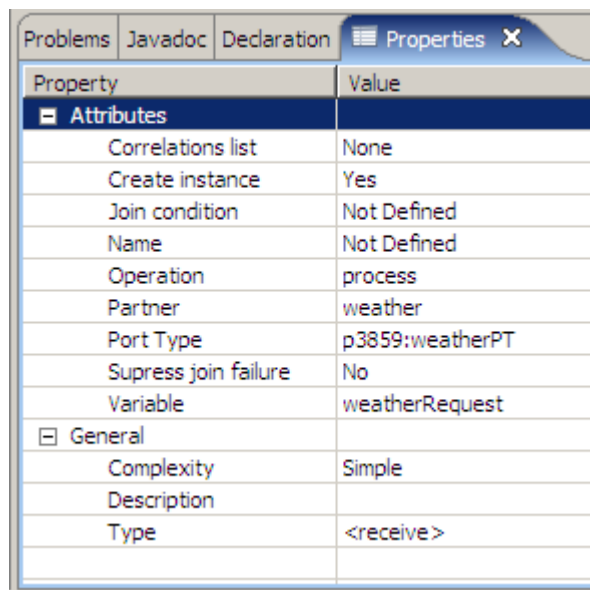
## Chapter 5. Examine Weather2 Steps in Zenflow

This section will use Zenflow to examine the BPEL activities and data associated with the weather2 BPEL process. Each section below describes one of the “steps” identified in Section 3, Figure 3-1. Various aspects of the BPEL and WSDL relations will be described. However, this is not intended to be an exhaustive analysis of the process. The descriptions concentrate primarily on the data flow within the process.

### 5.1 Step 1 – BPEL Process Receives Input From Client Application

A BPEL process receives data from clients in the context of BPEL “receive” activities. In the Bull Orchestra engine, the receive activity, together with a reply activity, constitute a method with input and output messages in a Web service that is represent by the BPEL process. Displaying the properties for the “receive” activity in weather2 shows the following:

Figure 5-1. “Receive” Activity Properties for Weather2



Property	Value
<b>Attributes</b>	
Correlations list	None
Create instance	Yes
Join condition	Not Defined
Name	Not Defined
Operation	process
Partner	weather
Port Type	p3859:weatherPT
Supress join failure	No
Variable	weatherRequest
<b>General</b>	
Complexity	Simple
Description	
Type	<receive>

The important properties are as follows:

**Create Instance = Yes** – This property tells the BPEL engine that a new process instance is to be created when a client program calls the method represented by this receive activity.

**Partner = weather** – This property is the reference to the partner link. Use the main menu Zenflow -> PartnerLinks to display this dialog:

Figure 5-2. Partner Links Definition Dialog

Name	My Role	Partner Role	Partner link type
NationalWeatherService		service	p3859:NWSService
weather	service		p3859:weatherPLT
geocoder		service	p3859:GeoCoderService

Name:  Partner link type:

My role:  Partner role:

Notice that a “weather” partner has been defined in which “myRole” is “service” and the partner Link Type is “p3859:weatherPLT”.

Now use the main menu Zenflow -> Link WSDL Files and select the “WSDL Element definition tab”. This displays all the WSDL elements that have been loaded as a result of importing the WSDL files. Notice the following lines:

```

PartnerLinkType:(Ref:1)p3859:weatherPLT
  Role:(Ref:p3859)service Ref:6
    PortType (Ref:6):Name: 6:weatherPT
      Operation:process
        input:6:weatherSoapRequest
        output:6:weatherSoapResponse

```

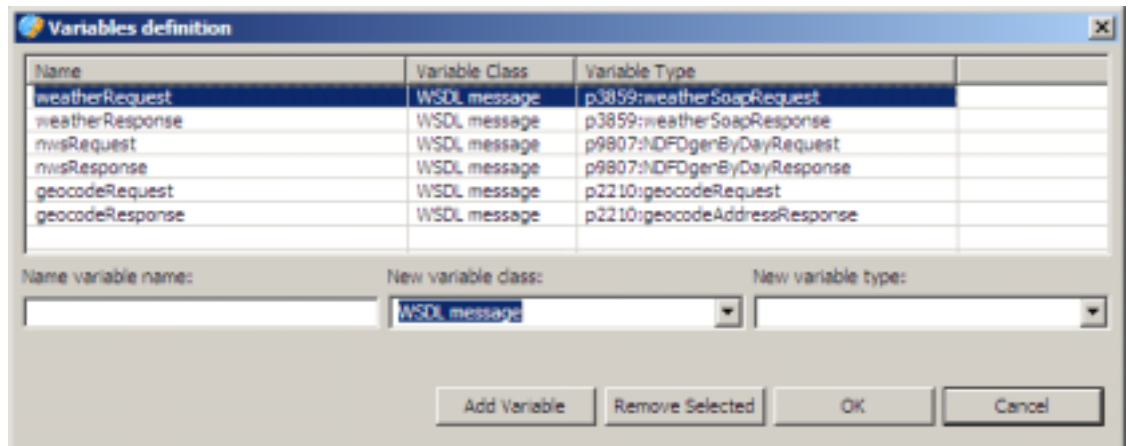
This indicates that the Web service implementing weatherPLT has a “service” role that defines a “weatherPT” port type, which in turn has a “process” operation that has input/output parameters of weatherSoapRequest and weatherSoapResponse respectively.

Collectively, this means that the “receive” statement in the weather2 BPEL process is implementing the input side of the above function definition.

**Port Type = p3859:weatherPT** – This is the reference to the port type. As seen in the WSDL elements above, this indicates that this receive statement is implementing the port type defined in the WSDL.

**Variable = weatherRequest** – This is the BPEL variable that will receive the data passed as a parameter by the client application. The variables type must be weatherSoapRequest, as defined in the WSDL element above. Use the main menu Zenflow -> Variables to display the variable definitions for the process as shown:

Figure 5-3. Variables Definition Dialog

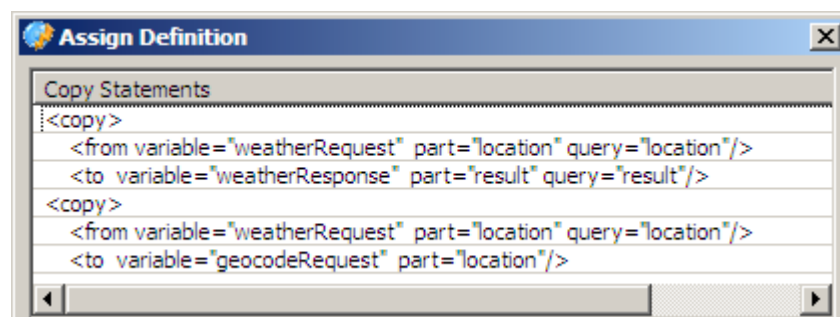


As can be seen, the “weatherRequest” variable referenced in the receive activity is of type “p3859:weatherSoapRequest” as is needed to satisfy the requirements of the interface.

## 5.2 Step 2 – BPEL Process Passes US Address to the Geocoder Web Service

The first “Assign” activity copies the address from the input variable to the variable used to invoke the Geocoder Web service. Select the “Assign” activity and look at the “Assign definition”. The copy statements are shown below:

Figure 5-4. Assign Definition Dialog



The first copy statement is for debug purposes only. It copies the input address from the “location” part of the input message to the “result” part of the output message.

The second copy statement copies the input address to the “location” part of the geocodeRequest variable. This variable is of type “p2210:geocodeRequest”. This type can be seen by using Zenflow -> Link WSDL Files and select the “WSDL Element definition tab”. The message definition is shown below:

```

message (Ref:13):geocodeRequest Ref:13NamRef:p2210
  part name:location Ref:13 type:string Ref:9

```

The geocodeRequest variable is passed to the Geocoder via the “Invoke” activity. This can be observed in the properties of the “Invoke” activity as shown below:

Figure 5-5. “Invoke” Activity Properties Dialog

Property	Value
<b>Attributes</b>	
Correlations list	None
Input Variable	geocodeRequest
Join condition	Not Defined
Name	Not Defined
Operation	geocode
Output Variable	geocodeResponse
Partner	geocoder
Port Type	p2210:GeoCode_PortType
Supress join failure	No
<b>General</b>	
Complexity	Simple
Description	Call geocoder to get latitude/longitude of the location.
Type	<invoke>

In addition, notice that the “Output variable” is “geocodeResponse”. These variables can be seen in the variables dialog as shown in the following figure.

Figure 5-6. Output Variable in Variables Definition Dialog

Name	Variable Class	Variable Type
weatherRequest	WSDL message	p3859:weatherSoapRequest
weatherResponse	WSDL message	p3859:weatherSoapResponse
nwsRequest	WSDL message	p9807:NDFDgenByDayRequest
nwsResponse	WSDL message	p9807:NDFDgenByDayResponse
geocodeRequest	WSDL message	p2210:geocodeRequest
geocodeResponse	WSDL message	p2210:geocodeAddressResponse

Name variable name:  New variable class:  New variable type:

## 5.3 Step 3 – Geocoder Returns Latitude and Longitude

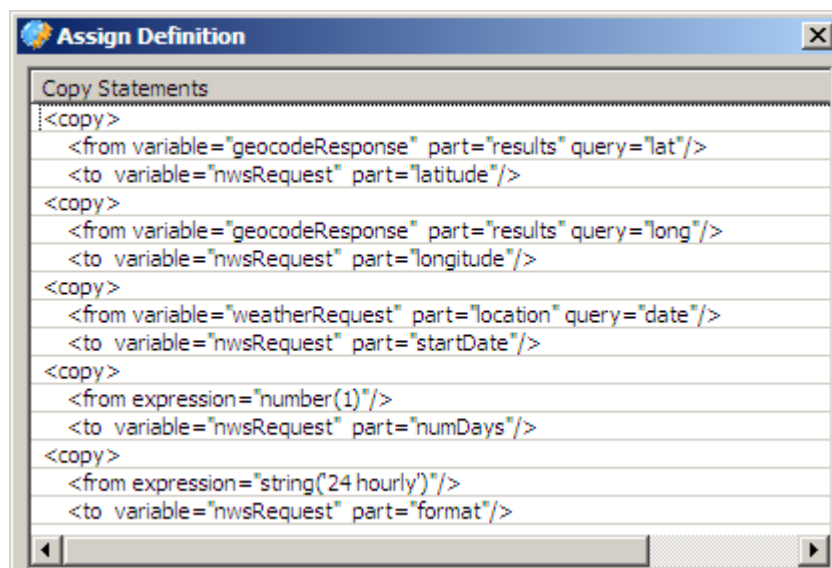
The Geocoder Web service returns its results in the geocoderResponse variable. The geocodeResponse variable is of type "p2210:geocodeAddressResponse". Again displaying the WSDL element in the WSDL dialog shown the following message definition:

```
message (Ref:13):geocodeAddressResponse Ref:13NamRef:p2210
    part name:results Ref:13 type:ArrayOf_GeocoderAddressResult Ref:13
```

## 5.4 Step 4 – BPEL Process Passes Latitude, Longitude, and Date to NWS

Now examine the properties for the assign statement following the geocode invoke, and examine the "Assign definition". The copy statements are shown below:

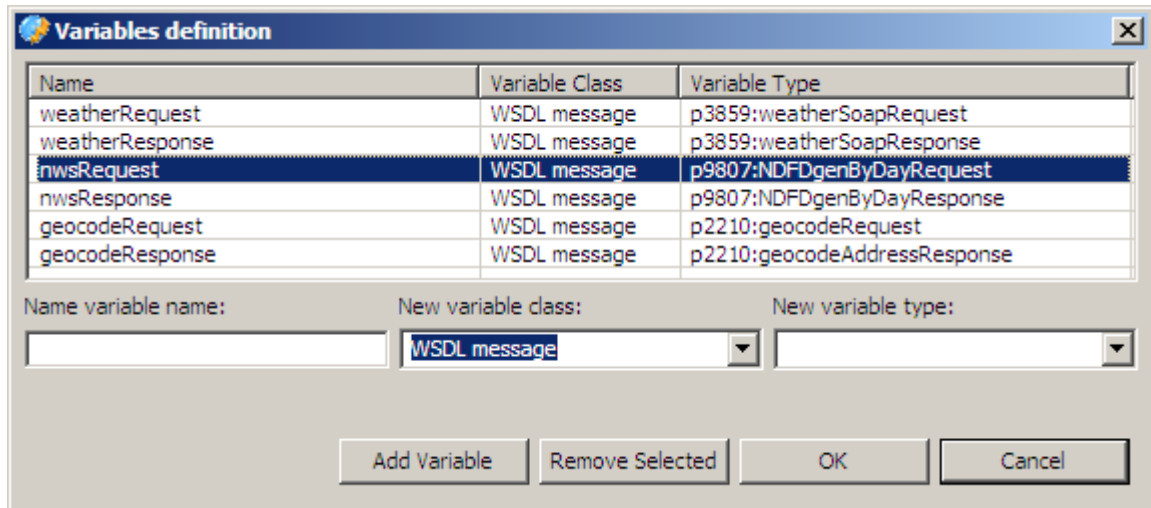
Figure 5-7. Assign Statement Following the "Invoke" Activity



The first two copy statements extract the latitude and longitude from the geocodeResponse variable to the nwsRequest variable. The third copy statement copies the current data from the weatherRequest input variable to the nwsRequest variable. The final two copy statements set up the other required data in the nwsRequest variable.

The variables dialog can be used to examine the type of each variable as shown below:

Figure 5-8. "nwsRequest" Variable in Variables Definition Dialog



It can be seen that the nwsRequest variable is of type "p9807:NDFDgenByDayRequest". This type can be seen by using Zenflow -> Link WSDL Files and select the "WSDL Element definition tab". The message definition is shown below:

```
message (Ref:6):NDFDgenByDayRequest Ref:6NamRef:p9807
  part name:latitude Ref:6 type:decimal Ref:9
  part name:longitude Ref:6 type:decimal Ref:9
  part name:startDate Ref:6 type:date Ref:9
  part name:numDays Ref:6 type:integer Ref:9
  part name:format Ref:6 type:string Ref:9
```

It can be seen that there is a copy statement to initialize each part of the message definition. It is necessary to consult the NWS Web service definition and WSDL to fully understand this interface.



## 5.5 Step 5 – NWS Returns Weather Forecast

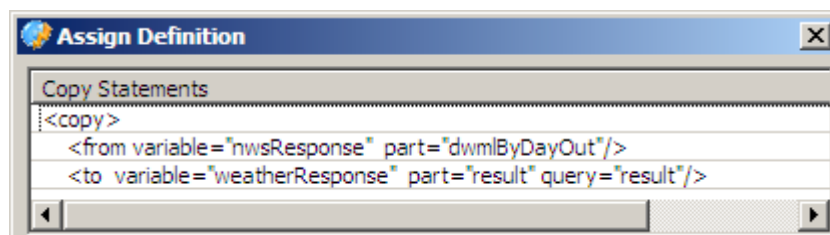
The final invoke passes the above information to the NWS Web service. The returned weather forecast is returned in a string containing an XML document. This string is returned in the `nwsResponse` variable. This variable is of type `"p9807:NDFDgenByDayResponse"`. This message type can be seen by using Zenflow -> Link WSDL Files and select the "WSDL Element definition tab". The message definition is shown below:

```
message (Ref:6):NDFDgenByDayResponse Ref:6NamRef:p9807
    part name:dtmlByDayOut Ref:6 type:string Ref:9
```

## 5.6 Step 6 – BPEL Process Returns Weather Forecast to Client Application

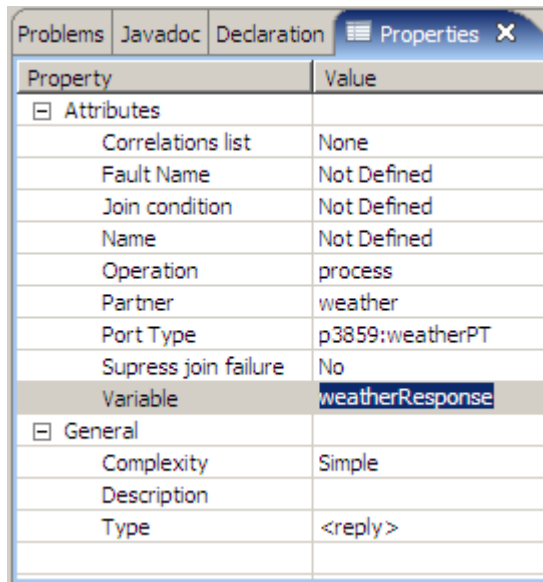
The final "Assign" activity copies the NWS return data to the variable that is returned to the client of the BPEL process. The copy statement for this is shown below:

Figure 5-9. Assign Definition Dialog Showing Final "Assign" Activity



As can be seen, this copy statement extracts the dwmlByDayOut part and puts it into the weatherResponse "result" part. This is returned to the original client in the final "Reply" activity. The properties for the final reply are shown below:

Figure 5-10. Properties Dialog for Final "Reply" Activity



Properties		✕
Property	Value	
[-] Attributes		
Correlations list	None	
Fault Name	Not Defined	
Join condition	Not Defined	
Name	Not Defined	
Operation	process	
Partner	weather	
Port Type	p3859:weatherPT	
Suppress join failure	No	
Variable	weatherResponse	
[-] General		
Complexity	Simple	
Description		
Type	<reply>	

---

## Chapter 6. Deploying and Running the “weather2” BPEL Process

This section explains how to run the “weather2” BPEL process that has been described in this tutorial. This assumes that the user has the BSOA Orchestra engine running (use “bsoap start”, if it is not), and the computer has Internet connectivity to [www.weather.gov](http://www.weather.gov) and [rpc.geocoder.us](http://rpc.geocoder.us) in order to actually call these external Web services.



### Note:

Check the Orchestra proxy settings in the [“Orchestra install dir”/bpel/build.properties](#) to ensure that access to external networks is properly configured.

### 6.1 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 weather2 -samples
```



### Note:

The Deploy action described in Section 4.2, “Deploy the New “weather2” Process,” does a “clean” operation as part of its processing.

### 6.2 Deploy the New “weather2” Process

The weather2 process can be deployed into the BPEL engine. This causes the WSDL and BPEL files to be scanned and the corresponding code is generated and compiled.

From the “Orchestra installation location”\BPEL\samples\weather2 directory, use the following command line:

```
bsoap deploy -p weather2 -samples nws/ndfdXML.wsdl geocoder/GeoCoder.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 to be typographical errors.

## 6.3 Run the “weather2” Process

Now it is time to run the weather2 BPEL process. To do so, it is necessary to have a client program that accesses the first "Receive" statement by sending the appropriate message. The client is provided for the weather2 sample. The source code for this client can be seen at:

["Orchestra installation location"](#)  
[\BPEL\samples\weather2\](#)  
[org\objectweb\orchestra\samples\weather2\Weather2Client.java](#)

This client program takes one input parameter, and builds the input message needed by the weather2.bpel process. The input parameter is either an integer from the following list, or a US address in single quotes. If no parameter is entered, or an invalid parameter is entered, then the following is displayed:

```
>>>> ERROR: Enter a single integer value to select an address from the
following list:
>>>>          1 = 1600 Pennsylvania Avenue, Washington DC
>>>>          2 = 601 Whitehead St, Key West, Fl
>>>>          3 = 13430 N Black Canyon Highway, Phoenix Az
>>>>          4 = 400 Broad St, Seattle Wa
>>>>          5 = 233 Wacker Drive, Chicago Il
>>>>          6 = 350 Fifth Ave, New York NY
>>>>
>>>>          Or enter a complete US address within single quotes.
```

For example, the following command line will display the weather forecast for the US Capitol:

```
bsoap launch -p weather2 -cc
org.objectweb.orchestra.samples.weather2.Weather2Client 1
```

The following is an example of entering an address directly:

```
bsoap launch -p weather2 -cc
org.objectweb.orchestra.samples.weather2.Weather2Client '233 Wacker
Drive, Chicago Il'
```



### Note:

This address should be entered on one line. It is displayed on two lines here due to the page-size limitation.

When a valid parameter is entered, the BPEL process is invoked, which in turn sends the data along to the Geocoder and National Weather Service Web services as described in this tutorial. The resulting xml document, which contains the weather forecast for the selected location, is displayed.

## Technical publication remarks form

**Title:** BSOA Orchestra

**Reference No.:** 86 A2 50ET 00

**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](mailto: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 50ET 00**