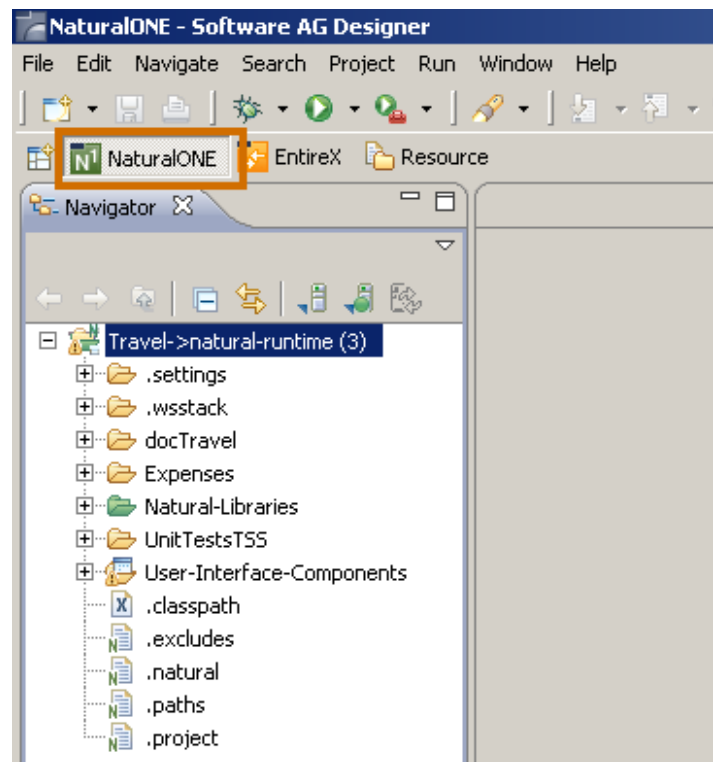


NaturalONE Tutorial

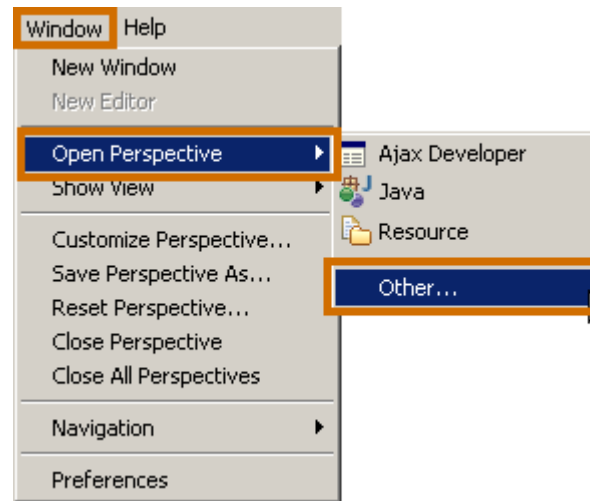
How to publish a NaturalONE subprogram as a Web service?

- 1 Prepare your Eclipse environment
 - 1.1 Activate the **NaturalONE** perspective
 - 1.2 Be sure you activated the **NaturalONE** perspective by clicking on **NaturalONE**.

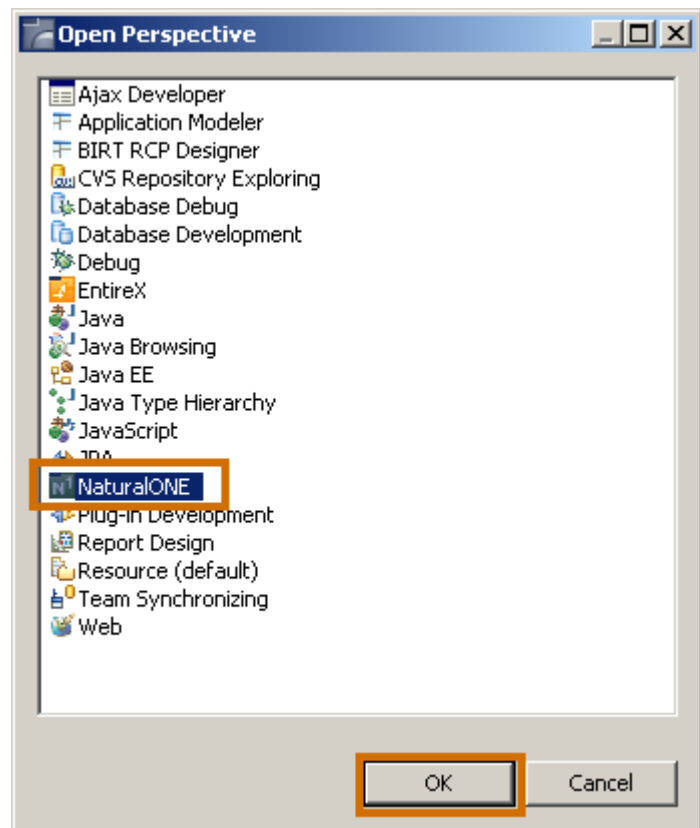
If the **NaturalONE** perspective is not yet visible, follow the next steps.



- 1.3 Select the menu **Window >> Open Perspective >> Other...**



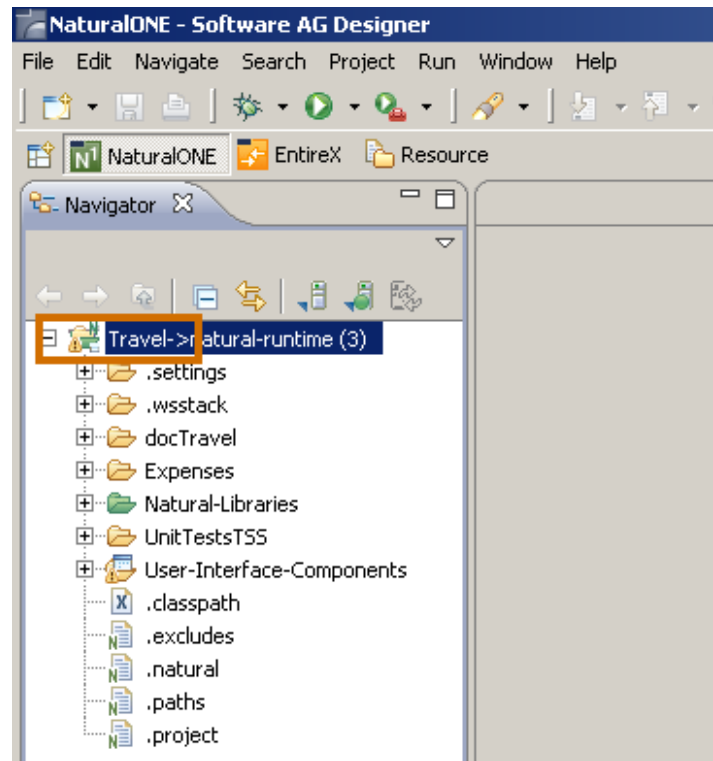
- 1.4 Select **NaturalONE** and press **OK**.



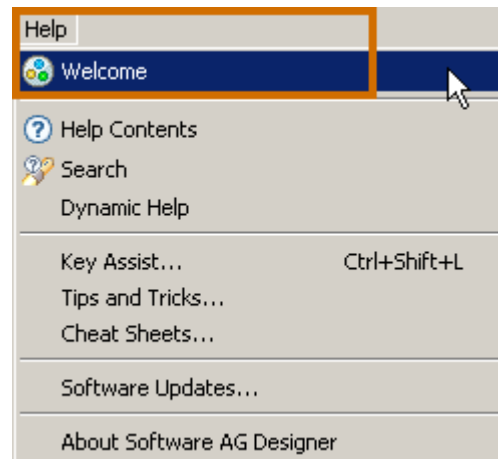
- 1.5 Open the **Travel** sample project

1.6 Be sure the **Travel** project is opened.

If the **Travel** project is not yet open, follow the next steps.



1.7 Select the menu **Help >> Welcome**.



- 1.8 Click on the symbol above **Samples**.



Samples

Try out the samples

- 1.9 Click on **Travel approval application**.

NaturalONE

Learn more about NaturalONE by installing prefabricated samples in your workspace.

After installation, you have to build the sample project in your workspace with "Build Project" or "Rebuild Project".



Hello World application

This sample demonstrates a Natural project implementing a Hello World Natural for Ajax application.



Travel approval application

This sample demonstrates a Natural project implementing a business travel approval application.



Database access application

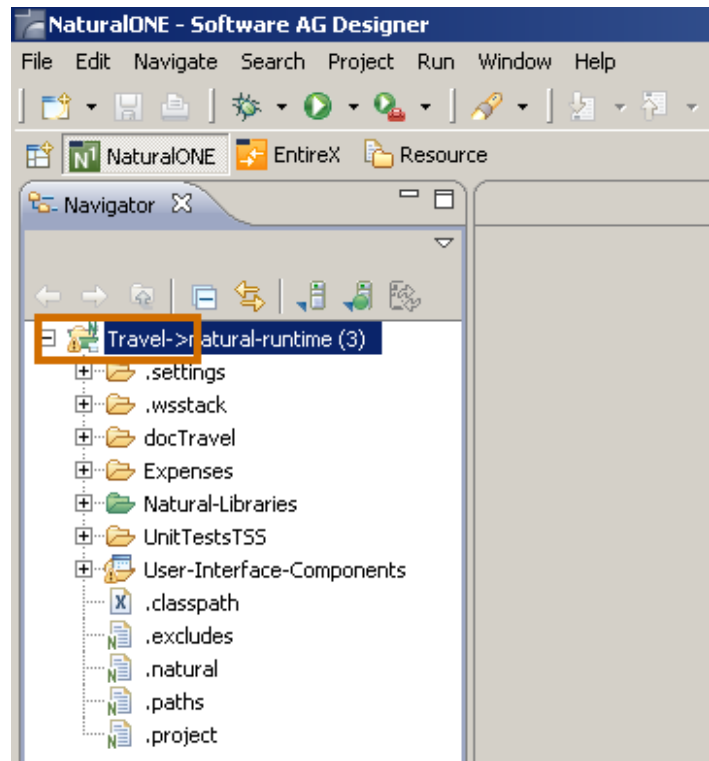
This sample demonstrates a Natural project which implements database access to MySQL and Adabas. It also demonstrates how to access a CSV file.

- 1.10 You can close the **Welcome** view by clicking on the cross 'x'.

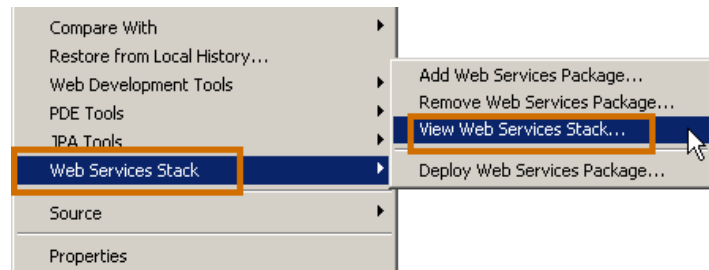


- 1.11 Open the Web services stack view.

- 1.12 Right click on the **Travel** project to open the context menu.



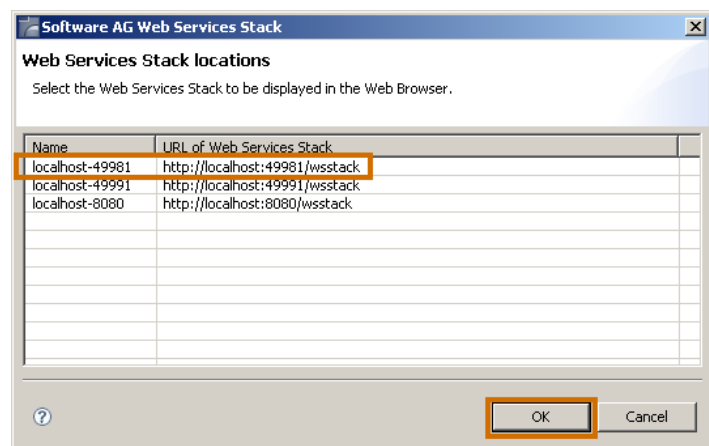
- 1.13 Select **Web Services Stack >> View Web Services Stack...**



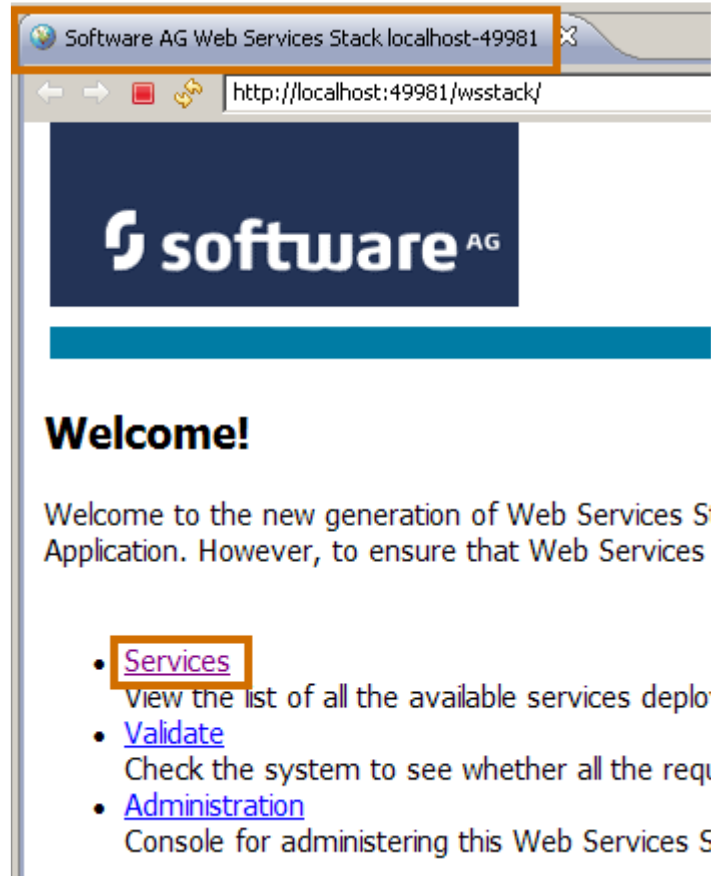
- 1.14 Select the line **localhost-49981** and press **OK**.

Port 49981 is the default NaturalONE Web service stack.

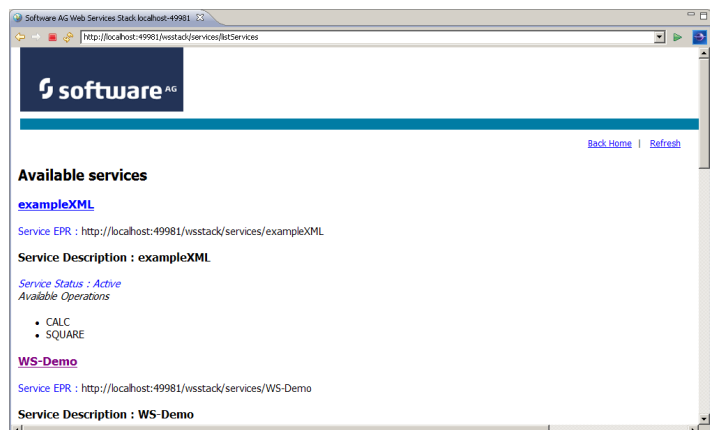
The other entries exist for convenience and compatibility:
 Port 8080 is the Tomcat default.
 Port 49991 is for previous versions of the Web service stack.



- 1.15 In the Web service stack view (i.e. Software AG Web Services Stack localhost-49981) click **Services**.

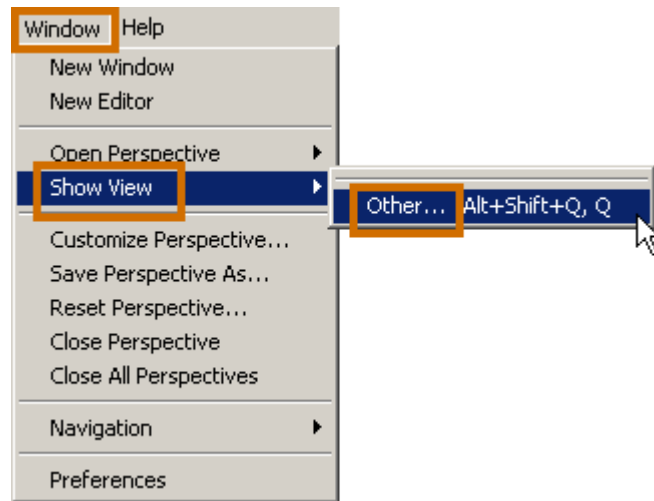


- 1.16 You can now see the services that are already deployed on the Web service stack.

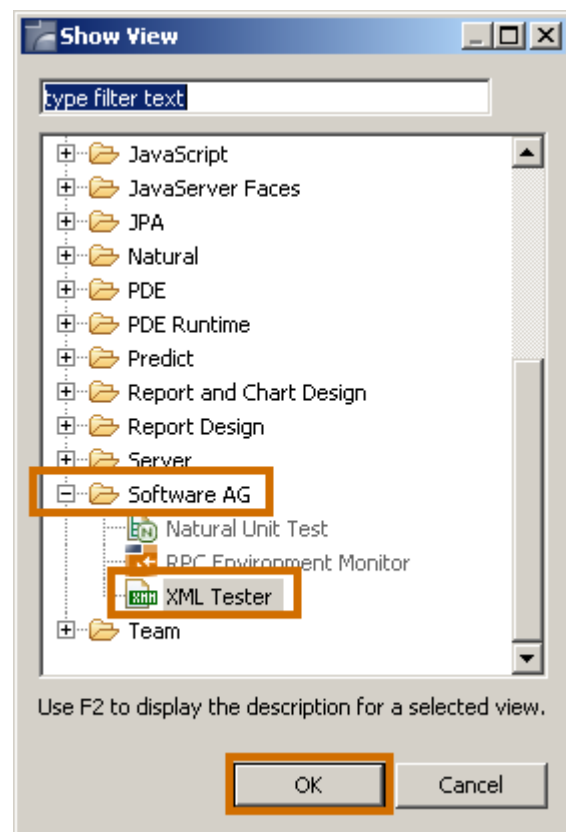


- 1.17 Open the XML Tester view

- 1.18 Select the menu
Window >> Show View >> Other...

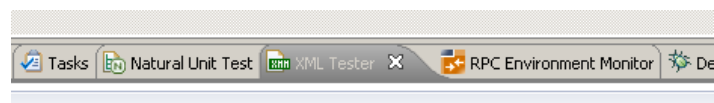


- 1.19 Expand the tree and select
Software AG >> XML Tester.
 Press **OK**.



- 1.20 You can drag the view to different locations by dragging the tab *XML Tester*.

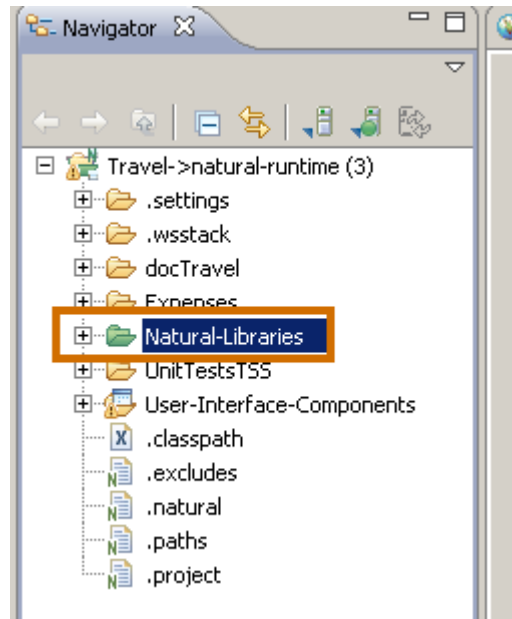
You can even detach it to float freely or open it as a fast view. Check your Eclipse documentation about these



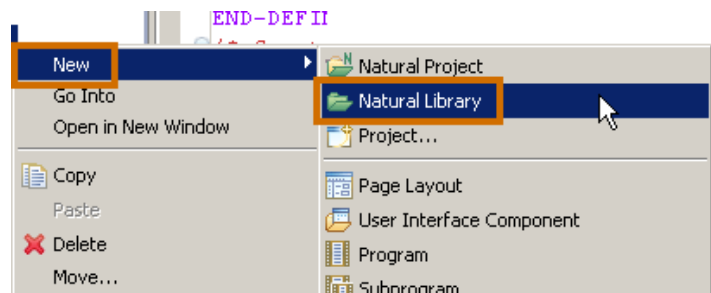
advanced possibilities.

1.21 Create a new Natural library and RES folder for the WS assets

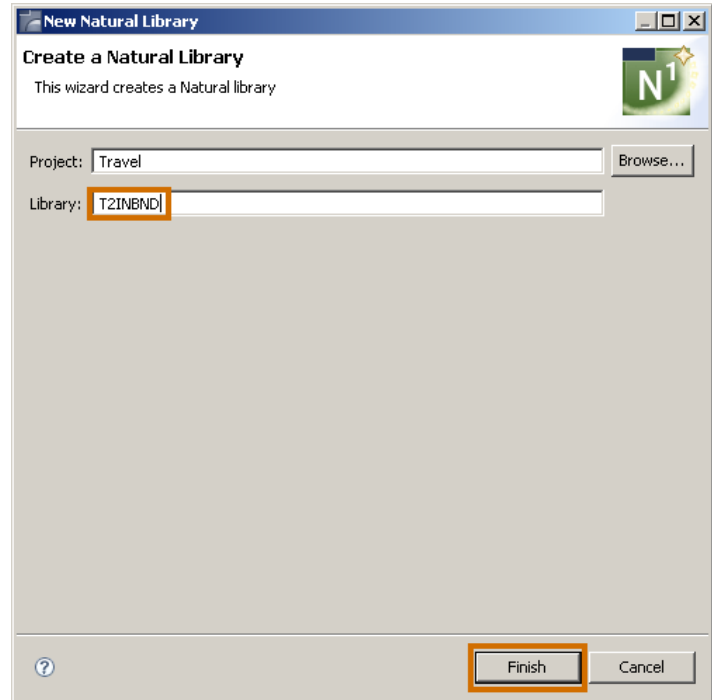
1.21 Right click on **Natural-Libraries** to open the context menu.



1.22 Select menu **New >> Natural Library**.



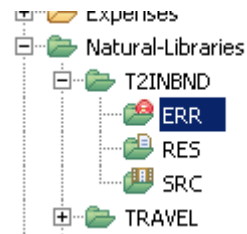
- 1.23 Enter “**T2INBND**” for Library and press **Finish**.



- 1.24 Expand **Natural-Libraries** to see the new library **T2INBND**.

It contains 3 folders: ERR, **RES** and SRC.

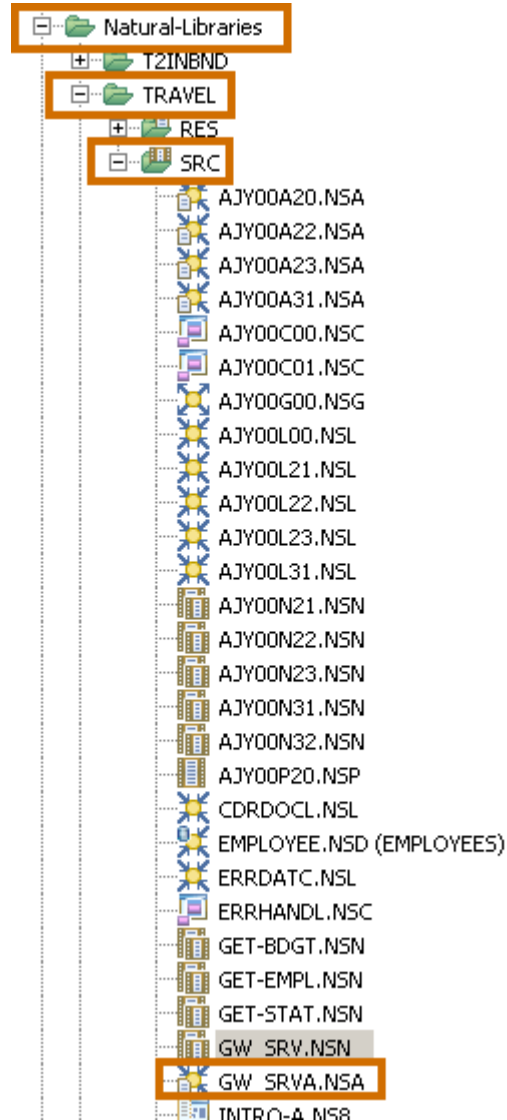
We will use **T2INBND >> RES** later.



- 1.25 Check and prepare your PDA (Parameter Data Area)

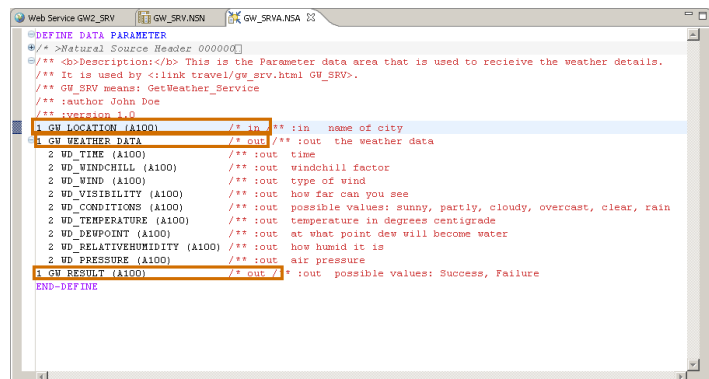
1.26 Expand **Natural-Libraries >> TRAVEL >> SRC.**

Open the PDA **GW_SRV.NSA** by double clicking on it.



1.27 Edit the PDA to define in and out directions for the parameters.

This is shown in more detail below.



1.28 Ensure that `/* in` is written directly after `GW_LOCATION (A100)`.

Ensure that `/* out` is written directly after `GW_WEATHER_DATA`.

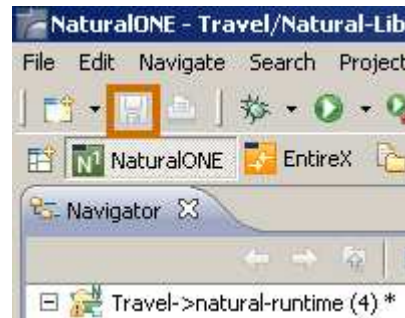
Ensure that `/* out` is written directly after `GW_RESULT (A100)`.

```

/** GW_SRV means: GetWeather_Service
/** :author John Doe
/** :version 1.0
1 GW_LOCATION (A100) /* in /** :
1 GW_WEATHER_DATA /* out /** :
2 WD_TIME (A100) /** :out t:
2 WD_WINDCHILL (A100) /** :out w:
2 WD_WIND (A100) /** :out ty
2 WD_VISIBILITY (A100) /** :out hc
2 WD_CONDITIONS (A100) /** :out pc
2 WD_TEMPERATURE (A100) /** :out te
2 WD_DEWPOINT (A100) /** :out at
2 WD_RELATIVEHUMIDITY (A100) /** :out hc
2 WD_PRESSURE (A100) /** :out a:
1 GW_RESULT (A100) /* out /* :
END-DEFINE

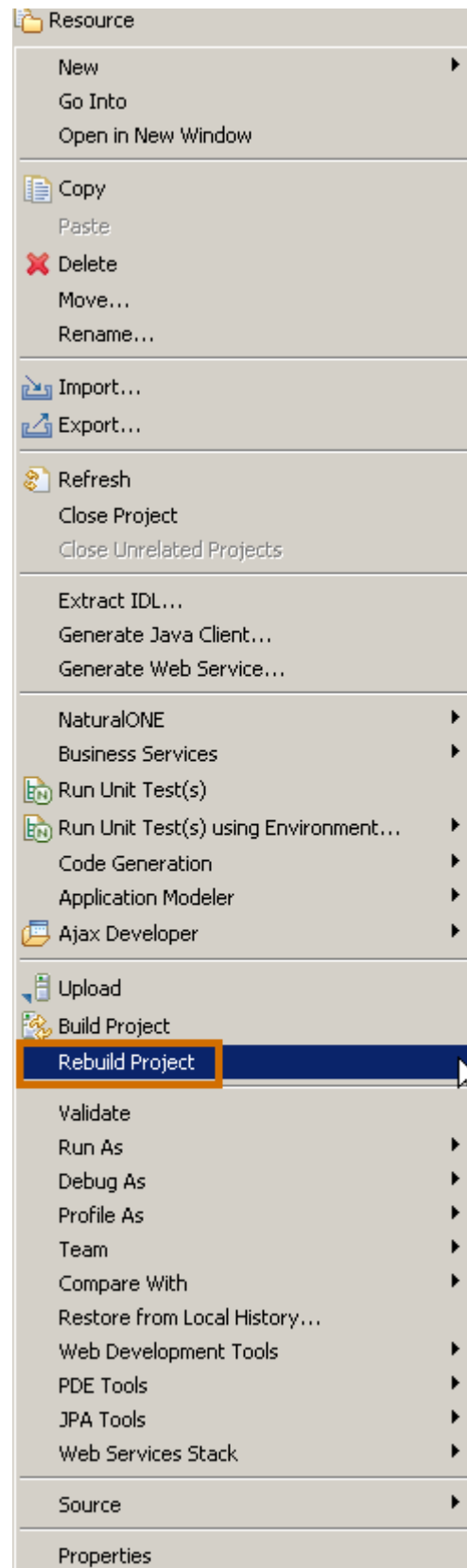
```

1.29 Save the PDA by pressing the Save button.



1.30 Right click on the Travel project to get the context menu.

Select **Rebuild Project**.

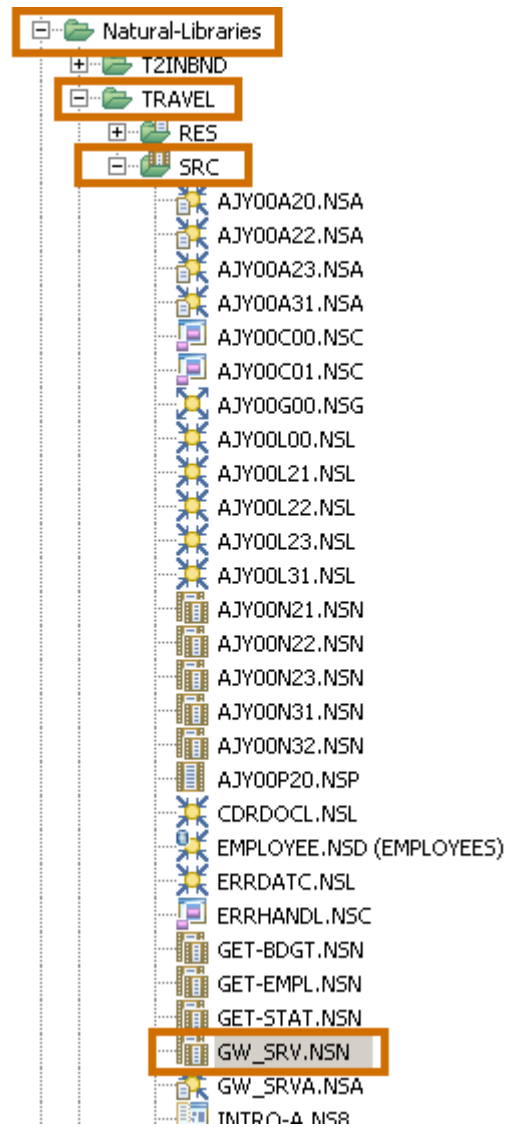


2 Create the Web service

2.1 Open the subprogram

2.2 Expand *Natural-Libraries* >> *TRAVEL* >> *SRC*.

Open the subprogram **GW_SRV.NSN** by double clicking on it.



2.3 The editor view for **GW_SRV.NSN** is opened.

```

Software AG Web Services Stack localhost-49981  GW_SRV.NSN
>Natural Source Header 000000
/** <b>Description:</b> This is the subprogram that is used to initialize the weather data.
/** (GW_SRV means: GetWeather_Service)
/** :author John Doe
/** :Version 1.0
@DEFINE DATA
PARAMETER USING GW_SRVA
END-DEFINE
/* Create a default weather data
MOVE '12:30 pm' TO WEATHER_DATA.LOCATION
MOVE '5°C' TO GW_WEATHER_DATA.WD_TIME
MOVE '3' TO GW_WEATHER_DATA.WD_WINDCHILL
MOVE 'clear' TO GW_WEATHER_DATA.WD_WIND
MOVE 'clear' TO GW_WEATHER_DATA.WD_VISIBILITY
MOVE 'partly sunny' TO GW_WEATHER_DATA.WD_CONDITIONS
MOVE '10°C' TO GW_WEATHER_DATA.WD_TEMPERATURE
MOVE '22°C' TO GW_WEATHER_DATA.WD_DEWPOINT
MOVE '50 %' TO GW_WEATHER_DATA.WD_RELATIVEHUMIDITY
MOVE '1013' TO GW_WEATHER_DATA.WD_PRESSURE
*
MOVE 'Success' TO GW_RESULT
@
/* Give Washington a different weather
IF GW_LOCATION = 'SCAN Washington'
MOVE 'clear' TO GW_WEATHER_DATA.WD_CONDITIONS
MOVE '4°C' TO GW_WEATHER_DATA.WD_TEMPERATURE

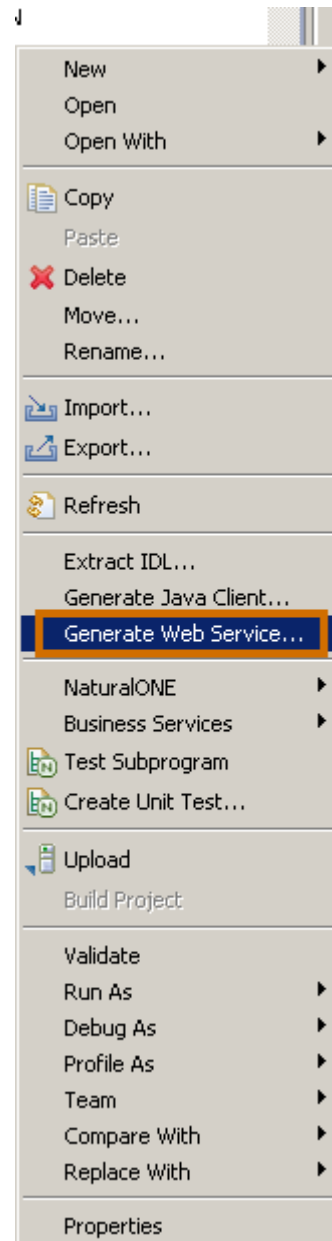
```

2.4 Create the Web service

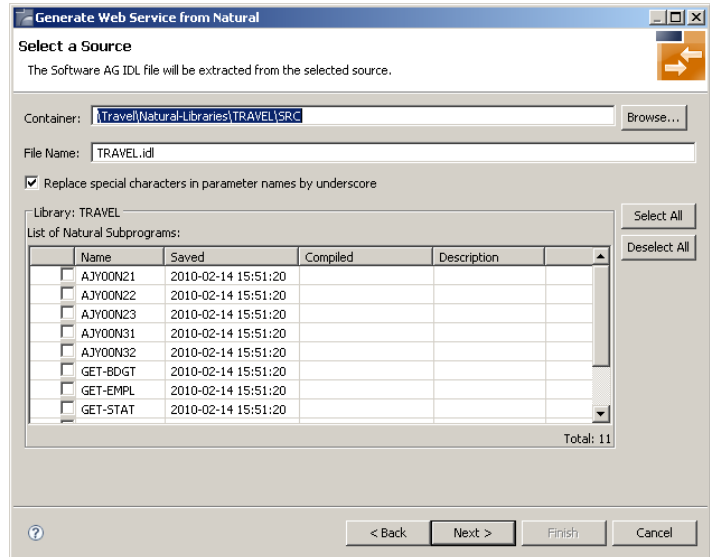
2.5 Right click on **GW_SRV.NSN** in the **Navigator** view to open the context menu.

Select

Generate Web Service...



2.6 The **Generate Web Service from Natural** wizard is opened.

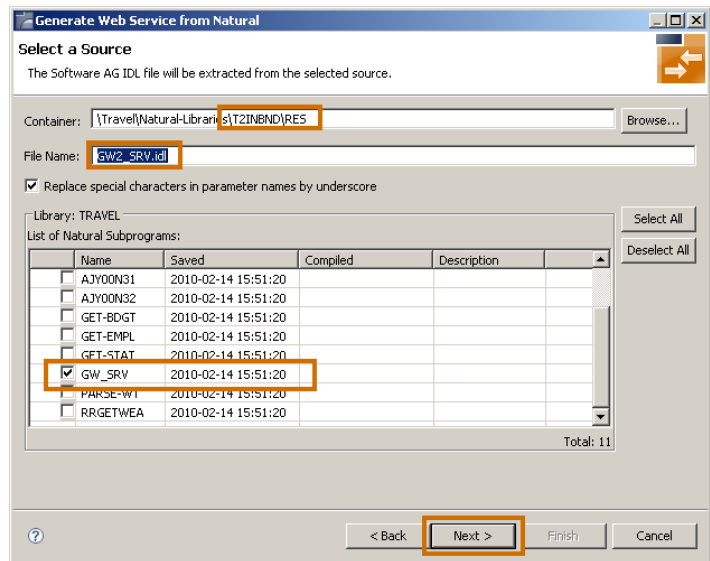


2.7 Change the **Container** to “\\Travel\Natural-Libraries\T2INBND\RES”.

Change the **File Name** to “GW2_SRV.idl”.

Ensure that **GW_SRV** is selected.

Press **Next >**.

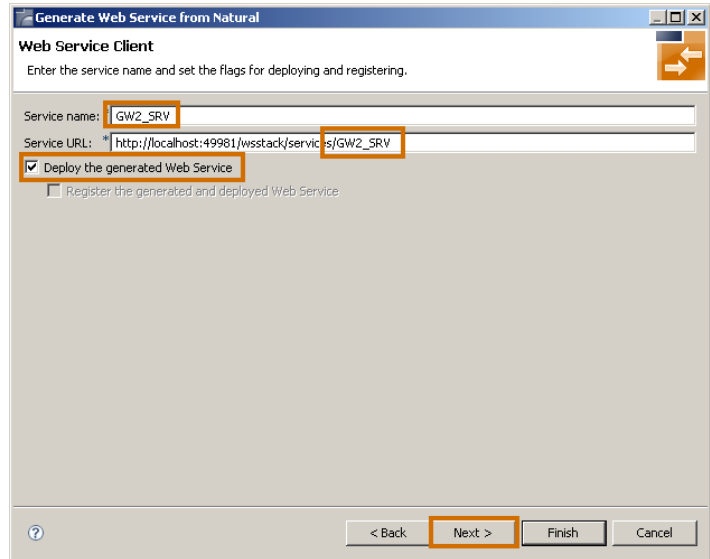


2.8 Change the *Service name* to “**GW2_SRV**”.

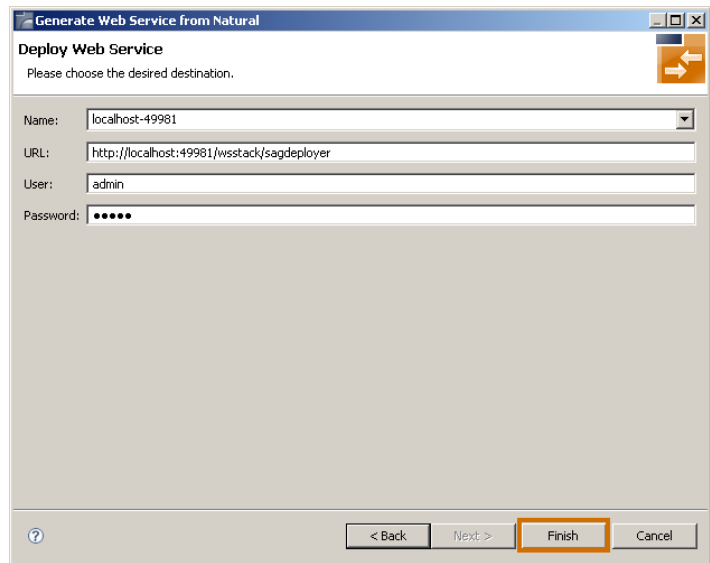
Change the end of the *Service URL* to “**GW2_SRV**”.

Check the **Deploy the generated Web Service** check box.

Press **Next >**.



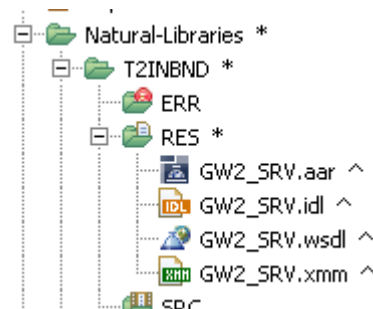
2.10 Press **Finish**.



2.11 Check created assets

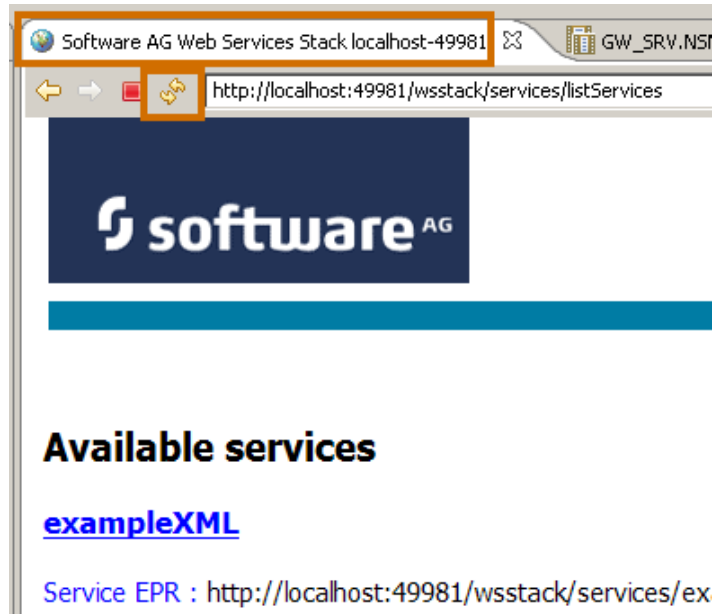
2.12 4 assets have been created in the **T2INBND >> RES folder**:

1. An application archive (.aar).
2. An interface definition (.idl).
3. A Web service definition (.wsdl).
4. An XML mapping file (.xmm)



The textual decorators will disappear after **Rebuild Project**.

- 2.13 Click the **Refresh** button in the Web service stack view.



- 2.14 Scroll down to see your newly created **GW2_SRV** service.

[GW2_SRV](#)

Service EPR : http://localhost:49981/wsstack/services/GW2_SRV

Service Description : GW2_SRV

Service Status : Active
Available Operations

- GW_SRV

- 2.15 **Get the WSDL**

- 2.16 Click on **GW2_SRV** to get the WSDL of your Web service

[GW2_SRV](#)

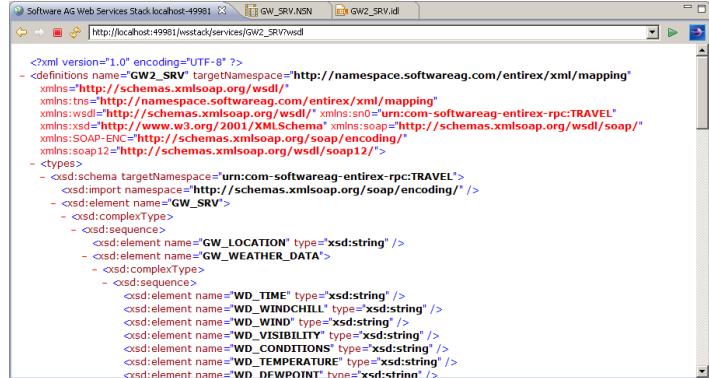
Service EPR : http://localhost:49981/wsstack/services/GW2_SRV

Service Description : GW2_SRV

Service Status : Active
Available Operations

- GW_SRV

2.17 The WSDL is displayed in the Web service stack view.



```

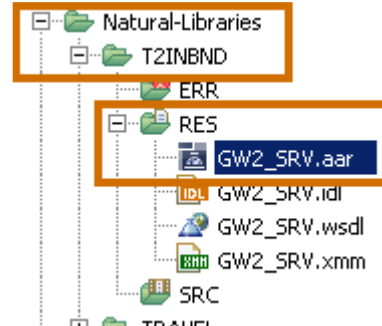
Software AG Web Services Stack localhost:49981
GW_SRV.NSN
GW2_SRV.idl
http://localhost:49981/wstacd/services/GW2_SRV/vwsdl

<?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="GW2_SRV" targetNamespace="http://namespace.softwareag.com/entirex/xml/mapping"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://namespace.softwareag.com/entirex/xml/mapping"
  xmlns:wsd="http://schemas.xmlsoap.org/wsdl/" xmlns:sn0="urn:com-softwareag-entirex-rpc:TRAVEL"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/">
- <types>
- <xsd:schema targetNamespace="urn:com-softwareag-entirex-rpc:TRAVEL">
  <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
- <xsd:element name="GW_SRV">
- <xsd:complexType>
  - <xsd:sequence>
    <xsd:element name="GW_LOCATION" type="xsd:string" />
    <xsd:element name="GW_WEATHER_DATA">
  - <xsd:complexType>
    - <xsd:sequence>
      <xsd:element name="WD_TIME" type="xsd:string" />
      <xsd:element name="WD_WINDCHILL" type="xsd:string" />
      <xsd:element name="WD_WIND" type="xsd:string" />
      <xsd:element name="WD_VISIBILITY" type="xsd:string" />
      <xsd:element name="WD_CONDITIONS" type="xsd:string" />
      <xsd:element name="WD_TEMPERATURE" type="xsd:string" />
      <xsd:element name="WD_DEWPOINT" type="xsd:string" />

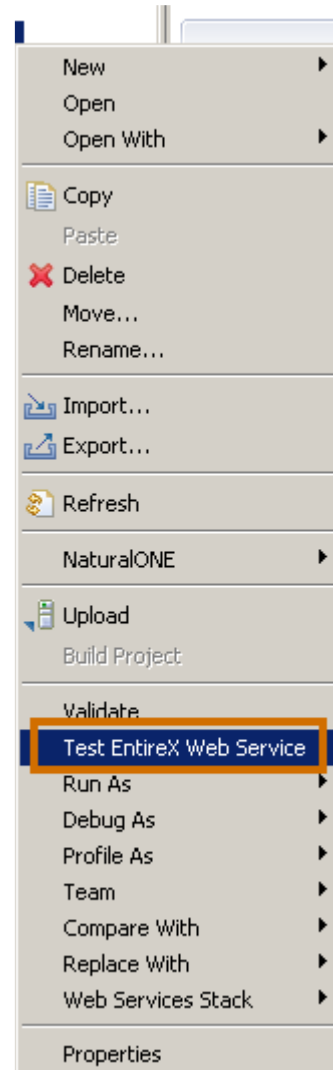
```

3 Test the Web service

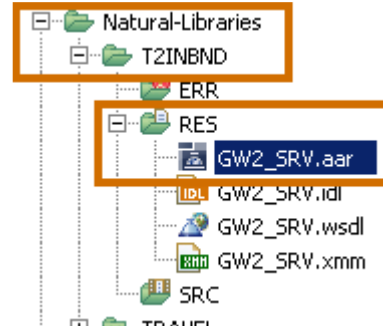
3.1 Right click on **GW2_SRV.aar** to get the context menu.



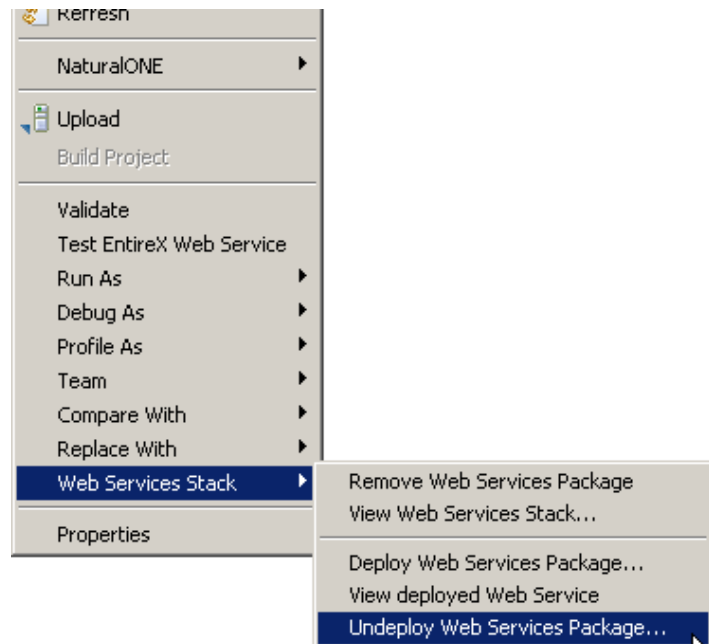
3.2 Select **Test EntireX Web Service**.



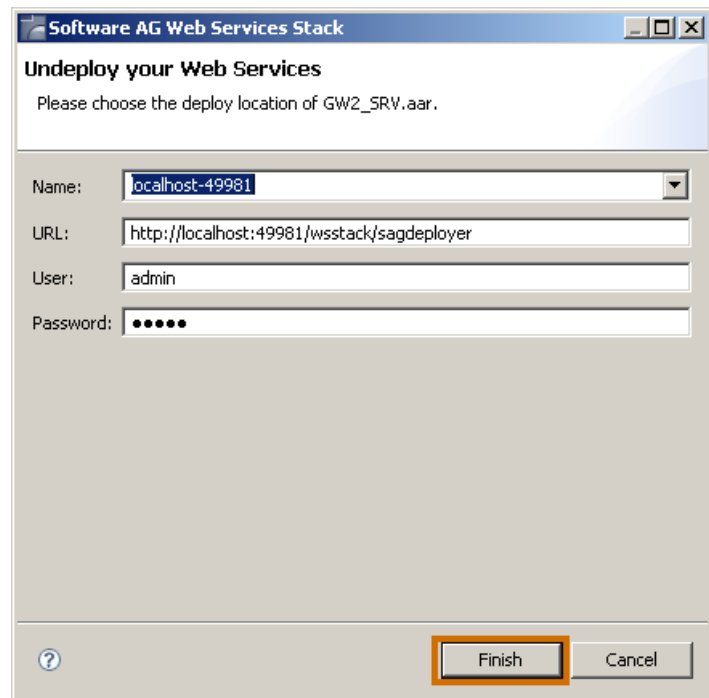
- 4 Undeploy the Web service
- 4.1 Right click on **GW2_SRV.aar** to get the context menu.



- 4.2 Select **Web Services Stack >> Undeploy Web Services Package...**



4.3 Click **Finish**.



4.4 Click the **Refresh** button in the Web service stack view.

Neither the WSDL nor the GW2_SRV service is displayed in the list of Available services.

