IBM® Express Runtime Wrapper for IBM Informix® Dynamic Server Workgroup Edition 10.00 on Windows™

Document version 1.0

Sundar Shunmugam
IBM Informix Product Development
Lenexa KS 66213

The most current version of this white paper and the *IBM Express Runtime Wrapper for IBM Informix Dynamic Server Workgroup Edition 10.00 on Linux* white paper is available at http://www.expressenablement.com.

Contents

Figures

Introduction

One of the fastest-growing areas of IT spending today is in midsize businesses, generally defined as companies with 100 to 1000 employees. Companies in this segment often address their IT needs by obtaining business applications from an Independent Software Vendor (ISV). ISVs serving the mid-market realize that being an on-demand business requires an on-demand operating environment—a cost-effective infrastructure that enables customers to tackle their business priorities and adapt to changing conditions in real time.

Additionally, as ISVs enhance their IT infrastructure, they want to make the most of their existing investments. After all, they have spent hard-earned money building their IT infrastructures. One powerful combination that can help extend their IT investments while realizing their on-demand objectives is IBM Informix® Dynamic Server (IDS) and IBM Express Runtime. IBM Express Runtime is targeted at ISVs who need to quickly deliver a total solution to their mid-market customers. The IBM Express Runtime product contains IBM DB2® UDB Express Edition. This IDS and Express Runtime solution provides a comprehensive, integrated e-business infrastructure that works together seamlessly for small and medium businesses.

This document addresses ISVs, Solution Developers (SD), and Solution Integrators (SI) who would like the capability to create the IBM Express Runtime deployment framework to deliver IBM Informix Dynamic Server Workgroup Edition (IDS WE) 10.00 or IBM Informix Dynamic Server Entry Level Edition 10.00 (IDS EL).

In IBM Express Runtime terminology, an *application* is one of the components of a solution. Examples of applications include DB2, IDS, or the actual application program package the developer wrote. Applications are contained in *application wrappers*, and one or more application wrappers are combined to create a *solution* within the *solution wrapper*.

The most current version of this whitepaper and the *IBM Express Runtime Wrapper for IBM Informix Dynamic Server Workgroup Edition 10.00 on Linux* white paper are available at http://www.expressenablement.com. Follow these instructions to access the white papers:

1. Log in to the Virtual Innovation Center at http://www.expressenablement.com.
2. Click the **Delivering Solutions > IBM Integrated Runtime** tab.
3. Click the **IBM Integrated Runtime** twistie.
4. Click **IBM Express Runtime Education** from the IBM Express Runtime dropdown menu.
5. Click **Technical Resources** from the My Enablement Resources section.

6. Click on the appropriate white paper title to open the PDF document you want to view.

Prerequisites

This white paper assumes that you have entitlement to use the IBM Express Runtime 2.1 and purchased the following Informix Dynamic Server products for Windows platform:

- IDS Workgroup 10.00 (PID 5724L23, p/n C81V2NA / C81X6NA)

Please see "IBM Informix Dynamic Server Installation Guide for Microsoft Windows", Version 10.00 (G251-2288-00) for more details regarding installation requirements or visit the IBM Informix Dynamic Server information center Web site http://publib.boulder.ibm.com/infocenter/ids9help/index.jsp, navigate to **Getting Started > Installation Guide for Microsoft Windows.**

Methodology

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

- Generating 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, as well as 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, as well as 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 through these exercises.
    - Configuration Exercises – The components are configured in these exercises.

- Control Exercises – The components are controlled in some way through these exercises.
- Optional Exercises – Some additional features or functions of the components are used during 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 shown in the following font and format:

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

IDS 10.00 CD

The source for creating the IDS 10.00 application component is on a CD that can be purchased from IBM through the PartnerWorld O.E.M. sales channel. This paper assumes that you have entitlement to use Express Runtime and license to allow the bundling of IDS 10.00 with your solution. Please contact your sales channel representative for licensing issues.

CD Contents

The IDS 10.00 CD has contents and directory structure similar to the following illustration:

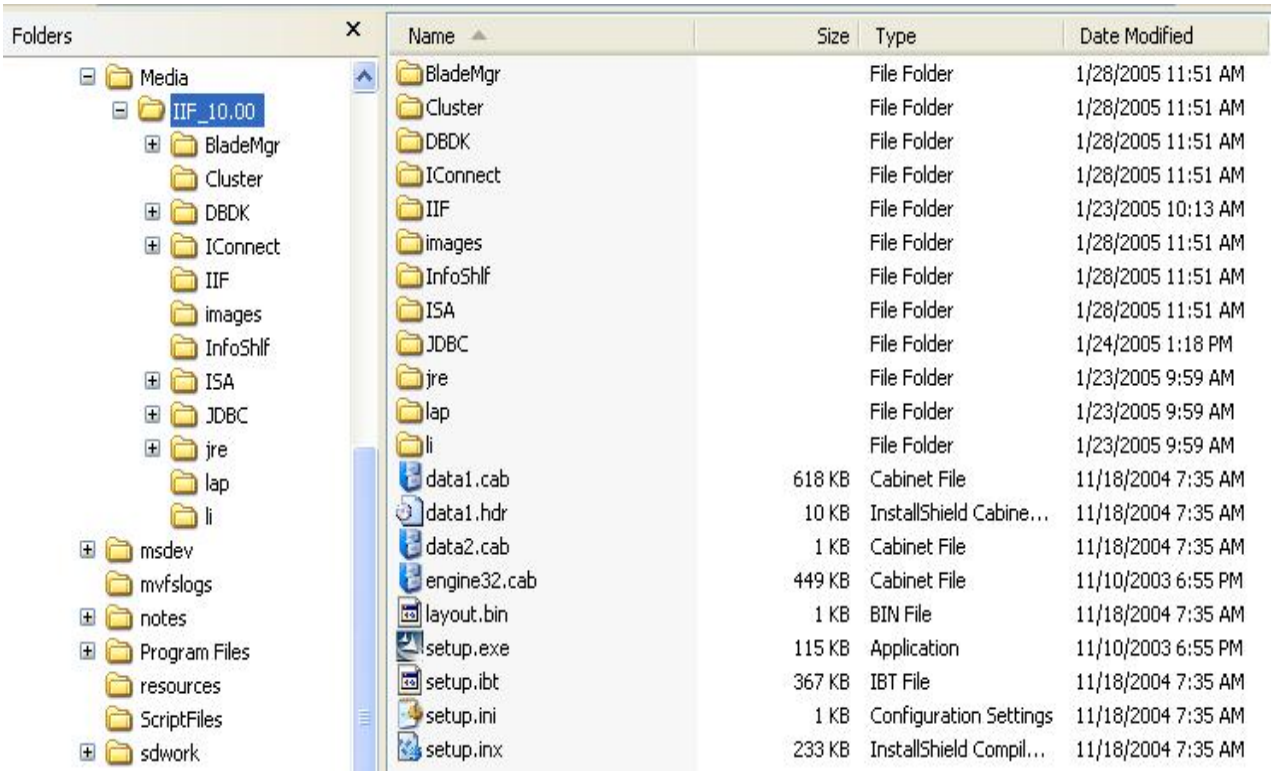

Figure 1. Directory structure of the installation image or CD

Response file

The IDS response file installation lets you install IDS without interactive user input. Response files contain installation choices and configuration data that the user would otherwise provide interactively on the installation graphical user interface (GUI). A sample response file is available on the IDS 10.00 installation CD in the `IIF\silent.in` directory.

```
[CIT]
        VERSION=1.00
        INTERNAL_D=
        [BUNDLE_COMMON]
        PRODUCTS=IDS
        INSTALL_DIR=C:\INFORMIX
        SUCCESS_FLAG=TRUE
[PRODUCT_IDS]
        NAME= IDS
        DESCRIPTION_RES_ID= 5001
        OS= WINNT
        NTFS_ONLY= Y
        IS_SELECTED=Y
        ACTUAL_INSTALL_DIR=C:\INFORMIX
        REBOOT_REQUESTED=Y
[server install]
        Shutdown_Services=1
        Upgrade=0
        Reinstall=0
        Configure_Instance=1
        Domain_Install=
        Cluster_Install=
        Typical=1
        Minimal=0
        Role_Separation= 0
        IXDBSA_Group=
        IXDBSSO_Group=
        IXAAO_Group=
        IXUSERS_Group=
        DBSA_User=
        DBSA_Passwd=
        SSO_User=
        SSO_Passwd=
        Account_Passwd=informix
        Create_Icons=1
[server instance]
        DBSERVERNAME=ol_server
        SERVERNUM=0
        DBSERVERALIASES=
        SQLHOSTS=
```

Figure 2. Partial contents of response file

Creating the IDS application component

In IBM Express Runtime terminology, an *application* is one of the components of a solution. Examples of applications include IDS or the actual application program package the developer wrote. Applications are contained in *application wrappers*, and one or more application wrappers are combined into a complete *solution* within a *solution wrapper*.

Create an application wrapper for IDS 10.00 and then use it in a test solution.

Create the IDS application project

1. Start the Express Runtime developer by clicking **Start > IBM Express Runtime 2.1 > Express Runtime Developer.**

Figure 3. Launching the Express Runtime developer

2. Verify your perspective is set to Express Runtime Developer. To set this perspective select **Window > Open Perspective > Other**. Select **Express Runtime Developer** and click **OK**.
3. Right-click anywhere in the Package Explorer and select **New > Application Project**.

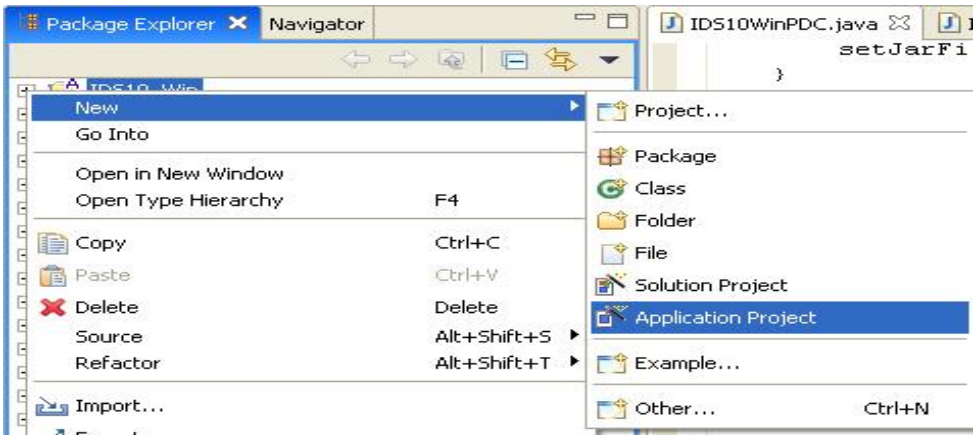

Figure 4. Create an application project

4. Enter the project name IDS10_Win. For the remainder of this document, IDS10_Win is the name of the application. Click **Next** to proceed.

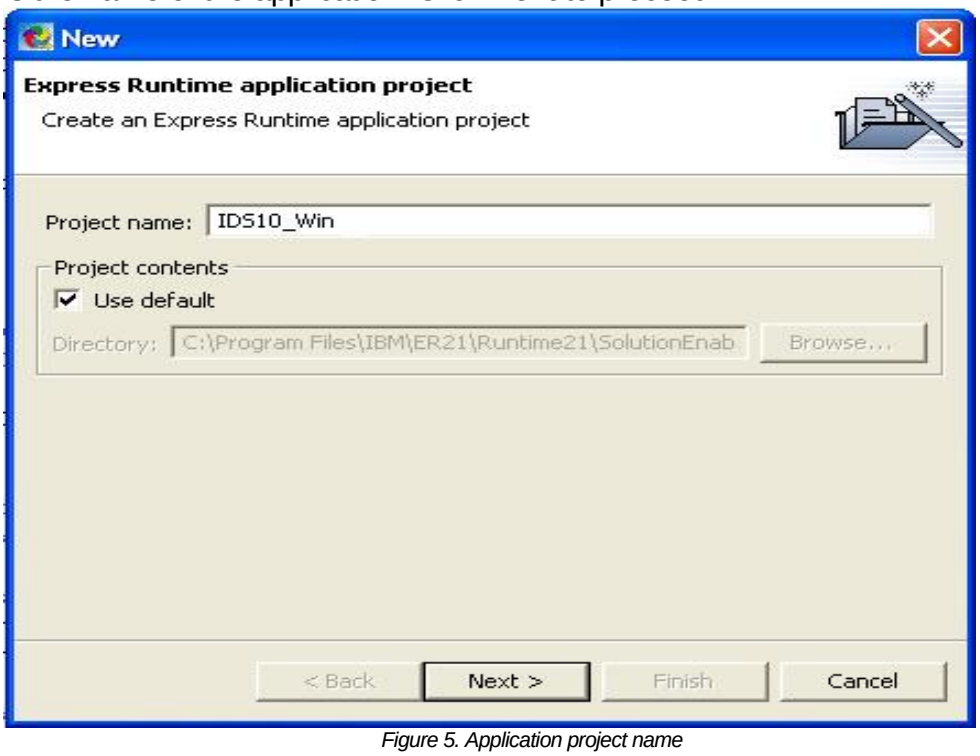

Figure 5. Application project name

5. In the subsequent application project panel, the Application ID field already contains the project name you entered in the previous panel, such as IDS10_Win. Type a version number (for the purpose of reference, the server version number is used) and estimated installation time for the application. Most of the files generated will use the

application ID as part of the name. Click **Finish**.

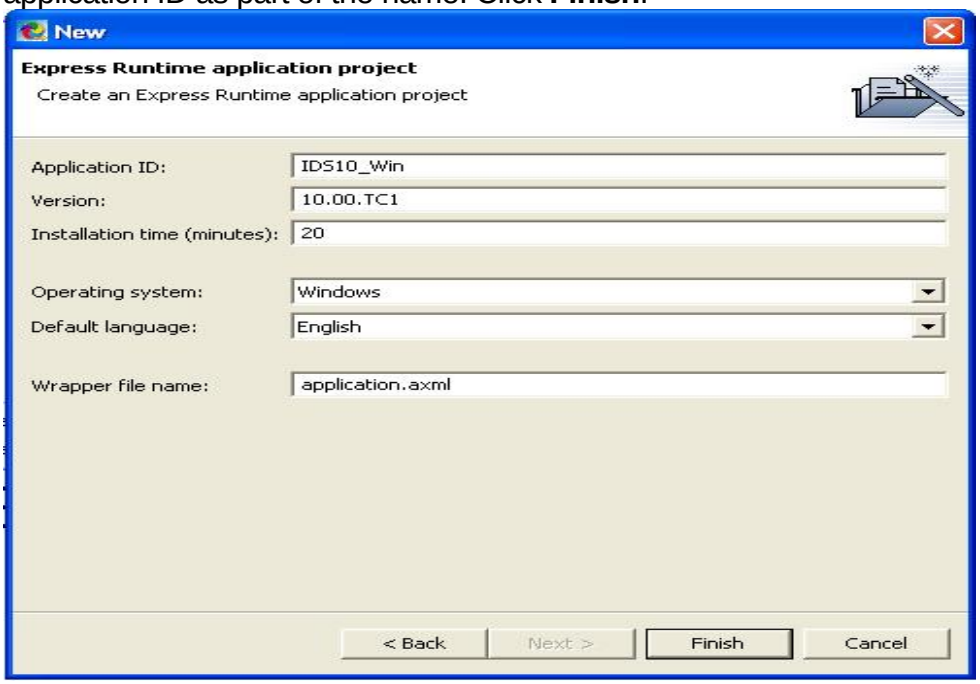

Figure 6. Application specific details

The application.axml opens automatically. The project starts with a welcome screen along with series of tabs (Welcome, General, Programs, Variables, Files, Libraries and Source) as shown, which help in setup for the application and its deployment.

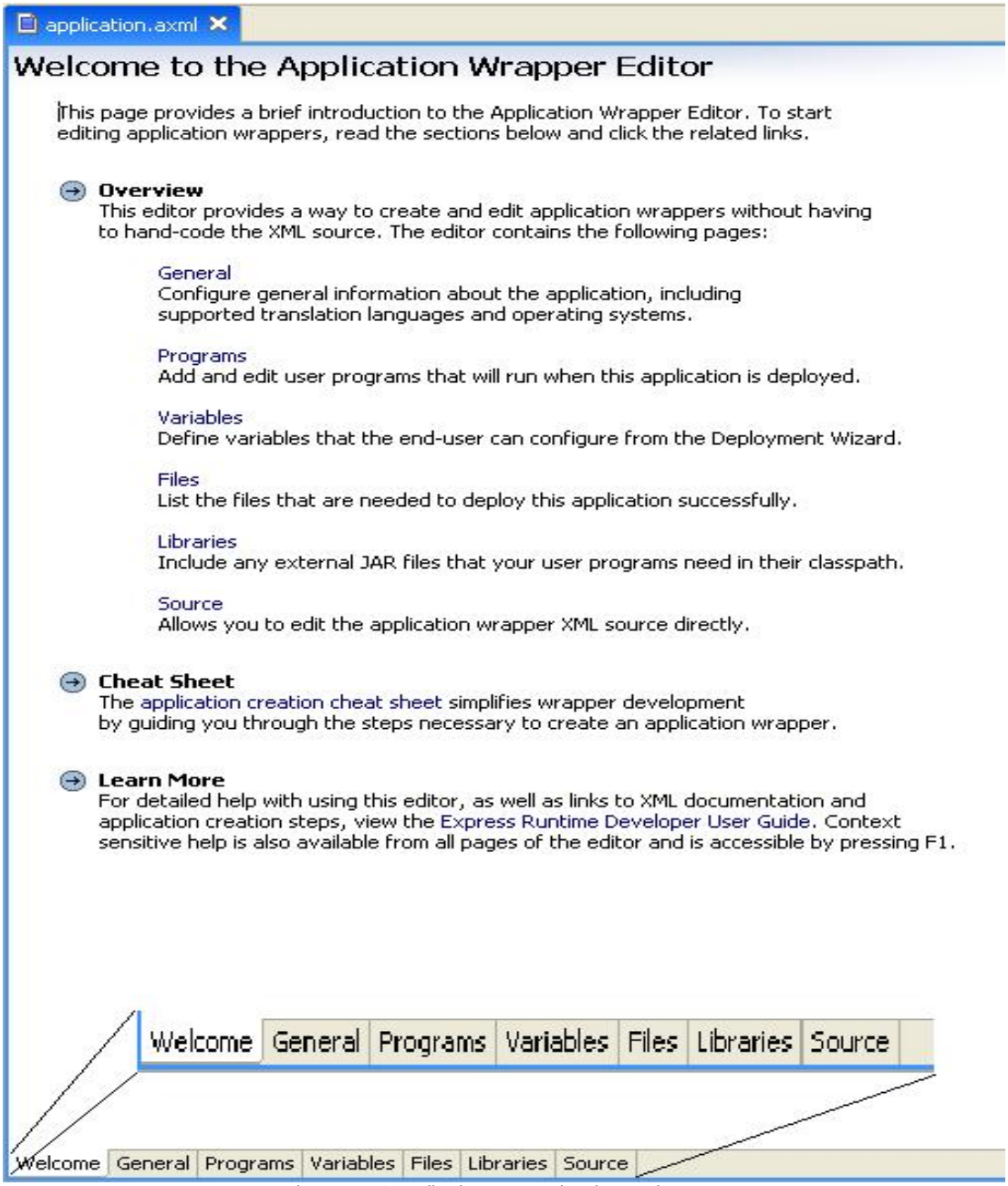

Figure 7. IDS application – general project settings

Click the **Source** tab. It contains basic code to define the application deployment logic, which will be enhanced to include necessary validation.

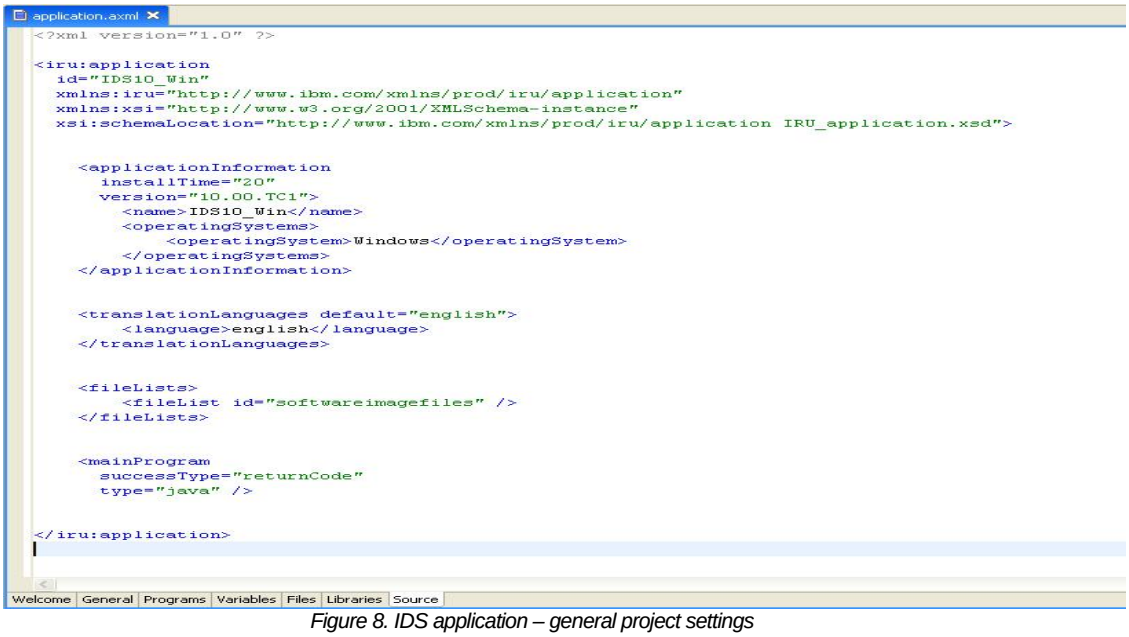

Figure 8. IDS application – general project settings

6. The path of the Source to be deployed is defined using the Files tab. Set the source directory for Software Image by clicking on **Click to set**. Browse through the folders where the IDS installation image resides and select the main folder containing the files. This must be set to define the folder where files will be picked for deployment.

   Note: Place the IDS installation image (software image files) on a local drive instead of on a mapped drive. It will be necessary to access it multiple times. To proceed with this setup, create a folder and copy the entire source into the new folder.

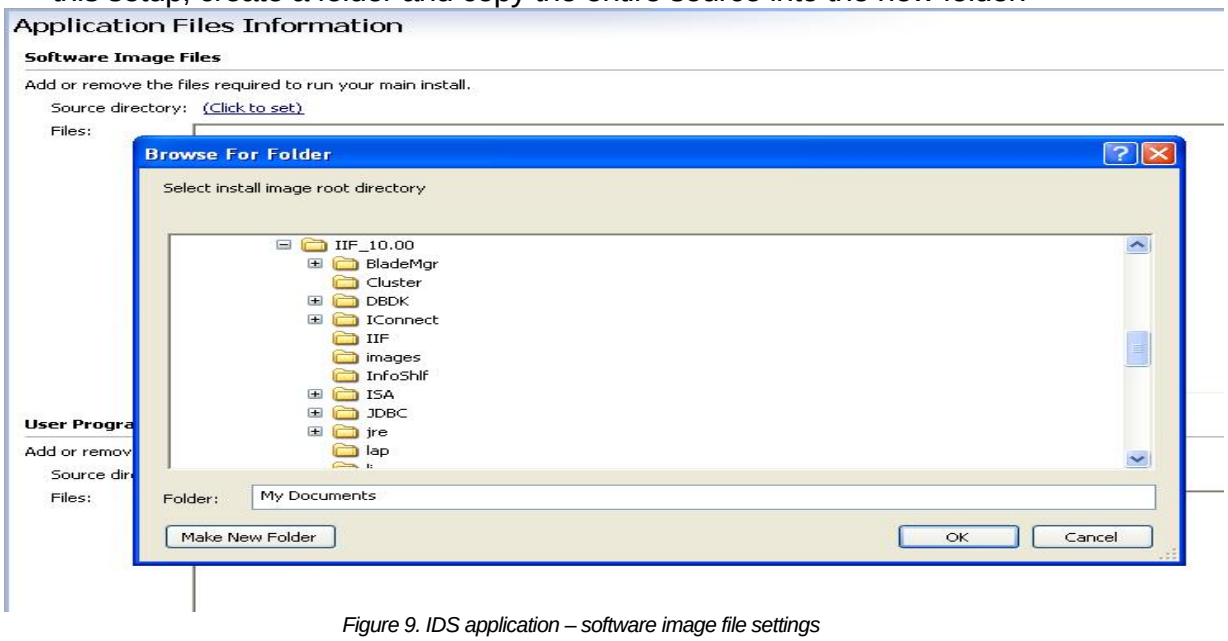

Figure 9. IDS application – software image file settings

7. To add the list of files, click the **Add** button (you might have to scroll the screen to the right) for Files option. Select the IIF, JDBC, LI, LAP and JRE directories by holding down the Control (Ctrl) key and clicking the folders. Click **Finish** to complete.

Figure 10. Selecting the source files

The selected files are now included in the files list.

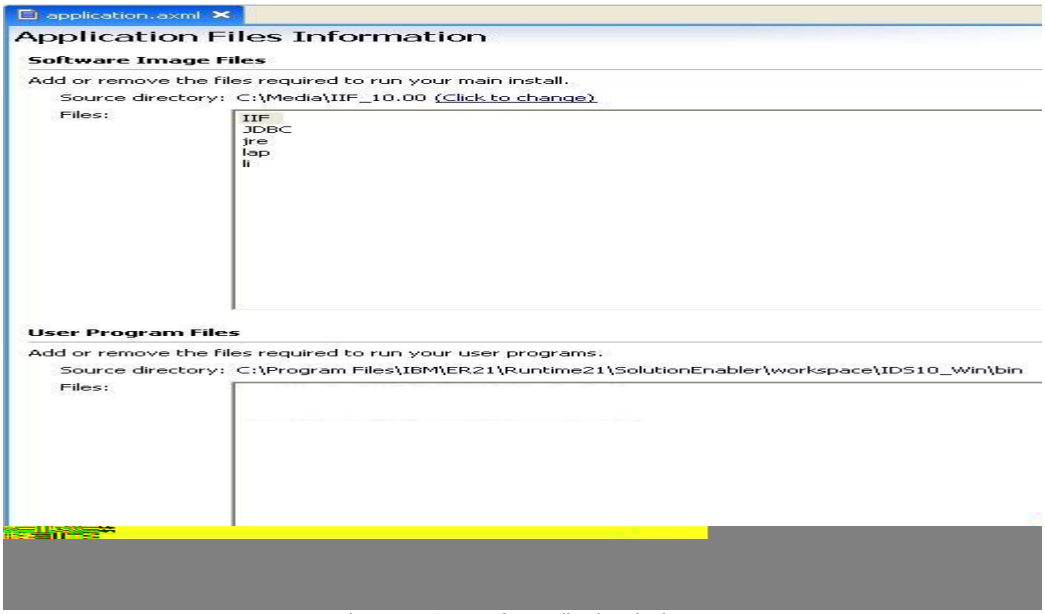

Figure 11. Source for application deployment

Sample deployment logic is explained in the subsequent section.

Edit the IDS project

Complete the following steps:

1. Using Windows Explorer, find \MEDIA\IIF_10.00\IIF and copy the response file silent.ini from \MEDIA\IIF_10.00\IIF\silent.ini to the same location as application.axml.
2. Add necessary sections to the application.axml file. Open the application.axml file by clicking the Source tab. Copy the individual sections, or cut and paste the entire application.xml file that is listed in the appendix of this paper. To view or download the entire application.axml file, see the IBM Informix Software: Portfolio Update and Future Directions CD-ROM or the IBM® Informix Software: IBM Express Runtime Wrapper for IBM Informix® Dynamic Server Workgroup Edition 10.00 CD-ROM.
    a. Define the predeployment checker (PDC). The PDC checks whether IDS is already installed on the target system.
    b. Define the main program section. The main program section does the installation.
    c. Define a validation program that performs the installation and verifies its completion.
    d. Define the exit program section that installs the JDBC driver and verify that the database server is up and running.
3. Edit or cut and paste the entire application_english.xml file that is listed in the appendix. Any field in the application.axml that requires a string can also be coded as `translatedKey="name"` in which case the value of “name” must be defined in the application_english.xml.
   NOTE: The name of this file reflects the language it is written for. The complete file application_english.xml is available through the location listed on the front cover.
4. Create the user programs.
    a. Navigate to the src/IDS10_Win/userPrograms folder. Right-click the folder and select **New > Package**.

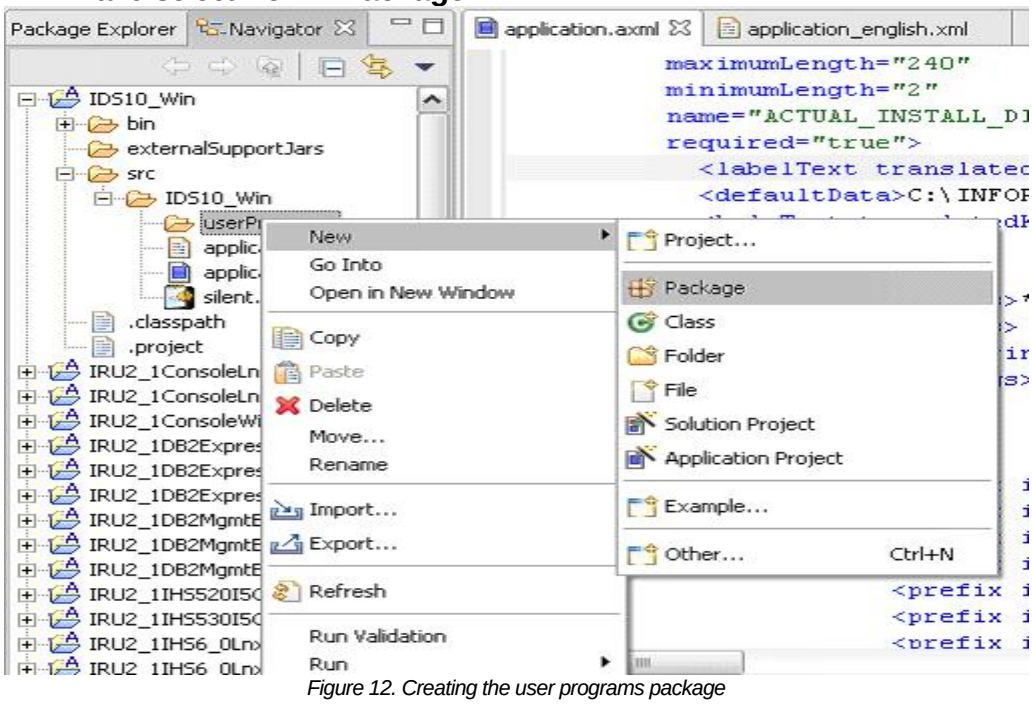

Figure 12. Creating the user programs package

b. Enter the package name `com.ibm.iru.ids10win` and click **Finish**.

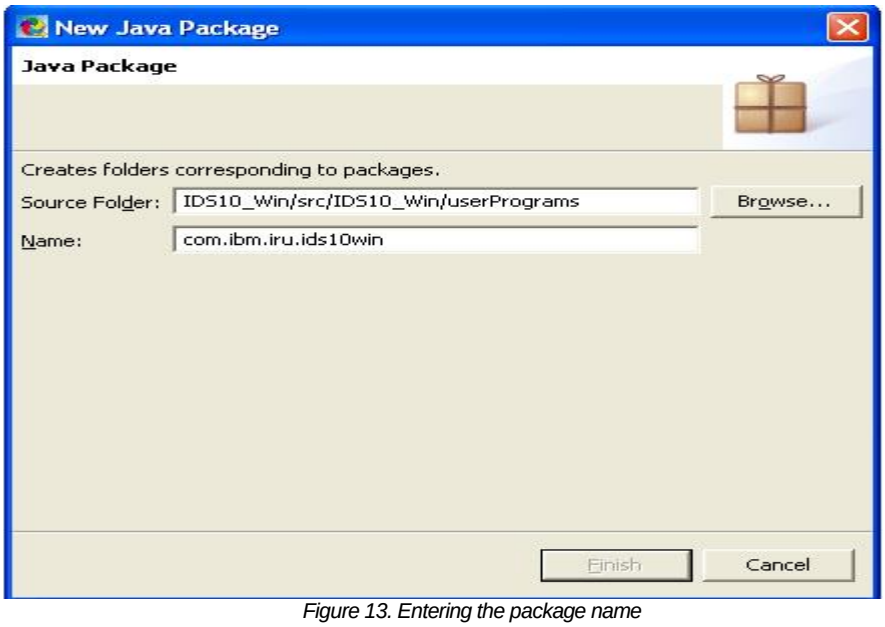

Figure 13. Entering the package name

c. Create the Java® programs in the folder. Right-click the package name and select **New > Class**.

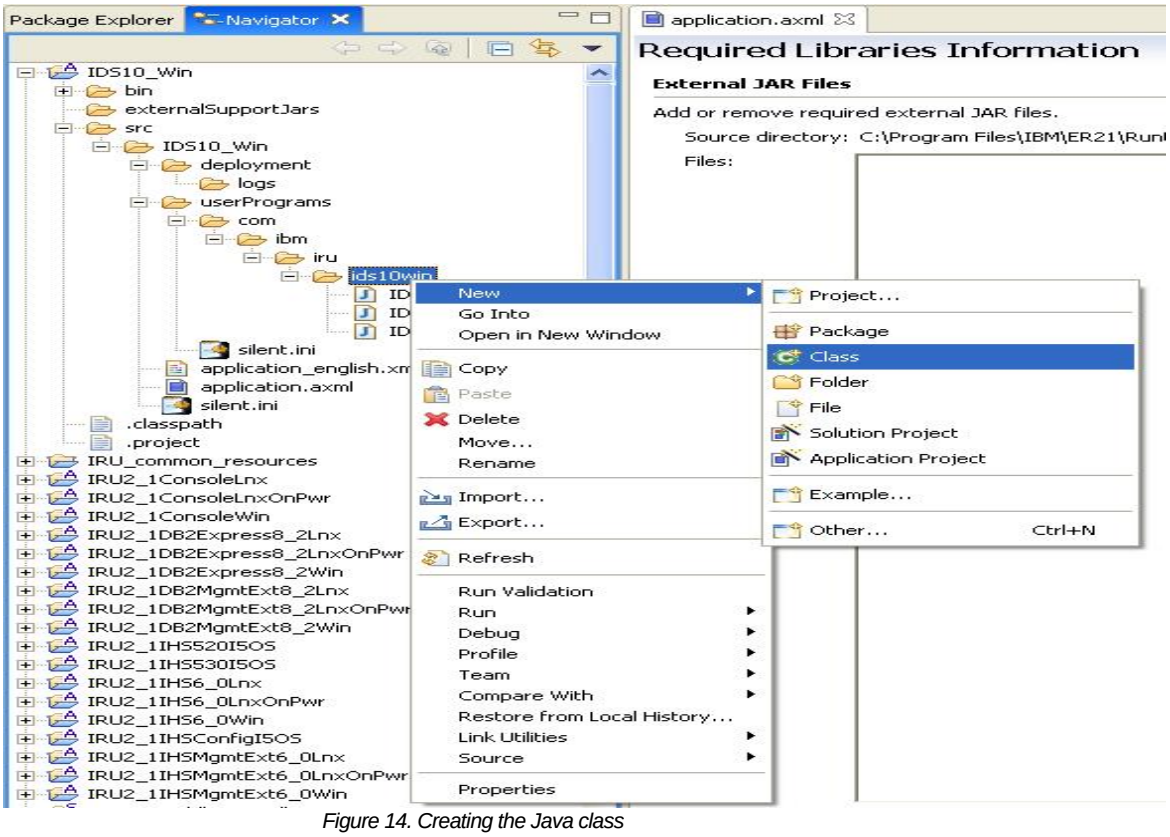

Figure 14. Creating the Java class

d. Enter the class name IDS10WinPDC.

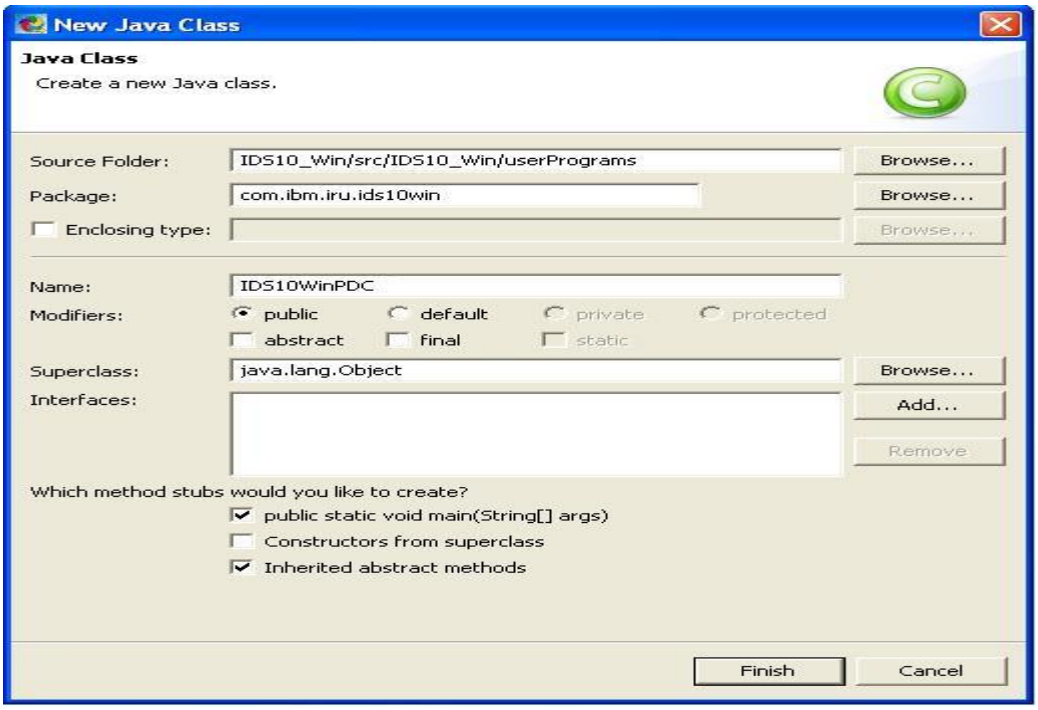

Figure 15. Entering the class name

e. Select public as the modifier.
f. Select method stubs of public and inherited.
g. Click **Browse** and select the Super class. In the Superclass Selection dialog box, enter supportw for type and select the SupportWindowsBase class. Click **OK**.

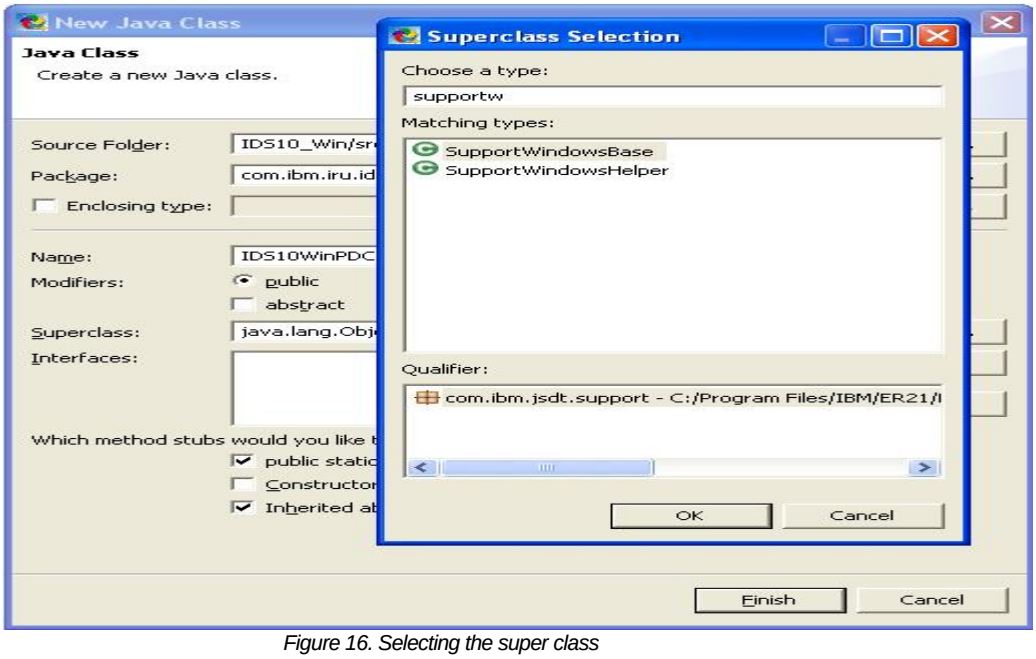

Figure 16. Selecting the super class

h. Click **Finish**.

i. Perform the same steps to create IDS10WinExit and IDSWinMain.
j. Create an IDS10Constants class that has java.lang.Object as its super class. This step does not require adding super class SupportWindowsBase.
k. The complete source code for the four classes are listed in Appendixes C, D, E and F; IBM Informix Software: Portfolio Update and Future Directions CD; and the IBM Informix Software: IBM® Express Runtime Wrapper for IBM Informix® Dynamic Server Workgroup Edition 10.00 CD.
l. Ensure that required code changes to the project and classes are complete.

5. Verify the changes. This can be done by traversing the tabs provided in the main project screen (application.axml), which was explained in the project creation section. Clicking on the Programs tab displays the list of new classes added to the project. For this sample, they are PreDeploymentChecker, Main, Validation and Exit Programs. This screen also facilitates checking the return values (return code, string check) and response file (if any) to be used.

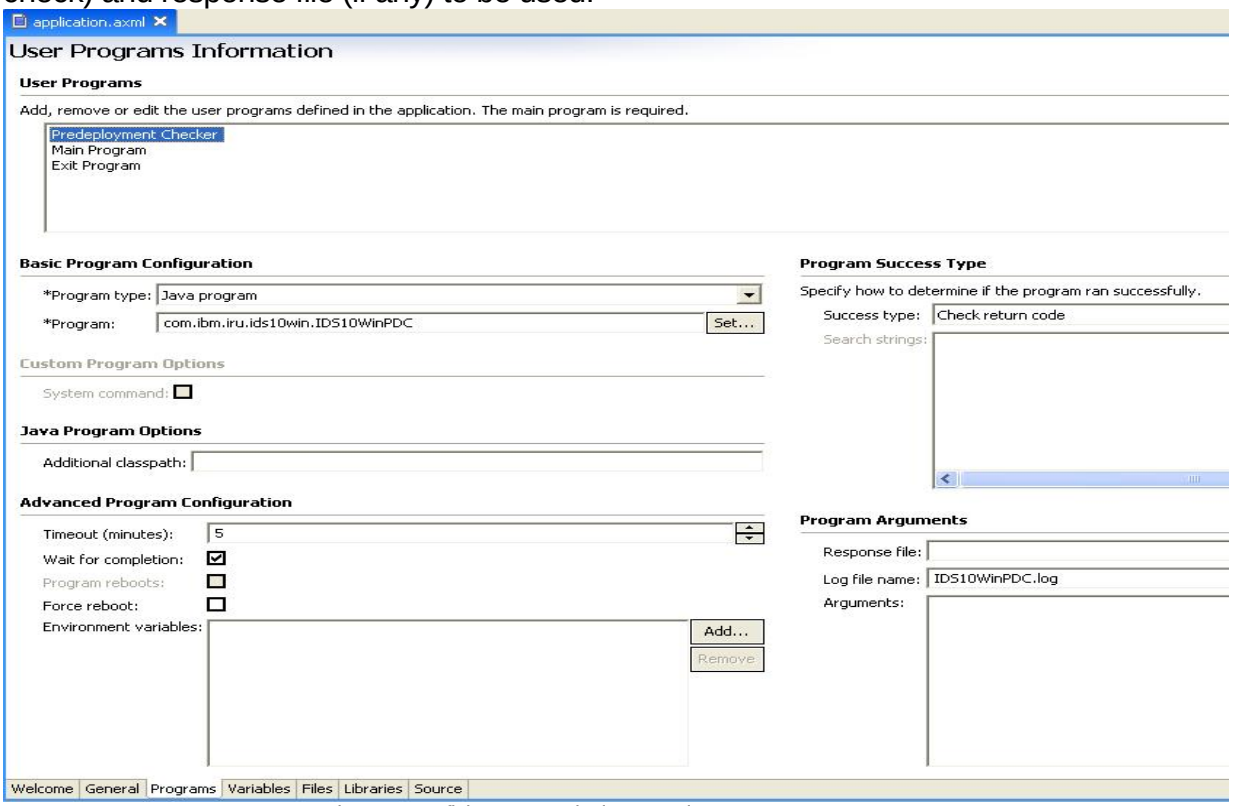

Figure 17. Defining custom deployment classes

6. Click the **Variables** tab to display the list of variables defined in the main class.

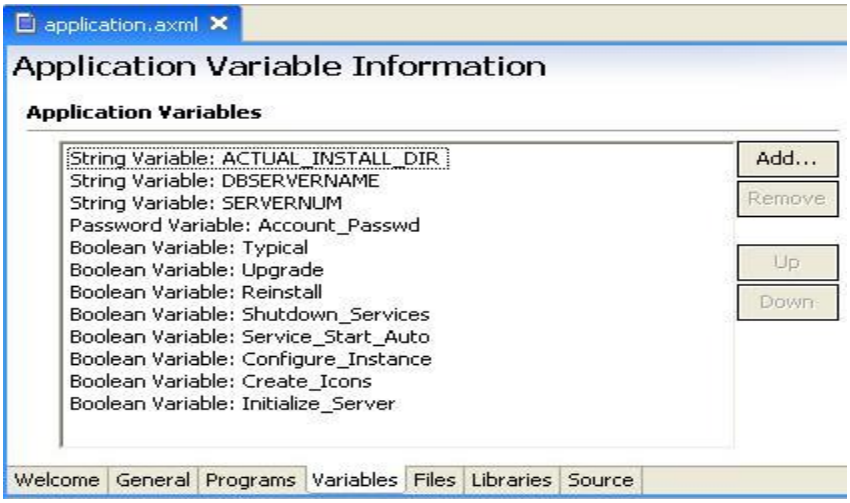

Figure 18. Application variables

Based on the selected variable, its associated definition and attributes are displayed.

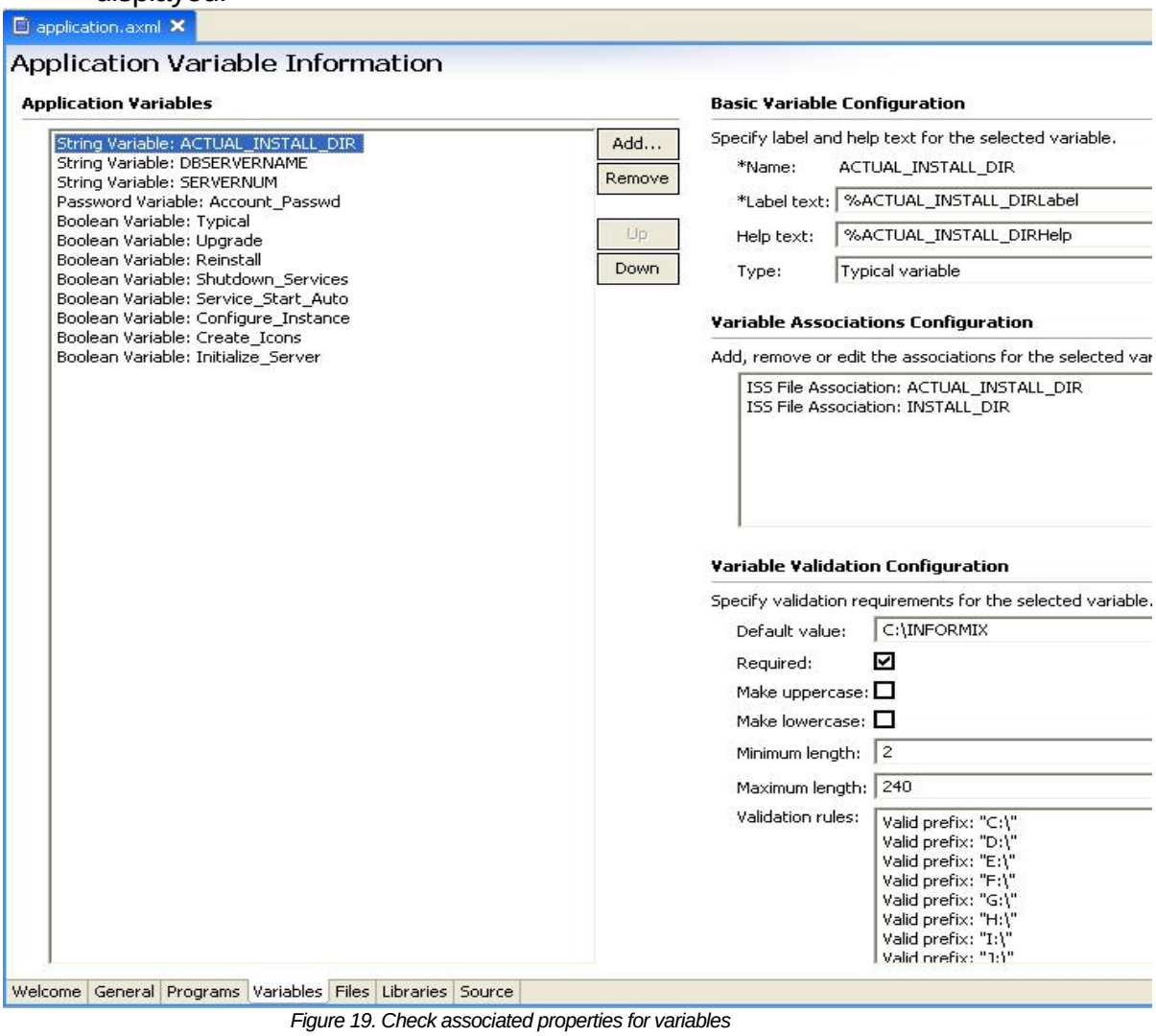

Figure 19. Check associated properties for variables

7. Right-click the application project and select **Generate Application** to build the application.
8. Verify that there were no error messages during the build.

Create a test solution

This test solution contains only the IDS application. In an actual deployment, it would also contain other application components such as WebSphere® Application Server - Express and an ISV application.

1. Right-click anywhere in the Package Explorer panel and select **New > Solution Project.** The solution wizard begins.

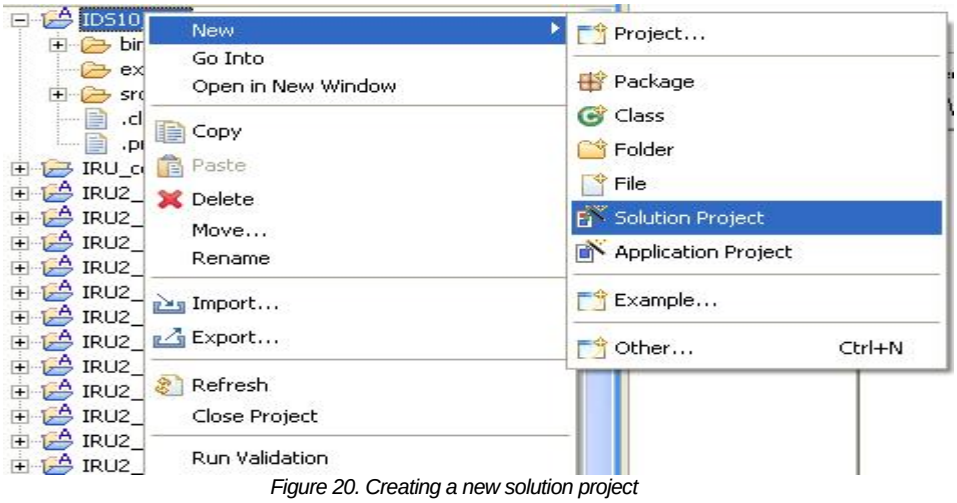

Figure 20. Creating a new solution project

2. Type a Project name (for example, TestIDS). Click **Next**.

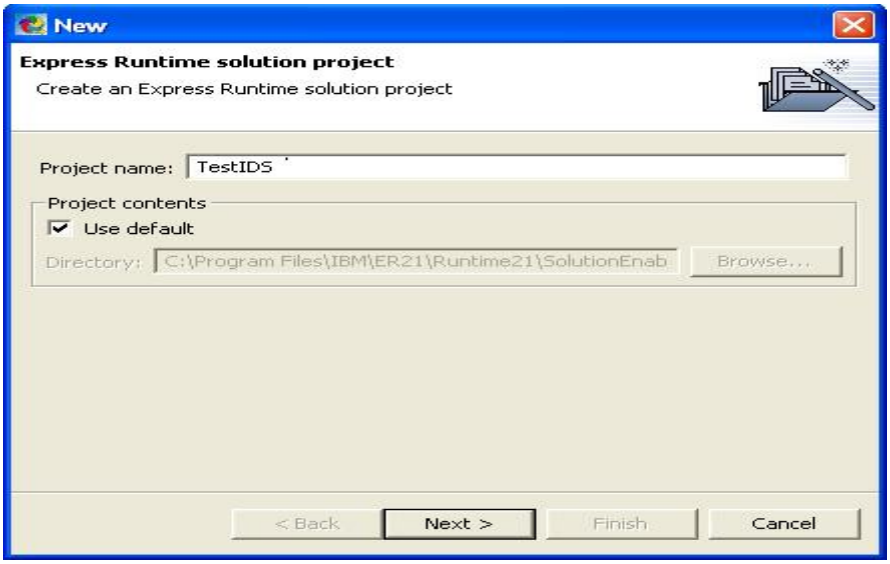

Figure 21. Entering the solution project name

3. Enter the data for the remaining fields in the panel. The fields are already pre-populated. The Solution ID defines the final name of one of the parts that will be deployed. For Solution title, type **A Test for IDS Wrapper**. This title will be used as heading during deployment. Select the language. Click **Finish**.

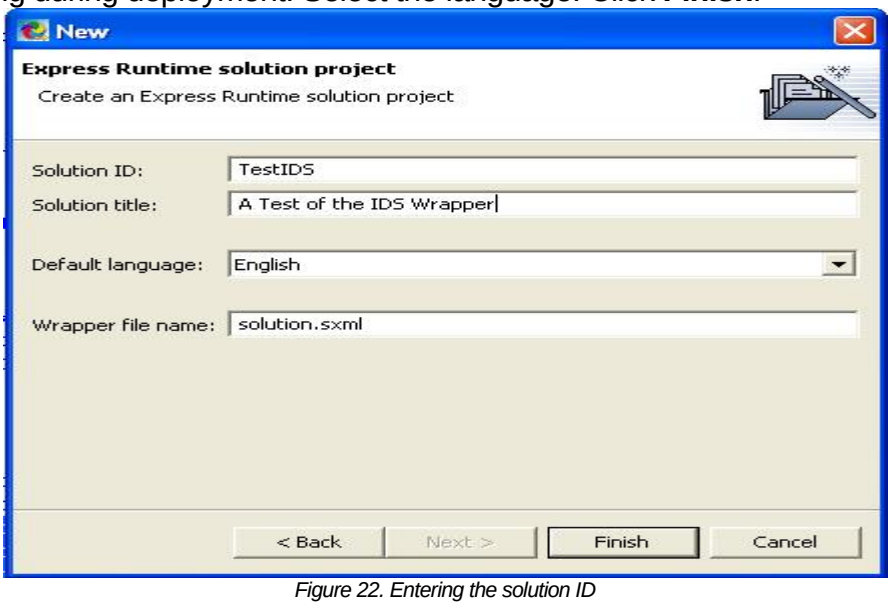

Figure 22. Entering the solution ID

4. In the Resource view, the solution contains a welcome screen with different tabs to define deployment details.

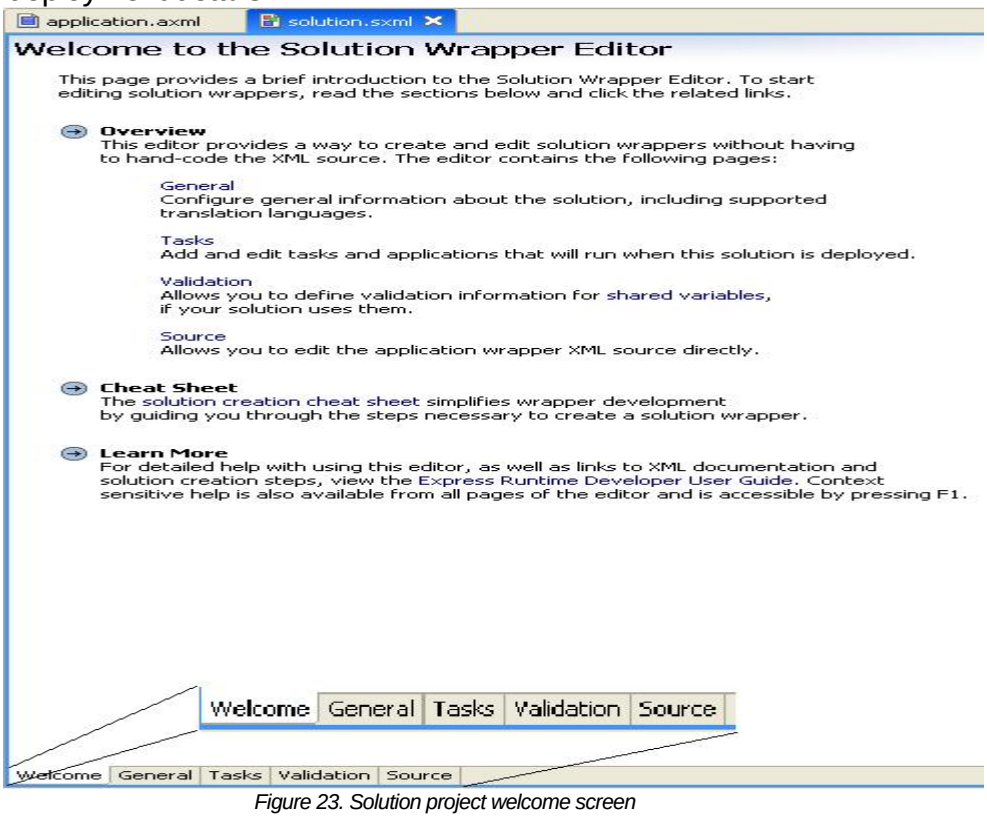

Figure 23. Solution project welcome screen

5. Click on the Tasks tab to define the project details that will be deployed. Click the **Add** button to display a panel to define which type of task you are adding. Select **Install task**. Click **Next**.

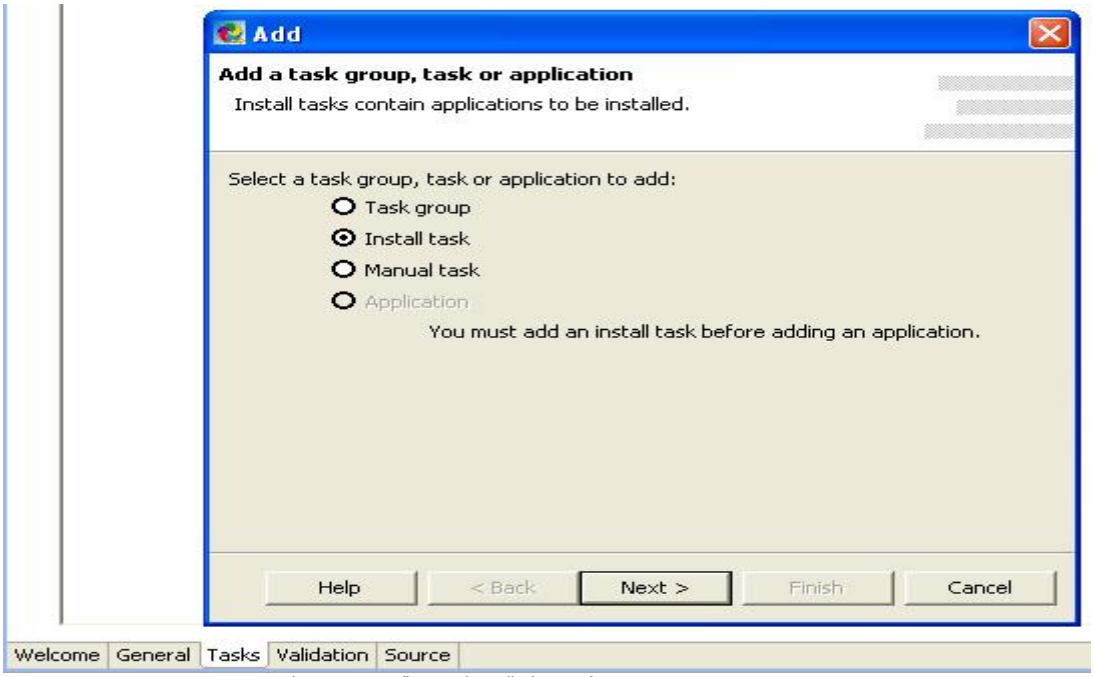

Figure 24. Define an installation task

6. Type a task description, such as “Deploy IDS solution.” The “Launch the Add Application” box is selected by default for you to define the application to be deployed. Click **Finish**.

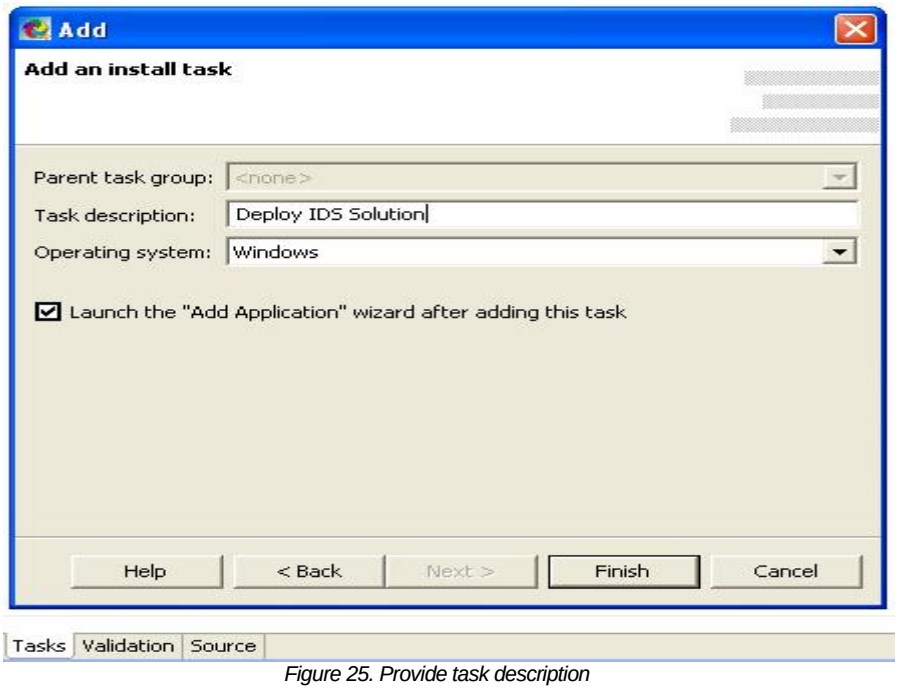

Figure 25. Provide task description

7. Select IDS10Win as the application that should be deployed from the Add Applications panel. Click **Finish**.

.

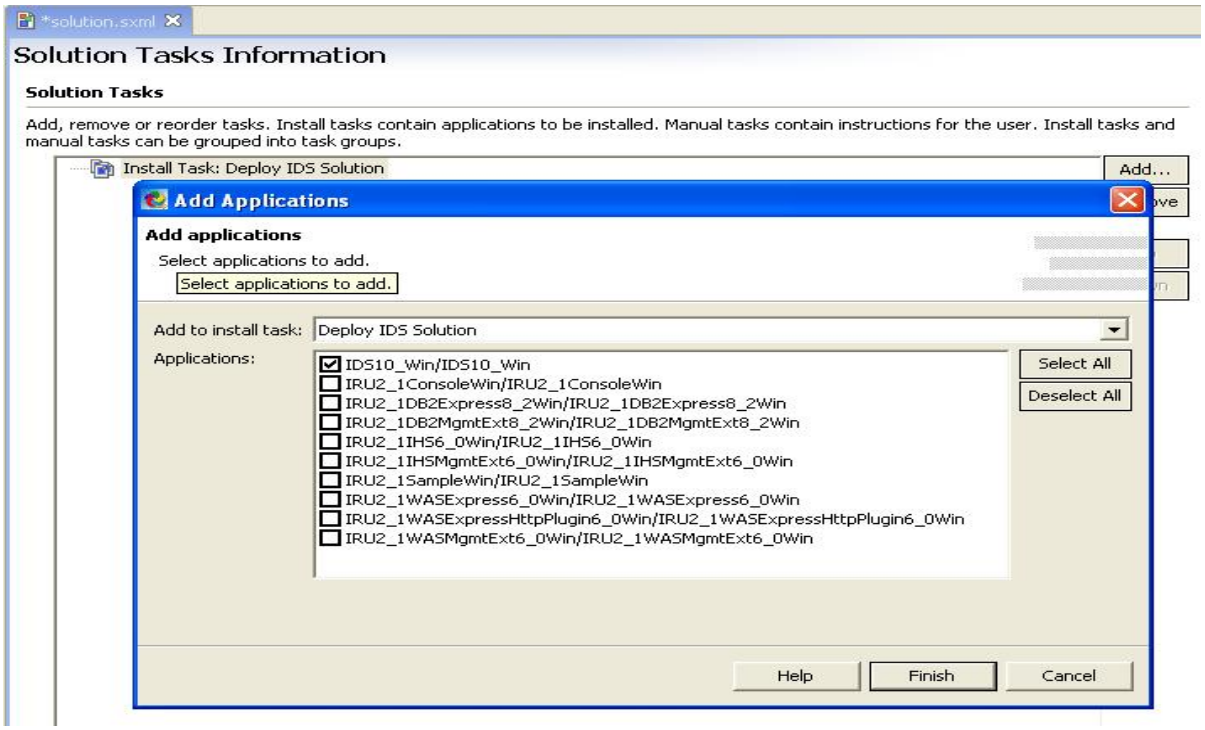

Figure 26. Select application to deploy

The Deployment link is displayed in the following interface

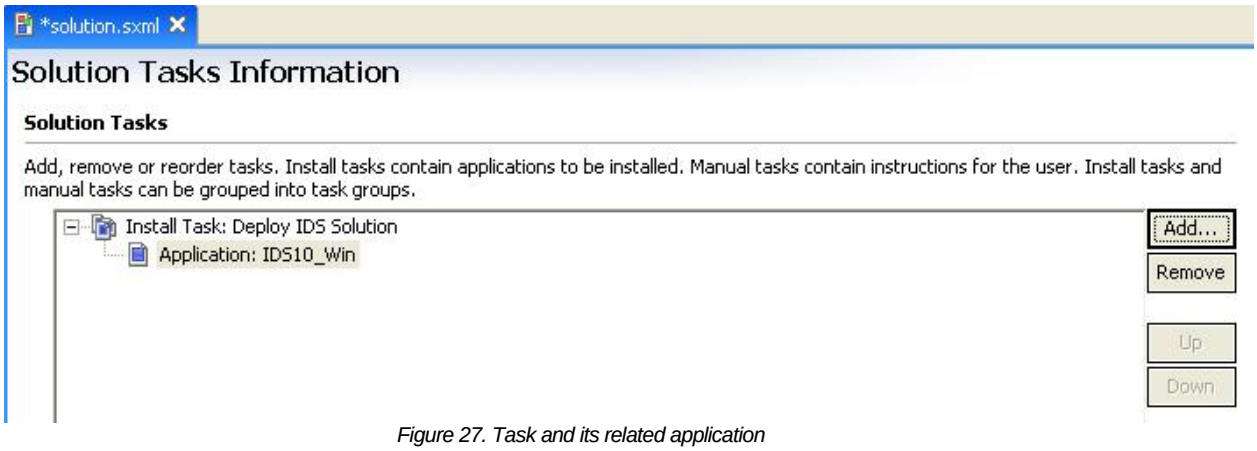

Figure 27. Task and its related application

8. Save the changes by selecting **File > Save** or by clicking the **Save** button in the toolbar. Build the solution by right-clicking the solution project, such as Deploy IDS Solution, and select **Generate Solution**.
9. Verify that there were no error messages during the build.

Deploy the solution

1. Right-click the solution project, such as TestIDS from the Project Explorer view, and select **Export**.

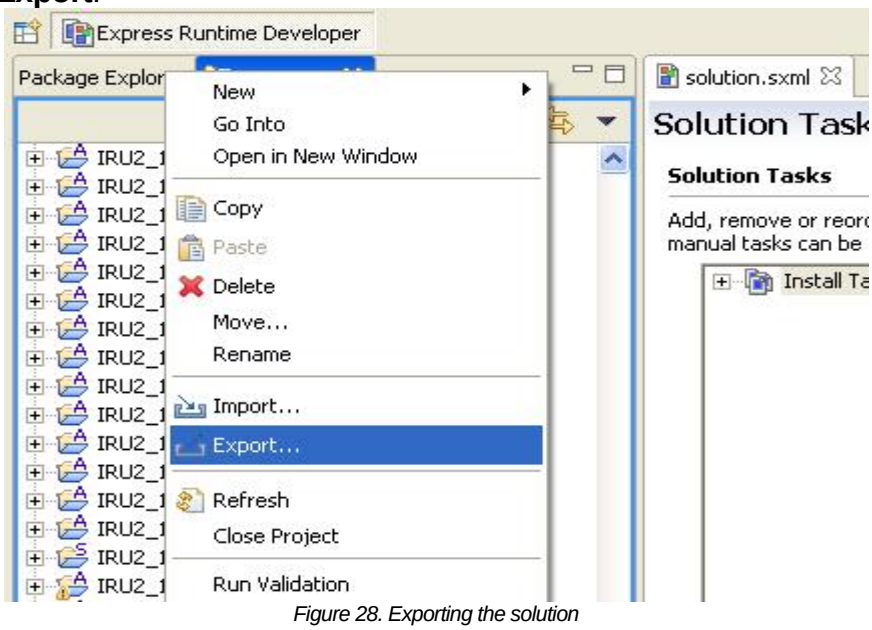

Figure 28. Exporting the solution

2. Select **Express Runtime Solution** from the list.
3. Project TestIDS is automatically selected from the list. Browse to <ER21 Install Path>\SolutionEnabler as the destination directory. Click **Finish**.

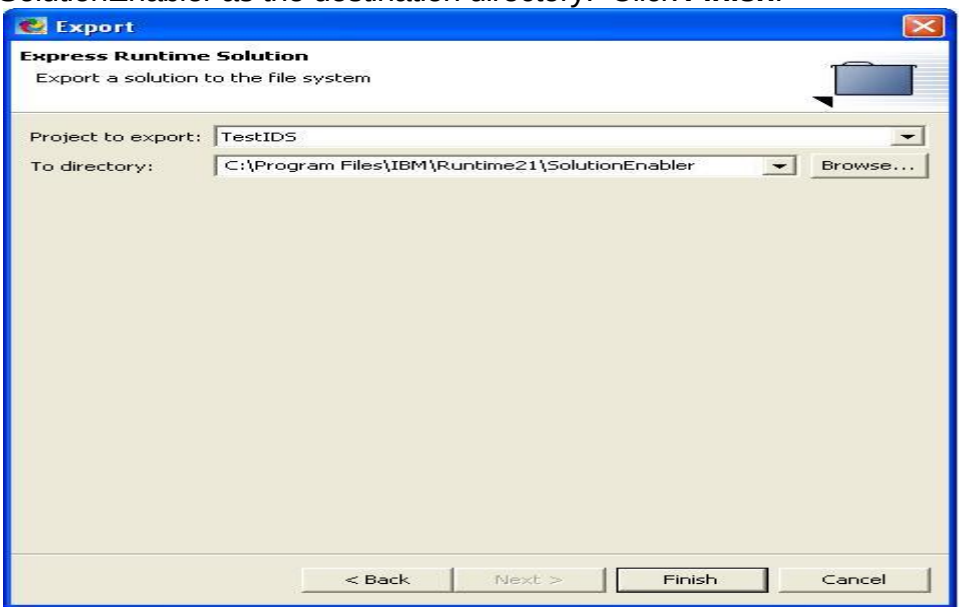

Figure 29. Export a solution

The TestIDS.ser file is located in the <ER21 Install Path>\SolutionEnabler directory.

4. Right-click on the application project and select **Generate Deployment Packages**.

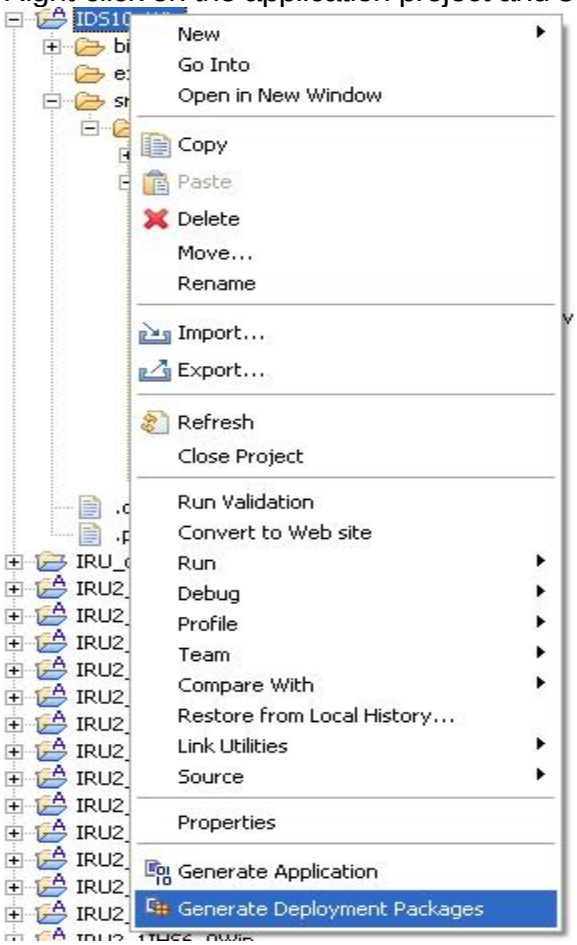

Figure 30. Start deployment package generation

5. After the generation is complete, click **OK**. This creates a jar package in <ER21 Install Path>\SolutionEnabler\Workspace\IRU_common_resources\mediaJar directory.

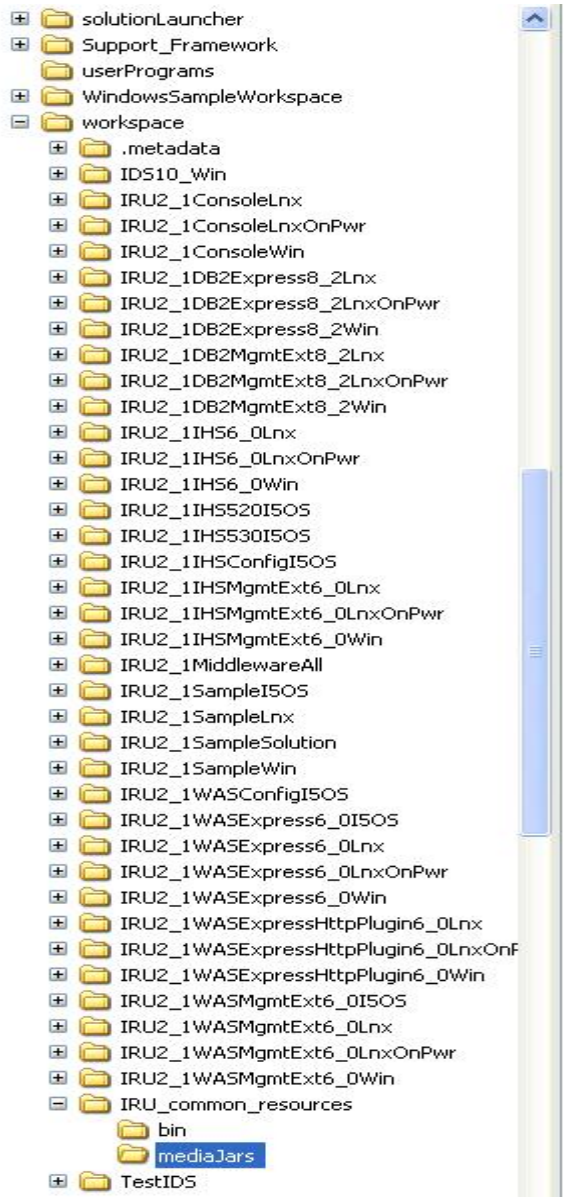

Figure 31. Path where deployment package is created

6. Open Windows Explorer and navigate to the <ER21 Install Path>\SolutionEnabler\Workspace\IRU_common_resources\mediaJar directory. Select the file ids10_win.xx.jar. Right-click and select **Copy** to copy the package to the clipboard.
7. Navigate to the <ER21 Install Path>\SolutionEnabler\com\ibm\jsdt\webserver\tree directory. Right-click and select **Paste** to create a copy of the package.

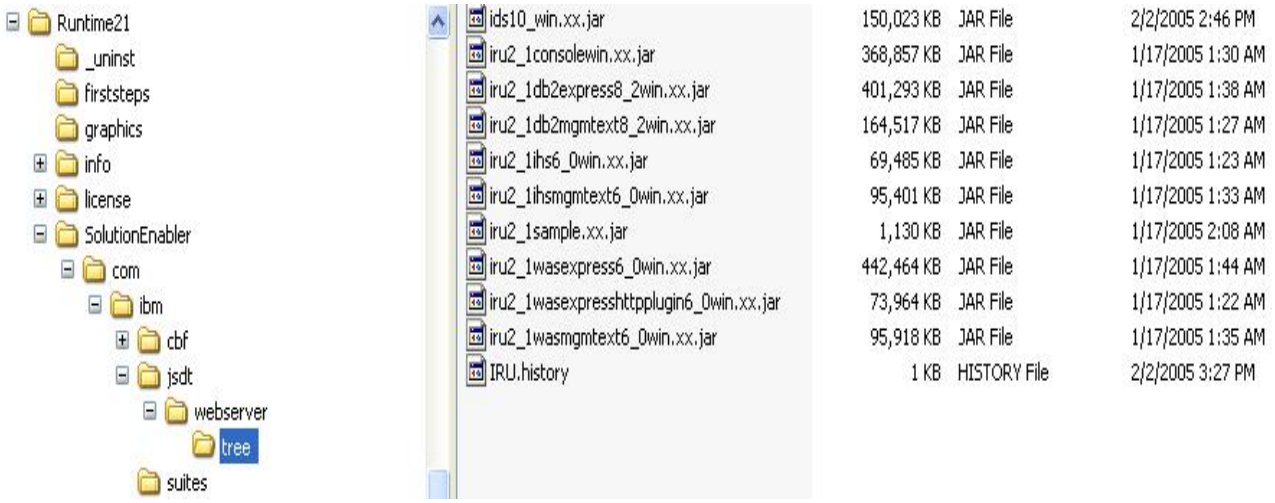

Figure 32. Path where deployment package must be copied

8. Start the deployment wizard from the Start menu.

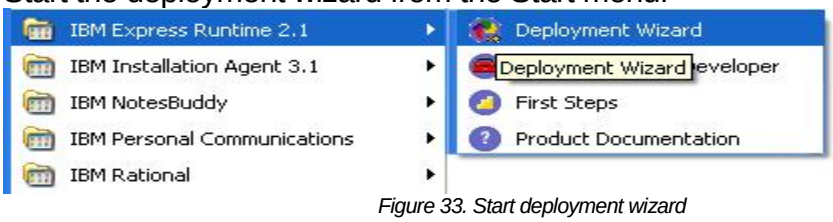

Figure 33. Start deployment wizard

9. Click **File > Open**. Navigate to the TestIDS.ser file and open it.

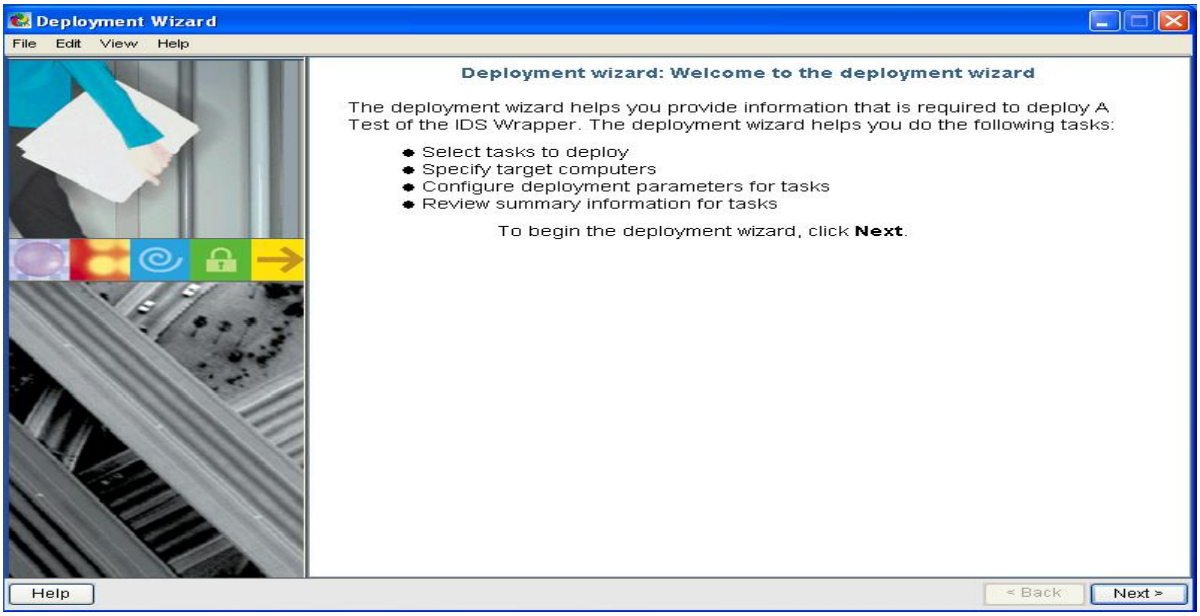

Figure 34. Deployment wizard interface

10. Select the solution to be deployed, such as Deploy IDS Solution.

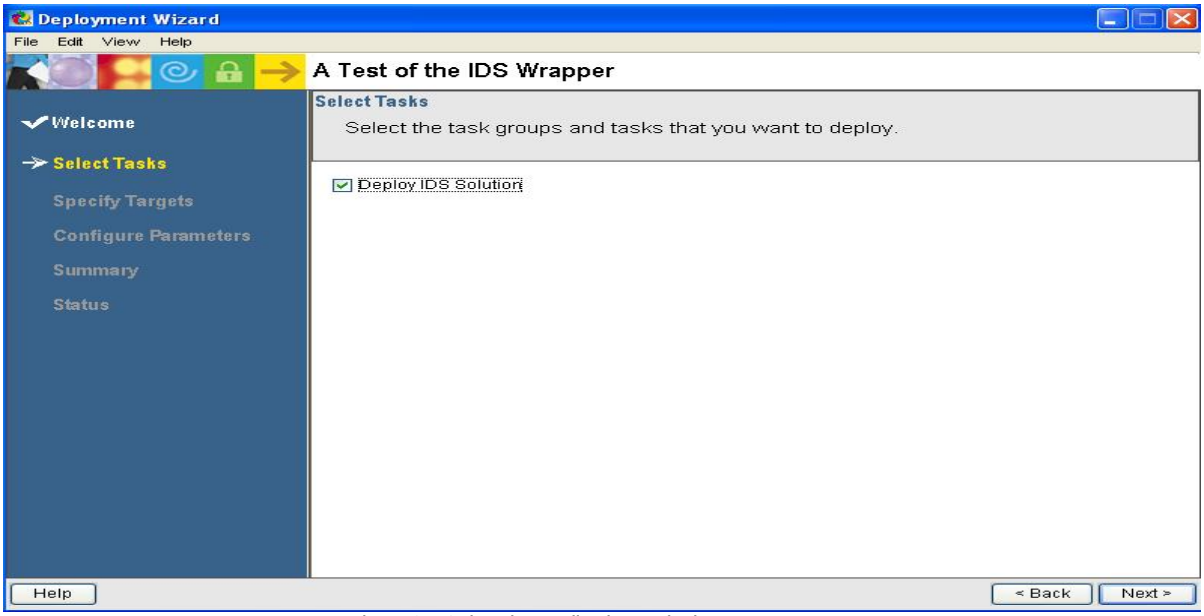

Figure 35. Select the application to deploy

11. Type the host name in the Target Computer text box and click **Add**. The host now appears in the Selected target computers list.

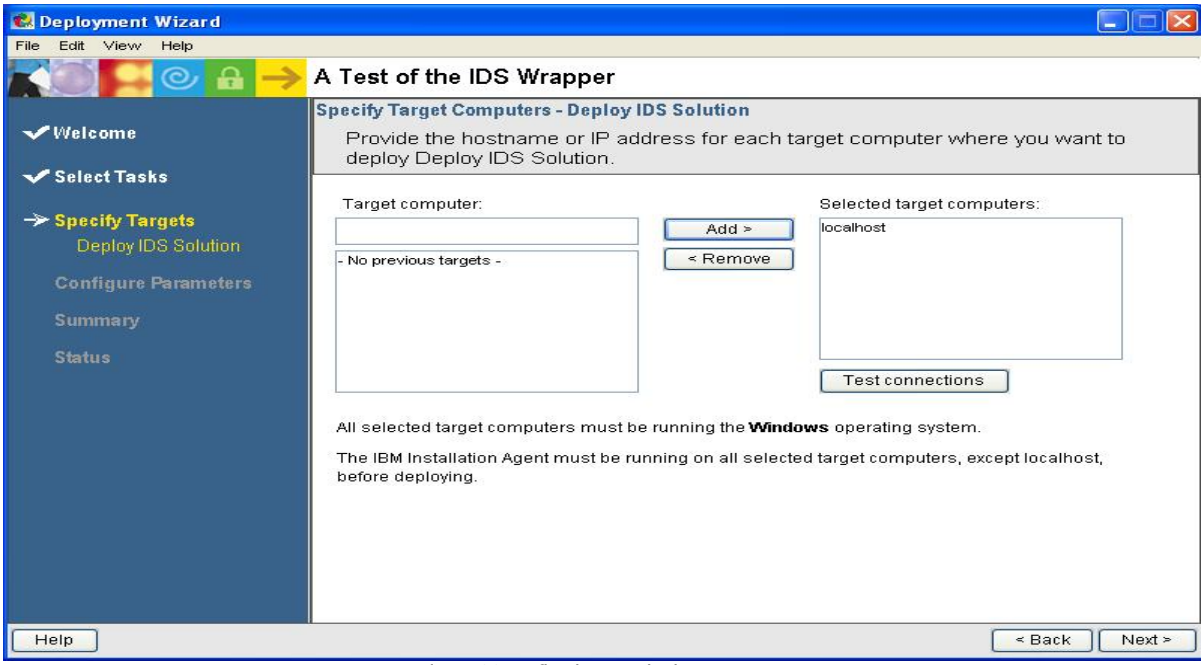

Figure 36. Define host to deploy

12. Click the **Test connections** button to ensure connectivity. Click **Next**.

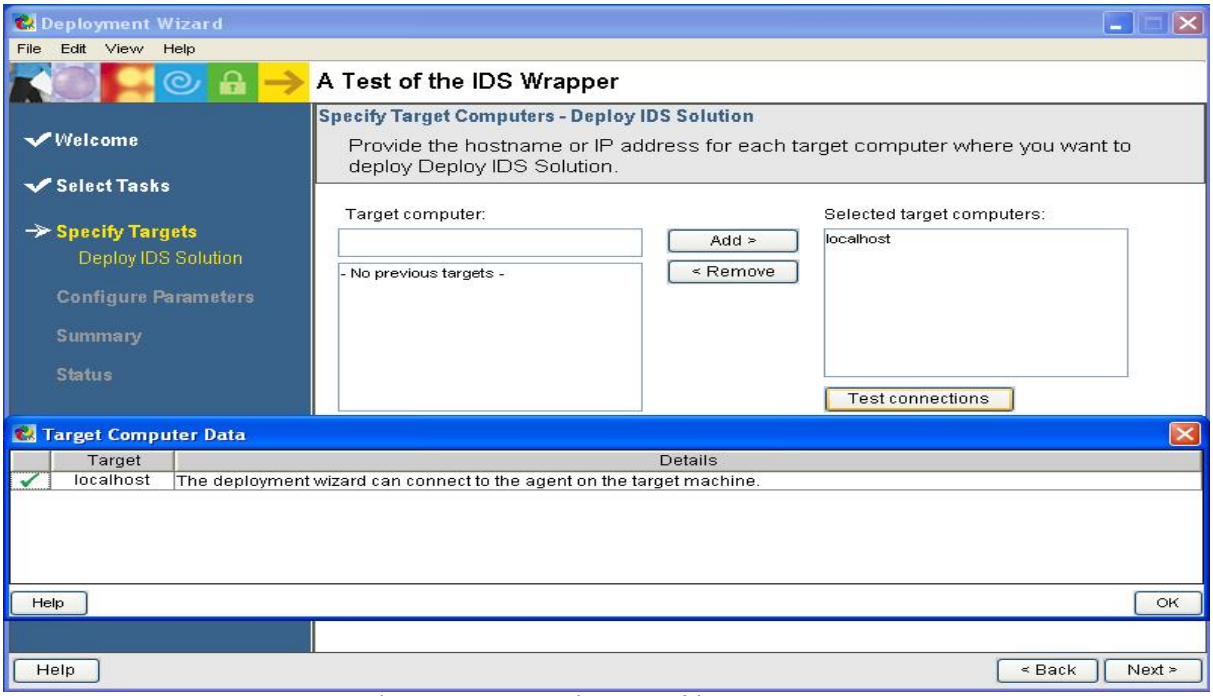

Figure 37. Test connection successful

Note: The IBM Installation Agent must be installed for remote deployment. If this connection test fails, ensure that the IBM Installation Agent is installed on the target computer.

13. Verify and make any changes to the product settings. Click **Next**.

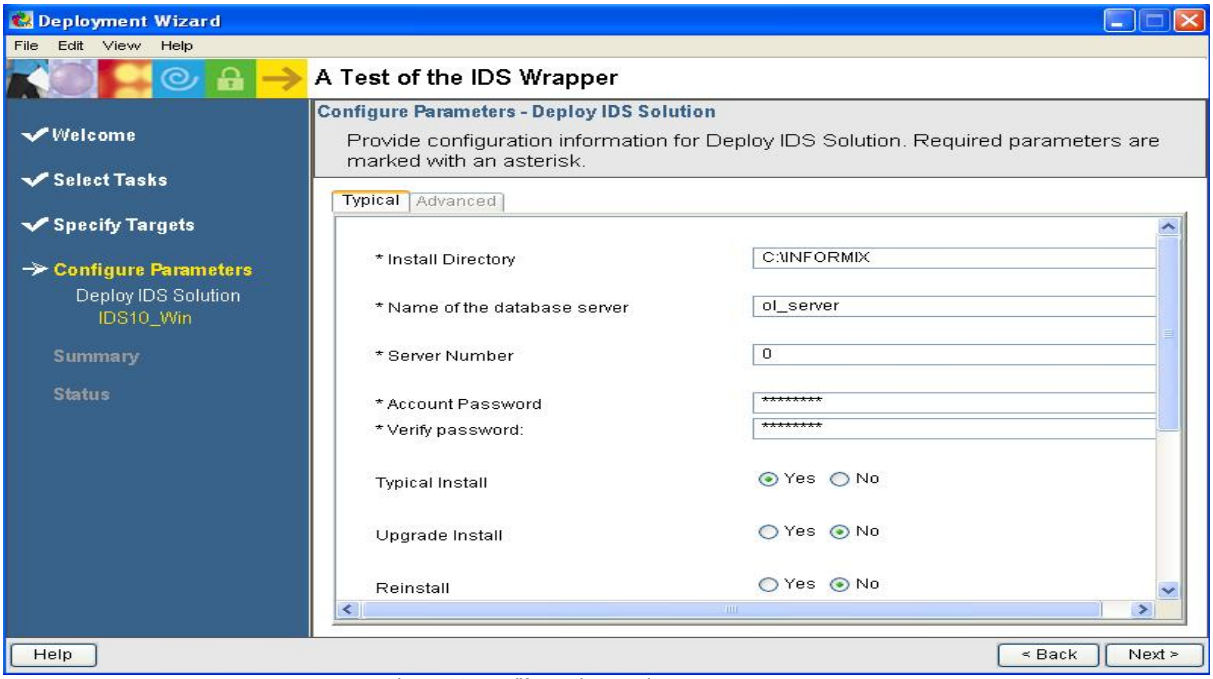

Figure 38. Modify product settings

14. Click **Deploy Task** to start deployment. Click **Agree** to the license prompt to start deployment.

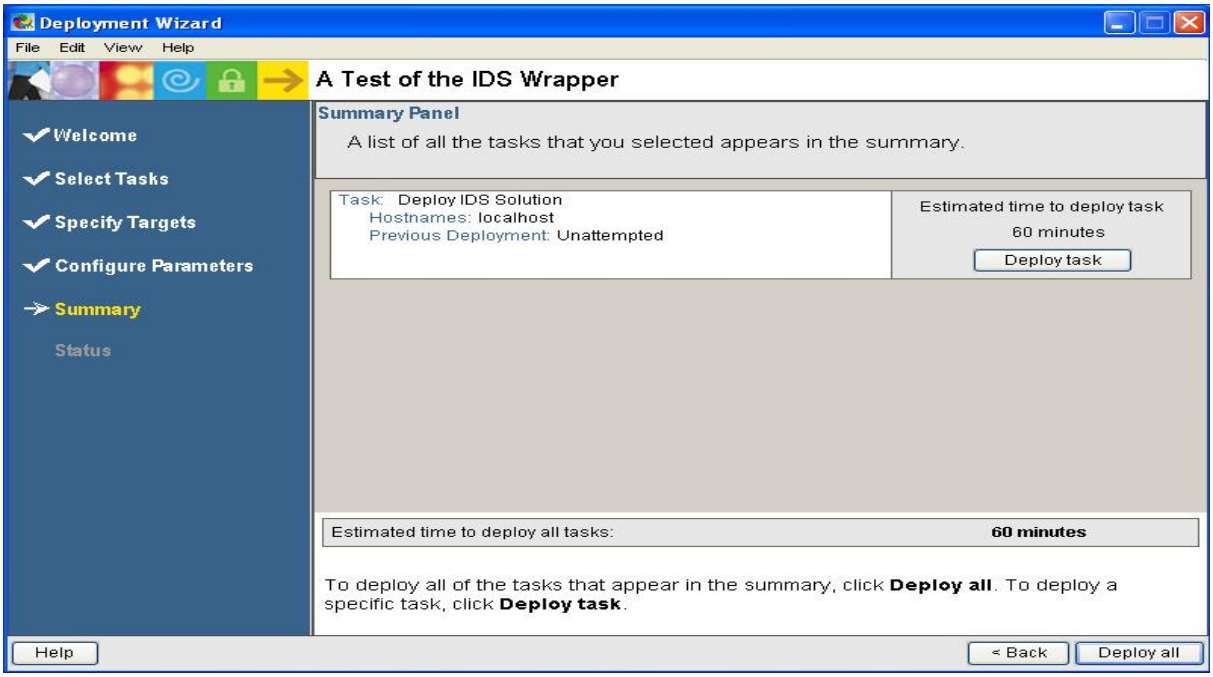

Figure 39. Start deployment

Deployment is complete when a message similar to the following is displayed:

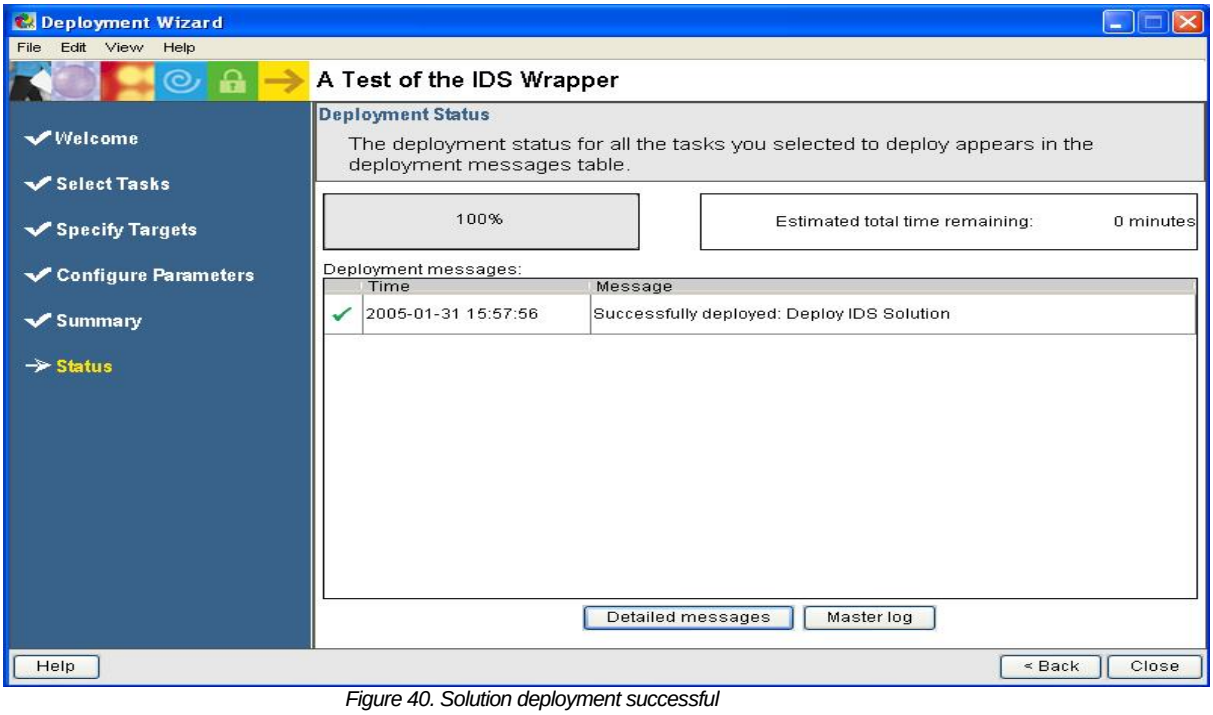

Figure 40. Solution deployment successful

15. Optionally click **Detailed messages** to check the stages of deployment.

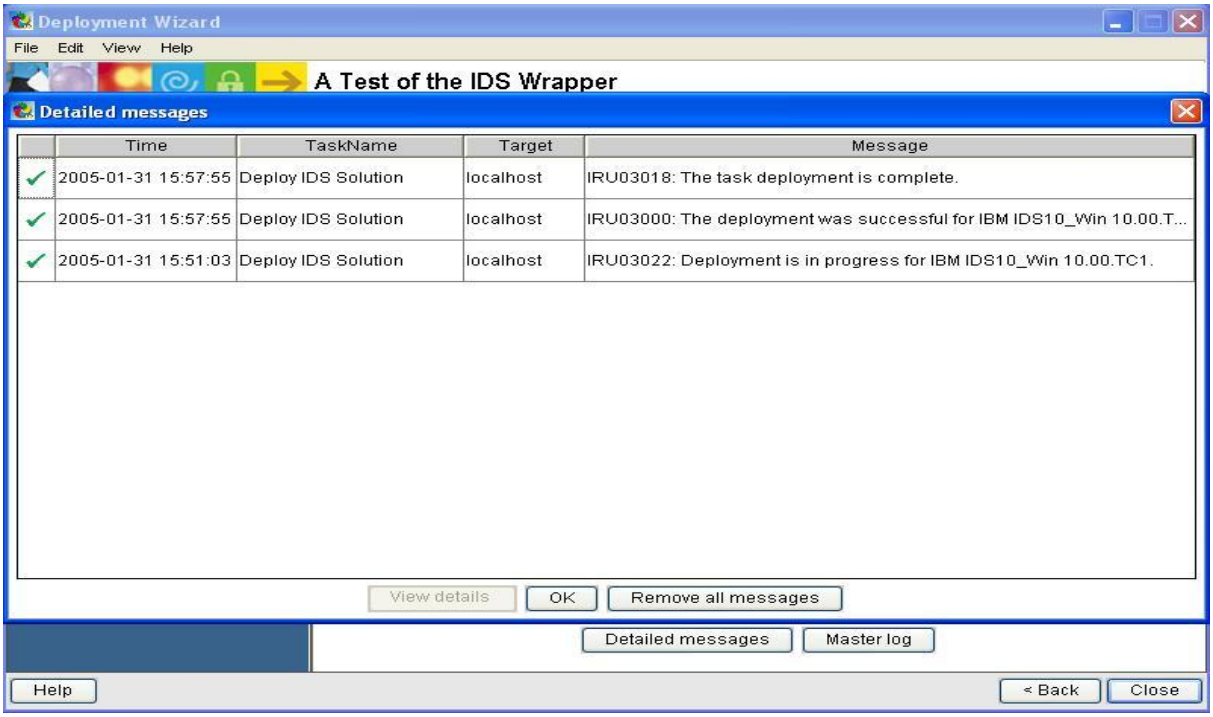

Figure 41. Stages of deployment

When deployment is finished, you have a complete, tested IDS 10.00 installation on the target system.

Troubleshooting

This section has two parts: things to consider while developing the solution and things to consider during and after solution deployment.

Troubleshooting while developing a solution

- To debug your Java application in the Express Runtime developer, do these steps:
    1. Open the user program Java source file (for example, IDS10WinPDC.java) you want to debug, and set up breakpoints in the code where you want to pause during execution.
    2. Deploy the solution in the test environment by right-clicking the solution project (for example, TestIDS) and selecting **Test Deployment Wizard**. You are prompted with Confirm Perspective Switch when the solution reaches a breakpoint. Select **Yes**. Deployment stops with the debugger waiting on the breakpoints. Follow normal Eclipse debugging procedures.

       Note: If you get a message stating "User program debugging cannot be enabled because port 1099 is in use," you must free the port 1099 and restart the deployment in the test environment. The IBM Installation Agent also uses port 1099; you cannot simultaneously debug the user programs and run the IBM Installation Agent on the same machine.
- If any changes are made to the application wrapper or any of the XML files, regenerate the application or solution wrapper.
- Several status log files are generated during the processes of generating application wrappers, solution wrappers, and deployment packages. The activity is written to the Express Runtime developer console and to the following log files located in the <ER install_directory>\Runtime21\SolutionEnabler\workspace directory:
    - Application generation status is in <application name>\bin\<application id>\log\applicationBuilder.log
    - Deployment package generation status is in <application name>\bin\<application id>\log\softwareImageDeploymentPkgBuilder.log
    - Solution generation status is in <solution id>\bin\log\solutionBuilder.log

Troubleshooting during and after solution deployment

- The problem determination files are located under <ER install _directory>\Runtime21\SolutionEnabler\logs as follows:
  - The IRU_DeploymentWizard.log file provides details about deployment status. In most cases, problem determination can be performed using this file.
  - The IRU_IIATrace.log file provides data from Java System output and error messages written inside application wrapper user programs and the Express Runtime framework classes.
  - The IDS10silent.log file contains IDS-installation-specific information and is located under <ER install_directory>\Runtime21\SolutionEnabler\Deployment\logs.
- Additional problem determination files are removed from the system once deployment is complete. Those files can be left on the system if the deployment wizard or the IBM Installation Agent (in case of remote install) is started in a debug mode:
  - To run the deployment wizard in debug mode, copy the Deployment Wizard shortcut from the **Start** menu and place it on the desktop. Right-click it and select **Properties > Shortcut**. Modify the **Target** field by appending the `-leavefiles` option at the end of the existing command line.
  - To run the agent in debug mode, start the agent using <IIA install_directory>\IRU_DebugInstallationAgent.bat on Windows.

  Additional files might be found in the following directories:

  - <ER install drive>:\iru<x> where x is optional, but if present, x will be an integer value. This directory contains files and directories unpacked during solution deployment. In case of any abrupt end of deployment, check the status of the unpacked files and directories.
  - <ER install directory>\Runtime21\SolutionEnabler\deployment\logs – local deployment log directory. The ibmnsi.log file contains the complete command the deployment wizard uses to launch the last Java wrapper program. The IDS10silent.log file contains IDS installation specific information and IDS10WinExit.log file contains IDS server status.
  - <IIA install_directory>\deployment\logs – remote deployment log directory. The ibmnsi.log file contains the complete command the deployment wizard uses to launch the last Java wrapper program. The IDS10silent.log file contains IDS installation specific information and IDS10WinExit.log file contains IDS server status.
- To enable a support framework trace and a deployment wizard trace on the staging server, in the deployment wizard select **Edit > Preferences > Diagnostic Trace.** Check the trace options you want to enable. Based on your selection, IRU_SupportTrace.log or IRU_DebugTrace.log in <ER install_directory>\Runtime21\SolutionEnabler\logs will be created. The IRU_SupportTrace.log contains specific trace messages from application

wrapper user programs. The IRU_DebugTrace.log contains specific trace messages from the Express Runtime framework classes.

- To enable a trace on a remote computer, start the agent with trace options: `<IIA install_directory>\IIAJRE\bin\java -jar DJT_ibmnsit.jar -task intallationAgent -enableSupportFrameworkTrace –enableSolutionDeployerTrace`. To run the trace-enabled agent in debugging mode, add the `–leavefiles` option to the end of the above command.
- Use the following invocation to provide debugging details and a trace. This provides information with respect to what arguments were passed and log file names where debug information is stored.

```
 C:\Program Files\IBM\Runtime21\SolutionEnabler>DJTJRE\bin\java -jar
DJT_ibmnsit.jar -task deployer -leavefiles -solutionFileName TestIDS.ser
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
args 0 = C:\Program Files\IBM\Runtime21\SolutionEnabler\DJTJRE\bin\javaw
args 1 = -Xrs
args 2 = -Xdebug
args 3 = -Xrunjdwp:transport=dt_socket
args 4 = -classpath
args 5 =
\C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\externalSupportJars/IDS10_Win/IRU_Suppor
t.jar;.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\R
UNTIM~1\SOLUTI~1\;C:\iru
args 6 = com.ibm.iru.ids10win.IDS10WinPDC

Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
args 0 = C:\Program Files\IBM\Runtime21\SolutionEnabler\DJTJRE\bin\javaw
args 1 = -Xrs
args 2 = -Xdebug
args 3 = -Xrunjdwp:transport=dt_socket
```

```
args 4 = -classpath
args 5 =
C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\externalSupportJars/IDS10_Win/IRU_Support
.jar;.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RU
NTIM~1\SOLUTI~1\;C:\iru
args 6 = com.ibm.iru.ids10win.IDS10WinMain
args 7 =
C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DEPLOY~1\logs\IDS10_Win4silent.ini
args 8 = C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DEPLOY~1\logs\IDS10silent.log

Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
Class path:
.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RUNTIM~
1\SOLUTI~1\;C:\iru
args 0 = C:\Program Files\IBM\Runtime21\SolutionEnabler\DJTJRE\bin\javaw
args 1 = -Xrs
args 2 = -Xdebug
args 3 = -Xrunjdwp:transport=dt_socket
args 4 = -classpath
args 5 =
C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\externalSupportJars/IDS10_Win/IRU_Support
.jar;.;C:\PROGRA~1\IBM\RUNTIM~1\SOLUTI~1\DJT_ibmnsit.jar;C:\PROGRA~1\IBM\RU
NTIM~1\SOLUTI~1\;C:\iru
args 6 = com.ibm.iru.ids10win.IDS10WinExit

IRU06172: Task 1 in solution TestIDS.ser deployed successfully.
IRU00097: See logfile IRU_DeploymentWizard.log for more details.
```

Configuring IDS 10.00 to work with WebSphere Application Server

This section provides basic information about configuring IDS 10.00 with IBM WebSphere Application Server 6.0. For more information, see the IBM Redbook "Using Informix Dynamic Server with WebSphere" - SG24-6948-00. You can download the redbook free of charge at http://www.redbooks.ibm.com/abstracts/sg246948.html?Open.

The IDS application component described in this paper places the IDS server and the IBM Informix JDBC drivers into the same installation location provided in the deployment wizard panel. If IDS and WebSphere Application Server were deployed to the same computer, use the ifxjdbcx.jar file from the IDS location.

If deployment was done to more than one computer, download the JAR file from the Web site mentioned in the Redbook under JDK and JDBC installation. The reference site is http://www-3.ibm.com/software/data/informix/. Copy it to the computer that runs WebSphere Application Server. The objective is to get the ifxjdbcx.jar file to a local file system on the WebSphere Application Server computer. Then use the JAR file with the following instructions:

1. Use the JACL code included with the IRU2_1SampleWin application project in the Express Runtime developer as a starting point.

2. Copy the IRU_WebSphereConfigProcs.jacl into the IRU2_1SampleWin application project.

3. During copy process a new name will be prompted for the file. Rename it to IDS_WebSphereConfigProcs.jacl.

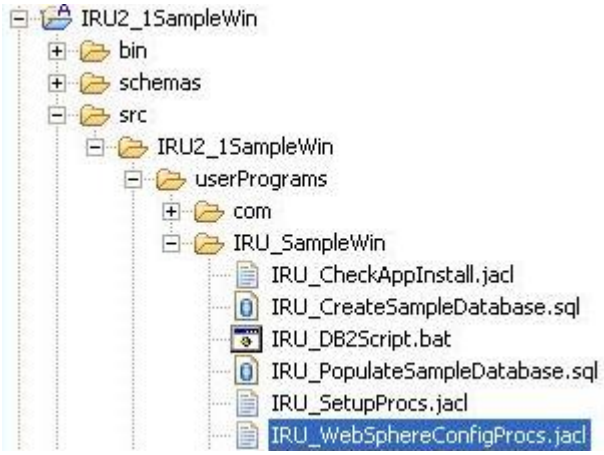

Figure 42. JACL procedure files preinstalled in your workspace

4. Make the following changes to the function that creates a JDBCProvider in the file IDS_WebSphereConfigProcs.jacl.

```
proc createJDBCProviderUsingTemplate {provName classPath nodeName serverName }
{
    puts "\nConfigProcs: createJDBCProviderUsingTemplate $provName $classPath
$nodeName $serverName"
    global AdminConfig  ;# Access the AdminConfig command

    if {[file exists $classPath]} {
        # get the path name upto the driver to use for
INFORMIX_JDBC_DRIVER_PATH
        set db2jdbcdriverpath [file dirname $classPath]
        # set the WebSphere environment variable INFORMIX_JDBC_DRIVER_PATH at
node scope
        updateVariableMap INFORMIX_JDBC_DRIVER_PATH $db2jdbcdriverpath
$serverName $nodeName
    } else {
        return -code error "Could not find the JDBC driver at the location
provided $classPath"
    }

    set parent [$AdminConfig getid /Node:$nodeName/]
    set pname_attr [list name $provName]
    set attrList [list $pname_attr]
    set templ [$AdminConfig listTemplates JDBCProvider "Informix JDBC Driver
(XA)("]
    #$AdminConfig createUsingTemplate JDBCProvider $parent {{name $provName}}
$templ
    puts " "
    puts "ConfigProcs: AdminConfig createUsingTemplate JDBCProvider $parent
$attrList $templ"
    puts " "
    $AdminConfig createUsingTemplate JDBCProvider $parent $attrList $templ

    # Create by template automatically creates a 4.0 and 5.0 DS with no name
    # We could modify them to suit us but it adds to the complexity of the
main script
    # so just delete the extra template datasources
    # ignore any errors

    catch [set dsToDelete [$AdminConfig getid
/JDBCProvider:$provName/DataSource:/]]
    catch [$AdminConfig remove $dsToDelete]

    catch [set dsToDelete [$AdminConfig getid
/JDBCProvider:$provName/WAS40DataSource:/]]
    catch [$AdminConfig remove $dsToDelete]

    puts "Please remember to issue a \$AdminConfig save if you wish to save
this configuration change"

}
```

5. Make the following changes to the function that creates a DB2 UDB Express DataSource.

```
 proc createDB2DataSource {bn serv JDBCProvName dsName dbName dbServerName
dbServerPort authAlias authMech
    cat desc IDSServer IDSHostname IDSportNumber } {
    global AdminConfig

    set jdbcProvId [$AdminConfig getid /JDBCProvider:$JDBCProvName/]
    set dsHelper "com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper"

    #-------------------------------------------------------
    # Set up the properties for a DB2 DataSource, set to Type 4
    #-------------------------------------------------------
    set dbservername_attr [list [list name serverName] [list value
$dbServerName] [list type java.lang.String] [list required false] [list
description "The DB2 server name"]]
    set dbserverport_attr [list [list name portNumber] [list value
$dbServerPort] [list type java.lang.String] [list required false] [list
description "The listening port of the DB2 server"]]
    set dstype_attr [list [list name driverType] [list value 4] [list type
java.lang.String] [list required true] [list description "Set to a Type4
driver"]]
    set dbname_attr [list [list name databaseName] [list value $dbName] [list
type java.lang.String] [list required true] [list description "The DB2
database name"]]
    set serverName_attr [list [list name serverName] [list value $IDSServer]
[list type java.lang.String] [list required true] [list description "The
informix server name e.g. ol_server"]]
    set ifxIFXHOST_attr [list [list name ifxIFXHOST] [list value $IDSHostname]
[list type java.lang.String] [list required true] [list description "DNS name
of IDS machine"]]
    set portNumber_attr [list [list name portNumber] [list value
$IDSportNumber] [list type java.lang.String] [list required true] [list
description "port number for IDS JDBC connection"]]
    set newdbprops [list $dbname_attr $serverName_attr $ifxIFXHOST_attr
$portNumber_attr]
    set resprops [list resourceProperties $newdbprops]
    set dsProp_attrs [list propertySet [list $resprops]]
    #-------------------------------------------------------
    # Set up the attributes for a connection pool
    #-------------------------------------------------------
```

If the IBM WebSphere Application Server Express product is installed, test the IDS_WebSphereConfigProcs.jacl file by copying it to the C:\Program Files\IBM\WebSphere\AppServer\bin directory. Invoke the following commands in a command prompt window. Replace the node name `sundar-portNode01` with the node information from `wsadmin` command output.

```
C:\Program Files\IBM\WebSphere\AppServer\bin>wsadmin
WASX7209I: Connected to process "server1" on node sundar-portNode01 using SOAP
connector;  The type of process is: UnManagedProcess
WASX7029I: For help, enter: "$Help help"
wsadmin>source IDS_WebSphereConfigProcs.jacl

wsadmin>createJDBCProviderUsingTemplate IDSProvider
c:/INFORMIX/lib/ifxjdbcx.jar sundar-portNode01 server1
ConfigProc: updateVariableMap INFORMIX_JDBC_DRIVER_PATH c:/INFORMIX/lib
server1 sundar-portNode01
Setting INFORMIX_JDBC_DRIVER_PATH to c:/INFORMIX/lib

wsadmin> AdminConfig createUsingTemplate JDBCProvider sundar-
portNode01(cells/sundar-portNode01Cell/nodes/sundar-
portNode01|node.xml#Node_1) {name IDSProvider} "Informix JDBC Driver
(XA)(templates/system|jdbc-resource-provider-
templates.xml#JDBCProvider_Informix_2)"

Please remember to issue a $AdminConfig save if you wish to save this
configuration change

wsadmin>createDB2DataSource sundar-portNode01 server1 IDSProvider
myIDSDatasource SAMPLE ol_server 1526 myAuthAlias BASIC_PASSWORD myAPPds
"Create by myApplication" ol_server sundar-port.lenexa.ibm.com 1526

ConfigProcs: AdminConfig create DataSource IDSProvider(cells/sundar-
portNode01Cell/nodes/sundar-
portNode01|resources.xml#JDBCProvider_1109723106476) {name myIDSDatasource}
{jndiName jdbc/myIDSDatasource} {datasourceHelperClassname
com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper}
{authMechanismPreference BASIC_PASSWORD} {authDataAlias myAuthAlias}
{description {Create by myApplication}} {category myAPPds} {connectionPool
{{agedTimeout 0} {connectionTimeout 180} {maxConnections 10} {minConnections
1} {purgePolicy EntirePool} {reapTime 180} {unusedTimeout 1800}}}
{relationalResourceAdapter {"WebSphere Relational Resource
Adapter(cells/sundar-portNode01Cell/nodes/sundar-
portNode01|resources.xml#builtin_rra)"}} {statementCacheSize 10} {propertySet
{{resourceProperties {{{name databaseName} {value SAMPLE} {type
java.lang.String} {required true} {description {The DB2 database name}}}
{{name serverName} {value ol_server} {type java.lang.String} {required true}
{description {The informix server name e.g. ol_server}}} {{name ifxIFXHOST}
{value sundar-port.lenexa.ibm.com} {type java.lang.String} {requiredtrue}
{description {DNS name of IDS machine}}} {{name portNumber} {value 1526} {type
java.lang.String} {required true} {description {port number for IDS JDBC
connection}}}}}}

myIDSDatasource(cells/sundar-portNode01Cell/nodes/sundar-
portNode01|resources.xml#DataSource_1109723134841)
wsadmin>$AdminConfig save
```

Perform the following steps for verification:

1. Start the administrative console by selecting **IBM WebSphere > Application Server – Express V6 > Profiles > Default > Administrative Console**.
2. Log in with your name or a user ID.

3. Click on **Resources** in the left pane**.**
4. Click on **JDBC Providers**.
5. In the right pane, click **IDSProvider** from the list of providers.
6. Click **DataSources** below **Additional Properties**.
7. Click **myIDSDatasource** and check the parameters.

Summary

It is easy to package a new application into the delivery structure of the IBM Express Runtime product. Specifically this paper described how to package IDS 10.00 into a deployable component. The IBM Express Runtime program is your middleware deployment solution that pays for itself through productivity gains alone. It truly takes care of deployment issues and lets you concentrate on your application. Find out more by visiting http://www.ibm.com/partnerworld/expressruntime and http://www.ibm.com/software/data/informix/.

Appendixes

Appendix A application.axml

```
 <?xml version="1.0" ?>

<!--
Licensed Materials - Property of IBM


 US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.


      (C) Copyright IBM Corp. 2003, 2005

      The sample source code provided in this white paper is licensed to you
under the IBM International License Agreement for Non-Warranted Programs for
the IBM Express Runtime Sample Code for Application Wrappers, Solution
Wrappers and User Programs, which is found in the license.html file on the IBM
Informix Software: Portfolio Update and Future Directions CD and the IBM
Informix Software: IBM(R) Express Runtime Wrapper for IBM Informix(R) Dynamic
Server Workgroup Edition 10.00 CD.
      If you do not agree to the terms of this agreement, do not access or use
this code.
-->
<iru:application
  deploymentPackageProtected="false"
  id="IDS10_Win"
  xmlns:iru="http://www.ibm.com/xmlns/prod/iru/application"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/xmlns/prod/iru/application
IRU_application.xsd">


    <applicationInformation
      installTime="20"
      version="10.00.TC1">
        <name>IDS10_Win</name>
        <operatingSystems>
            <operatingSystem>Windows</operatingSystem>
        </operatingSystems>
        <license>Licensed Materials - Property of IBM

US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.

&#9;
&#9;(C) Copyright IBM Corp. 2003, 2005
&#9;
&#9;The sample source code provided in this white paper is licensed to you
under the IBM International License Agreement for Non-Warranted Programs for
```

```
the IBM Express Runtime Sample Code for Application Wrappers, Solution
Wrappers and User Programs, which is found in the license.html file on the IBM
Informix Software: Portfolio Update and Future Directions CD and the IBM
Informix Software: IBM&amp;#174; Express Runtime Wrapper for IBM Informix&amp;#174;
Dynamic Server Workgroup Edition 10.00 CD.
&amp;#9;If you do not agree to the terms of this agreement, do not access or use
this code.
</license>
        <providerName>IBM</providerName>
    </applicationInformation>


    <translationLanguages default="english">
        <language>english</language>
    </translationLanguages>


    <fileLists>
        <fileList id="softwareimagefiles">
            <file>IIF</file>
            <file>JDBC</file>
            <file>jre</file>
            <file>lap</file>
            <file>li</file>
        </fileList>
        <fileList
          id="userprogramfiles"
          userPrograms="true">
            <file>com/ibm/iru/ids10win/IDS10Constants.class</file>
            <file>com/ibm/iru/ids10win/IDS10WinExit.class</file>
            <file>com/ibm/iru/ids10win/IDS10WinMain.class</file>
            <file>com/ibm/iru/ids10win/IDS10WinPDC.class</file>
        </fileList>
    </fileLists>


    <preDeploymentChecker
      logFile="IDS10WinPDC.log"
      programName="com.ibm.iru.ids10win.IDS10WinPDC"
      successType="returnCode"
      timeout="5"
      type="java" />


    <mainProgram
      externalCommand="true"
      logFile="IDS10silent.log"
      programName="com.ibm.iru.ids10win.IDS10WinMain"
      responseFile="silent.ini"
      successType="returnCode"
      timeout="20"
      type="java">
        <arguments>
            <argument responseFile="true" />
            <argument logFile="true" />
        </arguments>
    </mainProgram>
```

```
<exitProgram
  logFile="IDS10WinExit.log"
  programName="com.ibm.iru.ids10win.IDS10WinExit"
  successType="returnCode"
  timeout="10"
  type="java" />


<variables>
    <stringVariable
      maximumLength="240"
      minimumLength="2"
      name="ACTUAL_INSTALL_DIR"
      required="true">
        <labelText translatedKey="ACTUAL_INSTALL_DIRLabel" />
        <defaultData>C:\INFORMIX</defaultData>
        <helpText translatedKey="ACTUAL_INSTALL_DIRHelp" />
        <inputValidation>
            <invalid>
                <characters>*?&amp;quot;;/|&amp;lt;&amp;gt;</characters>
                <substrings>
                    <substring>\\</substring>
                </substrings>
            </invalid>
            <valid>
                <prefixes>
                    <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>
        </inputValidation>
        <issFileAssociations>
            <issFileAssociation
```

```
                keyword="ACTUAL_INSTALL_DIR"
                responseFileName="silent.ini"
                section="PRODUCT_IDS" />
            <issFileAssociation
                keyword="INSTALL_DIR"
                responseFileName="silent.ini"
                section="BUNDLE_COMMON" />
        </issFileAssociations>
    </stringVariable>
    <stringVariable
      lowerCase="true"
      maximumLength="128"
      minimumLength="3"
      name="DBSERVERNAME"
      required="true"
      upperCase="false">
        <labelText translatedKey="DBSERVERNAMELabel" />
        <helpText translatedKey="DBSERVERNAMEHelp" />
        <issFileAssociations>
            <issFileAssociation
              keyword="DBSERVERNAME"
              responseFileName="silent.ini"
              section="server instance" />
        </issFileAssociations>
        <defaultData>ol_server</defaultData>
        <inputValidation>
            <valid>
                <prefixes>
                    <prefix ignoreCase="false">ol_</prefix>
                </prefixes>
            </valid>
            <invalid>
                <characters>@; -</characters>
            </invalid>
        </inputValidation>
    </stringVariable>
    <stringVariable
      minimumLength="1"
      name="SERVERNUM"
      required="true">
        <labelText translatedKey="SERVERNUMLabel" />
        <helpText translatedKey="SERVERNUMHelp" />
        <issFileAssociations>
            <issFileAssociation
              keyword="SERVERNUM"
              responseFileName="silent.ini"
              section="server instance" />
        </issFileAssociations>
        <defaultData>0</defaultData>
        <inputValidation>
            <valid>
                <ranges>
                    <range>0 to 255</range>
                </ranges>
            </valid>
        </inputValidation>
    </stringVariable>
```

```
        <passwordVariable
          advanced="false"
          maximumLength="20"
          minimumLength="2"
          name="Account_Passwd"
          required="true">
            <labelText translatedKey="Account_PasswdLabel" />
            <helpText translatedKey="Account_PasswdHelp" />
            <issFileAssociations>
                <issFileAssociation
                  keyword="Account_Passwd"
                  responseFileName="silent.ini"
                  section="server install" />
            </issFileAssociations>
            <defaultData>informix</defaultData>
        </passwordVariable>
        <booleanVariable
          name="Typical"
          required="false">
            <labelText translatedKey="TypicalLabel" />
            <helpText translatedKey="TypicalHelp" />
            <issFileAssociations>
                <issFileAssociation
                  keyword="Typical"
                  responseFileName="silent.ini"
                  section="server install"
                  valueIfFalse="0"
                  valueIfTrue="1" />
                <issFileAssociation
                  keyword="Minimal"
                  responseFileName="silent.ini"
                  section="server install"
                  valueIfFalse="1"
                  valueIfTrue="0" />
            </issFileAssociations>
            <defaultData>true</defaultData>
        </booleanVariable>
        <booleanVariable
          name="Upgrade"
          required="false">
            <labelText translatedKey="UpgradeLabel" />
            <helpText translatedKey="UpgradeHelp" />
            <issFileAssociations>
                <issFileAssociation
                  keyword="Upgrade"
                  responseFileName="silent.ini"
                  section="server install"
                  valueIfFalse="0"
                  valueIfTrue="1" />
            </issFileAssociations>
            <defaultData>false</defaultData>
        </booleanVariable>
        <booleanVariable
          name="Reinstall"
          required="false">
            <labelText translatedKey="ReinstallLabel" />
            <helpText translatedKey="ReinstallHelp" />
```

```
            <issFileAssociations>
                <issFileAssociation
                  keyword="Reinstall"
                  responseFileName="silent.ini"
                  section="server install"
                  valueIfFalse="0"
                  valueIfTrue="1" />
            </issFileAssociations>
            <defaultData>false</defaultData>
        </booleanVariable>
        <booleanVariable
          name="Shutdown_Services"
          required="false">
            <labelText translatedKey="Shutdown_Services" />
            <helpText translatedKey="Shutdown_Services" />
            <issFileAssociations>
                <issFileAssociation
                  keyword="Shutdown_Services"
                  responseFileName="silent.ini"
                  section="server install"
                  valueIfFalse="0"
                  valueIfTrue="1" />
            </issFileAssociations>
            <defaultData>true</defaultData>
        </booleanVariable>
        <booleanVariable
          name="Service_Start_Auto"
          required="false">
            <labelText translatedKey="Service_Start_AutoLabel" />
            <helpText translatedKey="Service_Start_AutoHelp" />
            <issFileAssociations>
                <issFileAssociation
                  keyword="Service_Start_Auto"
                  responseFileName="silent.ini"
                  section="server instance"
                  valueIfFalse="0"
                  valueIfTrue="1" />
            </issFileAssociations>
            <defaultData>true</defaultData>
        </booleanVariable>
        <booleanVariable
          name="Configure_Instance"
          required="false">
            <labelText translatedKey="Configure_InstanceLabel" />
            <helpText translatedKey="Configure_InstanceHelp" />
            <issFileAssociations>
                <issFileAssociation
                  keyword="Configure_Instance"
                  responseFileName="silent.ini"
                  section="server install"
                  valueIfFalse="0"
                  valueIfTrue="1" />
            </issFileAssociations>
            <defaultData>true</defaultData>
        </booleanVariable>
        <booleanVariable
          name="Create_Icons"
```

```
          required="false">
            <labelText translatedKey="Create_IconsLabel" />
            <helpText translatedKey="Create_IconsHelp" />
            <issFileAssociations>
                <issFileAssociation
                  keyword="Create_Icons"
                  responseFileName="silent.ini"
                  section="server install"
                  valueIfFalse="0"
                  valueIfTrue="1" />
            </issFileAssociations>
            <defaultData>true</defaultData>
        </booleanVariable>
        <booleanVariable
          name="Initialize_Server"
          required="false">
            <labelText translatedKey="Initialize_ServerLabel" />
            <helpText translatedKey="Initialize_ServerHelp" />
            <issFileAssociations>
                <issFileAssociation
                  keyword="Initialize_Server"
                  responseFileName="silent.ini"
                  section="server instance"
                  valueIfFalse="0"
                  valueIfTrue="1" />
            </issFileAssociations>
            <defaultData>true</defaultData>
        </booleanVariable>
    </variables>


</iru:application>
```

Appendix B application_english.xml

```
 <?xml version="1.0" ?>
<!-- Licensed Materials - Property of IBM


 US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.


      (C) Copyright IBM Corp. 2003, 2005

      The sample source code provided in this white paper is licensed to you
under the IBM International License Agreement for Non-Warranted Programs for
the IBM Express Runtime Sample Code for Application Wrappers, Solution
Wrappers and User Programs, which is found in the license.html file on the IBM
Informix Software: Portfolio Update and Future Directions CD and the IBM
Informix Software: IBM(R) Express Runtime Wrapper for IBM Informix(R) Dynamic
Server Workgroup Edition 10.00 CD.
      If you do not agree to the terms of this agreement, do not access or use
this code. -->

<IDS10_Win>

      <name>Informix&#174; Dynamic Server</name>
      <providerName>IBM</providerName>
      <buildHelp>Fill in the location where the CD or unzipped install image
is located. (e.g. C:\Download\IDS10.00)</buildHelp>
      <buildPrompt>Specify the fully qualified path name where the IBM
Informix&#174; Dynamic Server install image is located.</buildPrompt>
      <configureText>Provide/Verify the following information and click Next
or OK to proceed. Click Field Help for additional information about the
fields. NOTE: Required fields are marked with *</configureText>

    <Account_PasswdHelp>Enter the password for the informix
user.</Account_PasswdHelp>
    <Account_PasswdLabel>Account Password</Account_PasswdLabel>

    <ACTUAL_INSTALL_DIRHelp>The directory where you plan to install the
product files</ACTUAL_INSTALL_DIRHelp>
    <ACTUAL_INSTALL_DIRLabel>Install Directory</ACTUAL_INSTALL_DIRLabel>

    <Configure_InstanceHelp>Check to create a new instance of the database
server. If unchecked, an instance is not configured in the case of a fresh
install.</Configure_InstanceHelp>
    <Configure_InstanceLabel>Configure an instance ?</Configure_InstanceLabel>

    <Create_IconsHelp>Creates icons for the installed programs. Uncheck to
disable icon creation.</Create_IconsHelp>
    <Create_IconsLabel>Create Icons in the start menu ?</Create_IconsLabel>

    <DBSERVERNAMEHelp>If you do not specify a name, a default name of
ol_server is assigned. You cannot change the database server name after it is
installed.</DBSERVERNAMEHelp>
    <DBSERVERNAMELabel>Name of the database server</DBSERVERNAMELabel>
```

```
    <Initialize_ServerHelp>Whether the database server is to be initialized
after creation. Check for the program to attempt to initialize the database
server. Otherwise you must perform the initialization
manually.</Initialize_ServerHelp>
    <Initialize_ServerLabel>Initialize Server</Initialize_ServerLabel>

    <ReinstallHelp>You can re-install the same version but not a older
version.</ReinstallHelp>
    <ReinstallLabel>Reinstall</ReinstallLabel>

    <SERVERNUMHelp>Specify a server number that is unique across all instances
of the database server configured on the computer. Make the value as small as
possible. You can use the default server number only for the first instance
that you are  configuring. This parameter sets the corresponding parameter in
the onconfig file, which contains the configuration parameters for the
database server.</SERVERNUMHelp>
    <SERVERNUMLabel>Server Number</SERVERNUMLabel>

    <Service_Start_AutoHelp>Check to set up the database server to start
automatically on reboot. If you do not check this option, you must start the
database server manually on every reboot.</Service_Start_AutoHelp>
    <Service_Start_AutoLabel>IDS service autostarts</Service_Start_AutoLabel>

    <Shutdown_Services>Check to shut down the services that must be shut down
for the installation to proceed. These services include any services that the
computer must upgrade, such as the database server and other services like
OnSNMP.</Shutdown_Services>
    <Shutdown_Services>Shutdown services prior to install</Shutdown_Services>

    <TypicalHelp>Uncheck to do a minimal install</TypicalHelp>
    <TypicalLabel>Typical Install</TypicalLabel>

    <UpgradeHelp>Check this if you already have an older version of IDS
installed and wish to upgrade it to IDS 10.0</UpgradeHelp>
    <UpgradeLabel>Upgrade Install</UpgradeLabel>

</IDS10_Win>
```

Appendix C IDS10WinPDC.java

```
/* Licensed Materials - Property of IBM


US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.


      (C) Copyright IBM Corp. 2003, 2005

      The sample source code provided in this white paper is licensed to you
under the IBM International License Agreement for Non-Warranted Programs for
the IBM Express Runtime Sample Code for Application Wrappers, Solution
Wrappers and User Programs, which is found in the license.html file on the IBM
Informix Software: Portfolio Update and Future Directions CD and the IBM
Informix Software: IBM® Express Runtime Wrapper for IBM Informix® Dynamic
Server Workgroup Edition 10.00 CD.
      If you do not agree to the terms of this agreement, do not access or use
this code.
*/
package com.ibm.iru.ids10win;

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

/**
 * This class reports to the deployment whether to continue with the
installation.
 * It checks whether IDS is already installed on the target.
 * If it is then it returns an exit code of IDS10Constants.EXISTS to skip the
installation
 * otherwise returns IDS10Constants.DOESNOTEXIST to continue with the
installation
 *
 * @author Administrator
 */
public class IDS10WinPDC extends SupportWindowsBase {

      private static final String copyright0 = "Licensed Materials - Property
of IBM";
      private static final String copyright1 = "5724-J10";
      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 SupportWindowsHelper ivHelper;

      /**
       * ctor
       */
      public IDS10WinPDC() {
            super();
            ivHelper = getWindowsHelper();
            // this log file must match the one set for the PDC in the
```

```
application.axml
            // it is here so we can write to it
            // it is in application.axml so deployer can display it.
            // we do not log anything in this class. Code is here to
facilitate any later addition of logging.
            setLogFileName("IDS10WinPDC.log");
            // we do need to set the message bundle and JAR file for the
resource strings
            setMainResources(IDS10Constants.ivRuntimeMessagesClassString);
            setJarFile(ivHelper.getProductInstallingId(this));
      }

      public static void main(String[] args) {

            IDS10WinPDC checker = new IDS10WinPDC();
            if (checker.checkVersion() == IDS10Constants.EXISTS) {
                  System.exit(IDS10Constants.EXISTS);
            }

            System.exit(IDS10Constants.DOESNOTEXIST);

      }

      /**
       * This method checks the Windows registry for IDS
       * @return EXISTS if IDS is found else DOESNOTEXIST
       */
      public int checkVersion() {
            int rc = IDS10Constants.DOESNOTEXIST;

            // The programming style used is to set the search value and then
invoke a method
            // instead of passing in the search value as an argument

            // e.g. set the key we want to look for in the base class
            setRegistryKey(IDS10Constants.HKEY_LOCAL_MACHINE);
            setRegistrySubKey(IDS10Constants.ivIDSRegistryKeyString);

            // now call the method to look for the key and pass a reference in
the call
            if (ivHelper.doesRegKeyExist(this)) {
                  // return the value to terminate install
                  rc = IDS10Constants.EXISTS;
            } else {
                  rc = IDS10Constants.DOESNOTEXIST;
            }
            return rc;
      }

}
```

Appendix D IDS10WinMain.java

```
package com.ibm.iru.ids10win;
/* Licensed Materials - Property of IBM


US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.


      (C) Copyright IBM Corp. 2004, 2005

      The sample source code provided in this white paper is licensed to you
under the IBM International License Agreement for Non-Warranted Programs for
the IBM Express Runtime Sample Code for Application Wrappers, Solution
Wrappers and User Programs, which is found in the license.html file on the IBM
Informix Software: Portfolio Update and Future Directions CD and the IBM
Informix Software: IBM® Express Runtime Wrapper for IBM Informix® Dynamic
Server Workgroup Edition 10.00 CD.
      If you do not agree to the terms of this agreement, do not access or use
this code.
*/

import com.ibm.iru.message.NLSKeys;
import com.ibm.jsdt.support.SupportWindowsBase;
import com.ibm.jsdt.support.SupportWindowsHelper;


public class IDS10WinMain extends SupportWindowsBase {
      private static final String copyright0 = "Licensed Materials - Property
of IBM";
      private static final String copyright1 = "5724-J10";
      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 SupportWindowsHelper ivHelper;
      /**
       * ctor
       */
      public IDS10WinMain() {
            ivHelper = getWindowsHelper();

            // we do need to set the message bundle and jar file for the
resource strings
            setMainResources(IDS10Constants.ivRuntimeMessagesClassString);

      }

      public static void main(String[] args) {

            IDS10WinMain idsMain = new IDS10WinMain();
            if (idsMain.setFiles(args) == IDS10Constants.FAIL){
                  System.exit(IDS10Constants.FAIL);
            }
```

```
            if (idsMain.install() == IDS10Constants.FAIL) {
                  System.exit(IDS10Constants.FAIL);
            }

            // report all clear
            System.exit(0);

      }

      private int setFiles(String[] args){
        if (args.length != 2) {
            setMessage(getResourceString(NLSKeys.BAD_NUMBER_PGM_ARGS, new
String []{"IDS10WinMain", "1", Integer.toString(args.length-1)}));
            ivHelper.log(this);
           return IDS10Constants.FAIL;
        }
        // Assign response and log files
        setResponseFileName(args[0]);
        setLogFileName(args[1]);
        return IDS10Constants.OK;
      }


      private int install(){
            int rc = IDS10Constants.OK;
            String unpackedDir = ivHelper.getUnpackedDir(this);
        try {
            String cmd = "cmd /C " + "setup.exe -s " + getResponseFileName() +
" -l " + getLogFileName();
                  setCommand(cmd);
                  setCommandLaunchDirectory( unpackedDir  + "IIF");
                  setMessage(getResourceString(NLSKeys.CMDINVOKED,
getCommand()));
                  ivHelper.log(this);
                  String output = ivHelper.getSystemCommandOutput(this);

                  if ((output.indexOf("A fatal error occurred during the
installation!") == -1) ||
                        (output.indexOf("IBM Informix Dynamic Server has been
successfully installed on your system.") != -1)) {
                        setMessage(getResourceString(NLSKeys.CMD_SUCCESS));
                        rc = IDS10Constants.OK;
                  } else {
                        setMessage(getResourceString(NLSKeys.CMD_FAIL,
output));
                        rc = IDS10Constants.FAIL;
                  }
                  setMessage(output);
                  ivHelper.log(this);
        } catch (Exception e) {
            rc = IDS10Constants.FAIL;
            setMessage(getResourceString(NLSKeys.CMD_EXCEPTION,
e.toString()));
            ivHelper.log(this);
        }
            return rc;
```

```
        }
}
```

Appendix E IDS10WinExit.java

```
/* Licensed Materials - Property of IBM


US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.


      (C) Copyright IBM Corp. 2003, 2005

      The sample source code provided in this white paper is licensed to you
under the IBM International License Agreement for Non-Warranted Programs for
the IBM Express Runtime Sample Code for Application Wrappers, Solution
Wrappers and User Programs, which is found in the license.html file on the IBM
Informix Software: Portfolio Update and Future Directions CD and the IBM
Informix Software: IBM® Express Runtime Wrapper for IBM Informix® Dynamic
Server Workgroup Edition 10.00 CD.
      If you do not agree to the terms of this agreement, do not access or use
this code.
*/
package com.ibm.iru.ids10win;


import com.ibm.iru.message.NLSKeys;
import com.ibm.jsdt.support.SupportWindowsBase;
import com.ibm.jsdt.support.SupportWindowsHelper;

/**
 * This class performs post install functions e.g. installing the JDBC driver
and
 * verifying that the server is up and working
 *
 * @author Administrator
 *
 */

public class IDS10WinExit extends SupportWindowsBase {

      private static final String copyright0 = "Licensed Materials - Property
of IBM";
      private static final String copyright1 = "5724-J10";
      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 SupportWindowsHelper ivHelper;

      /**
       * ctor
       */
      public IDS10WinExit() {
            super();
            ivHelper = getWindowsHelper();
            // this log file must match the one set for the PDC in the
```

```
application.axml
            // it is here so we can write to it
            // it is in application.axml so deployer can display it.
            setLogFileName("IDS10WinExit.log");
            // we do need to set the message bundle and jar file for the
resource strings
            setMainResources(IDS10Constants.ivRuntimeMessagesClassString);
            setJarFile(ivHelper.getProductInstallingId(this));
      }


      public static void main(String[] args) {

            if (new IDS10WinPDC().checkVersion() ==
IDS10Constants.DOESNOTEXIST) {
                  // big problem, no registry entries found.
                  // IDS install did not work
                  System.exit(IDS10Constants.FAIL);
            }
            IDS10WinExit idsExit = new IDS10WinExit();

            if (idsExit.installJDBC() == IDS10Constants.FAIL) {
                  System.exit(IDS10Constants.FAIL);
            }

            if (idsExit.verifyInstallation() == IDS10Constants.FAIL) {
                  System.exit(IDS10Constants.FAIL);
            }

            // report all clear
            System.exit(0);

      }

      /**
       * Retrieve the bin directory for the WAS installation from the registry
       * @return String - The location of the installed WAS bin directory
       */
      private String getIDS_HOME() {
            setRegistryKey(IDS10Constants.HKEY_LOCAL_MACHINE);
            setRegistrySubKey(IDS10Constants.ivIDSRegistryKeyString);
            if (getWindowsHelper().doesRegKeyExist(this)) {
                  setRegistryStringValue("Install_Dir");
                  // we want a path like C:\PROGRA~1\....not "C:\Program
files"
                  setPath(getWindowsHelper().getRegistryValue(this));
                  return ivHelper.getWindowsShortPath(this);
            } else {
                  return "";
            }
      }

      /**
       * Runs an install verification test provided by IDS. Scans return code
from the batch file
       * to determine status of test.
       * @return OK if it works else FAIL
```

```
        */
       private int verifyInstallation() {
             int rc = 0;
             // get the server name from the modified response file.
             setVariableName ("DBSERVERNAME");
             String serverName = ivHelper.getIbmNsiPropValue(this);

             // cmd /c "C:\PROGRA~1\Informix\ol_dsra && onstat"
             setCommand("cmd /c \"" + getIDS_HOME() + sSLASH + serverName + "
&& onstat" + "\"");
             setMessage(getResourceString(NLSKeys.CMDINVOKED, getCommand()));
             ivHelper.log(this);

             // Starting server
             String output = ivHelper.getSystemCommandOutput(this);
             if (output.indexOf("-- On-Line -- Up") != -1) {
                   setMessage(getResourceString(NLSKeys.CMD_SUCCESS));
                   rc = IDS10Constants.OK;
             } else {
                   setMessage(getResourceString(NLSKeys.CMD_FAIL, output));
                   rc = IDS10Constants.FAIL;
             }

             ivHelper.log(this);
             setMessage(output);
             ivHelper.log(this);
             return rc;
       }

       /**
        * This method places the JDBC drivers needed by WAS to access Informix
        * @return OK if it works else FAIL
       */
       private int installJDBC() {
             int rc = 0;
             String getUnpackedDir = ivHelper.getUnpackedDir(this);

             // we want to execute a program like this
             // we will ignore the log file, the # will echo to stdout and we
will log stdout
             // the @err is to get notification of error messages only.
             //C:\>"C:\Program
Files\IBM\Runtime\SolutionEnabler\DJTJRE\bin\java" -jar
             //JDBC\2_21\setup.jar -silent -P
product.installLocation=D:\INFORMIX -log # !jnk
             //.log @err
             setCommand(System.getProperty("java.home") + "/bin/javaw -jar " +
getUnpackedDir + "JDBC/setup.jar -silent -P product.installLocation=" +
getIDS_HOME() + " -log # !jdbcinstall.log @err");

             setMessage(getResourceString(NLSKeys.CMDINVOKED, getCommand()));
             ivHelper.log(this);

             // Start JDBC install
             String output = ivHelper.getSystemCommandOutput(this);
             // no messages means success
             if (output.length() == 0) {
```

```
                    setMessage(getResourceString(NLSKeys.CMD_SUCCESS));
                    rc = IDS10Constants.OK;
              } else {
                    setMessage(getResourceString(NLSKeys.CMD_FAIL, output));
                    rc = IDS10Constants.FAIL;
              }

              ivHelper.log(this);
              setMessage(output);
              ivHelper.log(this);
              return rc;
        }
}
```

Appendix F IDS10Constants.java

```
/* Licensed Materials - Property of IBM


US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.


      (C) Copyright IBM Corp. 2003, 2005

      The sample source code provided in this white paper is licensed to you
under the IBM International License Agreement for Non-Warranted Programs for
the IBM Express Runtime Sample Code for Application Wrappers, Solution
Wrappers and User Programs, which is found in the license.html file on the IBM
Informix Software: Portfolio Update and Future Directions CD and the IBM
Informix Software: IBM® Express Runtime Wrapper for IBM Informix® Dynamic
Server Workgroup Edition 10.00 CD.
      If you do not agree to the terms of this agreement, do not access or use
this code.
*/
package com.ibm.iru.ids10win;

/**
 * @author Administrator
 *
 */
public class IDS10Constants {
      public static final int HKEY_LOCAL_MACHINE = 1;
      public static final int DOESNOTEXIST = 0;
      public static final int EXISTS = 1;
      public static final int OK = 31275; // random number
      public static final int FAIL = -1;

      public static final String ivRuntimeMessagesClassString =
"com.ibm.iru.message.MessagesNLS";
      public static final String ivIDSRegistryKeyString =
"SOFTWARE\\Informix\\DBMS\\10.00";

}
```

REFERENCES

[1] IBM Informix Dynamic Server Installation Guide for Microsoft Windows, Version 10.00 (G251-2288-00). Performing a Silent Installation. Go to http://publib.boulder.ibm.com/infocenter/ids9help/index.jsp and navigate to **Getting Started > Installation Guide for Microsoft Windows > Appendix A. Setting Up Specialized Installations**. Click on Performing a Silent Installation for details.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank the following people for their contributions to this document:

- **Srini Bhagavan, Amit Dandekar, Rajesh Nair and Srinivas Peddi** for providing information, validation and help related to this document.
- **Sahdev P Zala** for invaluable technical help to make this wrapper possible.
- **Paul A Smith** for following all the steps laid out in this document and verifying the accuracy of the instructions in this paper.
- **Kelley Greeson** for reviewing the document.

ABOUT THE AUTHOR

Sundar Shunmugam is a developer on the IBM Informix Development team in Lenexa, Kansas. He has eight years of experience mainly with the IBM Informix Dynamic Server and Client products. He is also an Informix Certified Professional. Sundar can be contacted at sundars@us.ibm.com.

IBM United States of America

Produced in the United States of America