Wrapper for WebSphere Application Server 5.1 on Windows

Document version 1.0

Har M Puri
Staff Software Engineer
IBM Integrated Runtime development
Rochester MN 55901

Sahdev Zala
Software Engineer
IBM Integrated Runtime development
Rochester MN 55901

Contents

Figures

Introduction

The IBM Integrated Runtime (IR) product contains WebSphere Application Server – Express, Version 5.1. Some of the larger Independent Software Vendors (ISV), Solution Developers (SD), and Solution Integrators (SI) want a mechanism to use the IBM IR deployment framework to deliver WebSphere Application Server 5.1 (WAS) and to take advantage of the increased capability of the base WAS product or its licensing terms.

This document is targeted to the ISV personnel or other developer who wants to create a wrapper for the base WAS product for use in a solution. This application wrapper is developed on Windows. It can be developed on the Linux platform with some changes.

To keep this component as simple as possible, the IBM HTTP Server (IHS) included with the WebSphere Application Server 5.1 product is used. The disadvantage of this approach is that the deployer cannot target separate machines for the IHS and WebSphere Application Server installations. If this approach presents a problem, the solution developer can chose to use the IHS Integrated Runtime component and disable installation of the IHS in the WebSphere Application Server response file.

Methodology

This paper is separated into several sections to assist in completing the following tasks:

- Modifying a response file.
- Building the wrapper.
- Testing the wrapper.

Each section might include the following:

- **Description** – A general description of the section or exercise.
- **Interactions** – The dependencies of the component within the solution, and the products that are depending on the component. Descriptions of interactions between other components within the solution and the subject component.
- **Control** – How to start and stop the component, and the methods used to determine if the component is running.
- **Configuration** – A description of the methods used to configure the component.

- **Problem Determination** – How to determine the cause of problems with the component.
- **Additional Information** – Where to find additional information about the component.
- **Exercises** – Hands-on exercises with the component. Exercises are separated into four types:
    - Installation Exercises – The components are installed in these exercises.
    - Configuration Exercises – The components are configured in these exercises.
    - Control Exercises – The components are controlled in some way in these exercises.
    - Optional Exercises – Some additional features or functions of the components are used in these exercises.

Conventions

This paper uses the following conventions to illustrate various types of information.

- All exercises are numbered, and each step within the exercise is numbered. It is important to complete the steps and exercises in the order they are provided. If an exercise or step is not required, it is indicated as optional.
- Throughout the exercises, informational text is formatted like this current paragraph.
- Commands are shown in the following font and format and should be typed exactly as shown, including letter case.
- `Type this exactly - This command should be typed as though on a single command line. If part of the line is already present, do not type it. For example the drive d:\> may already be displayed in the command prompt and there is no need to type it again.`

  ```
  install.bat d:\
  ```

- Code samples are show in the following font and format.

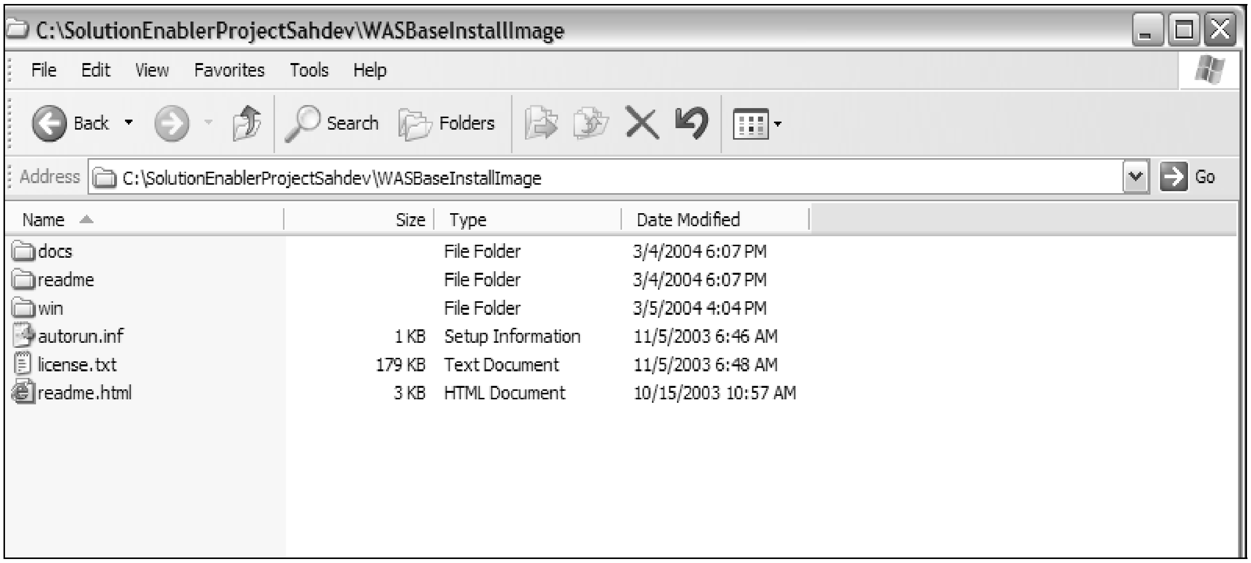
  ```
  <html>
  <body>
  <p>This is sample code.</p>
  </body>
  </html>
  ```

WebSphere Application Server CD

The source for creating the WebSphere Application Server application is a CD that you can purchase from IBM through the Partner World O.E.M sales channel. You must purchase an appropriate license to help bundle WebSphere Application Server with a solution. Please contact a channel sales representative for licensing issues.

CD Contents

The contents and directory structure of the WebSphere Application Server CD are similar to the following illustration:



C:\SolutionEnablerProjectSahdev\WASBaseInstallImage

Address: C:\SolutionEnablerProjectSahdev\WASBaseInstallImage

Name	Size	Type	Date Modified
docs		File Folder	3/4/2004 6:07 PM
readme		File Folder	3/4/2004 6:07 PM
win		File Folder	3/5/2004 4:04 PM
autorun.inf	1 KB	Setup Information	11/5/2003 6:46 AM
license.txt	179 KB	Text Document	11/5/2003 6:48 AM
readme.html	3 KB	HTML Document	10/15/2003 10:57 AM

Figure 1 WebSphere Application Server CD directory

Response file

A WebSphere Application Server response file installation helps you install WebSphere Application Server without interactive user input. Response files contain installation choices and configuration data that would otherwise be provided by the user interactively on the installation Graphical User Interface (GUI). This is useful because it automates the process and helps you to embed the WebSphere Application Server installation process transparently inside an installation program. The response file installation is one of the ways that WebSphere Application Server provides a complete solution for your integrated application server needs.

A sample response file (responsefile.txt) is available on the WebSphere Application Server installation CD in the win\ directory.

```
# *******************************************
#
# Response file for WebSphere Application Server 5.1 Install
#
# Please follow the comments to use the response file and
# understand the various options.  You must carefully complete
# or change the various values. If the values are not completed
# properly, the install may be unsuccessful.
#
# NOTE: This file is for silent install only.
#
                                             #
# IMPORTANT: ALL VALUES MUST BE ENCLOSED IN DOUBLE QUOTES ( "" ).
#
# *******************************************

# *******************************************
# This value is required.
# PLEASE DO NOT REMOVE THIS VALUE.
# *******************************************

-W setupTypes.selectedSetupTypeId="custom"


# *******************************************
# Below is the beginning of the response file that needs to be
# filled in by the user.
# *******************************************

# *******************************************
# The below value specifies silent install. This value
# indicates that the install will be silent.
# *******************************************

                                          -silent


# *******************************************
# WebSphere Application Server Install Location
#
```

Figure 2 Partial content of a generated response file

The first step of the response file installation process is to modify the response file. When installing WebSphere Application Server using the interactive setup utility, you are prompted to make installation choices and provide necessary configuration data before installation takes place. Using the following instructions you can create a response file that contains the choices you would make during installation. Later, during the application development, you can encode a process to replace the response file variables (for example, *WebSphere Application Server node*) with text you can use to prompt the user during the deployment of the solution.

Create the response file

1. Copy the responsefile.txt file from the win\ directory to the My Documents directory. Rename it `baseresponsefile.txt`.
2. Modify `baseresponsefile.txt` to disable installation of the embedded JMS server. figure 3 shows the changes that should be made: Please note that in the interest of making some strings fit in the table partial strings can be used.

Original	Change to
-P mqSeriesBean.active="true"	-P mqSeriesBean.active="false"
-P mqSeriesServerBean.active="true"	-P mqSeriesServerBean.active="false"
-P mqSeriesServerBean.installLocation="C:\Program Files\IBM\W	#-P mqSeriesServerBean.installLocation="C:\Program Files\IBM\
-P mqSeriesClientBean.installLocation="C:\Program Files\IBM\W	#-P mqSeriesClientBean.installLocation="C:\Program Files\IBM\
-P mqSeriesSamplesBean.active="true"	-P mqSeriesSamplesBean.active="false"
-P samplesBean.active="true"	-P samplesBean.active="false"
-P performanceAndAnalysisToolsBean.active="true"	-P performanceAndAnalysisToolsBean.active="false"
-P tivoliPerfBean.active="true"	-P tivoliPerfBean.active="false"
-P DCMBean.active="true"	-P DCMBean.active="false"
-P performanceServletBean.active="true"	-P performanceServletBean.active="false"
-P logAnalyzerBean.active="true"	-P logAnalyzerBean.active="false"
-P javadocBean.active="true"	-P javadocBean.active="false"
-W serviceSettingsWizardBean.userName="YOUR_USER_NAME"	-W serviceSettingsWizardBean.userName=" **put user name here**"
-W serviceSettingsWizardBean.password="YOUR_PASSWORD"	-W serviceSettingsWizardBean.password="**put passwaord here**"
#-W coexistenceOptionsBean.doCoexistence="true"	-W coexistenceOptionsBean.doCoexistence="true"
#-W coexistencePanelBean.useIhs="true"	-W coexistencePanelBean.useIhs="true"
#-W coexistencePanelBean.bootstrapPort="2810"	-W coexistencePanelBean.bootstrapPort="2809"
#-W coexistencePanelBean.ihsPort="81"	-W coexistencePanelBean.ihsPort="80"
#-W coexistencePanelBean.httpTransportPort="9086"	-W coexistencePanelBean.httpTransportPort="9086"
#-W coexistencePanelBean.httpsTransportPort="9044"	-W coexistencePanelBean.httpsTransportPort="9044"
#-W coexistencePanelBean.adminConsolePort="9091"	-W coexistencePanelBean.adminConsolePort="9091"
#-W coexistencePanelBean.secureAdminConsolePort="9444"	-W coexistencePanelBean.secureAdminConsolePort="9444"
#-W coexistencePanelBean.soapConnectorAddress="8881"	-W coexistencePanelBean.soapConnectorAddress="8881"

Figure 3 Changes to the default response file

3. If you want to expose additional parameters, they can also be uncommented, but you need to provide defaults or modify additional files to accept user input during deployment.
4. This process leaves a default response file (`baseresponsefile.txt`) in the My Documents directory.

Creating the WebSphere Application Server application

In IBM Integrated Runtime terminology, an *application* is one of the components of a *solution*. Examples of applications include WebSphere Application Server or the actual application program package written by the developer. Applications are contained in *application wrappers*. One or more application wrappers are combined into a complete solution by creating a *solution wrapper*.

First you must create an application wrapper for WebSphere Application Server and then package it into a test solution. Since the wrapper for the WebSphere Application Server Express Edition is similar to the WebSphere Application Server Workgroup Server Edition, it makes a good starting point. You can modify a copy to make a wrapper for WebSphere Application Server.

Copy the WebSphere Application Server EE project

1. Start the Solution Developer by selecting **Start > IBM Integrated Runtime > Solution Developer**.

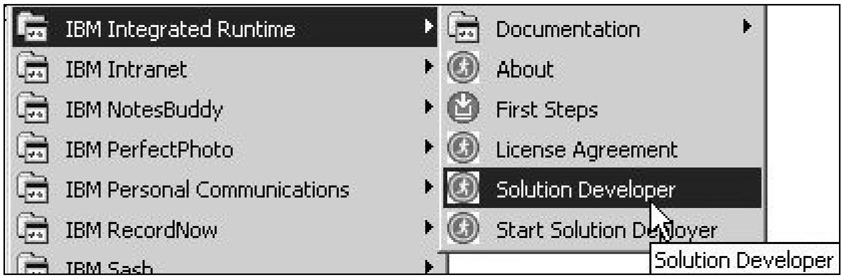


Figure 4 Solution developer

2. Right-click IRU_WASExpress5_1Win and select **Copy**.
3. Right-click again and select **Paste**.
4. Enter the new name for the application in the dialog box. For the remainder of this document, the name of the application is IRU_WAS5_1Win.
5. You have the following tree structure in Solution Developer:

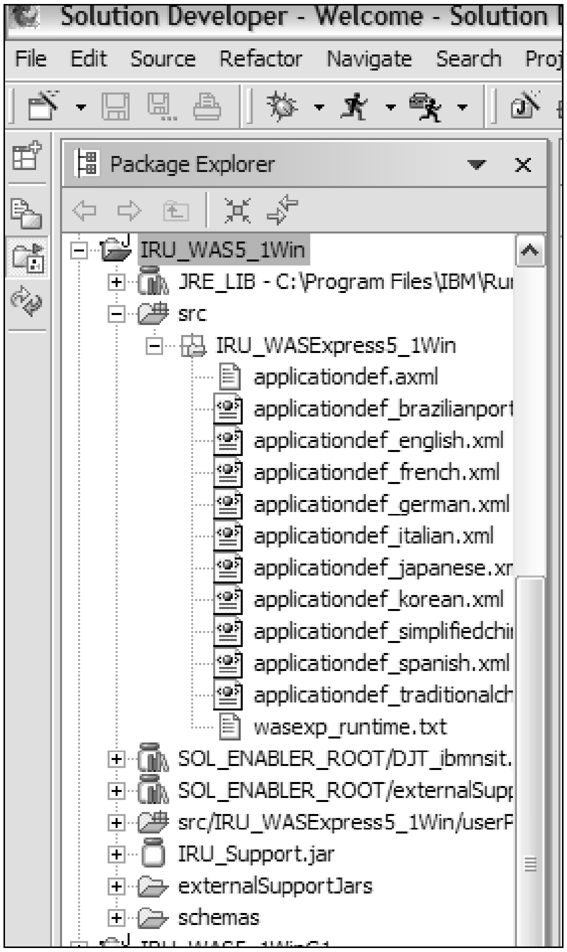

Solution Developer - Welcome - Solution
File Edit Source Refactor Navigate Search Proj
Package Explorer
IRU_WAS5_1Win
JRE_LIB - C:\Program Files\IBM\Ru
src
IRU_WASExpress5_1Win
applicationdef.axml
applicationdef_brazilianport
applicationdef_english.xml
applicationdef_french.xml
applicationdef_german.xml
applicationdef_italian.xml
applicationdef_japanese.xr
applicationdef_korean.xml
applicationdef_simplifiedchi
applicationdef_spanish.xml
applicationdef_traditionalch
wasexp_runtime.txt
SOL_ENABLER_ROOT/DJT_ibmnsit.
SOL_ENABLER_ROOT/externalSupp
src/IRU_WASExpress5_1Win/userP
IRU_Support.jar
externalSupportJars
schemas

Figure 5 Structure of the new project after the copy operation

Modify the IRU_WAS5_1Win project

Modify the copied files to fit the new project by performing the following steps:

1. Delete the response file wasexp_runtime.txt from the directory src/IRU_WASExpress5_1Win. From the Windows Explorer, select baseresponsefile.txt from the folder my documents and copy it to the location of deleted response file, wasexp_runtime.txt. You can also select **File > Import > File System** to import the file instead of using the copy and paste method
2. Open the applicationdef.axml file in an editor and make the following changes:
   a. Search for and delete all occurrences of the string `Express (ignore case)`.
   b. Change the name of the response file from wasexp_runtime.txt to `baseresponsefile.txt`.
   c. Delete the following lines of code:
      ```
      <file>IRU_WAS5_1Win/IRU_UpdateAdminConsolePorts5_1Win.bat</file>
      <file>IRU_WAS5_1Win/IRU_UpdatePorts5_1Win.bat</file>
      ```
   d. Replace the following lines as directed:
      1. `<file>com/ibm/iru/iru_was5_1win/WAS5_1WinExit.class</file>`

becomes

```
<file>com/ibm/iru/iru_was5_1win/WAS5_1WinMain.class</file>
```

2.

```
<mainProgram type="custom"
      programName="util/syncInstall.bat"
            logFile="IRU_WAS5_1WinMain.log"
            responseFile="baseresponsefile.txt">
            <arguments>
                  <argument>-silent</argument>
                  <argument>-options</argument>
                  <argument responseFile="true" />
                  <argument>-log</argument>
                  <argument>@ALL</argument>
                  <argument>!../logs/IRU_WAS5_1WinMain.log
                  </argument>
            </arguments>
</mainProgram>
```

becomes

```
<mainProgram type="java"
      programName="com.ibm.iru.iru_was5_1win.WAS5_1WinMain"
      successType="returnCode"
      logFile="IRU_WAS5_1WinMain.log"
      responseFile="baseresponsefile.txt">
      <arguments>
            <argument responseFile="true" />
      </arguments>
</mainProgram>
```

e. Remove the exit program definition by deleting the following lines of code:

```
  <!-**********************************************************--
>
  <!-- Specify an exit program to be run *after* the install.  -->
  <!-- ******************************************************** --
>
  <exitProgram
    type="java"
    successType="returnCode"
programName="com.ibm.iru.iru_was5_1win.WAS5_1WinExit"
    logFile="IRU_Was5_1WinExit.log">
  </exitProgram>
```

f. Add or modify a number of variables. These variables are displayed on the task details pane of the Solution Deployer. You have some discretion here on which variables you want the person deploying the solution to be able to change. For example, if you do not want the deployer to change any port assignments, do not define the variables here. Assign default values in the `baseresponsefile.txt` file. The entire variables section (content between `<variables>` and `</variables> tag`) of the response file is provided in figure 6:

```
<!-- ********************************************************* -->
<!-- Specify a variable to expose the node name.       -->
<!-- ********************************************************* -->
<stringVariable name="nodeNameBean.nodeName" minimumLength="4">
      <defaultData>DefaultNode</defaultData>
      <labelText translatedKey="nodeLabel" />
      <helpText translatedKey="nodeHelp" />
      <propertiesAssociations>
            <propertiesAssociation keyword="nodeNameBean.nodeName" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard" propertyKey="nodeNameBean.nodeName" />
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the node name.     -->
      <!-- ******************************************************* -->
      <inputValidation>
            <valid>
                  <characters
ignoreCase="true">abcdefghijklmnopqrstuvwxyz0123456789</characters>
            </valid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************* -->
<!-- Specify a variable to expose the host name.       -->
<!-- ********************************************************* -->
<stringVariable name="nodeNameBean.hostName" minimumLength="4">
      <defaultData>localhost</defaultData>
      <labelText translatedKey="hostLabel" />
      <helpText translatedKey="hostHelp" />
      <propertiesAssociations>
            <propertiesAssociation keyword="nodeNameBean.hostName" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard" propertyKey="nodeNameBean.hostName" />
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the host name.     -->
      <!-- ******************************************************* -->
      <inputValidation>
            <valid>
                  <characters
ignoreCase="true">.abcdefghijklmnopqrstuvwxyz0123456789</characters>
            </valid>
      </inputValidation>
</stringVariable>

<stringVariable
```

```
        name="installDir"
        minimumLength="4">
        <defaultData>C:\Program Files\WebSphere\AppServer</defaultData>
        <labelText translatedKey="dirLabel"/>
        <helpText translatedKey="dirHelp"/>
        <ismpFileAssociations>
          <ismpFileAssociation
            responseFileName="baseresponsefile.txt"
            propertyKeyType="product"
            propertyKey="wasBean.installLocation"/>
        </ismpFileAssociations>
        <propertiesAssociations>
          <propertiesAssociation keyword="installDir"/>
        </propertiesAssociations>
        <inputValidation>
          <valid>
            <prefixes>
              <prefix ignoreCase="true">A:\</prefix>
              <prefix ignoreCase="true">B:\</prefix>
              <prefix ignoreCase="true">C:\</prefix>
              <prefix ignoreCase="true">D:\</prefix>
              <prefix ignoreCase="true">E:\</prefix>
              <prefix ignoreCase="true">F:\</prefix>
              <prefix ignoreCase="true">G:\</prefix>
              <prefix ignoreCase="true">H:\</prefix>
              <prefix ignoreCase="true">I:\</prefix>
              <prefix ignoreCase="true">J:\</prefix>
              <prefix ignoreCase="true">K:\</prefix>
              <prefix ignoreCase="true">L:\</prefix>
              <prefix ignoreCase="true">M:\</prefix>
              <prefix ignoreCase="true">N:\</prefix>
              <prefix ignoreCase="true">O:\</prefix>
              <prefix ignoreCase="true">P:\</prefix>
              <prefix ignoreCase="true">Q:\</prefix>
              <prefix ignoreCase="true">R:\</prefix>
              <prefix ignoreCase="true">S:\</prefix>
              <prefix ignoreCase="true">T:\</prefix>
              <prefix ignoreCase="true">U:\</prefix>
              <prefix ignoreCase="true">V:\</prefix>
              <prefix ignoreCase="true">W:\</prefix>
              <prefix ignoreCase="true">X:\</prefix>
              <prefix ignoreCase="true">Y:\</prefix>
              <prefix ignoreCase="true">Z:\</prefix>
            </prefixes>
          </valid>
          <invalid>
            <substrings>
              <substring>\\</substring>
            </substrings>
            <characters>*?";/|~<![CDATA[<>]]></characters>
          </invalid>
        </inputValidation>
      </stringVariable>

<!-- ******************************************************** -->
<!-- Specify a variable to expose the bootstrap port.         -->
```

```
<!-- ********************************************************** -->
<stringVariable name="coexistencePanelBean.bootstrapPort" minimumLength="1"
maximumLength="5">
      <defaultData>2809</defaultData>
      <labelText translatedKey="rmiPortLabel" />
      <helpText translatedKey="rmiPortHelp" />
      <propertiesAssociations>
            <propertiesAssociation
keyword="coexistencePanelBean.bootstrapPort" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard" propertyKey="coexistencePanelBean.bootstrapPort" />
      </ismpFileAssociations>
      <!-- ******************************************************** -->
      <!-- Specify validation rules for the bootstrap port.        -->
      <!-- ******************************************************** -->
      <inputValidation>
            <valid>
                  <ranges>
                        <range>1 to 65535</range>
                  </ranges>
            </valid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************** -->
<!-- Specify a variable to expose the soap port.               -->
<!-- ********************************************************** -->
<stringVariable name="coexistencePanelBean.soapConnectorAddress"
minimumLength="1" maximumLength="5">
      <defaultData>8880</defaultData>
      <labelText translatedKey="soapPortLabel" />
      <helpText translatedKey="soapPortHelp" />
      <propertiesAssociations>
            <propertiesAssociation
keyword="coexistencePanelBean.soapConnectorAddress" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard"
propertyKey="coexistencePanelBean.soapConnectorAddress" />
      </ismpFileAssociations>
      <!-- ******************************************************** -->
      <!-- Specify validation rules for the port.                  -->
      <!-- ******************************************************** -->
      <inputValidation>
            <valid>
                  <ranges>
                        <range>1 to 65535</range>
                  </ranges>
            </valid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************** -->
```

```
<!-- Specify a variable to expose the http port.               -->
<!-- ********************************************************* -->
<stringVariable name="coexistencePanelBean.httpTransportPort"
minimumLength="1" maximumLength="5">
      <defaultData>9080</defaultData>
      <labelText translatedKey="httpPortLabel" />
      <helpText translatedKey="httpPortHelp" />
      <propertiesAssociations>
            <propertiesAssociation
keyword="coexistencePanelBean.httpTransportPort" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard" propertyKey="coexistencePanelBean.httpTransportPort"
/>
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the port.                  -->
      <!-- ******************************************************* -->
      <inputValidation>
            <valid>
                  <ranges>
                        <range>1 to 65535</range>
                  </ranges>
            </valid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************* -->
<!-- Specify a variable to expose the https port.              -->
<!-- ********************************************************* -->
<stringVariable name="coexistencePanelBean.httpsTransportPort"
minimumLength="1" maximumLength="5">
      <defaultData>9443</defaultData>
      <labelText translatedKey="httpsPortLabel" />
      <helpText translatedKey="httpsPortHelp" />
      <propertiesAssociations>
            <propertiesAssociation
keyword="coexistencePanelBean.httpsTransportPort" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard" propertyKey="coexistencePanelBean.httpsTransportPort"
/>
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the port.                  -->
      <!-- ******************************************************* -->
      <inputValidation>
            <valid>
                  <ranges>
                        <range>1 to 65535</range>
                  </ranges>
            </valid>
      </inputValidation>
</stringVariable>
```

```
<!-- ********************************************************* -->
<!-- Specify a variable to expose the admin port.              -->
<!-- ********************************************************* -->
<stringVariable name="coexistencePanelBean.adminConsolePort" minimumLength="1"
maximumLength="5">
      <defaultData>9090</defaultData>
      <labelText translatedKey="adminPortLabel" />
      <helpText translatedKey="adminPortHelp" />
      <propertiesAssociations>
            <propertiesAssociation
keyword="coexistencePanelBean.adminConsolePort" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard" propertyKey="coexistencePanelBean.adminConsolePort"
/>
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the port.                  -->
      <!-- ******************************************************* -->
      <inputValidation>
            <valid>
                  <ranges>
                        <range>1 to 65535</range>
                  </ranges>
            </valid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************* -->
<!-- Specify a variable to expose the admins port.             -->
<!-- ********************************************************* -->
<stringVariable name="coexistencePanelBean.secureAdminConsolePort"
minimumLength="1" maximumLength="5">
      <defaultData>9043</defaultData>
      <labelText translatedKey="adminsPortLabel" />
      <helpText translatedKey="adminsPortHelp" />
      <propertiesAssociations>
            <propertiesAssociation
keyword="coexistencePanelBean.secureAdminConsolePort" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard"
propertyKey="coexistencePanelBean.secureAdminConsolePort" />
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the port.                  -->
      <!-- ******************************************************* -->
      <inputValidation>
            <valid>
                  <ranges>
                        <range>1 to 65535</range>
                  </ranges>
            </valid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************* -->
<!-- Specify a variable to allow IHS to install            -->
<!-- ********************************************************* -->
<booleanVariable name="ihsFeatureBean.active" minimumLength="1"
maximumLength="5">
      <defaultData>true</defaultData>
      <labelText translatedKey="ihsLabel" />
      <helpText translatedKey="ihsHelp" />
      <propertiesAssociations>
            <propertiesAssociation keyword="ihsFeatureBean.active" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="product" propertyKey="ihsFeatureBean.active" />
      </ismpFileAssociations>
</booleanVariable>

<!-- ********************************************************* -->
<!-- Specify a variable to expose the IHS install directory.       -->
<!-- ********************************************************* -->
<stringVariable name="ihsFeatureBean.installLocation" minimumLength="4">
      <defaultData>C:\Program Files\IBMHTTPServer</defaultData>
      <labelText translatedKey="IHSdirLabel" />
      <helpText translatedKey="IHSdirHelp" />
      <propertiesAssociations>
            <propertiesAssociation keyword="ihsFeatureBean.installLocation" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="product" propertyKey="ihsFeatureBean.installLocation" />
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the install directory.     -->
      <!-- ******************************************************* -->
      <inputValidation>
      <valid>
            <prefixes>
              <prefix ignoreCase="true">A:\</prefix>
              <prefix ignoreCase="true">B:\</prefix>
              <prefix ignoreCase="true">C:\</prefix>
              <prefix ignoreCase="true">D:\</prefix>
              <prefix ignoreCase="true">E:\</prefix>
              <prefix ignoreCase="true">F:\</prefix>
              <prefix ignoreCase="true">G:\</prefix>
              <prefix ignoreCase="true">H:\</prefix>
              <prefix ignoreCase="true">I:\</prefix>
              <prefix ignoreCase="true">J:\</prefix>
              <prefix ignoreCase="true">K:\</prefix>
              <prefix ignoreCase="true">L:\</prefix>
              <prefix ignoreCase="true">M:\</prefix>
              <prefix ignoreCase="true">N:\</prefix>
              <prefix ignoreCase="true">O:\</prefix>
              <prefix ignoreCase="true">P:\</prefix>
              <prefix ignoreCase="true">Q:\</prefix>
              <prefix ignoreCase="true">R:\</prefix>
              <prefix ignoreCase="true">S:\</prefix>
              <prefix ignoreCase="true">T:\</prefix>
              <prefix ignoreCase="true">U:\</prefix>
              <prefix ignoreCase="true">V:\</prefix>
              <prefix ignoreCase="true">W:\</prefix>
              <prefix ignoreCase="true">X:\</prefix>
              <prefix ignoreCase="true">Y:\</prefix>
              <prefix ignoreCase="true">Z:\</prefix>
            </prefixes>
          </valid>
          <invalid>
            <substrings>
              <substring>\\</substring>
            </substrings>
            <characters>*?";/|~<![CDATA[<>]]></characters>
          </invalid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************* -->
<!-- Specify a variable to expose the IHS httpd.conf directory.       -->
<!-- ********************************************************* -->
<stringVariable name="defaultIHSConfigFileLocationBean.value"
minimumLength="4">
      <defaultData>C:\Program
Files\IBMHTTPServer\conf\httpd.conf</defaultData>
      <labelText translatedKey="IHSconfLabel" />
      <helpText translatedKey="IHSconfHelp" />
      <propertiesAssociations>
            <propertiesAssociation
keyword="defaultIHSConfigFileLocationBean.value" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard" propertyKey="defaultIHSConfigFileLocationBean.value"
/>
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the install directory.     -->
      <!-- ******************************************************* -->
      <inputValidation>
      <valid>
            <prefixes>
              <prefix ignoreCase="true">A:\</prefix>
              <prefix ignoreCase="true">B:\</prefix>
              <prefix ignoreCase="true">C:\</prefix>
              <prefix ignoreCase="true">D:\</prefix>
              <prefix ignoreCase="true">E:\</prefix>
              <prefix ignoreCase="true">F:\</prefix>
              <prefix ignoreCase="true">G:\</prefix>
              <prefix ignoreCase="true">H:\</prefix>
              <prefix ignoreCase="true">I:\</prefix>
              <prefix ignoreCase="true">J:\</prefix>
              <prefix ignoreCase="true">K:\</prefix>
              <prefix ignoreCase="true">L:\</prefix>
              <prefix ignoreCase="true">M:\</prefix>
              <prefix ignoreCase="true">N:\</prefix>
              <prefix ignoreCase="true">O:\</prefix>
              <prefix ignoreCase="true">P:\</prefix>
              <prefix ignoreCase="true">Q:\</prefix>
              <prefix ignoreCase="true">R:\</prefix>
              <prefix ignoreCase="true">S:\</prefix>
              <prefix ignoreCase="true">T:\</prefix>
              <prefix ignoreCase="true">U:\</prefix>
              <prefix ignoreCase="true">V:\</prefix>
              <prefix ignoreCase="true">W:\</prefix>
              <prefix ignoreCase="true">X:\</prefix>
              <prefix ignoreCase="true">Y:\</prefix>
              <prefix ignoreCase="true">Z:\</prefix>
            </prefixes>
          </valid>
          <invalid>
            <substrings>
              <substring>\\</substring>
            </substrings>
            <characters>*?";/|~<![CDATA[<>]]></characters>
          </invalid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************* -->
<!-- Specify a variable to expose the ihs port.               -->
<!-- ********************************************************* -->
<stringVariable name="coexistencePanelBean.ihsPort" minimumLength="1"
maximumLength="5">
      <defaultData>80</defaultData>
      <labelText translatedKey="ihsPortLabel" />
      <helpText translatedKey="ihsPortHelp" />
      <propertiesAssociations>
            <propertiesAssociation keyword="coexistencePanelBean.ihsPort" />
      </propertiesAssociations>
      <ismpFileAssociations>
            <ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="wizard" propertyKey="coexistencePanelBean.ihsPort" />
      </ismpFileAssociations>
      <!-- ******************************************************* -->
      <!-- Specify validation rules for the port.                  -->
      <!-- ******************************************************* -->
      <inputValidation>
            <valid>
                  <ranges>
                        <range>1 to 65535</range>
                  </ranges>
            </valid>
      </inputValidation>
</stringVariable>

<!-- ********************************************************* -->
<!-- Specify a variable to allow samples to install           -->
```

```
<!-- ********************************************************* -->
<booleanVariable name="samplesBean.active" minimumLength="1"
maximumLength="5">
	<defaultData>false</defaultData>
	<labelText translatedKey="samplesLabel" />
	<helpText translatedKey="samplesHelp" />
	<propertiesAssociations>
		<propertiesAssociation keyword="samplesBean.active" />
	</propertiesAssociations>
	<ismpFileAssociations>
		<ismpFileAssociation responseFileName="baseresponsefile.txt"
propertyKeyType="product" propertyKey="samplesBean.active" />
	</ismpFileAssociations>
</booleanVariable>
```

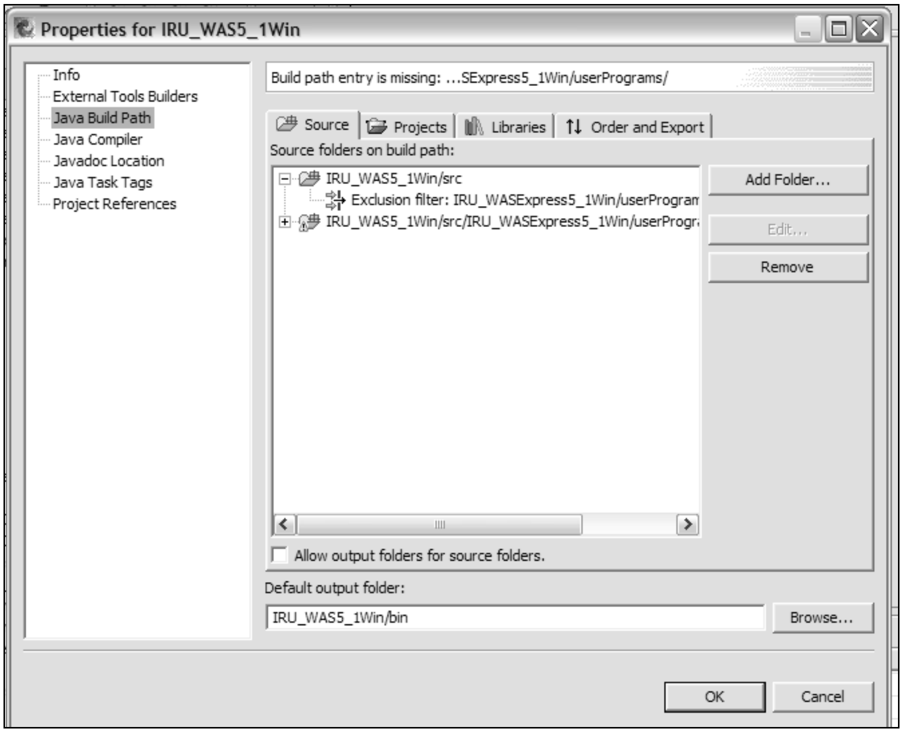
Figure 6 Variables for applicationdef.axml

g. Verify that the application id under the tag `iru:application` is successfully changed from IRU_WASExpress5_1Win to IRU_WAS5_1Win

```
<iru:application
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:iru="http://www.ibm.com/xmlns/prod/iru/application"
xsi:schemaLocation="http://www.ibm.com/xmlns/prod/iru/application
../../schemas/DJT_application.xsd"
debug="true"
id="IRU_WAS5_1Win"
responseFileTemplate="baseresponsefile.txt"
builderVersion="3.0">
```

h. Save the applicationdef.axml file.

3. Rename the packages
    a. Expand the src/IRU_WAS5_1Win/userPrograms folder, right-click the com.ibm.iru.iru_wasexpress5_1win package and select **Refactor > Rename**. Enter com.ibm.iru.iru_was5_1win for the new package name and click **OK**.
    b. Similarly, right-click each Java program in the folder and rename WASExpress5_1WinExit.java to WAS5_1WinMain.java and WASExpress5_1WinVC.java to WAS5_1WinVC.java.
    c. Click the **Navigator** tab and navigate to the src/IRU_WASExpress5_1Win directory. Right-click it and rename it to src/IRU_WAS5_1Win.
    d. Right-click the IRU_WASExpress5_1Win folder that contains the IRU_UpdateAdminConsolePorts5_1Win.bat and IRU_UpdateExpressPorts5_1Win.bat files. Delete the folder.
    e. Click the **Package Explorer** tab.
4. Modify the project build path
    a. Right-click the IRU_WAS5_1Win project, select **Properties > Java Build Path**. The properties editor opens. Click the **Source** tab. Click the folder with a caution sign next to it and click **Edit** to edit that path.



Properties for IRU_WAS5_1Win

- Info
- External Tools Builders
- Java Build Path
- Java Compiler
- Javadoc Location
- Java Task Tags
- Project References

Build path entry is missing: ...SExpress5_1Win/userPrograms/

Source folders on build path:

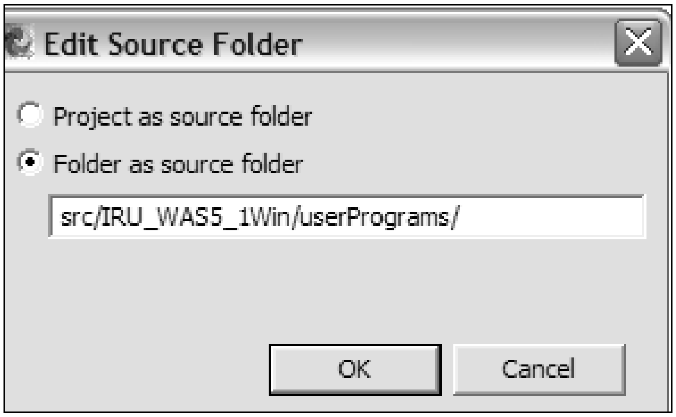
```
⊟ IRU_WAS5_1Win/src
    Exclusion filter: IRU_WASExpress5_1Win/userProgram
⊞ IRU_WAS5_1Win/src/IRU_WASExpress5_1Win/userProgr
```

□ Allow output folders for source folders.

Default output folder: IRU_WAS5_1Win/bin

Figure 7 Edit the properties for the project

b. Delete the word “Express” from the path, so that the source folder is now named src/IRU_WAS5_1Win/userPrograms. Click **OK**.



Edit Source Folder

○ Project as source folder

● Folder as source folder

src/IRU_WAS5_1Win/userPrograms/

Figure 8 Edit the source folder location

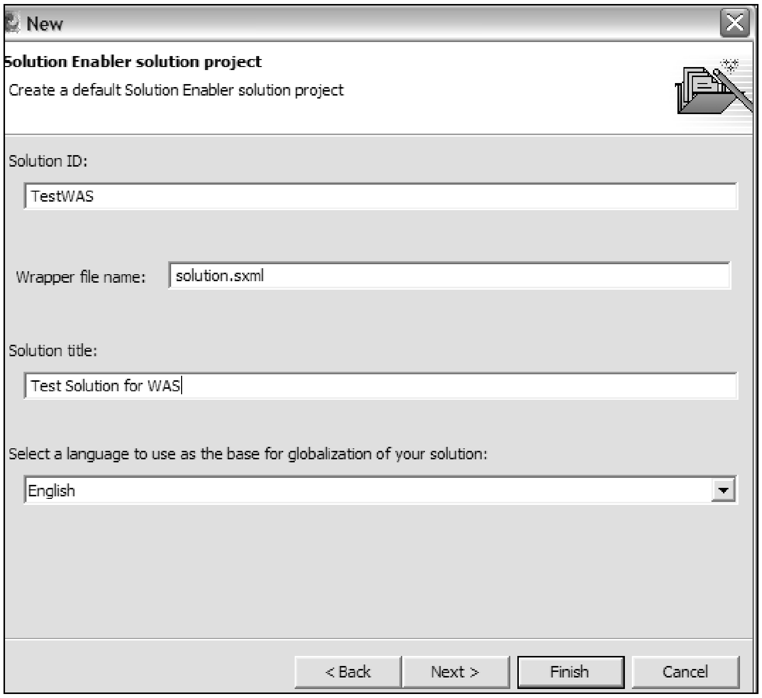
c. Highlight the exclusion filter for the IRU_WebSphere Application Server WSE8_1Win/src directory and click **Edit**. Click **Edit** again, browse to and select the userPrograms folder, and click **OK** on the next three dialog boxes, so that the name of the IRU_WASExpress5_1Win/userPrograms folder changes to IRU_WAS5_1Win/userPrograms. Click **OK**.

5. Modify the entry and main programs

   a. In the Package Explorer panel of the Solution Developer perspective, expand the src/IRU_AS5_1Win/userPrograms folder and then expand the com.ibm.iru.iru_was5_1win package.
   b. Replace the contents of WAS5_1WinVC.java with the code listed in Appendix B.
   c. Replace the contents of WAS5_1WinMain.java with the code listed in Appendix C.
   d. Add the programs WASMessagesNLS.java and WASNLSKeys.java with the listings in appendixes D and E, respectively.
6. Replace the content of applicationdef_english.xml in the src/IRU_WAS5_1Win directory with with the code listed in Appendix F.
7. Define the software image root
   a. In the Package Explorer panel of the Solution developer perspective, expand the src and IRU_WAS5_1Win folders.
   b. Right-click applicationdef.axml and click **Properties > Application Software Image Root**.
   c. Browse to and select the directory that contains the Install.exe file for WebSphere Application Server. Click **OK**.
8. Build the application
   a. Right-click the application project and select **Solution Enabler Tools > Build Application Definition**.
9. Verify that no error messages appear in the console during the build.

Create a test solution

1. Right-click in the Package Explorer panel of the Solution Developer perspective. Select **New > Default Solution Project Template**.
2. Name the project (for example, TestWAS) and click **Next**.
3. Enter the rest of the attributes as shown in the following image:



New

Solution Enabler solution project

Create a default Solution Enabler solution project

Solution ID: TestWAS

Wrapper file name: solution.sxml

Solution title: Test Solution for WAS

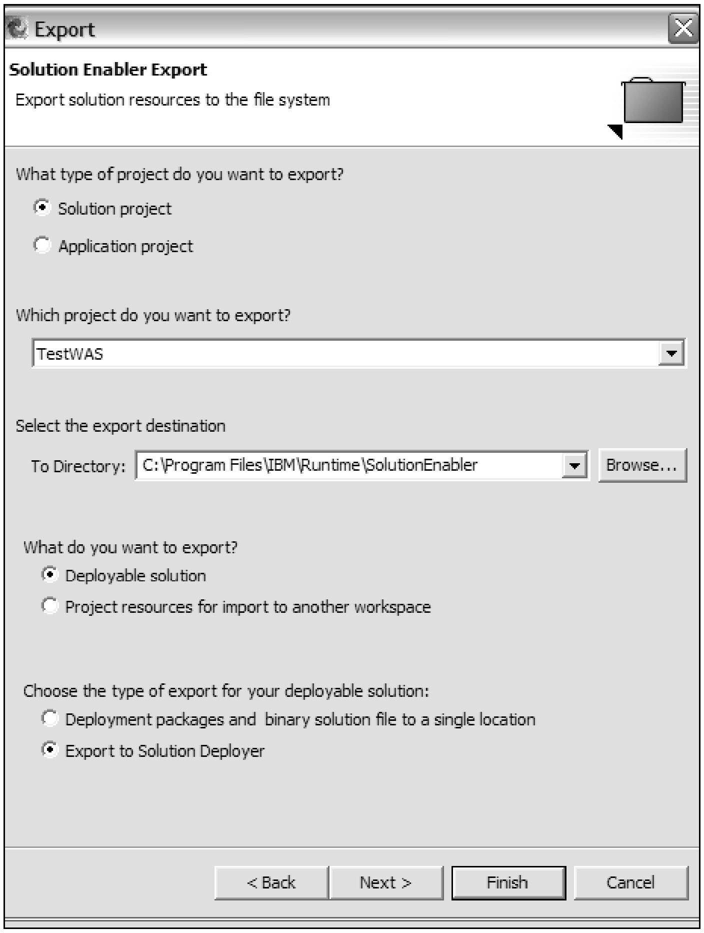
Select a language to use as the base for globalization of your solution: English

Figure 9 Creating a test solution

4. Click **Next**.
5. In the **Default export destination** field, click the **Browse** and select the C:\Program Files\IBM\Runtime\SolutionEnabler directory. Click **Next**.
6. Select the IRU_WAS5_1Win application to include in the solution and click **Finish**.
7. Solution.sxml opens in the editor. Find the `<description>` tag and replace "`Put your description here`" with "`Test Solution for Websphere Application Server.`"
8. Build the solution
   a. Right-click the solution project and select **Solution Enabler Tools > Build Solution**.
9. Verify that no error messages appear in the console during the build.

Deploy the solution

1. Export the solution
   a. Right-click the solution project and select **Export**.
   b. Select **Solution Developer** and click **Next**.
   c. Accept the default settings.



Export

Solution Enabler Export

Export solution resources to the file system

What type of project do you want to export?

● Solution project

○ Application project

Which project do you want to export?

TestWAS

Select the export destination

To Directory: C:\Program Files\IBM\Runtime\SolutionEnabler

What do you want to export?

● Deployable solution

○ Project resources for import to another workspace

Choose the type of export for your deployable solution:

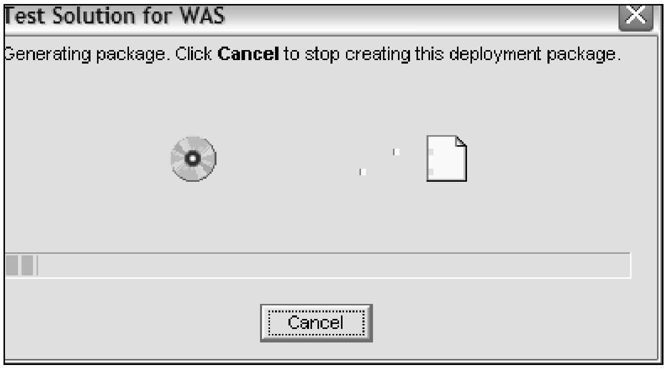
○ Deployment packages and binary solution file to a single location

● Export to Solution Deployer

Figure 10 Solution export dialog box

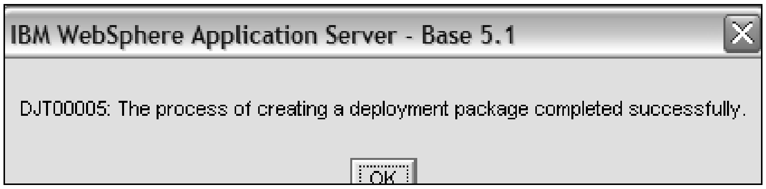
   d. Click **Finish**.
2. The TestWAS.ser file is located in the C:\Program Files\IBM\Runtime\SolutionEnabler directory and the user JAR file is placed in the directory where IBM Integrated Runtime is installed. You can now deploy the solution.
3. Select the Solution Deployer from the **Start** menu
4. Select **File > Open**.
5. Navigate to the TestWAS.ser file and open it.
6. Configure the deployment task
   a. Right-click **Task 1** and select **Configure Task 1**.

b. In the Task 1 configuration window, click **Target Hostnames**.
c. Enter `localhost` in the **New Target Hostname** field and click **Add**.
d. Click **WebSphere Application Server** under the deployment parameters and enter the DNS hostname.
e. Click **WebSphere Application Server** under the deployment packages and click **Create Package**.
f. Browse to the location of the WebSphere Application Server installation image or CD and click **OK**.
g. The entire directory tree is archived into a JAR file and placed in the installation location of IBM Integrated Runtime.



Generating package. Click **Cancel** to stop creating this deployment package.

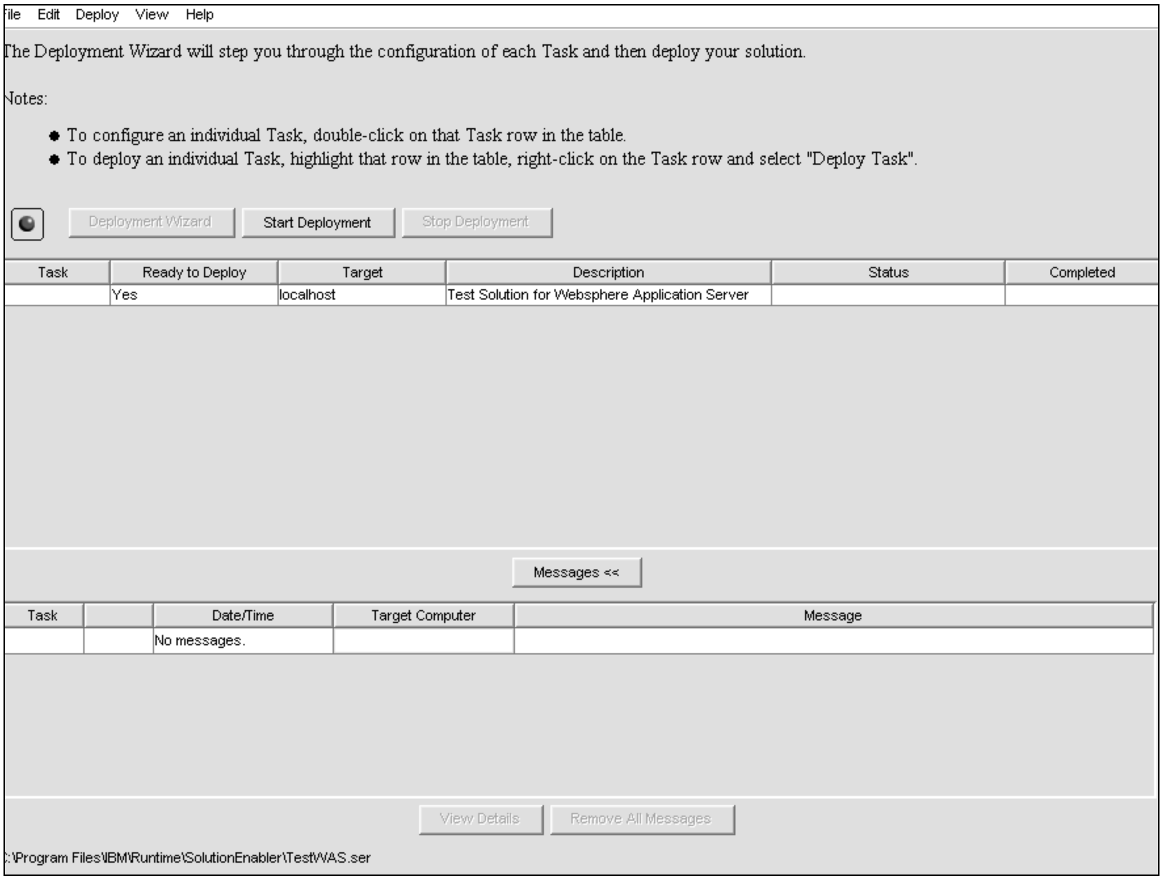
Figure 11 Deployment package creation in progress



DJT00005: The process of creating a deployment package completed successfully.

Figure 12 Deployment package completion

7. When the process is complete, click **OK** twice. The Start Deployment button becomes active and the ”Ready to deploy” entry for the task changes to `yes`. Click **Start Deployment**.



The Deployment Wizard will step you through the configuration of each Task and then deploy your solution.

Notes:

- To configure an individual Task, double-click on that Task row in the table.
- To deploy an individual Task, highlight that row in the table, right-click on the Task row and select "Deploy Task".

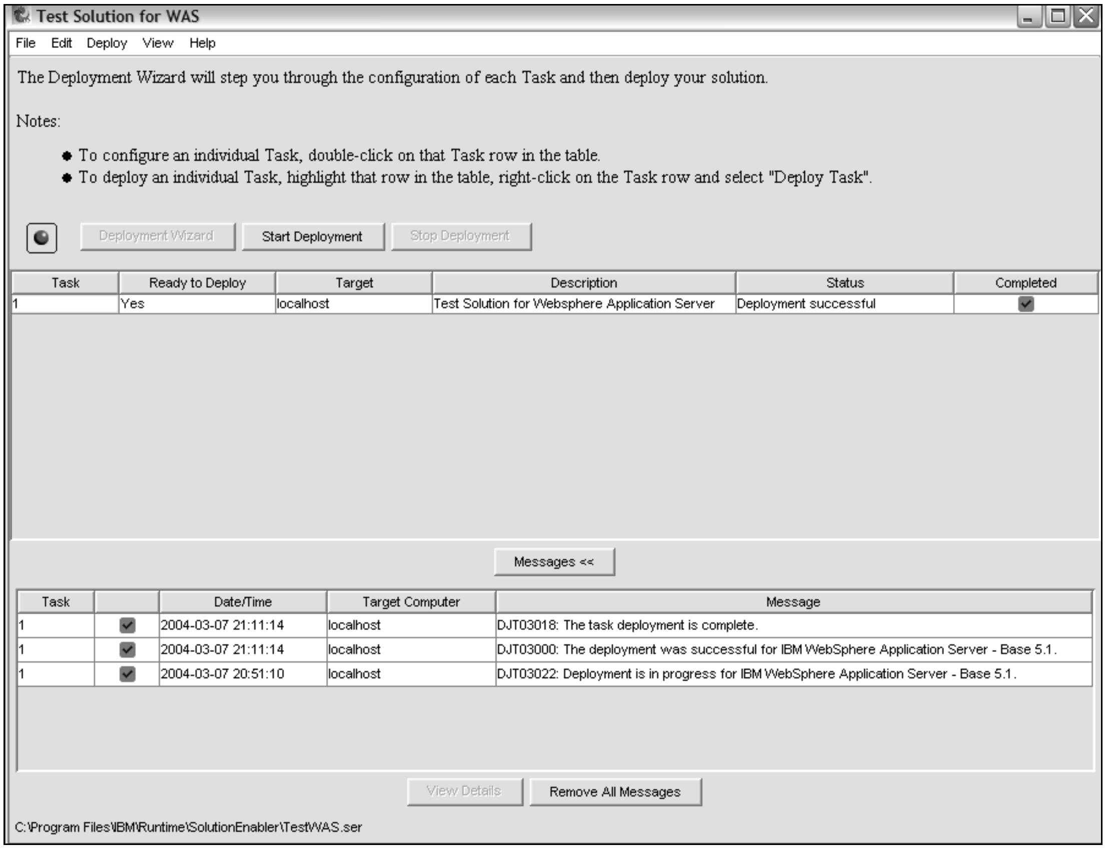
Task	Ready to Deploy	Target	Description	Status	Completed
	Yes	localhost	Test Solution for Websphere Application Server		

Task		Date/Time	Target Computer	Message
		No messages.		

C:\Program Files\IBM\Runtime\SolutionEnabler\TestWAS.ser

Figure 13 Solution Deployment Interface

8. Deployment is complete when the animation of the icon stops and the following message, indicating completion of deployment is displayed:
   `DJT03000: The deployment was successful for IBM WebSphere Application Server 5.1.`



The Deployment Wizard will step you through the configuration of each Task and then deploy your solution.

Notes:

- To configure an individual Task, double-click on that Task row in the table.
- To deploy an individual Task, highlight that row in the table, right-click on the Task row and select "Deploy Task".

Task	Ready to Deploy	Target	Description	Status	Completed
1	Yes	localhost	Test Solution for Websphere Application Server	Deployment successful	☑

Task		Date/Time	Target Computer	Message
1	☑	2004-03-07 21:11:14	localhost	DJT03018: The task deployment is complete.
1	☑	2004-03-07 21:11:14	localhost	DJT03000: The deployment was successful for IBM WebSphere Application Server - Base 5.1.
1	☑	2004-03-07 20:51:10	localhost	DJT03022: Deployment is in progress for IBM WebSphere Application Server - Base 5.1.

C:\Program Files\IBM\Runtime\SolutionEnabler\TestWAS.ser

Figure 14 Successful deployment

In addition, the Status field for the task changes to Deployment successful.

When deployment is complete, you have a complete WebSphere Application Server installation on the system.

Troubleshooting

You can perform the following activities to debug any problems:

- Test the version checker entry program by running it in the Solution Developer. You might have to configure the JRE to get it to work:
  1. Click **Window > Preferences** to open the preferences editor.
  2. Expand **Java > Installed JRE > Search**.
  3. Browse to the installation location of IBM Integrated Runtime.
  4. Select **DJTJRE** and click **OK**. Wait for the entire workspace to recompile.
- Copy the Solution Deployer shortcut from the **Start** menu, and place it on the desktop.
  - Modify the **Target** field of the shortcut to append the `-leavefiles` option at the end of the Deployer command. The log files and intermediate files are left on the machine
- The DJT_IIAtrace.log and DJT_SolutionEnabler.log, located in the C:\Program Files\IBM\Runtime\SolutionEnabler directory, are important files.
- If you used the `-leavefiles` option, additional logs are located in the C:\Program Files\IBM\Runtime\SolutionEnabler\logs directory. Notice the edited response file in that directory. It is created from the options entered during the Deployer task configuration.
- If you make any changes to the application wrapper or any of the XML files, remember to rebuild the solution and export it again.

Summary

You have seen how easy it is to package a new application into the delivery structure of the IBM Integrated Runtime. The IBM Integrated Runtime is a middleware solution that can pay for itself through productivity gains alone. It takes care of deployment issues and lets you concentrate on your application. Buy it now by visiting http://www-306.ibm.com/software/smb and selecting your geography.

Appendix A

To verify that the installation of WebSphere Application Server is successful, perform the following test:

1. Start the application server.
2. Attempt to access http://localhost:9090/admin. You should see the administrative console. Note that this port number can be changed during installation. You should use the same port you specified in the response file.
3. If you have installed the predefined samples, they appear by visiting http://localhost/WSSamples. Note that in this document, we use the default setting and do not install the samples.

Appendix B

```
package com.ibm.iru.iru_was5_1win;
/**
 * @(#)WASExpress5_1WinVC.java
 *
 * Licensed Materials - Property of IBM
 *
 * 5724-F71
 *
 * (C) Copyright IBM Corp. 2004
 *
 * US Government Users Restricted Rights - Use, duplication or disclosure
 * restricted by GSA ADP Schedule Contract with IBM Corp.
 *
 * This is sample code made available for use in accordance with terms set
 * forth in the license agreement document for the IBM Integrated Runtime.
 */

import com.ibm.jsdt.support.SupportWindowsBase;
import com.ibm.jsdt.support.SupportWindowsHelper;

public class WAS5_1WinVC extends SupportWindowsBase
{
  private static final String copyright0="Licensed Materials - Property of
IBM";
  private static final String copyright1="5724-F71";
  private static final String copyright2="(C) Copyright IBM Corp. 2004 All
Rights Reserved.";
  private static final String copyright3="US Government Users Restricted
Rights - Use, duplication or disclosure restricted by GSA ADP Schedule
Contract with IBM Corp.";

  private static final String WASEXPRESS_NAME = "WebSphere Application
Server";
  private static final String WASEXPRESS_VERSION = "5.1.0.0";

  private SupportWindowsHelper helper;


  /**
   * This function checks the Windows registry for WebSphere Express
   * Edition and writes the installed version to standard output
   * if one is found. The function returns 0 if WebSphere Express is found.
   * A value of 1 is returned otherwise.
   */

  private int checkVersion()
  {
    int rc = 0;
    setRegistrySubKey("SOFTWARE\\IBM\\" + WASEXPRESS_NAME + "\\" +
WASEXPRESS_VERSION);
    if (getWindowsHelper().doesRegKeyExist(this))
    {
      setRegistryStringValue("MajorVersion");
      if (getWindowsHelper().getRegistryValue(this).equals("5"))
      {
        rc=1;
        System.out.println("5.1");
      }
    }
    return rc;
  }

  /*********************************
   ******  Main Routine   ***********
   ********************************/
  public static void main(String args[])
  {
    WAS5_1WinVC checker = new WAS5_1WinVC ();
    System.exit(checker.checkVersion());
  }

}
```

Figure 15 WAS5_1WinVC Program

Appendix C

```
package com.ibm.iru.iru_was5_1win;
/**
 * @(#)WAS5_1WinMain.java
 *
 * Licensed Materials - Property of IBM
 *
 * 5724-F71
 *
 * (C) Copyright IBM Corp. 2004
 *
 * US Government Users Restricted Rights - Use, duplication or disclosure
 * restricted by GSA ADP Schedule Contract with IBM Corp.
 *
 * This is sample code made available for use in accordance with terms set
 * forth in the license agreement document for the IBM Integrated Runtime.
 */
import com.ibm.iru.common.*;
import com.ibm.jsdt.support.*;
import com.ibm.iru.message.NLSKeys;

public class WAS5_1WinMain extends SupportWindowsBase
{
  private static final String copyright0="Licensed Materials - Property of
IBM";
  private static final String copyright1="5724-F71";
  private static final String copyright2="(C) Copyright IBM Corp. 2004 All
Rights Reserved.";
  private static final String copyright3="US Government Users Restricted
Rights - Use, duplication or disclosure restricted by GSA ADP Schedule
Contract with IBM Corp.";

  private final String PortsDir = "bin" + CommonConstants.SLASH + "Ports";
  private final String defaultServer = "server1";
  private static final String RUNTIME_MESSAGES =
"com.ibm.iru.message.MessagesNLS";
  private static final String LOG_FILE = "IRU_WAS5_1WinMain.log";
  private static final String WAS_NAME = "WebSphere Application Server";
  private static final String WAS_VERSION = "5.1.0.0";
  private static final String MYDIRECTORY = "iru_was5_1win";

  private String serverName = null;
  private SupportWindowsHelper helper;
  private String wasInstallDir = null;

  private String unpackedDir = null;
  private String RmiPort = "";
  private String SoapPort = "";
  private String HttpPort = "";
  private String HttpsPort = "";
  private String AdminPort = "";
  private String AdminsPort = "";
  /**
   * Constructor
   */
  public WAS5_1WinMain()
  {
    setLogFileName(LOG_FILE);
    setMainResources(RUNTIME_MESSAGES);
  }
  /**
   * Get the arguments from the properties file and invoke the installSP bat
   * Use the default server name during installation to install the admin
console
   * Stop and restart the app server to activate the admin console app
   */
  private int install()
  {
    int rc = 0;
    setServerName(defaultServer);
    helper = (SupportWindowsHelper)getHelper();
    setUnpackedDir(helper.getUnpackedDir(this));

    String installDIR = determineWASInstallDir();

    // get the WAS install directory from the Windows registry
    // use the helper to create a short path name, then append AppServer
    setPath(installDIR);
    setWASInstallDir(helper.getWindowsShortPath(this) +
CommonConstants.SLASH + "AppServer");

    // get information from the properties file
    rc = getProperties(helper);

    if(rc != 0)
    {
      return rc;
    }

    if (installDIR == null)
    {
      rc = 1;
      return rc;
    }

      // actual installation
      rc = installWas();

      // verify installtion
      rc = checkInstallationStatus();
      if(rc == 1)
      {
            return rc;
      }

    return rc;
  }
  private int installWas()
      {
        int rc = 0;

            String Test = getResponseFileName();
            String Test2 = getUnpackedDir();

            String command = getUnpackedDir() +
                                     "win\\Install.exe -options " +
getResponseFileName();

            rc = invokeCommand(helper,command);

        //if (rc == 0)
        //{
            //rc = checkInstallationStatus();
        //}

        return rc;
      }


  /**
   * Get install root from the properties file
   */
  private int getProperties(SupportWindowsHelper helper)
  {
    int ret = 0;
    setJarFile(getHelper().getProductInstallingId(this));
    //    setVariableName("installDir");
    //    setPath(helper.getIbmNsiPropValue(this));  // set up to get short
name
    //    setWASInstallDir(helper.getWindowsShortPath(this));  // get short
name
    setMessage(getResourceString(NLSKeys.WASEXPRESS_INSTALL_DIR,
getWASInstallDir()));
    helper.log(this);

    // get the port values
             setVariableName("installDir");
             wasInstallDir = helper.getIbmNsiPropValue(this);
             setMessage(getResourceString(WASNLSKeys.WAS_INSTALL_DIR,
wasInstallDir));
             helper.log(this);

             setVariableName("coexistencePanelBean.bootstrapPort");
             RmiPort = helper.getIbmNsiPropValue(this);
             setMessage(getResourceString(WASNLSKeys.WAS_PORT_ASSIGNMENT,
"RMI," + RmiPort));
             helper.log(this);

             setVariableName("coexistencePanelBean.soapConnectorAddress");
             SoapPort = helper.getIbmNsiPropValue(this);
             setMessage(getResourceString(WASNLSKeys.WAS_PORT_ASSIGNMENT,
"SOAP," + SoapPort));
```

```
             helper.log(this);

             setVariableName("coexistencePanelBean.httpTransportPort");
             HttpPort = helper.getIbmNsiPropValue(this);
             setMessage(getResourceString(WASNLSKeys.WAS_PORT_ASSIGNMENT,
"HTTP," + HttpPort));
             helper.log(this);

             setVariableName("coexistencePanelBean.httpsTransportPort");
             HttpsPort = helper.getIbmNsiPropValue(this);
             setMessage(getResourceString(WASNLSKeys.WAS_PORT_ASSIGNMENT,
"HTTPS," + HttpsPort));
             helper.log(this);

             setVariableName("coexistencePanelBean.adminConsolePort");
             AdminPort = helper.getIbmNsiPropValue(this);
             setMessage(getResourceString(WASNLSKeys.WAS_PORT_ASSIGNMENT,
"ADMIN," + AdminPort));
             helper.log(this);

             setVariableName("coexistencePanelBean.secureAdminConsolePort");
             AdminsPort = helper.getIbmNsiPropValue(this);
             setMessage(getResourceString(WASNLSKeys.WAS_PORT_ASSIGNMENT,
"ADMINS," + AdminsPort));
             helper.log(this);



    // ensure there is a value for each property
    if ((getWASInstallDir() == "" || RmiPort=="" || SoapPort=="" ||
         HttpPort=="" || HttpsPort=="" || AdminPort=="" || AdminsPort=="")
        || (getWASInstallDir() == null || RmiPort==null || SoapPort==null ||
            HttpPort==null || HttpsPort==null || AdminPort==null ||
AdminsPort==null))
    {

      setMessage(getResourceString(NLSKeys.GET_PROPERTIES_FAILED));
      helper.log(this);
      ret = 1;
    }
    return ret;
  }

  /**
    * Invoke the command, msgText is the text to be logged
    */
  private int invokeCommand(SupportWindowsHelper helper, String command)
  {
    int rc = 0;
    setMessage(getResourceString(NLSKeys.CMDINVOKED,command));
    helper.log(this);

    try
    {
      Process p = Runtime.getRuntime().exec(command);
      rc = p.waitFor();
    }
    catch (Exception e)
    {
      rc = 1;
      setMessage(getResourceString(NLSKeys.CMD_EXCEPTION, e.toString()));
      helper.log(this);
    }

    if (rc == 0)
    {
      setMessage(getResourceString(NLSKeys.CMD_SUCCESS));
      helper.log(this);
    }
    else
    {
      setMessage(getResourceString(NLSKeys.CMD_FAIL,String.valueOf(rc)));
      helper.log(this);
    }

    return rc;
  }

  /**
   * Check / Varify Installation
   * @return
   */

  private int checkInstallationStatus()
  {
      int rc = 0;
      setRegistrySubKey("SOFTWARE\\IBM\\" + WAS_NAME + "\\" + WAS_VERSION);
      if (getWindowsHelper().doesRegKeyExist(this))
      {
        setRegistryStringValue("MajorVersion");
        if (getWindowsHelper().getRegistryValue(this).equals("5"))
        {
            rc=0;
        }
      }
      return rc;
  }
  /**
   * Determine the WAS install directory name.
   */
  private String determineWASInstallDir()
  {
    String installDir="";

    setRegistrySubKey("SOFTWARE\\IBM\\" + WAS_NAME + "\\" + WAS_VERSION);
    if (getWindowsHelper().doesRegKeyExist(this))
    {
      setRegistryStringValue("InstallLocation");
      installDir = getWindowsHelper().getRegistryValue(this);
    }

    return installDir;
  }
  /**
****************************************************************************
    *
    * Standard getters and setters
    *
****************************************************************************
    */
  /**
    * Get the name of the server
    */
  private String getServerName()
  {
    return serverName;
  }
  /**
   * Set the name of the server
   */
  private void setServerName(String name)
  {
    serverName = name;
  }
  /**
    * Get the unpacked directory name
    */
  private String getUnpackedDir()
  {
    return unpackedDir;
  }
  /**
   * Set the WAS unpacked directory name.
   */
  private void setUnpackedDir(String myDir)
  {
    unpackedDir = myDir;
  }
  /**
    * Get the WAS install directory name.
    */
  private String getWASInstallDir()
  {
    return wasInstallDir;
  }
  /**
   * Set the WAS install directory name.
   */
  private void setWASInstallDir(String myDir)
  {
    wasInstallDir = myDir;
  }
  /**
   * Get the WAS bin directory name.
   */
  private String getWASBinDir()
  {
    return getWASInstallDir() + CommonConstants.SLASH + "bin";
  }
  /*********************************
   ******  Main Routine   ***********
   ********************************/
  public static void main(String args[])
  {
    WAS5_1WinMain wasMain = new WAS5_1WinMain();
    if (args.length == 1){
      wasMain.setResponseFileName(args[0]);
    }
    int rc  = wasMain.install();
    System.exit(rc);
  }
}
      }
}
```

Figure 16 WAS5_1WinMain Program

Appendix D

```
package com.ibm.iru.iru_was5_1win;
/**
* Licensed Materials - Property of IBM
*
* 5724-F71
*
* (C) Copyright IBM Corp. 2004
*
* US Government Users Restricted Rights - Use, duplication or disclosure
* restricted by GSA ADP Schedule Contract with IBM Corp.
*
* This is sample code made available for use in accordance with terms set
* forth in the license agreement document for the IBM Integrated Runtime.
*/
import java.util.ListResourceBundle;
/**
 * WASMessagesNLS contains the US/English literal objects for WAS
 */
public class WASMessagesNLS extends ListResourceBundle {
      private static final String copyright0 = "Licensed Materials - Property
of IBM";
      private static final String copyright1 = "5724-F71";
      private static final String copyright2 = "(C) Copyright IBM Corp. 2004
All Rights Reserved.";
      private static final String copyright3 = "US Government Users Restricted
Rights - Use, duplication or disclosure restricted by GSA ADP Schedule
Contract with IBM Corp.";
      /**
       * Localize the second argument in each pair of brackets.
       * @since JDK1.3
       */
      static final Object[][] messages = {

            // ---------------------------------------------------------------
-
            // MESSAGES for WAS - range 2000-2099
            // ---------------------------------------------------------------
-
            { WASNLSKeys.INSTALLING_WAS, "IRU02000: Installation of WebSphere
is in progress." }, {
                  WASNLSKeys.INSTALL_WAS_SUCCESS, "IRU02001: Installation of
WebSphere  was successful." }, {
                  WASNLSKeys.INSTALL_WAS_FAIL, "IRU02002: Installation of
WebSphere  failed." }, {
                  WASNLSKeys.CONFIGURE_WAS_PORTS, "IRU02003: Configuring
WebSphere ports." },
            // {0} represents the port type (RMI, SOAP, etc.) and {1}
represents the port value
            {
                  WASNLSKeys.WAS_PORT_ASSIGNMENT, "IRU02004: Port assignment
{0} = {1}." }, {
```

```
                  WASNLSKeys.CONFIGURE_WAS_ADMIN_PORTS, "IRU02005: Configuring
WebSphere Administration ports." }, {
                  WASNLSKeys.INSTALLING_WAS_ADMIN_CONSOLE, "IRU02006:
Installation of WebSphere Administration console is in progress." }, {
                  WASNLSKeys.INSTALL_WAS_ADMIN_SUCCESS, "IRU02007:
Installation of WebSphere Administration console was successful." }, {
                  WASNLSKeys.INSTALL_WAS_ADMIN_FAIL, "IRU02008: Installation
of WebSphere Administration console failed." }, {
                  WASNLSKeys.START_WAS, "IRU02009: Starting WebSphere
Application Server." }, {
                  WASNLSKeys.STOP_WAS, "IRU02010: Stopping WebSphere
Application Server." },
            // {0} represents the directory where WebSphere  was installed
            {
                  WASNLSKeys.WAS_INSTALL_DIR, "IRU02011: WebSphere  install
directory: {0}." }, {
                  WASNLSKeys.CONFIGURE_WAS_PORTS_FAIL, "IRU02012: Configuring
WebSphere ports failed." }, {
                  WASNLSKeys.CONFIGURE_WAS_ADMIN_PORTS_FAIL, "IRU02013:
Configuring WebSphere Administration ports failed." }, {
                  WASNLSKeys.START_WAS_SUCCESS, "IRU02014: Started WebSphere
Application Server." }, {
                  WASNLSKeys.STOP_WAS_SUCCESS, "IRU02015: Stopped WebSphere
Application Server." }, {
                  WASNLSKeys.START_WAS_FAIL, "IRU02016: Failed to start
WebSphere  Application Server." }, {
                  WASNLSKeys.STOP_WAS_FAIL, "IRU02017: Failed to stop
WebSphere  Application Server." },
            // {0} represents the name of the application server (eg:
"MyAppServer")
            {
                  WASNLSKeys.CREATE_WAS_SUCCESS, "IRU02018: Creation of
WebSphere  Application Server {0} was successful." },
            // {0} represents the name of the application server (eg:
"MyAppServer")
            {
                  WASNLSKeys.CREATE_WAS_FAIL, "IRU02019: Creation of WebSphere
Application Server {0} failed." },
            // {0} represents the name of the application server (eg:
"MyAppServer")
            {
                  WASNLSKeys.CREATE_WAS_DUPLICATE, "IRU02020: There is already
a WebSphere  Application Server named {0} on the target system." }, {
                  WASNLSKeys.SNOOP_WAS_SUCCESS, "IRU02021: Deployment of
sample application snoop was successful." }, {
                  WASNLSKeys.SNOOP_WAS_FAIL, "IRU02022: Deployment of sample
application snoop failed." }, {
                  WASNLSKeys.CREATE_WAS_BAD_PORT, "IRU02023: The range of
WebSphere ports {0} overlaps with the HTTP server port {1}." },

            // End Messages translations.
      };

      /**
       */
      public Object[][] getContents() {
            return getMessages();
      }

      /**
       * Convenience static method to get the messages array.  It is
       * public because JUnitMessageAbstraction needs access to this
       * method.
       */
      public static Object[][] getMessages() {
            return messages;
      }
}
```

Figure 17 WASMessagesNLS.java Program

Appendix E

```
package com.ibm.iru.iru_was5_1win;
/**
 * @(#)NLSKeys.java
 *
 * Licensed Materials - Property of IBM
 *
 * 5724-F71
 *
 * (C) Copyright IBM Corp. 2004
 *
 * US Government Users Restricted Rights - Use, duplication or disclosure
 * restricted by GSA ADP Schedule Contract with IBM Corp.
 *
 * This is sample code made available for use in accordance with terms set
 * forth in the license agreement document for the IBM Integrated Runtime.
 */

/**
 * NLSKeys contains the statics references to the resources
* <P>
 * Static keys for the Resource Bundles
 */

public class WASNLSKeys {
      private static final String copyright0 = "Licensed Materials - Property
of IBM";
      private static final String copyright1 = "5724-F71";
      private static final String copyright2 = "(C) Copyright IBM Corp. 2004
All Rights Reserved.";
      private static final String copyright3 = "US Government Users Restricted
Rights - Use, duplication or disclosure restricted by GSA ADP Schedule
Contract with IBM Corp.";

      public static final String INSTALLING_WAS = "02000";
      public static final String INSTALL_WAS_SUCCESS = "02001";
      public static final String INSTALL_WAS_FAIL = "02002";
      public static final String CONFIGURE_WAS_PORTS = "02003";
      public static final String WAS_PORT_ASSIGNMENT = "02004";
      public static final String CONFIGURE_WAS_ADMIN_PORTS = "02005";
      public static final String INSTALLING_WAS_ADMIN_CONSOLE = "02006";
      public static final String INSTALL_WAS_ADMIN_SUCCESS = "02007";
      public static final String INSTALL_WAS_ADMIN_FAIL = "02008";
      public static final String START_WAS = "02009";
      public static final String STOP_WAS = "02010";
      public static final String WAS_INSTALL_DIR = "02011";
      public static final String CONFIGURE_WAS_PORTS_FAIL = "02012";
      public static final String CONFIGURE_WAS_ADMIN_PORTS_FAIL = "02013";
      public static final String START_WAS_SUCCESS = "02014";
      public static final String STOP_WAS_SUCCESS = "02015";
      public static final String START_WAS_FAIL = "02016";
      public static final String STOP_WAS_FAIL = "02017";
      public static final String CREATE_WAS_SUCCESS = "02018";
      public static final String CREATE_WAS_FAIL = "02019";
      public static final String CREATE_WAS_DUPLICATE = "02020";
      public static final String SNOOP_WAS_SUCCESS = "02021";
      public static final String SNOOP_WAS_FAIL = "02022";
      public static final String CREATE_WAS_BAD_PORT = "02023";
      public static final String MAX_CHARS_IN_DEST_DIR =
"MaxCharsInDestination";

}
```

Figure 18 WASNLSKeys.java Program

Appendix F

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Licensed Materials - Property of IBM                             -->
<!--                                                                  -->
<!-- 5724-F71                                                         -->
<!--                                                                  -->
<!-- (C) Copyright IBM Corporation 2004 All Rights Reserved           -->
<!--                                                                  -->
<!-- US Government Users Restricted Rights- Use, duplication or       -->
<!-- disclosure restricted by GSA ADP Schedule Contract with IBM      -->
<!-- Corp.                                                            -->
<!--                                                                  -->
<!-- This is sample code made available for use in accordance with    -->
<!-- terms set forth in the license agreement document for the IBM    -->
<!-- Integrated Runtime.                                              -->
<!-- ================================================================ -->

<!-- A translation XML document must be encoded in a codepage that is -->
<!-- supported by your XML parser. This codepage must be specified    -->
<!-- in an encoding declaration within the translation XML document.  -->
<!-- **************************************************************** -->

<IRU_WAS5_1Win>
  <configureText>Provide the following information and click Next or OK to
proceed. Click Field Help for additional information about the
fields</configureText>
  <providerName>IBM</providerName>
  <name>WebSphere Application Server - Base</name>
  <prompt>Specify the fully qualified path name where the WebSphere install
image is located (for example, /downloads/was).</prompt>

  <nodeLabel>WebSphere Node Name</nodeLabel>
  <nodeHelp>Please enter a node name and hostname for this installation. The
node name is used for administration, and must be unique within its group of
nodes (cell). You must replace the "DefaultNode" with the node name that you
want the default node to be.</nodeHelp>

  <hostLabel>DNS Hostname</hostLabel>
  <hostHelp>The hostname is the DNS name or IP address for this computer. You
must set it to the fully qualified, resolveable hostname of the target
machine.</hostHelp>

  <dirLabel>WebSphere destination directory</dirLabel>
  <dirHelp>The directory where you want to install WebSphere Application
Server.</dirHelp>

  <ihsLabel>Install IBM HTTP server ?</ihsLabel>
  <ihsHelp>Do you wish to install the IBM HTTP server(IHS). You would chose
yes unless you already have the IHS installed.</ihsHelp>

  <IHSdirLabel>HTTP server destination directory</IHSdirLabel>
  <IHSdirHelp>The directory where you want to install the IBM HTTP
```

```
server.</IHSdirHelp>

  <IHSconfLabel>IHS Config File Location</IHSconfLabel>
  <IHSconfHelp>Full path to the HTTP server configuration file
httpd.conf.</IHSconfHelp>

  <rmiPortLabel>RMI Connector/Bootstrap Port</rmiPortLabel>
  <rmiPortHelp>Please see the WebSphere  documentation for more
detail</rmiPortHelp>

  <soapPortLabel>SOAP Connector Port</soapPortLabel>
  <soapPortHelp>Please see the WebSphere  documentation for more
detail</soapPortHelp>

  <httpPortLabel>HTTP Transport Port</httpPortLabel>
  <httpPortHelp>WAS webcontainer services HTTP requests at this port. Please
see the WebSphere  documentation for more detail</httpPortHelp>

  <httpsPortLabel>Secure HTTP Transport Port</httpsPortLabel>
  <httpsPortHelp>WAS webcontainer services HTTPS requests at this portPlease
see the WebSphere  documentation for more detail</httpsPortHelp>

  <ihsPortLabel>IHS HTTP Port</ihsPortLabel>
  <ihsPortHelp>IBM HTTP server listens at this port. Strongly recommend
default. Please see the WebSphere documentation for more detail</ihsPortHelp>

  <adminPortLabel>Admin Console Port</adminPortLabel>
  <adminPortHelp>Please see the WebSphere documentation for more
detail</adminPortHelp>

  <adminsPortLabel>Admin Console secure Port</adminsPortLabel>
  <adminsPortHelp>Please see the WebSphere documentation for more
detail</adminsPortHelp>

  <samplesLabel>Install WAS samples</samplesLabel>
  <samplesHelp>These are sample applications that install into the WebSphere
Application Server</samplesHelp>
</IRU_WAS5_1Win>
```

Figure 19 applicationdef_english.xml

References

[1] Response file installation of WebSphere Application Server overview, Customizing the base options response file in the InfoCenter at http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp

About the authors

Har Puri is a developer on the IBM Integrated Runtime Development team in Rochester, Minnesota. He has seven years of Java development experience mainly with the WebSphere family of products, including significant time developing resource adapter code for the WebSphere Application Server. Additionally, Har developed sample applications for several Java projects, including IBM SanFrancisco (an enterprise-level Java framework). A technical paper that Har coauthored was presented at eBU 2004, a convention of IBM Software Group Sales Staff and Business Partners at Las Vegas (access is available to IBM's Business Partners through the **Partnerworld for Software System**). The paper may be viewed or downloaded at the IBM developerWorks Web site (http://www.ibm.com/developerworks/websphere/library/techarticles/0401_heins/heins.html). Har is certified on Red Hat Linux (609004545108107) and can be contacted at hpuri@us.ibm.com.

Sahdev Zala is a developer working on Runtime Integration at IBM Rochester, Minnesota.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PAPER "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Information in this paper as to the availability of products (including portlets) was believed accurate as of the time of publication. IBM cannot guarantee that identified products (including portlets) will continue to be made available by their suppliers.

This information could include technical inaccuracies or typographical errors. Changes may be made periodically to the information herein; these changes may be incorporated in subsequent versions of the paper. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this paper at any time without notice.

Any references in this document to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
4205 South Miami Boulevard
Research Triangle Park, NC 27709 U.S.A.

e business software