JD Edwards EnterpriseOne

Deployment Server Reference Guide for IBM i

9.2

JD Edwards EnterpriseOne
Deployment Server Reference Guide for IBM i

9.2

Part Number: E55540-10

Contents

Preface

Welcome to the JD Edwards EnterpriseOne documentation.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at *http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc* .

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info* or visit *http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs* if you are hearing impaired.

Related Information

For additional information about JD Edwards EnterpriseOne applications, features, content, and training, visit the JD Edwards EnterpriseOne pages on the JD Edwards Resource Library located at:

http://learnjde.com

Conventions

The following text conventions are used in this document:

Convention	Meaning
Bold	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
Italics	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`Monospace`	Monospace type indicates commands within a paragraph, URLs, code examples, text that appears on a screen, or text that you enter.
> Oracle by Example	Indicates a link to an Oracle by Example (OBE). OBEs provide hands-on, step- by-step instructions, including screen captures that guide you through a process using your own environment. Access to OBEs requires a valid Oracle account.

1 Introduction

Understanding This Guide

This guide describes the JD Edwards EnterpriseOne Deployment Server with Tools Release 9.2 and Applications Release 9.x. This *Deployment Server Reference Guide* supersedes the guides from previous JD Edwards EnterpriseOne tools releases known as the *Tools Release Reference Guide*, and before that the *Tools Release Installation Guide*. All information in this guide is relevant to actions unique to and performed on the Deployment Server. This includes setup of the Development Client, which is deployed from the Deployment Server.

You should always check the Oracle Technology Network for revisions to this guide subsequent to the initial release, which coincides with the General Availability of *JD Edwards EnterpriseOne Tools Release 9.2* .

> **Note:** The JD Edwards Applications Release 9.2 is installed on the Deployment Server using a separate installer. Refer to "Chapter 3, Installing the Deployment Server," in the *JD Edwards EnterpriseOne Applications Release 9.2 Installation Guides*. To access the guides, refer to the Installation and Upgrade Documentation Library at this link:
> *http://docs.oracle.com/cd/E24902_01/nav/installation.htm*

Understanding the Tools Release on the Deployment Server

Although the process to install the Deployment Server itself remains via an Oracle Universal Installer (OUI), for Release 9.2, you must use Server Manager to install the latest tools release on the Deployment Server. The following is the list of procedures that are documented in the *JD Edwards EnterpriseOne Server Manager Guide* , which is located in the *Installation* and *Upgrade* libraries for JD Edwards EnterpriseOne on the Oracle Technology Network (OTN) at this link:

http://docs.oracle.com/cd/E61420_01/index.htm

> **Tip:** The links to the Server Manager Guide that are provided in the below steps are "live" only in the HTML version of this guide.

1. Install a Server Manager Management Agent on the Deployment Server. This should be the 32-bit Microsoft Windows version of the JD Edwards EnterpriseOne Management Agent.

   Refer to this chapter in the Server Manager Guide:

   Install a Management Agent
2. Register the Deployment Server with the Server Manager Console.

   Refer to this chapter in the Server Manager Guide:

   Register a JD Edwards Deployment Server as a New Managed Instance

3. Obtain and deploy the Tools Release software component for the Deployment Server.

   Refer to this chapter in the Server Manager Guide:

   Managed Software Components

Understanding Server Manager and This Guide

> **Note:** Management of *JD Edwards EnterpriseOne* Tools Release 8.96 or earlier must be done using the tools provided with those releases. *Server Manager* cannot be used to manage any *JD Edwards EnterpriseOne* tools release prior to 8.97.

For more information, see the *JD Edwards EnterpriseOne Tools Server Manager Guide* :

The JD Edwards EnterpriseOne *Server Manager* tool provides:

- Web Based System Management

  You can securely access and manage your *JD Edwards EnterpriseOne* installation from anywhere using a standard web browser.
- Remote Deployment and Upgrades

  You can install, uninstall, and update your *JD Edwards EnterpriseOne* servers regardless of their physical location or platform.
- Remote Operational Control

  You can start and stop any of your *JD Edwards EnterpriseOne* servers, Oracle J2EE application servers, or supported third party J2EE application servers directly from the *Management Console* .
- Secure Administrative Tasks

  Server Manager permits you to specify which existing *JD Edwards EnterpriseOne* users have access to the *Management Console* , control which *JD Edwards EnterpriseOne* servers the user may view, and specify which administrative tasks the user may perform on those servers.
- Configuration Management

  Server Manager provides a web-based interface for managing the configuration of all managed servers. The application presents each configuration item along with integrated help describing the configuration setting.

  > **Note:** Beginning with the availability of *Server Manager* , it is strongly advised that all changes to configuration files (such as jde.ini, jas.ini, jdbj.ini, jdelog.properties, etc.) for any *JD Edwards EnterpriseOne* server managed by *Server Manager* be accomplished using only the *Management Console* interface of *Server Manager* . In addition to providing usability improvements, using Server Manager reduces the risk of introducing configuration errors by providing dropdowns that contain only valid values where applicable. Further, the tool provides a useful *#unique_6/unique_6_Connect_42_BHAICJHF* for any modifications made to configurations using *Server Manager* .

- Configuration Comparison

  You can use *Server Manager* to compare the configuration of two or more servers to identify configuration differences. You can compare configurations through the *Management Console* application regardless of the

platform or location of the actual *JD Edwards EnterpriseOne* server. You can also compare individual servers with the default configuration of the corresponding server groups to which the servers belong.

- Audit History

  Server Manager maintains a history of changes made to the managed servers. This includes a history of each configuration change, each server start and stop, and each tools release update, including the user that performed the change or operation. The *Management Console* application provides mechanisms to query and view the audit history that is maintained.

- Integrated EnterpriseOne Software Management

  Use *Server Manager* to centrally maintain all your *JD Edwards EnterpriseOne* server tools releases, including the ability to copy the software to the remote server machines.

- Logical Server Grouping

  Server Manager allows you to group servers with a similar purpose. These groups can include any of the server types such as Enterprise Server, HTML Web Server , and so on. A default, or template, configuration is maintained for each server group.

- Application Release Independence

  Server Manager is delivered with every *JD Edwards EnterpriseOne* Tools Release 8.97 and greater and is compatible with any supported *JD Edwards EnterpriseOne* application release beginning with Application Release 8.9 through the currently supported release. No electronic software updates (ESUs) are required to support *Server Manager* .

- Self-Contained Installation

  The installation of *Server Manager* delivers all components that are required by the *Management Console* application. There are no third party requirements regardless of your existing or intended middleware topology (for example, WebLogic Server, WebSphere Application Server, or no application server).

- Tools Release Independence

  Newer versions of the *Server Manager* application will continue to support the management of earlier tools releases back to the initial general availability release of *JD Edwards EnterpriseOne* Tools Release 8.97.

Understanding Certifications

Customers must conform to the supported platforms for the release as detailed in the Oracle JD Edwards EnterpriseOne Certifications. In addition, JD Edwards EnterpriseOne may integrate, interface, or work in conjunction with other Oracle products. Refer to the following link for cross-reference material in the Program Documentation for Program prerequisites and version cross-reference documents to assure compatibility of various Oracle products.

http:\\www.oracle.com\corporate\contracts\index.html

Refer to the Certifications tab on My Oracle Support and search for this product:

- **JD Edwards EnterpriseOne Deployment Server**

For additional information on using Certifications, refer to this document on My Oracle Support (***https://support.oracle.com***):

- **Certifications FAQ for JD Edwards EnterpriseOne [Article ID 1525328.1]**

https://support.oracle.com/epmos/faces/DocumentDisplay?id=745831.1

Microsoft Windows Security

When installing, uninstalling/deinstalling, or running any JD Edwards EnterpriseOne product on Microsoft Windows operating systems, be sure to follow the below guidelines. This includes saving or restoring or deleting snapshots of EnterpriseOne using the SnapShot program. Not following these guidelines may cause unexpected errors to occur.

1. **Registry and directory permissions**
   Be sure that the user account into which you are signing into Microsoft Windows is in the Administrators group or a similar group that provides permissions to write to and read from registry hives (for example, HKEY_LOCAL_MACHINE\SOFTWARE) and disk subdirectories (`c:\Windows`) that are restricted from standard Windows users.
2. **"Run as administrator"**
   When starting any JD Edwards Enterprise One executable such as Change assistant, ESU setup, EnterpriseOne, SnapShot, client install, you must right click on the executable and choose **Run as administrator**. Failing to do this might result in incomplete or failed processes.
   If a file that you need to edit and then save is located in a directory with access restricted by UAC, you will need to right-click on your text editor program's icon, select **Run as administrator**, and open the file from within the editor. If you do not follow those steps and Microsoft's User Access Control (UAC) is turned on, you may not be able to save the modified file.
3. **Remote Desktop connections**
   When connecting to a remote Microsoft Windows computer using Microsoft's Remote Desktop, run Remote Desktop with the "/admin" flag. Not running with this setting may cause subtle, hard-to-explain errors in the program. You can do this using these steps:
   a. Right-click on the Remote Desktop shortcut.
   b. Select properties.
   c. Click on the Shortcut tab.
   d. At the end of the Target field, add a space, a forward slash, and "admin" (without the quotes) to the end of the Target. The screen shot sample below illustrates this setting:

Minimizing Locked Files

To minimize the possibility that the Tools Release installer or SnapShot might fail due to locked files, follow these guidelines:

1. Exit from all running programs. You can use Microsoft Windows Task Manager to verify that the programs are stopped. In particular, be sure that EnterpriseOne, JDeveloper, and SQL Developer are stopped, but other programs may also need to be stopped. Exceptions to this rule include the database service(s) which should be running.
2. Verify that no files in the Deployment Server's installation directory or subdirectories are open in any file editors.

3. Verify that neither the installation directory nor any of its subdirectories are open in Microsoft Windows Explorer.
4. Verify that no Command Prompt window has as its current working directory either the Deployment Server's installation directory or any of its subdirectories.

2 Setting Up a Development Client Installer on the Deployment Server

Understanding the JD Edwards Clients (Development and Web)

The JD Edwards EnterpriseOne Development Client (also known as a Web Development Client, "Fat" Client, Administrative Client, Windows client, or Workstation) contains components that run as standard Microsoft Windows applications (for example, Active Console, Form Design Aid (FDA), and Report Design Aid (RDA)) and components that run in a web browser.

Note: This document uses the following terminology when discussing JD Edwards EnterpriseOne clients:

- **Web Client**

  Components that run in a web browser.

- **Development Client**

  Composed of standard Windows components and Web Client.

The Web Client part of the Development Client runs inside an Application Server, with the user interface displayed in a browser. For Tools Releases prior to 9.2.9, the supported Application Servers installed on the Development Client machine are:

- Oracle WebLogic Server (WLS)
- IBM WebSphere Application Server (WAS)

The version of the Web Client is known by either of these names:

- WLSH4A (WebLogic Server HTML for Applications)
- WASH4A (WebSphere Server HTML for Applications)

Note: If you are not using WLSH4A or WASH4A, you can ignore references to it in the subsequent sections.

Prior to Tools Release 9.2.9, the Application Server such as WebLogic or WebSphere was installed on the Development Client computer to run the Web Client in a browser. With Tools Release 9.2.9, EnterpriseOne connects to a remote Application Server (such as, WebLogic or WebSphere) installed on a separate server known as a Development HTML Server.

The information needed to connect to this remote Development HTML Server is configured by the EnterpriseOne administrator in the `jde.ini` on the Deployment Server. This `jde.ini` is copied to the Development Client machine during installation of the EnterpriseOne Development Client software

This chapter describes the setup of the JD Edwards EnterpriseOne Development Client installer and installation package on an EnterpriseOne Deployment Server. This setup on the Deployment Server must be done before a user can install a

JD Edwards EnterpriseOne Development Client on a workstation. The installation package specifies the components to install and may or may not include the Web Client.

If you are on Tools Releases prior to 9.2.9, you can find instructions on how to install Web Clients for both Application Servers and their WLSH4A and WASH4A features, as well as the installation of the WebLogic and WebSphere Application Servers in the *JD Edwards EnterpriseOne Development Client Installation Guide* .

Note: These instructions assume that all prerequisites for running EnterpriseOne on the Deployment Server are installed.

Note: The EnterpriseOne Deployment Server requires that the E1Local Oracle database be installed on the Deployment Server machine. Some Tools Releases also require that the E1Local Oracle database be installed on the Development Clients as well. Starting with Tools Release 9.2.5.0, the E1Local Oracle database will not be used on the Development Clients and can be uninstalled. It is important to note that the database is still required on the Deployment Server.Installation of the JD Edwards Deployment Server provides installers for these components that can be installed on a Development Client:

- Development Client
- WLSH4A (Tools Releases prior to 9.2.9)
- WASH4A (Tools Releases prior to 9.2.9)

These installers are based on the Oracle Universal Installer (OUI) which is used to install several JD Edwards EnterpriseOne and non-JD Edwards Oracle products such as Oracle databases and Oracle Middleware. The JD Edwards EnterpriseOne administrator must perform a small number of configuration steps to prepare these installers for running on EnterpriseOne workstations.

Understanding that Installers are Not Backward Compatible

OUI is not compatible between OUI versions. This means that if the OUI version changes between Tools Releases, all the Development Client installations must be uninstalled and new installations of a full package using the new OUI must be performed. When a Development Client is uninstalled or saved by using SnapShot, the WLSH4A or WASH4A is automatically removed or saved.

See *Editing oraparam.ini for the Development Client Installer During an Update or Upgrade* for information about the OUI versions.

The following examples are of a few configurations that are not allowed:

- A package being installed with one version of OUI cannot be installed on an existing Development Client installation that used a different version of OUI.
- A WLSH4A or WASH4A installed with one version of OUI cannot be installed on an existing Development Client installation that used a different version of OUI.

- The OUI versions of Development Client, WLSH4A, and WASH4A installers cannot be different.
- The OUI installers' directories cannot contain any files from previous OUI installers.

Note: Avoid the above scenarios on the Development Client by saving with SnapShot or uninstalling the existing Development Client before attempting to install a full package using the new installers.

Configuring Installers

The following sections detail the requisite administrative tasks for the JD Edwards EnterpriseOne Administrator upon installation of the Deployment Server or update of the Deployment Server:

- Update the `oraparam.ini` for the Development Client Installer
- Update the `oraparam.ini` for the WLSH4A Installer (Tools Releases prior to 9.2.9)
- Update the `oraparam.ini` for the WASH4A Installer (Tools Releases prior to 9.2.9)
- Edit the `jde.ini` template.
- Configure the Web Client (Tools Release 9.2.9)
- Copy JDBC Drivers and the `tnsnames.ora` file to the Deployment Server.
- Update the Database Drivers in the `JDBJ.ini` file.
- Update the `InstallManager.htm` File
- Understand the Local Oracle Database Installer
- Additional OEE Setup

Note: If you are not using a WLSH4A or WASH4A installer, you can ignore references to it in the following steps.

Configuring oraparam.ini for the Development Client Installer

The Development Client installer for Tools Releases prior to 9.1 uses the Exemplar installer framework. Installation parameters for Exemplar-based installers are stored in the `install.inf` file. When the installer was migrated to use Oracle Universal Installer (OUI) with Tools Release 9.1, the installation parameters were moved to the `oraparam.ini` file.

When updating a Deployment Server to Tools Release 9.2, the `oraparam.ini` file is not updated from the existing install.inf file (for Exemplar-based installers) or the oraparam.ini file (for OUI-based installers prior to Tools Release 9.2.3.4). In those cases, you must manually update some settings in the Tools Release 9.2 `oraparam.ini file` as described in this section.

Starting with Tools Release 9.2.3.4, Server Manager performs most of the updates to the oraparam.ini file automatically when updating from Tools Releases 9.2.0.0 and later. However, some initial settings must be configured as listed below:

The `oraparam.ini` file resides with the OUI executable setup.exe in this directory:

```
<E1_dep_svr_install_directory>\OneWorld Client Install\install
```

The `oraparam.ini` file follows the standard format of ini files for Microsoft Windows and it contains sections with names inside square brackets ([ ]) and key/value pairs. The keys and their values are separated by equals signs (=).

Editing the Initial oraparam.ini for the Development Client Installer

When a Deployment Server is first installed, a few settings in the oraparam.ini for the Development Client must be configured manually. Follow these steps:

1. Navigate to the directory:

   <E1_dep_svr_install_directory>\OneWorld Client Install\install
2. Make a backup copy of **oraparam.ini**.
3. Edit the **oraparam.ini** and set the values from the below table:

If the section or keys in the table do not exist in your oraparam.ini, you must add them along with the values indicated. Be sure that any paths on the Deployment Server can be reached from the Development Client machine. Do not use local paths; rather use the Uniform Naming Convention (UNC) format for paths such as:

\\<deployment server name>\<release>\xxx

Section	Key	Value Description	Value Example
[Oracle]	**DEFAULT_HOME_ LOCATION**	Default Oracle Home path into which Development Client will be installed. Users can change this path at installation time.	**E920**
[Oracle]	**DEFAULT_HOME_NAME**	Default Oracle Home name of path into which Development Client will be installed. Users can change this name at installation time.	**JDE_E920_Client**
[FileLocations]	**PackageInfs**	UNC path to package inf files.	**\ \depsvr2\E920\pack inf**
[FileLocations]	**currentReleaseMaster**	UNC path including deployment server and share.	**\ \depsvr2\E920**
[FileLocations]	**JdbcPath**	Optional - Used when installing WLSH4A or WASH4A (Tools Releases prior to 9.2.9) or the Development Client (Tools Release 9.2.9) - UNC path to JDBC files - If not specified, installer looks for a directory called JDBC. • Tools Releases prior to 9.2.9: This JDBC directory is expected up one level from the directory containing the **setup.exe** file for WLSH4A or WASH4A. • Tools Release 9.2.9: This JDBC directory is expected to be under **OneWorld Client Install\ThirdParty**.	**\ \depsvr2\E920\My_J**
[Attributes]	**DataByPathCode**	(Tools Releases prior to 9.2.9) Optional - Used when installing WLSH4A or WASH4A to create the datasource name that points to local data.	**1**

		=1 means that local data is specific to individual pathcodes. =0 means that local data is shared among all installed pathcodes. For details, see the appendix of this guide entitled *Data by Pathcode*.	
`[Attributes]`	`LocalDS`	(Tools Releases prior to 9.2.9) Optional - Used when installing WLSH4A or WASH4A to create the datasource name that points to local data For details, see the appendix of this guide entitled *Data by Pathcode*.	`OneWorldLocal`

Editing the oraparam.ini for the Development Client Installer During an Update or Upgrade

Prior to Tools Release 9.2.3.4, you must update the `oraparam.ini` for the Development Client installer using the following steps.

Starting with Tools Release 9.2.3.4, Server Manager automatically performs the following steps when updating or upgrading the Deployment Server to a new Tools Release.

1. Navigate to the directory:

   `<E1_dep_svr_install_directory>\OneWorld Client Install\install`
2. Make a backup copy of `oraparam.ini`.
3. Edit the `oraparam.ini` and set the values from the below table based on the Tools Release:

TOOLS RELEASE	OUI_VERSION	BOOTSTRAP_COMPS
9.2.9 and later	12.2.1.6.0	"oracle.swd.oui:12.2.1.6.0,oracle.swd.oui.core:1
9.2.3.4 and later	12.2.0.6.0	"oracle.swd.oui:12.2.0.6.0,oracle.swd.oui.core
9.2.3.3	12.2.0.9.0	"oracle.swd.oui:12.2.0.9.0,oracle.swd.oui.core:
9.2.2.6 and later	12.2.0.2.2	"oracle.swd.oui:12.2.0.2.2,oracle.swd.oui.core:
Prior to 9.2.2.6	11.2.0.2.0	"oracle.swd.oui:11.2.0.2.0,oracle.swd.oui.core:

   Note: The version of OUI decreased (downgraded) with Tools Release 9.2.3.4.

4. Save the `oraparam.ini` file.

Note: As noted in the table, the versions of the JRE or JDK that are provided with the Development Client installer may not be the latest versions available. See the appendix of this guide entitled *Updating the JRE/JDK Used by the Installers*.

Updating the oraparam.ini for the WLSH4A Installer (Tools Releases Prior to 9.2.9)

If you are not using WLSH4A, skip this section.

Prior to Tools Release 9.2.3.4, you must update the `oraparam.ini` for the WLSH4A installer using the following steps.

Starting with Tools Release 9.2.3.4, Server Manager automatically updates `oraparam.ini` for the WLSH4A installer when updating or upgrading the Deployment Server to a new Tools Release.

1. Navigate to this directory:

   `<E1_dep_svr_install_directory>\OneWorld Client Install\ThirdParty\WebDevFeature\WLSH4A\install`
2. Make a backup copy of `oraparam.ini`.
3. Edit the `oraparam.ini` and set the values from the below table based on the Tools Release:

TOOLS RELEASE	OUI_VERSION	BOOTSTRAP_COMPS
9.2.3.4 and later	12.2.0.6.0	"oracle.swd.oui:12.2.0.6.0,oracle.swd.oui.core
9.2.3.3	12.2.0.9.0	"oracle.swd.oui:12.2.0.9.0,oracle.swd.oui.core:
9.2.2.6 and later	12.2.0.2.2	"oracle.swd.oui:12.2.0.2.2,oracle.swd.oui.core:
Prior to 9.2.2.6	11.2.0.2.0	"oracle.swd.oui:11.2.0.2.0,oracle.swd.oui.core:

   Note: The version of OUI decreased (downgraded) with Tools Release 9.2.3.4.

4. Save the `oraparam`.ini file.

Note: As noted in the table, the versions of the JRE or JDK that are provided with the Development Client installer may not be the latest versions available. See the appendix of this guide entitled *Updating the JRE/JDK Used by the Installers*.

Updating the oraparam.ini for the WASH4A Installer (Tools Releases Prior to 9.2.9)

If you are not using WASH4A, skip this section.

Prior to Tools Release 9.2.3.4, you must update the `oraparam.ini` for the WASH4A installer using the following steps.

Starting with Tools Release 9.2.3.4, Server Manager automatically updates the `oraparam.ini` for the WASH4A installer when updating or upgrading the Deployment Server to a new Tools Release.

1. Navigate to this directory:

   `<E1_dep_svr_install_directory>\OneWorld Client Install\ThirdParty\WebDevFeature\WASH4A\install`
2. Make a backup copy of `oraparam.ini`.

3. Edit the `oraparam.ini` and set the values from the below table based on the Tools Release:

TOOLS RELEASE	OUI_VERSION	BOOTSTRAP_COMPS
9.2.3.4 and later	12.2.0.6.0	"oracle.swd.oui:12.2.0.6.0,oracle.swd.oui.core
9.2.3.3	12.2.0.9.0	"oracle.swd.oui:12.2.0.9.0,oracle.swd.oui.core:
9.2.2.6 and later	12.2.0.2.2	"oracle.swd.oui:12.2.0.2.2,oracle.swd.oui.core:
Prior to 9.2.2.6	11.2.0.2.0	"oracle.swd.oui:11.2.0.2.0,oracle.swd.oui.core:

> **Note:** The version of OUI decreased (downgraded) with Tools Release 9.2.3.4

4. Save the `oraparam.ini` file.

> **Note:** As noted in the table, the versions of the JRE or JDK that are provided with the Development Client installer may not be the latest versions available. See the appendix of this guide entitled *Updating the JRE/JDK Used by the Installers*.

The installer in this folder is obsolete: `<E1_dep_svr_install_directory>\OneWorld Client Install\ThirdParty\WebDevFeature\H4A85`

You can delete this folder; however, there are no adverse effects if you keep it.

Editing jde.ini Template

A `jde.ini` file is copied from a template on the Deployment Server to the Development Client upon installation. Most settings in the template are not modified when the template is copied to the Development Client. Therefore, if you want to have the settings in `jde.ini` on each Development Client set to particular values, set them in the `jde.ini` template located in `<E1_dep_svr_install_directory>\OneWorld Client Install\misc`.

Alternatively, you can set up a `<pathcode>.inf` file in `<E1_dep_svr_install_directory>\OneWorld Client Install\misc` where `<pathcode>` is the name of the path code. This file can contain path code specific settings to be applied to `jde.ini` on the Development Client. When the Development Client installer is run, the `jde.ini` template file is copied to the Development Client. If a `<pathcode>.inf` file exists, any settings in this file will overwrite the corresponding settings in the `jde.ini` on the Development Client.

Editing jas.ini, jdbj.ini, jdelog.properties Templates (Tools Release 9.2.9)

Upon installation, the jas.ini, jdbj.ini, and jdelog.properties files are copied from templates on the Deployment Server to the Development Client. Some settings in these files are updated by the installer. If you need to update any settings for all future installations, follow the steps in this section. Be careful to make only those edits that are required.

Extract the file(s) in the below jar files, edit it, and save it back into the jar file at the same location. "<installer_version>" in the below path names is the version of the installer such as 7.0.1.0.1.

- **<E1_dep_svr_install_directory>\OneWorld Client Install\stage\Components\com.e1.devclient.top \<installer_version>\1\DataFiles\filegroup1.1.1.jar**

  The contents of this jar are copied to **<E1_install_dir>\<pathcode>\ini\sbf** on the Development Client.

  - **jdeinterop.ini**
  - **jdelog.properties**
  - **LocalLogicCatalog.xml**

- **<E1_dep_svr_install_directory>\OneWorld Client Install\stage\Components\com.e1.devclient.top \<installer_version>\1\DataFiles\filegroup2.1.1.jar**

  The contents of this jar are copied to **<E1_install_dir>\system\classes** on the Development Client.

  - **jdbj.ini** (This file is also copied to **<E1_install_dir>\<pathcode>\ini\sbf**.)
  - **jdelog.properties**
  - **NT_jas.ini** (This file is renamed to **jas.ini** upon installation.)

Configuring the Web Client (Tools Release 9.2.9)

Prior to Tools Release 9.2.9, EnterpriseOne connected to an Application Server such as WebLogic or WebSphere installed on the Development Client computer to run the Web Client in a browser. With Tools Release 9.2.9, EnterpriseOne connects to a remote Application Server (such as, WebLogic or WebSphere) installed on a separate server known as a Development HTML Server.

The information needed to connect to this remote Application Server is configured by the EnterpriseOne administrator in the **jde.ini** template on the Deployment Server. This **jde.ini** is copied to the Development Client machine during installation of the EnterpriseOne Development Client software.

To specify the connection information in the **<E1_dep_svr_install_directory>\OneWorld Client Install\misc\jde.ini** template, update these values in the **[LOCALWEB]** section:

Key	Value	Comments
webhostname	**<Development HTML Server's name>**	Fully qualified name (includes domain) or IP address of Development HTML Server. If useSSL (seen below) is 1, then the server name must exactly match the name on the web browser's SSL certificate.
webport	**<Development HTML Server's port>**	Port number on which Development HTML Server listens
useSSL	**1 or 0**	1 = webport value is an https port 0 = webport value is an http port

		If this setting is omitted, the default value is 1.

Note: [LOCALWEB] AppServerInstalled is set automatically in the `jde.ini` on the Development Client during installation. Do not change it manually.

Note: If there are any firewalls between the Development Client(s) and the Development HTML Server(s) designated in the LOCALWEB section, the webport must be open to allow communication between the server(s)

and Development Client(s). The ports must also be open for the serviceNameListen value in the [JDENET] section of the Development Client JDE.INI and the BPMBroker value (default of 9876) in the registry under the following path: `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\JDEdwards\BPMBroker\port`

Copying JDBC Drivers and the tnsnames.ora file to the Deployment Server

Ensure that the required JDBC drivers and the `tnsnames.ora` file are automatically downloaded from the Deployment Server to the Development Client during the installation of the Development Client. To set up this process, the administrator must first obtain the required JDBC drivers for the supported platforms and databases as well as the `tnsnames.ora` file, if connecting to an Oracle database, and copy them to the Deployment Server as described in this procedure.

This section describes these tasks:

- Copying the JDBC Drivers (Tools Releases prior to 9.2.9)
- Copying the JDBC Drivers (Tools Releases 9.2.9)
- Copying the tnsnames.ora file

Copying the JDBC Drivers (Tools Releases Prior to 9.2.9)

To copy the JDBC drivers:

1. On the Deployment Server, create a \JDBC subdirectory under the folder that contains the Web Client installation program. For example:

   <E1_dep_svr_install_directory>\OneWorld Client Install\ThirdParty\WebDevFeature\WLSH4A\JDBC

   or

   <E1_dep_svr_install_directory>\OneWorld Client Install\ThirdParty\WebDevFeature\WASH4A\JDBC

   > **Note:** If you must copy the JDBC files to a different directory for some reason, you must add an entry to the oraparam.ini file for the WLSH4A or WASH4A installer to indicate the location of that JDBC directory. Edit the oraparam.ini file located in this directory: **\\<deployment server name>\<release>\OneWorld Client Install\ThirdParty\WebDevFeature\WLSH4A\Install** or **\\<deployment server name>\<release>\OneWorld Client Install\ThirdParty\WebDevFeature\WASH4A\Install**. Add this entry to the **[FileLocations]** section of the **oraparam.ini** file: **JdbcPath=<UNC_Path_to_JDBC_files>** For example: **[FileLocations] JdbcPath=\\depsvr2\E920\My_JDBC**
   >
   > > **Note:** If you leave this entry out, the WLSH4A or WASH4A installer will look for the JDBC directory up one level from the setup.exe program for WLSH4A or WASH4A.

   > **Note:** You must delete any obsolete driver files because all drivers in this directory will be loaded by EnterpriseOne. If obsolete drivers exist, they may be used instead of newer drivers. Check Oracle Certifications for the latest drivers.

2. Refer to *Understanding Certifications* for information on where to obtain the JDBC drivers.

   The required JDBC drivers for each platform are listed in the following table, but check Oracle Certifications for the latest drivers:

Database	Required Drivers	Comments
Oracle	ojdbc8.jar, ons.jar, ucp.jar	The JDBC driver for Oracle will automatically be downloaded to your system when you install an Oracle database client or server. If you install a different driver, the system may not function properly.
UDB/DB2	db2jcc4.jar	The JDBC driver for DB2/UDB is delivered with DB2/UDB server or client installation.
DB2/400 for IBMi	jt400.jar	The DB2/400 driver for the IBMi platform is delivered with the DB2/400 database installation.
SQL Server	mssql-	Microsoft provides a type 4 JDBC driver with full support at no

	jdbc-11.2.0.jre8.jar	additional charge. Download the JDBC driver directly from Microsoft's website. Use the installation documentation that comes with the download to install the JDBC driver into a temporary location.

3. Obtain the JDBC drivers and copy them to the JDBC directory you created in the first step. When the Web Client is deployed to a target machine, these drivers are automatically installed into the following directory:

   <Client Install Path>\misc

Copying the JDBC Drivers (Tools Release 9.2.9)

To copy the JDBC drivers:

1. On the Deployment Server, create a \JDBC subdirectory under the "ThirdParty" folder. For example:

   <E1_dep_svr_install_directory>\OneWorld Client Install\ThirdParty\JDBC

   > **Note:** If you must copy the JDBC files to a different directory for some reason, you must add an entry to the oraparam.ini file for the Development Client installer to indicate the location of that JDBC directory. Edit the **oraparam.ini** file located in this directory: **<E1_dep_svr_install_directory>\OneWorld Client Install\Install.** Add this entry to the [FileLocations] section of the **oraparam.ini** file: **JdbcPath=<UNC_Path_to_JDBC_files>** For example: **[FileLocations] JdbcPath=\\depsvr2\E920\My_JDBC**
   >
   > > **Note:** If you leave this entry out, the Development Client installer will look for the JDBC directory under **OneWorld Client Install\ThirdParty.**

   > **Note:** You must delete any obsolete driver files because all drivers in this directory will be loaded by EnterpriseOne. If obsolete drivers exist, they may be used instead of newer drivers. Check Oracle Certifications for the latest drivers.

2. Refer to *Understanding Certifications* for information on where to obtain the JDBC drivers.

   The required JDBC drivers for each platform are listed in the following table, but check Oracle Certifications for the latest drivers:

Database	Required Drivers	Comments
Oracle	ojdbc8.jar, ons.jar, ucp.jar	The JDBC driver for Oracle will automatically be downloaded to your system when you install an Oracle database client or server. If you install a different driver, the system may not function properly.

UDB/DB2	db2jcc4.jar	The JDBC driver for DB2/UDB is delivered with DB2/UDB server or client installation.
DB2/400 for IBMi	jt400.jar	The DB2/400 driver for the IBMi platform is delivered with the DB2/400 database installation.
SQL Server	mssql- jdbc-11.2.0.jre8.jar	Microsoft provides a type 4 JDBC driver with full support at no additional charge. Download the JDBC driver directly from Microsoft's website. Use the installation documentation that comes with the download to install the JDBC driver into a temporary location.

3. Obtain the JDBC drivers and copy them to the JDBC directory you created in the first step.

   When the Web Client is deployed to a target machine, these drivers are automatically installed into the following directory:

   ```
   <Client Install Path>\misc
   ```

Copying the tnsnames.ora File

The `tnsnames.ora` file is required only if the Development Client will connect to an Oracle database. However, if the `tnsnames.ora` file is not copied to the correct directory on the Deployment Server, the user will receive an error message during the Web Client (Tools Releases prior to 9.2.9) or Development Client (Tools Release 9.2.9) installation process.

To copy the `tnsnames.ora` file:

1. Obtain the `tnsnames.ora` file from an Oracle database configured to run with JD Edwards EnterpriseOne.
2. **Tools Releases prior to 9.2.5.0**: If the Development Client will use an Oracle database installed locally on the Development Client machine, you must use the following steps to ensure that the tnsnames.ora file has an entry to allow the Development Client to connect to the local database:
   - Edit the tnsnames.ora file.
   - If an entry for E1Local does not already exist, add this to the file:

     ```
     E1LOCAL = (DESCRIPTION =

     (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA =

     (SERVER = DEDICATED) (SERVICE_NAME = E1Local)

     ))
     ```
   - Save the `tnsnames.ora` file.
3. Copy the `tnsnames.ora` to this directory on the Deployment Server: `\\<deployment server name>\<release>\client` When the Development Client is installed on a target machine, this file is automatically installed to the following directories:

For WLSH4A (Tools Releases Prior to 9.2.9):

- **<Client Install Path>\system\JAS\EA_JAS_80.ear\webclient.war\WEB-INF\classes**
- **<Client Install Path>\<pathcode>\ini\sbf**

For WASH4A (Tools Releases Prior to 9.2.9):

- **<Client Install Path>\JAS\EA_JAS_80.ear\webclient.war\WEB-INF**
- **<Client Install Path>\<pathcode>\ini\sbf**

For Development Client (Tools Releases Prior to 9.2.5):

- **<64-bit Oracle database install path>\NETWORK\ADMIN**

For Development Client (Tools Releases 9.2.9):

- **<Client Install Path>\<pathcode>\ini\sbf**

Updating the Database Drivers in the JDBJ.ini File (Tools Releases Prior to 9.2.9)

The Web Client installer (WLSH4A or WASH4A) updates the `JDBJ.ini` file with settings for the databases to which the Web Client will connect based on the JDBC database drivers found in this directory on the Development Client:

<Client Install Path>\misc

As noted in the preceding section of this chapter entitled: Copying the JDBC Drivers (Tools Releases prior to 9.2.9), these JDBC database drivers are copied from this directory:

\\<deployment server name>\<release>\OneWorld Client Install\ThirdParty\WebDevFeature\<WLSH4A_or_WASH4A>\JDBC

> **Note:** If the user copies additional database drivers into the **<Client Install Path>\misc** directory after installing WLSH4A or WASH4A, they must update the list of drivers in the `JDBJ.ini` file. This update must occur on each Development Client because the jdbj.ini template is built into the WLSH4A and WASH4A installers. Instructions for doing this are in the *JD Edwards EnterpriseOne Development Client Installation Guide*.

Updating the Database Drivers in the JDBJ.ini File (Tools Release 9.2.9)

The Development Client installer updates the `JDBJ.ini` file with settings for the databases to which the Development Client will connect based on the JDBC database drivers found in this directory:

<Client Install Path>\misc

As noted in the preceding section of this chapter entitled: Copying the JDBC Drivers (Tools Release 9.2.9), these JDBC database drivers are copied from this directory:

\\<deployment server name>\<release>\OneWorld Client Install\ThirdParty\JDBC

Note: If the user copies additional database drivers into the `<Client Install Path>\misc` directory after installing the Development Client, they must update the list of drivers in the `JDBJ.ini` file. This update must occur on each Development Client because the jdbj.ini template is built into the Development Client installer. Instructions for doing this are in the *JD Edwards EnterpriseOne Development Client Installation Guide*.

Updating the installmanager.htm File

To install a local Oracle database for use by the Development Client or to install the Development Client itself, the user runs this program:

```
\\<deployment server name>\<release>\OneWorld Client Install\InstallManager.exe
```

This program displays a graphical user interface that provides several options for the user to perform. The options that InstallManager.exe displays are read from the file InstallManager.htm that resides in the same directory as `InstallManager.exe`.

The JD Edwards EnterpriseOne administrator can edit this file to rename the options or to add new options. Instructions are inside the file.

CAUTION: Although `InstallManager.htm` is in HTML format, it is a restricted HTML that the `InstallManager.exe` can read. Some text file editing programs add extraneous HTML tags that `InstallManager.exe` does not recognize. The Microsoft Windows program Notepad.exe is an example of a program that is known to not add these extra tags, making it safe to use. Before you make any changes to the file, save a copy in case any changes that you make are not recognized by `InstallManager.exe`.

Understanding the Local Oracle Database Installer

Tools Releases prior to 9.2.5.0 require a database to be installed onto the same machine on which the Development Client is to be installed. This local Oracle database on the Development Client contains local specs and other data. This database is also referred to as E1Local, OEE (which stands for Oracle Enterprise Edition), or JDELocal. With Tools Release 9.2.5.0, the EnterpriseOne object specs were moved from that local database to the Central Objects data source. Because of this, the need for the database on the Development Client was removed. This move saves the time and disk space required to install the local Oracle database. It also saves much time during the installation of the Development Client and the save/restore operations of SnapShot.

You can choose the database based on the Applications Release of EnterpriseOne and "bitness" of the Windows operating system that your users are using. Do not take the following information for granted - you should always check Oracle Certifications for the most current official information.

- Applications Release 9.0
  - 64-bit Oracle Enterprise Edition 12c (OEE)
  - Microsoft SQL Server 2005 Express Edition SP3 or later
- Applications Release 9.2
  - 64-bit Oracle Enterprise Edition 19c (OEE) or
  - 64-bit Oracle Enterprise Edition 12c (OEE)

To install a database onto a Development Client machine, the user runs this program:

\\<deployment server name>\<release>\OneWorld Client Install\InstallManager.exe

This program enables the user to install a local database or the JD Edwards EnterpriseOne Development Client. By default, the **InstallManager.htm** file is shipped from Oracle configured to install OEE.

Additional OEE Setup (Tools Releases Prior to 9.2.5)

The main installation program for OEE 19 is **OEE19Setup.exe**. For OEE 12 it is **OEE12Setup.exe**. Upon initial installation of Tools Release 9.2, this directory is created:

\\<deployment server name>\<release>\OneWorld Client Install\ThirdParty\ORACLE

The Tools Release 9.2 installation delivers these files in the above directory:

- **OEE19Setup.exe** or **OEE12Setup.exe**
- **deinstall.bat.tmpl**
- **deinstall_E1Local.rsp.tmpl**

You must manually copy the **E1Local** cloned database (for example, the .CAB files) to this directory:

\\<deployment server name>\<release>\OneWorld Client Install\ThirdParty\ORACLE

This preconfigured and cloned database can be downloaded from the Oracle Software Delivery Cloud site or the JD Edwards Update Center. Only a 64-bit version of the OEE 19c database is available in the download.

Inside the zip files that you downloaded are files containing the cloned database. These internal files are called **E1Local19c.cab**, **E1Local19c2.cab**, etc. Place the **.cab** files in the ORACLE directory specified above.

Removing the Local Oracle Database Installer for the Development Client (Tools Releases 9.2.5 and Later)

As noted above, the requirement for a local Oracle database on the Development Client was removed in Tools Release 9.2.5.

> **Note:** Although the need for E1Local was removed from the Development Clients, it is still required on the Deployment Server.

E1Local can remain installed on Development Client machines, or it may be removed following the steps in the

JD Edwards EnterpriseOne Development Client Installation Guide.

These steps to remove the E1Local installer from the Deployment Server are optional:

1. Edit this file:

   <E1_dep_svr_install_directory>\OneWorld Client Install\InstallManager.htm
2. Either comment out or remove the line on which OEE19Setup.exe exists.
3. Save the file.
4. Delete this directory:

   <E1_dep_svr_install_directory>\OneWorld Client Install\ThirdParty\ORACLE

Updating Installers

Applying New Tools Release Using Server Manager

You must use Server Manager to apply a new Tools Release to the Deployment Server. Instructions are provided in the *JD Edwards EnterpriseOne Server Manager Guide*.

Important: To avoid incompatibilities between versions of OUI on the Development Client, you can use SnapSshot or uninstall the existing Development Client before attempting to install a full package using the new installers. See *Editing the oraparam.ini for the Development Client Installer During an Update or Upgrade* in the *JD Edwards EnterpriseOne Tools Release Upgrade Process Guide* for information about the versions of OUI.

Manually Update Installers (Tools Release 9.2.3.3 Only)

If you are on Tools Release 9.2.3.3 only, you will have to perform the following steps manually. Server Manager was updated in Tools Release 9.2.3.4 to automatically perform the steps, therefore Oracle highly recommends that you update to at least Tools Release 9.2.3.4. If you cannot move to 9.2.3.4 or later and are moving to 9.2.3.3, continue to the steps below.

If you are moving to Tools Release 9.2.3.3, perform these steps:

1. On the Deployment Server, rename the directories of old installers BEFORE applying the Tools Release. The Tools Release 9.2.3.3 and later Development Client, WLSH4A, and WASH4A installers are not compatible with installers from prior Tools Releases; therefore, the new installers must be placed into empty directories on the Deployment Server.

   Rename the following directories under **<E1_dep_svr_install_directory>\OneWorld Client Install** (for example, add "**_backup**" to the end of their names). Ignore any that you are not using (for example, those for WLSH4A or WASH4A).

   - **\install**
   - **\stage**
   - **\ThirdParty\WebDevFeature\WLSH4A\install**
   - **\ThirdParty\WebDevFeature\WLSH4A\response**
   - **\ThirdParty\WebDevFeature\WLSH4A\stage**

- `\ThirdParty\WebDevFeature\WASH4A\install`
- `\ThirdParty\WebDevFeature\WASH4A\response`
- `\ThirdParty\WebDevFeature\WASH4A\stage`

2. In Server Manager Console, apply the new Tools Release to the Deployment Server. See the *JD Edwards EnterpriseOne Server Manager Guide* for details.
3. Update the `oraparam.ini` for the new Development Client installers with the section/key pairs shown in the table in *Configuring oraparam.ini for the Development Client Installer*.

Note: Ensure to start with the new oraparam.ini that comes with the latest Tools Release because the updated file contains values that must match with the new installer. You must update only those settings in the table.

No changes are required to the oraparam.ini or response files for the WLSH4A and WASH4A installers. Use the files that come with the installers.

Automatically Update Installers Using Server Manager (Tools Releases 9.2.3.4 and later)

Starting with Tools Release 9.2.3.4, Server Manager automatically performs the steps to rename the directories of the installers and updates the `oraparam.ini` file.

To update the installers, perform this step in the Server Manager Console: Apply the new Tools Release to the Deployment Server. Instructions are provided in the *JD Edwards EnterpriseOne Server Manager Guide*.

Updating JDBC Drivers

Ensure the WLSH4A or WASH4A installer (Tools Releases prior to 9.2.9) or Development Client installer (Tools Releases 9.2.9 and later) has the latest JDBC driver files for the currently supported database versions.

Follow the steps in *Copying JDBC Drivers and the tnsnames.ora file to the Deployment Server* to update the drivers.

Note: As noted in *Copying JDBC Drivers and the tnsnames.ora file to the Deployment Server*, you must delete any obsolete driver files. Check Oracle Certifications for the latest drivers.

Note: When installing the Web Client (WLSH4A or WASH4A) using installers prior to Tools Release 9.2.5.0, the JDBj-SPEC DATA SOURCE section in the `jdbj.ini` file is populated with information that points to the E1Local database as the location of the Serialized Objects tables (F989998 and F989999). Since E1Local is no longer supported with Tools Release 9.2.5.0, the Web Client installers comment out this entire section in the *jdbj.ini* file. This causes the Web Client to use OCM to determine the location of the Serialized Objects tables.

Administer JD Edwards Clients' (Development and Web) Installers

The initial installation of the JD Edwards Deployment Server provides installers for these components that can be installed on a Development Client:

- Development Client
- WLSH4A
- WASH4A

These installers are based on the Oracle Universal Installer (OUI) which is used to install a number of JD Edwards EnterpriseOne and non-JD Edwards Oracle products such as Oracle databases and Oracle Middleware.

The JD Edwards EnterpriseOne administrator must perform a small number of configuration steps to prepare these installers for running on EnterpriseOne workstations.

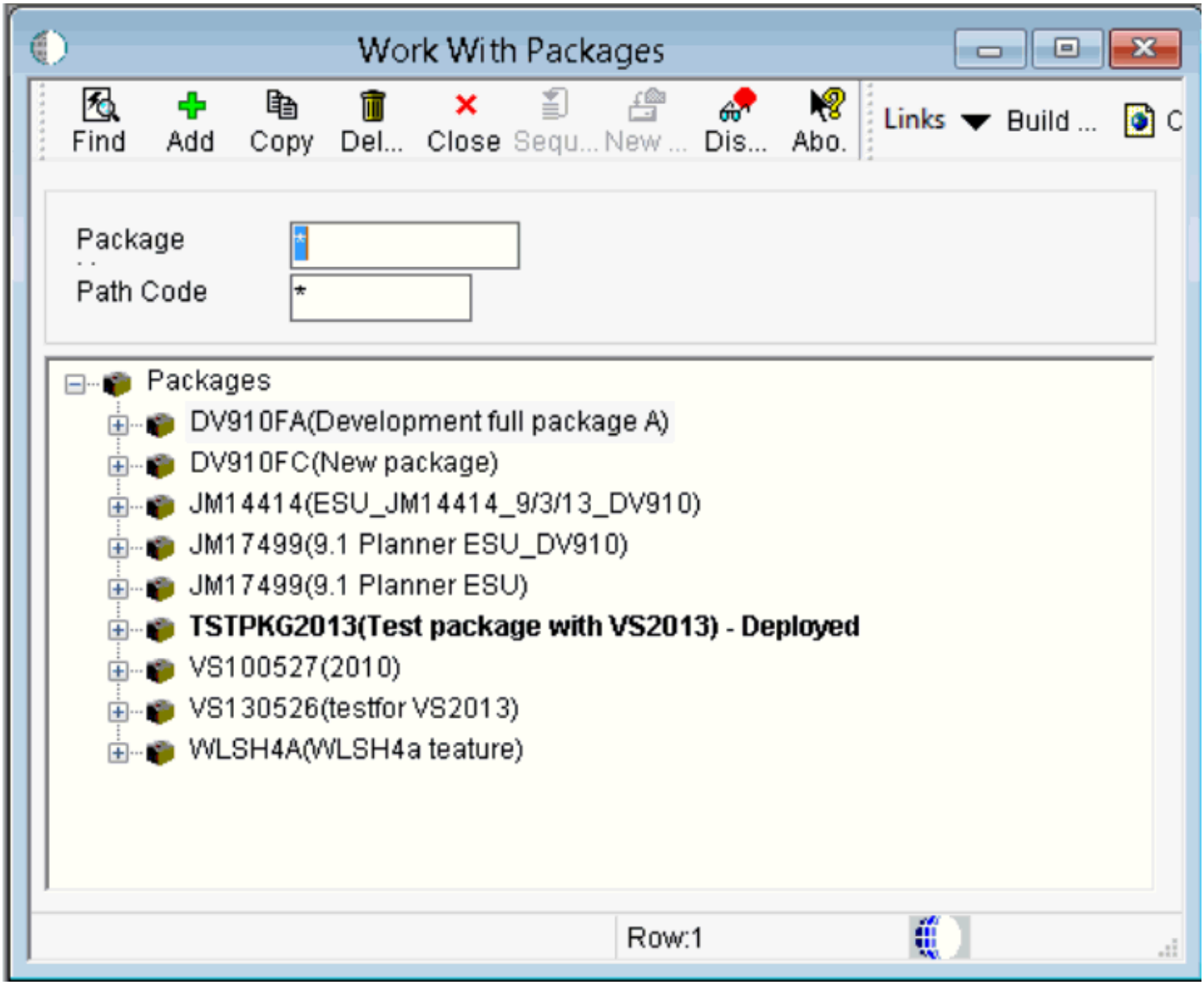
The JD Edwards product Server Manager (SM) updates the Deployment Server and Development Client with new Tools Releases. Because of changes in various Tools Releases, additional setup steps are required depending on the particular Tools Release being applied.

Creating the Web Development Feature (Tools Releases Prior to 9.2.9)

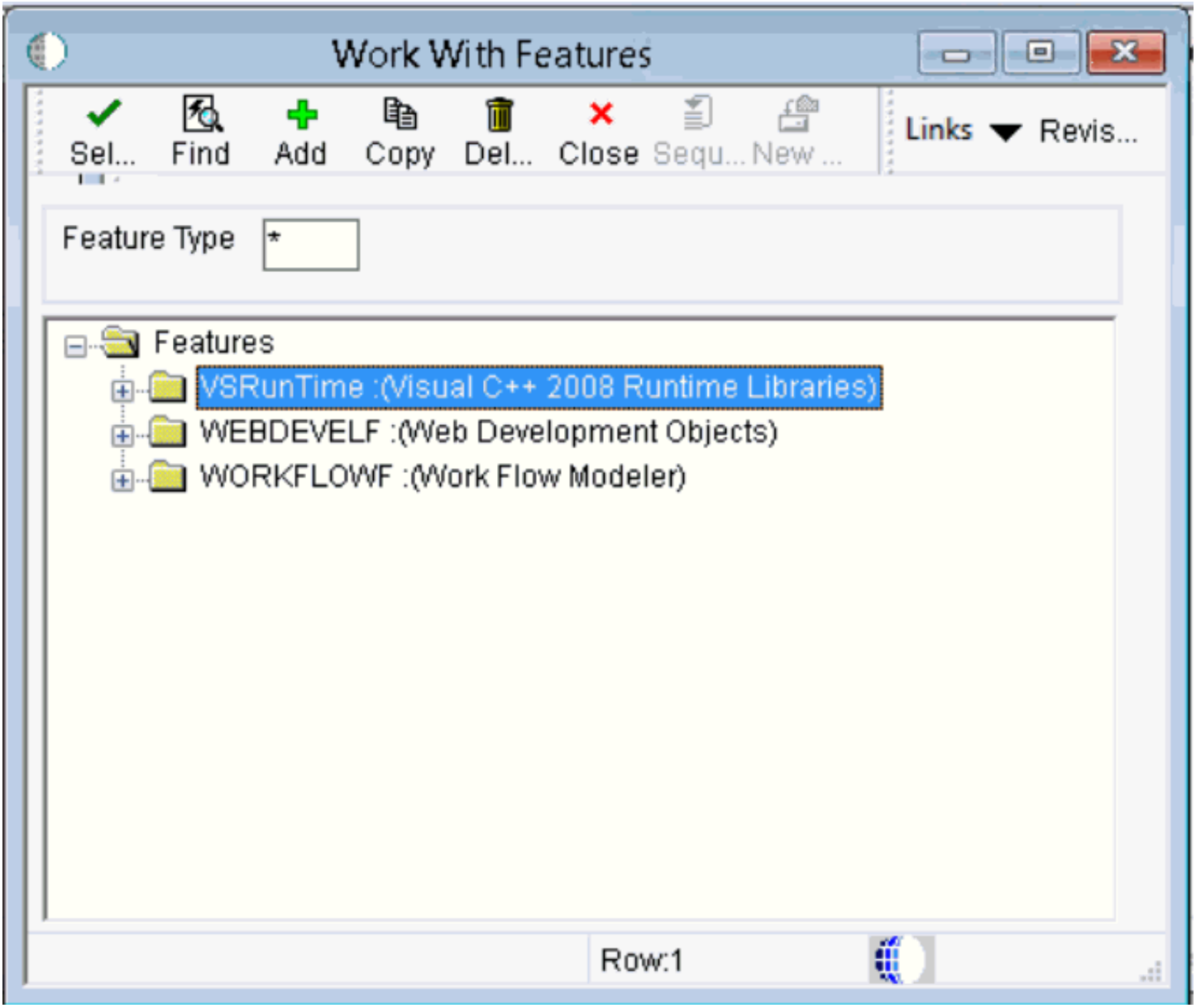
Complete this task to create a Web Client Installation Feature for your version of the application server. This feature must be included in the installation package before it is deployed to the Web Client machine.

1. On the Deployment Server, log on to the Deployment environment (for example, DEP920 for Applications Release 9.2).
2. Access the GH9083 Package Assembly application.

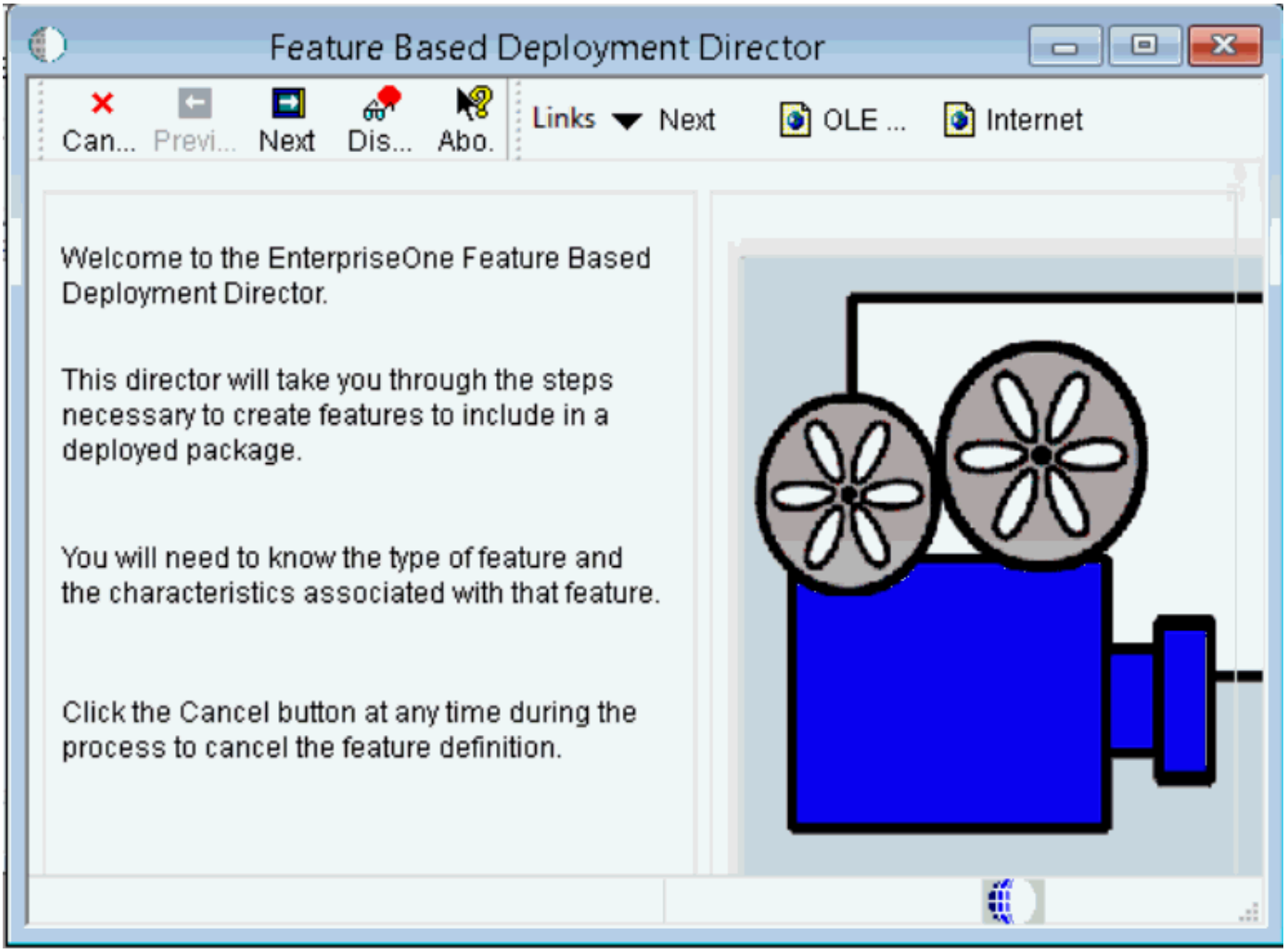
3. Select Form > Features.



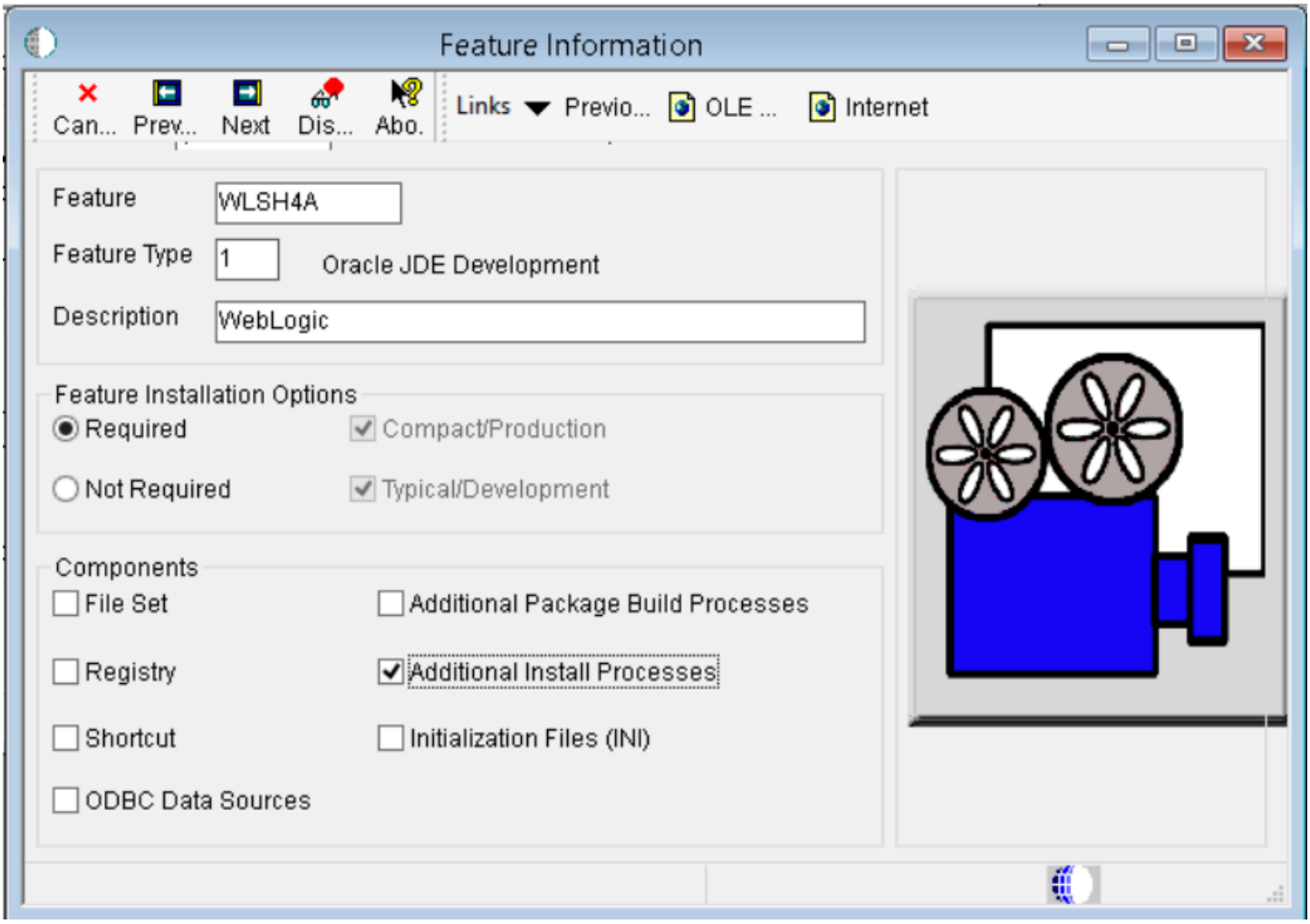
4. On Work With Packages, click the **Add** button.



5. On Work With Features, click the **Add** button.



6. On Feature Based Deployment Director, click the **Next** button.



7. On Feature Information, complete these fields:

   - *Feature*

     Enter a meaningful name for the feature. For example, if you are using Oracle WebLogic Server:

     WLSH4A

     If you are using IBM WebSphere Application Server, an example would be:

     WASH4A

     You can specify any name for the feature.

     > **Tip:** Alternatively, you can provide the WAS Express version number (for example, WAS85H4A).

     > **Tip:** This Feature Name is for display purposes only. It does not have to be any of these names:
     > - WLSH4A
     > - WASH4A
     > - WAS90H4A
     > - H4A85 (Obsolete)

   - *Feature Type*

     Enter the value **1**.

   - *Description*

     Enter a description. For example:

     Web Development Objects (WLSH4A) for WebLogic

     or

     Web Development Objects (WASH4A) for WebSphere

     You can specify any text string for the description.

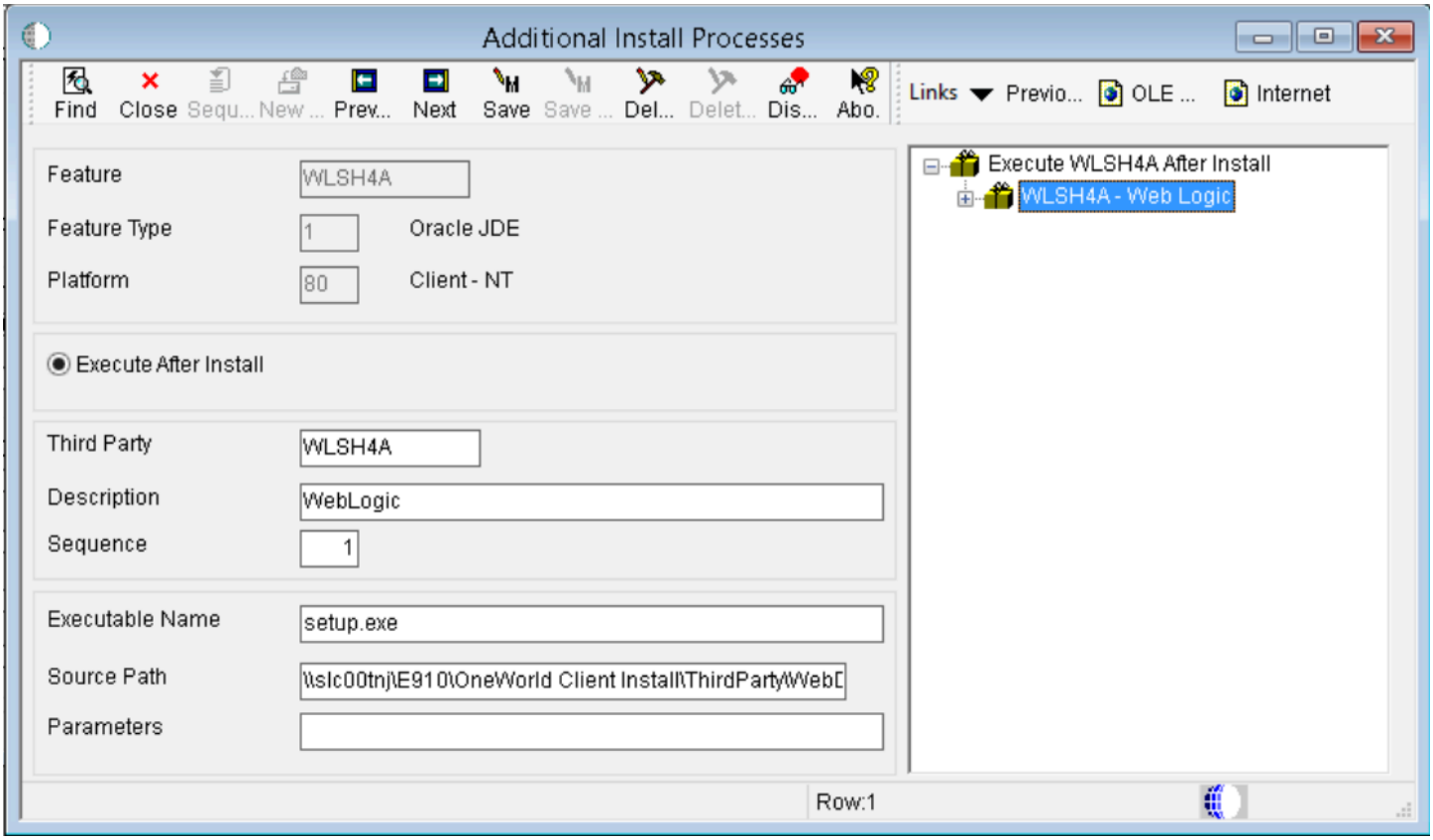
   - *Required*

     Ensure this option is selected.

   - *Additional Install Processes*

     Ensure this option is selected.

8. Click the **Next** button.



9. On Additional Install Processes, complete these fields:

   - *Execute After Install*

     For Tools Release 9.1 and greater, the Development Client installer does not use this flag. The default and unalterable behavior is to run the feature's installer after installing the Development Client.
   - *Third Party*

     Enter the name of the feature that you entered earlier. For example, if you are using WebLogic:

     WLSH4A

     If you are using WAS, an example would be:

     WASH4A
   - *Description*

     Enter a description. For example:

     Web Development Objects (WLSH4A) for WebLogic

     or

     Web Development Objects (WASH4A) for WebSphere

     You can specify any text string for the feature.
   - *Executable Name*

     Enter this value:

     setup.exe
   - *Source Path*

     Enter the path to the installation executable.

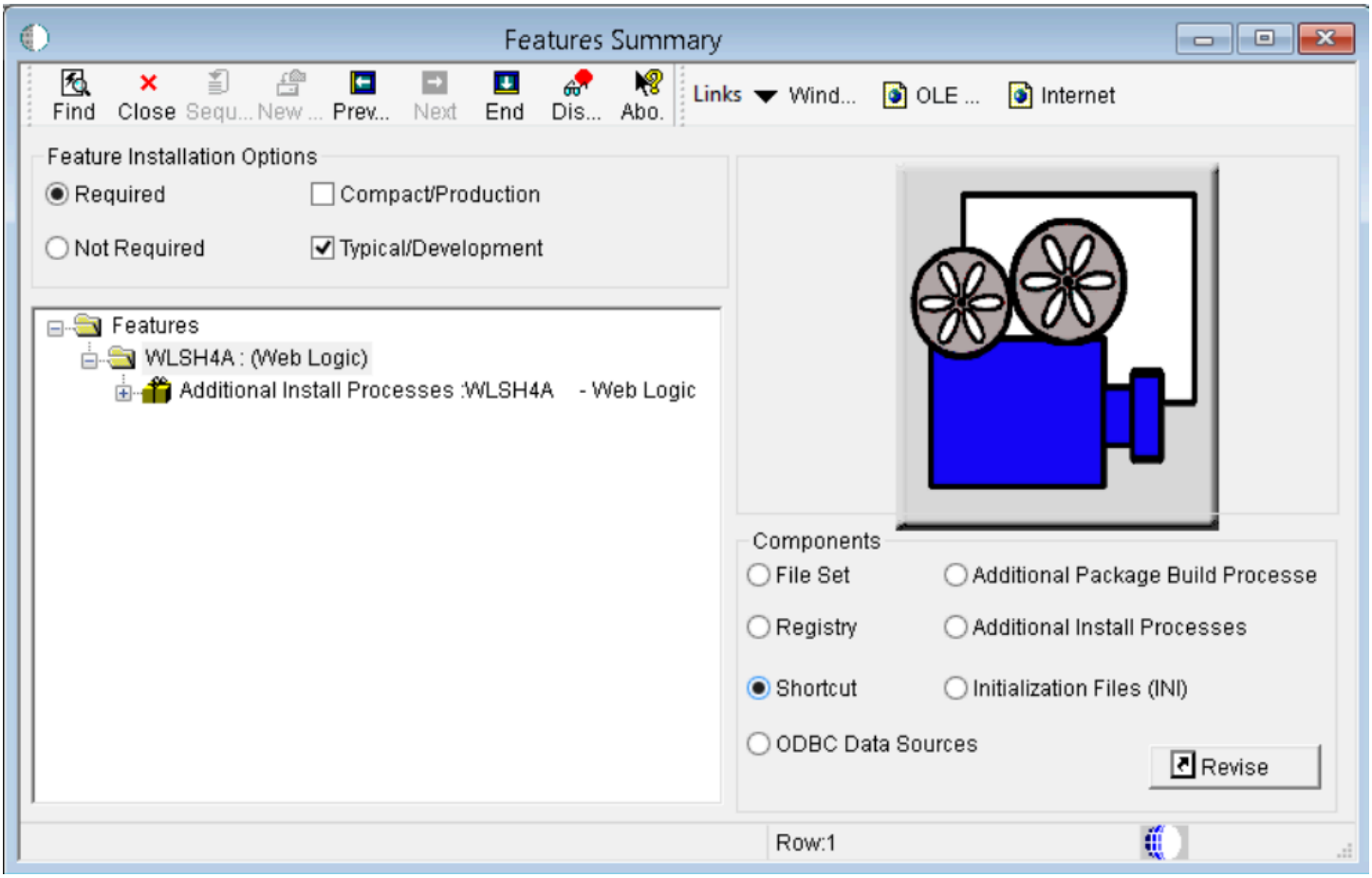
     For WLSH4A, the path would be:

     ```
     \\<deployment server name>\<release>\OneWorld Client Install\ThirdParty\WebDevFeature\WLSH4A
     \install
     ```

     For WASH4A, the path would be:

     ```
     \\<deployment server name>\<release>\OneWorld Client Install\ThirdParty\WebDevFeature\WASH4A
     \install
     ```
10. Click the **Save** button.

11. Click the **Next** button.



12. On Features Summary, click the **End** button.

Creating an Installation Package (Tools Releases Prior to 9.2.9)

This task describes how to add a Web Client Installation Feature to an installation package.

1. On the Deployment Server, run EnterpriseOne.

2. From the Package and Deployment Tools menu (GH9083), choose Package Assembly (P9601).
3. Choose an assembled package that you want to deploy to the Development Client, and click Package Revisions from the Row Exit menu, or click Add to create a new package.

   > **CAUTION:** Do not attempt to add a feature to an existing package that has already been built. When you deactivate a built package, EnterpriseOne deletes the package.

4. Click Features.
5. On the Features Component screen, click Browse.
6. On the Feature Component Selection screen, click Find.
7. Choose the Web Client Installation Feature for the Application Server you are using.

   > **CAUTION:** Do not include a feature for more than one of the Web Client installers (such as WLSH4A or WASH4A) in a single installation package. If you use more than one application server in your environment, create a separate package for each feature.

8. Click Select.

   A check mark displays to the left of the component.
9. Click Close.
10. On the Feature Component screen, click Close.
11. On Package Component Revisions, click OK.
12. Choose the package you just modified, and click **Activate/Inactive** in the Row Exit menu.
13. Click **Define Build** in the Row Exit menu.
14. On the Work with Package Build Definition screen, click Add.
15. Navigate through the Package Build Definition screens by clicking Next.
16. On the Build Features screen, click the *Build Feature INFs* check box.
17. Complete the Build Definition process.
18. On Work With Package Build Definition, click **Submit Build** in the Row Exit menu.

3 Working With the Full Client Package

Understanding the Full Client Package

The package build and assembly process includes many critical tasks that must be successfully completed to correctly install packages. See *JD Edwards EnterpriseOne Tools Package Management Guide* for details on building, assembling, and deploying packages. That guide lists most of these tasks and gives step-by-step instructions for completing them.

It is necessary to build a Full Client and Server Package when upgrading to the *Tools Release 9.2* . You must first upgrade the Enterprise Servers before you upgrade the Development Clients. If the Enterprise Servers are not upgraded first, they may not be able to properly receive and process requests from Development Clients that are upgraded.

> **CAUTION:** When run the OUI installer on the Development Client, OUI reuses components from previous installations. Because of this, users migrating to Tools Release 9.2 on Development Clients must first uninstall their existing client or save it with SnapShot and then install a full client package.

Creating the Full Client Package

To create the full client package:

1. Sign on to JD Edwards EnterpriseOne.
2. Go to GH9083.
3. Select **Package Assembly**.
4. Click the **Add** button.
5. On the Welcome screen, click the **Next** button.
6. Enter the Package name, Description and Pathcode, click the **Next** button.
7. If the package is using the Defaults then select **End**; otherwise, you can use the form icons to perform the functions in this procedure.
8. Activate the package.
9. Select **Define Build**.
10. On the Welcome screen, click the **Next** button.
11. Click the **Client and Server** button, and then click the **Next** button.
12. Click the **Client** button, and then click the **Next** button.
13. Select **End** if you want to take the defaults.
14. Activate the package.
15. Click the **Submit Build** button to build the package.
16. When the package build is complete, deploy the package to the client workstations or make it **Approved for Install Manager** for the client workstations.

4 Working With SnapShot on the Deployment Server

Understanding SnapShot

The SnapShot program (note the terminology uses upper cases in the naming convention for this program) manages multiple instances of the JD Edwards EnterpriseOne Deployment Server or Development Client on a single machine.

> **Note:** This chapter discusses using SnapShot on the Deployment Server. For details on using SnapShot on the Development Client, refer to the *JD Edwards EnterpriseOne Development Client Installation Guide* .

Using SnapShot you can save and restore copies of an installed Deployment Server. A saved copy is called a "snapshot" (note the terminology uses lower case in the naming convention for this entity).

> **Note:** The SnapShot program that comes with each Tools Release is backwards compatible and can save/restore EnterpriseOne Development Clients from previous Tools Releases that may or may not include E1Local. Use the latest Snapshot.exe that is available.

Below is an example of SnapShot managing multiple tools releases including E920 and two installations of E910 (B9_Perf and E910_aaa).

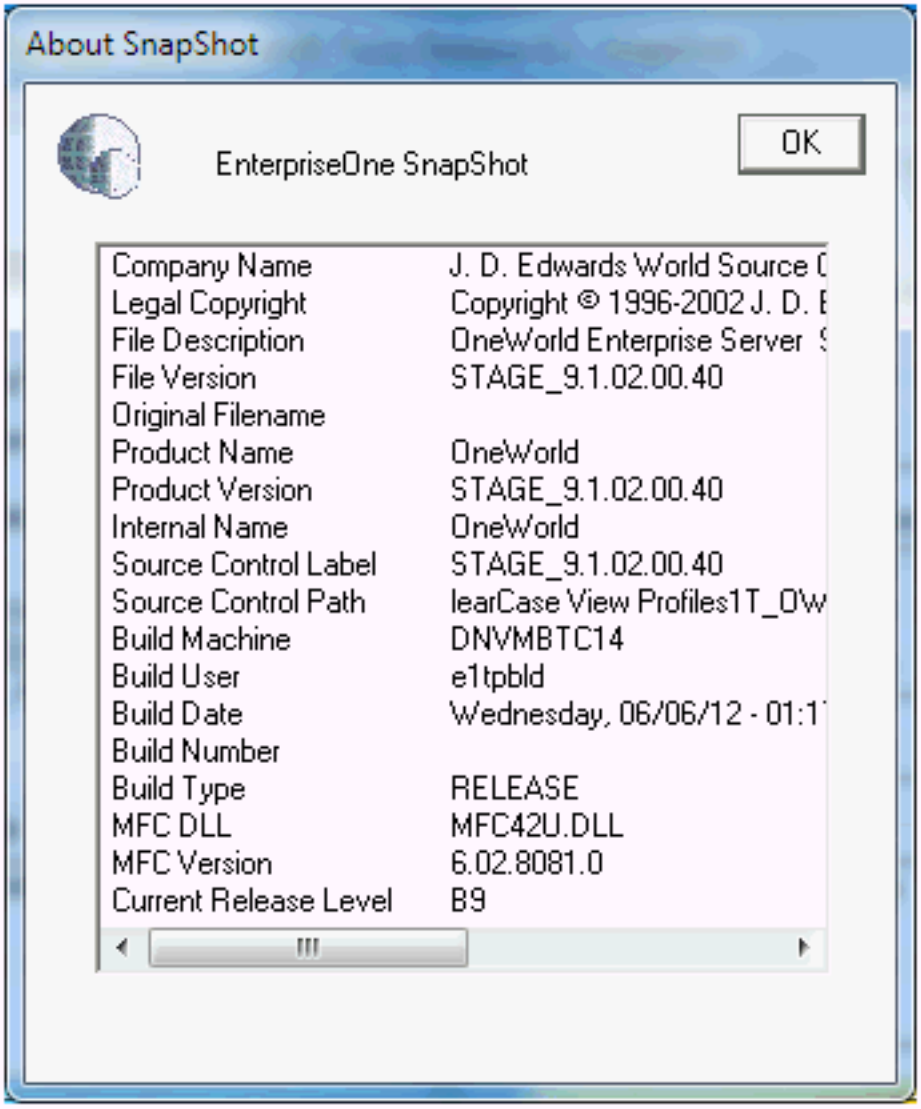
The preceding image shows the JD Edwards main SnapShot window. The main features of this window, from top to bottom, include:

- *Close (icon)*

  The Close icon is located in the upper right-hand corner. Click this icon to exit `SnapShot.exe`.

- *System (icon)*

  The System icon is located in the upper left-hand corner. Click on this icon or right-click on the title bar to display a drop-down menu that contains "About EnterpriseOne SnapShot...." Selecting this item displays version and build information about SnapShot as shown below:



- *Existing Version*

  This area contains these fields and buttons:

  - *Version*

    The existing installed Deployment Server. Initially, this is the release specified in the installed package. After you save and then restore a SnapShot, this field will be the name that you gave the SnapShot when you saved it.
  - *Location*

    The JD Edwards EnterpriseOne installation directory.
  - *Save Button*

    Saves the installed JD Edwards EnterpriseOne to a snapshot.
  - *Rename Environment Button*

    Renames the path code and environment names of the installed JD Edwards EnterpriseOne. This function is not available on the Deployment Server.
- *Saved Versions*

  The **Saved Versions** area contains a field that lists saved versions (also called snapshots). The example in this image shows the following versions: B9_Perf and E910_aaa.
- *Restore Button*

  Restores a saved version (snapshot) to a runnable JD Edwards EnterpriseOne installation.
- *Delete Button*

  Removes a saved version (snapshot).
- *Log File*

  Displays the log file path and name.
- *Close Button*

  Exits SnapShot.

Prerequisites

Before saving or restoring a JD Edwards EnterpriseOne installation using SnapShot, be sure that:

- You are signed into Microsoft Windows using an account with sufficient privileges (for example: read, write, execute) to the registry and to the JD Edwards EnterpriseOne installation and saved directories.
- All JD Edwards EnterpriseOne programs are closed.
- No applications (for example, Windows Explorer) have a file or subfolder open in either the installation or the saved directory or one of their subdirectories.
- The `SnapShot.exe` that you are running is not in either the installation or the saved directory or one of their subdirectories.

- The database(s) that both the JD Edwards EnterpriseOne installation and the saved SnapShot use is installed and running.
- Ensure that the Oracle product JDeveloper is not running.

Additional considerations:

- Before installing a new Deployment Server into a new Oracle Home, make sure you do not have any previous versions in the **Existing Version** field of SnapShot. All versions must be saved and should appear in the **Saved Versions** field.

Using SnapShot on the Deployment Server

To use SnapShot with multiple releases of the JD Edwards EnterpriseOne applications, you must use the most current version of SnapShot when switching between different releases of JD Edwards EnterpriseOne. For example, if you install the foundation code for both JD Edwards EnterpriseOne Applications *Release 9.2* and Applications Release 8.12, you must use the version of SnapShot corresponding to the most current JD Edwards EnterpriseOne tools release, in this case, *Tools Release 9.2* .

This section describes these tasks:

- *Starting SnapShot*
- *Saving a Snapshot*
- *Restoring a Snapshot*
- *Deleting a Snapshot*

Starting SnapShot

> **CAUTION:** Be sure to follow the guidelines in the preceding section of this guide entitled:
>
> - *Minimizing Locked Files*

> **Note:** You do not have to right click on the `SnapShot.exe` icon and select **Run as administrator**. This is because this version of `SnapShot.exe` is designed to automatically attempt to start with the elevated permissions. If you are not signed into Windows with an administrative account, you will be prompted to enter the credentials for an administrative account.

The SnapShot utility is delivered with the installation of both the JD Edwards EnterpriseOne Deployment Server and the Development Client workstations. It is located in this directory:

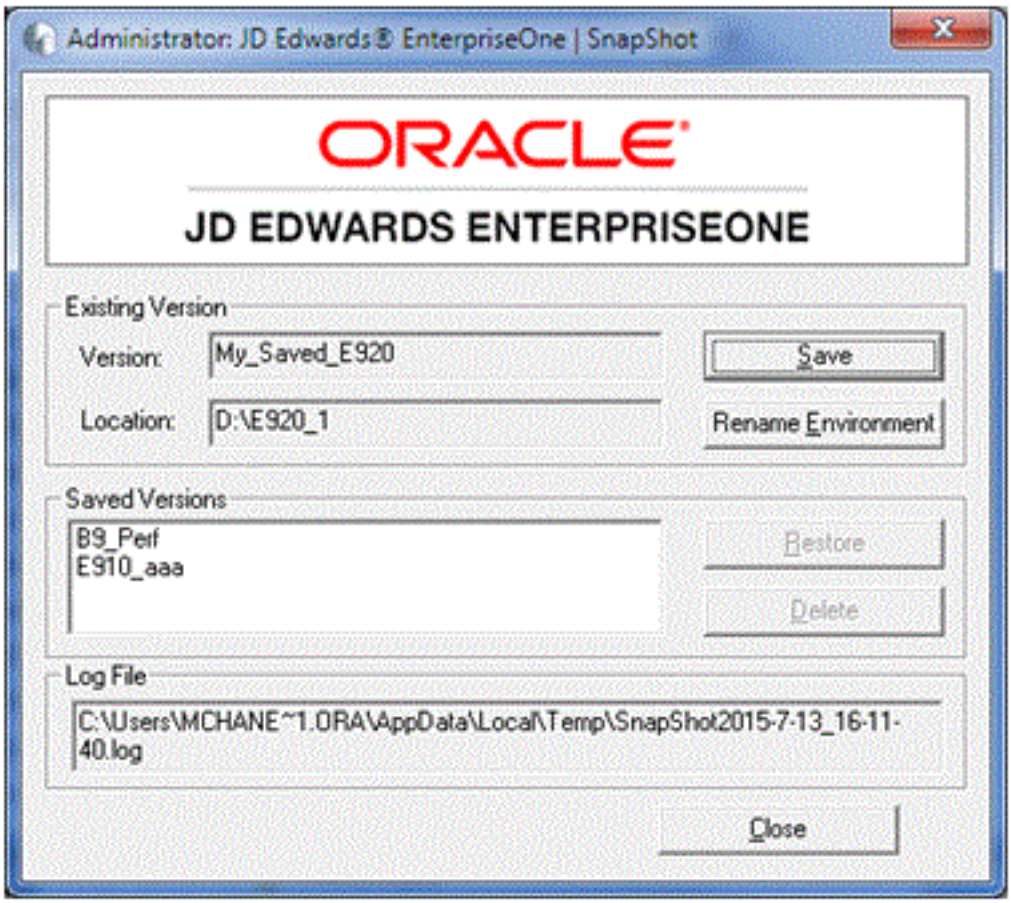
```
<JDE_dep_server_or_dev_client_installation_directory>\System\bin64 or bin32
```

To run the most current version of SnapShot for multiple releases, you must copy the `SnapShot.exe` program to a directory outside the Deployment Server installation directory. For example, the installation directory might be `c:\E920`. If you attempt to run SnapShot from within the Deployment Server installation directory (for example, `c:\E920`) the application will display an error message.

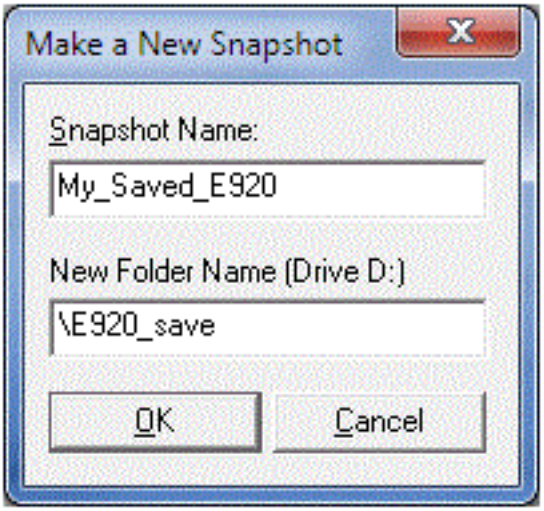
> **Tip:** You can create a shortcut to `SnapShot.exe`, but you still must copy `SnapShot.exe` to a location outside the original installation directory where it was delivered.

Saving a Snapshot

To save a snapshot:



1. On the main SnapShot window, click the Save button.



2. On Make a New Snapshot, complete this field:

   - *Snapshot Name*

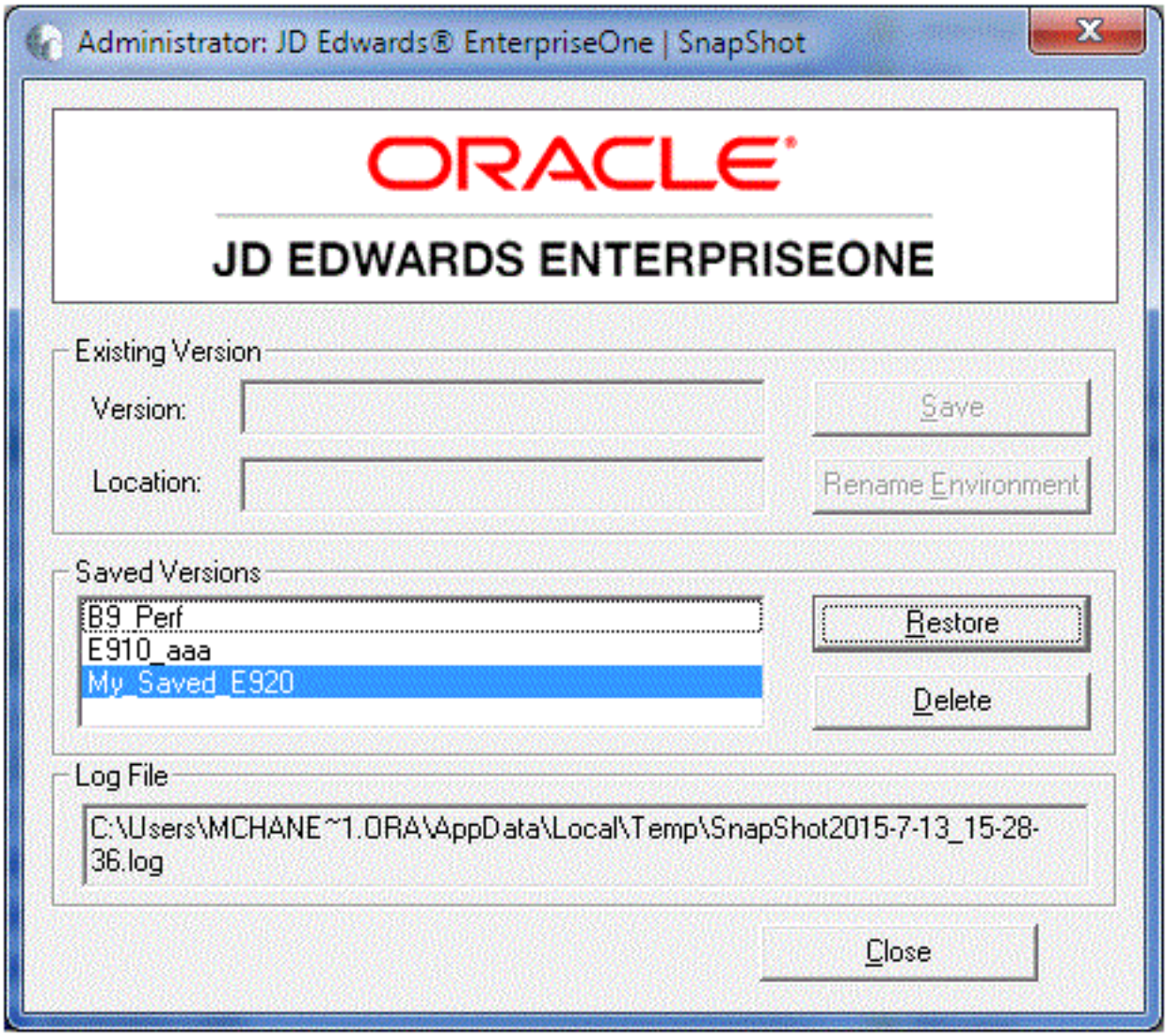
     Enter a name for the snapshot that will be saved. You may choose any name with the exceptions that the name cannot be empty and it cannot contain a backslash ('\'). A recommended scheme is to make it release specific, for example, Xe, B9, or E920.
   - *New Folder Name*

     SnapShot does not rename or move the installed directory on the Deployment Server; therefore, you cannot change the value in the New Folder Name field.
3. Click the **OK** button.

   SnapShot saves a snapshot of the Deployment Server.

Restoring a Snapshot

To restore a snapshot:

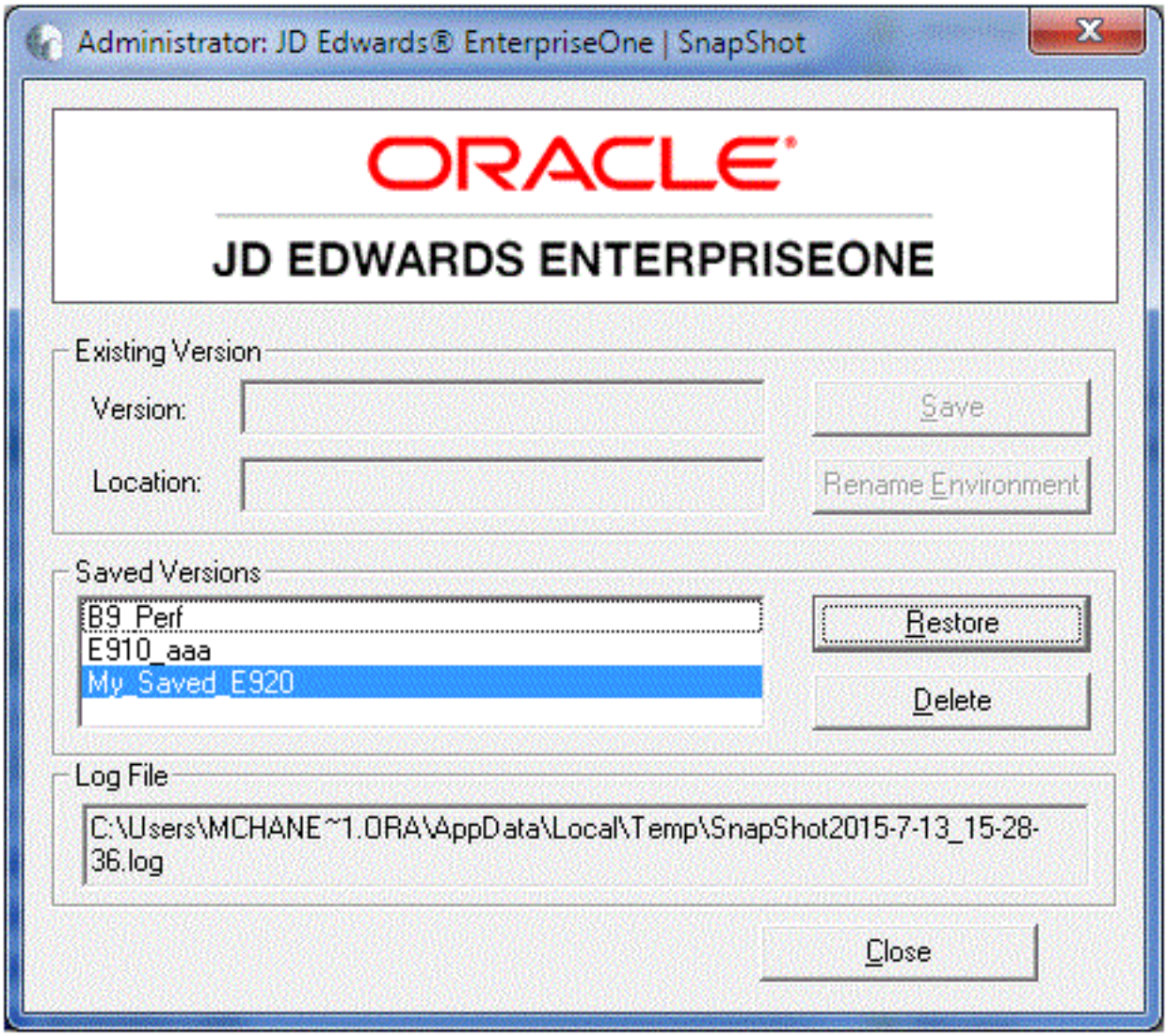


1. On the main SnapShot window and in the Saved Versions section, highlight the saved instance that you want to restore.
2. Click the **Restore** button.

   SnapShot restores the selected instance of JD Edwards EnterpriseOne and makes it active.

Deleting a Snapshot

To delete a snapshot:



1. On the main SnapShot window and in the Saved Versions section, highlight the instance that you want to delete.
2. Click the **Delete** button.

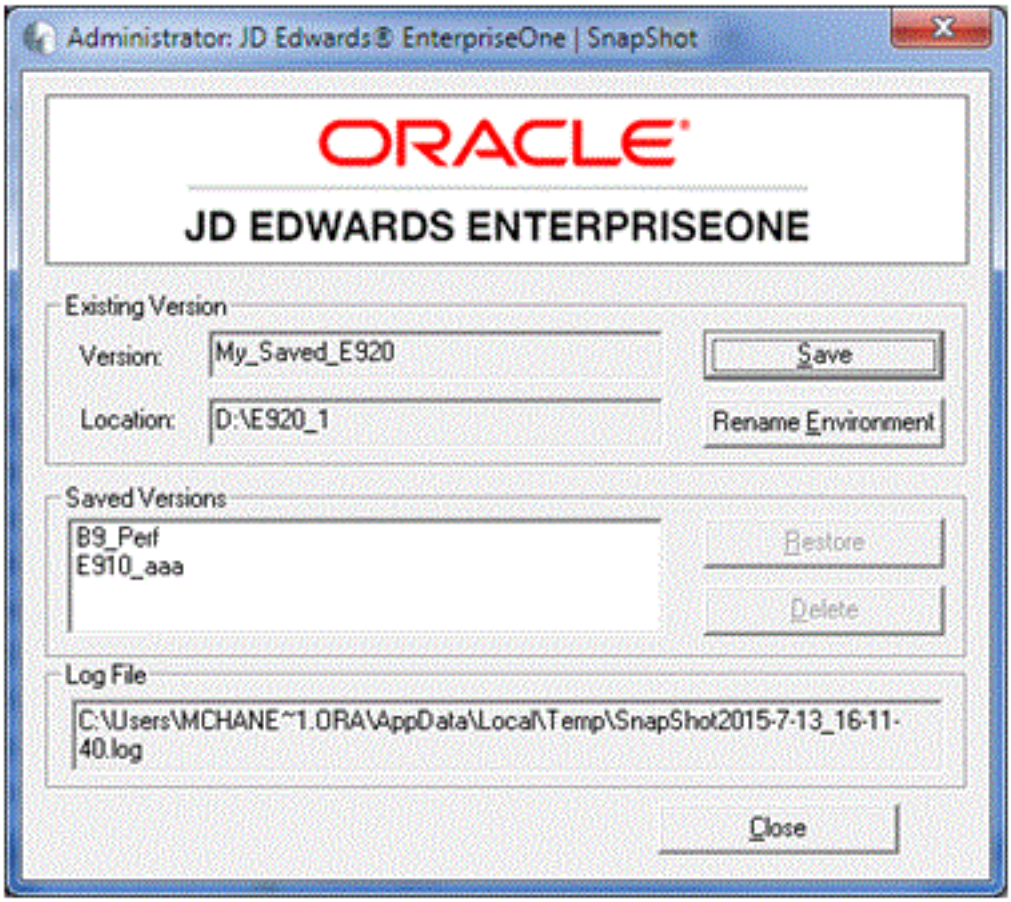
   SnapShot completely removes the saved instance from the JD Edwards EnterpriseOne Deployment Server.

Renaming an Environment

You can use SnapShot to rename the installed JD Edwards EnterpriseOne environment and path code. An example of a use case for this functionality is when a JD Edwards EnterpriseOne administrator needs to promote one path code/environment combination to another.

Note: When you use SnapShot the program does not update the database tables with the new path code and environment names. Instead, it simply updates the path code directory name and occurrences of the path code and environment in various files (for example, `jde.ini`, `jdbj.ini`, `jas.ini`, and `pathcode.inf`). If the new path code or environment does not exist in the JD Edwards EnterpriseOne system tables, you will need to update those tables yourself either through JD Edwards EnterpriseOne or by using a database editing tool such as Oracle SQLPlus.

To rename an environment in SnapShot:



1. On the main SnapShot window, ensure that an existing version of JD Edwards EnterpriseOne is active.
2. Click the **Rename Environment** button.
3. On Rename Environment, use the **Select an installed path code to change** drop-down to select the path code that you want to rename.

   Note: The **Old Values** section on the left side of the window displays old path code and environment names.
4. In the New Values section of the window, enter the new names for the environment you want to rename.
5. Click the **OK** button.

   Note: The section of the screen entitled **Select a view to change to** is available only to JD Edwards EnterpriseOne internal application developers.

Manually Backing Up Files and Settings

Once you have saved a snapshot for the Deployment Server, it is a good idea to manually backup the files to ensure that your settings for that Deployment Server are secure.

The files and subdirectories that need to be backed up are found in the folder of each saved snapshot. You should backup these files and subdirectories only while the **SnapShot.exe** program is not running. These files contain the information necessary to restore registry values and settings for each snapshot.

Set Logging for SnapShot Using the Registry

SnapShot always outputs the maximum amount of logging information. Furthermore, by design there is no way to turn off logging. You can specify the name and location of the log file that is generated by adjusting a single registry setting.

> **CAUTION:** Changes made to the Microsoft Windows registry happen immediately, and no backup is automatically made. Do not edit the Windows registry unless you are confident about doing so. Microsoft has issued the following warning with respect to the Registry Editor: "Using Registry Editor incorrectly can cause serious, system-wide problems that may require you to re-install Windows to correct them. Microsoft cannot guarantee that any problems resulting from the use of Registry Editor can be solved. Use this tool at your own risk."

By default, the SnapShot log is located in the `temp` directory of the Microsoft Windows user. This is the directory pointed to by the TEMP environment variable. You can determine the value of this TEMP variable in a number of ways including:

1. Control Panel > System
2. In a command prompt window, enter this command:

   ```
   echo %TEMP%
   ```
3. In the Address Bar of Windows Explorer, enter this string:

   ```
   %TEMP%
   ```

The default log file name is `SnapShot_<date-time>.log`.

Both the directory and name of the SnapShot log can be specified by editing the registry; however, the date-time stamp will always be inserted immediately before the period.

To change the log file directory and/or name:

1. Open the registry by clicking on Windows Start button and entering `regedit` in the search field.
2. Navigate to this node:

   ```
   \HKEY_LOCAL_MACHINE\Software\Wow6432Node\JDEdwards\Snapshot
   ```

   The first time that you run SnapShot, it creates a value (shown on the right-hand pane in regedit) called **LogFileName** with the default value **SnapShot.log**. You can change this value or, if it does not exist yet, you can add the value with name LogFileName, type string value, and the directory and/or file name of your choice.

3. The rules listed below are used for determining the final log file directory and name based on the value of LogFileName:

   - In all cases, the date and time that `SnapShot.exe` is run will be inserted immediately before the period in the extension of the file name.
   - If the value of **LogFileName** does not contain any backslashes, the name represents only the file name. The value of the Windows TEMP variable will be used as the directory.
   - If the value of **LogFileName** contains backslashes but does not start with a drive letter or a backslash, everything up to the last backslash represents a subdirectory or hierarchy of subdirectories below the directory designated by the Windows TEMP variable. Everything after the last backslash is the file name.
   - If the value of **LogFileName** starts with a backslash, it represents a subdirectory below the root directory of the drive (that is, directory "\"). The drive letter is derived from the drive specified in the Windows TEMP variable.
   - If the value of **LogFileName** starts with a letter followed by a colon and backslash, the letter is considered the drive letter where the log will reside. Any other applicable rule in the preceding rules above then apply.

   The following are examples of values of the **LogFileName** registry entry and the resulting log file path and name. In these examples, the assumed value of the TEMP variable is `C:\Users\John\AppData\Local\Temp` and assumes the log was created on June 5, 2012, at 3:46:9 PM.

Value for LogFileName	Resulting Log File
`SnapShot.log`	`C:\Users\John\AppData\Local\Temp\SnapShot_2012-6-5_15-46-9.log`
`MyFile.txt`	`C:\Users\John\AppData\Local\Temp\MyFile_2012-6-5_15-46-9.txt`
`MyTempDir\MyFile.txt`	`C:\Users\John\AppData\Local\Temp\MyTempDir\MyFile_2012-6-5_15-46-9.txt`
`MyTempDir\MyFile.txt`	`C:\MyTempDir\MyFile_2012-6-5_15-46-9.txt`
`D:\MyTempDir\MyFile.txt`	`D:\MyTempDir\MyFile_2012-6-5_15-46-9.txt`

4. Save the changes and exit the registry.

Troubleshooting

If SnapShot encounters a problem when trying to perform an action, it is designed to attempt to rollback the actions performed up to the point of failure. This means that during a save operation, SnapShot attempts to restore the JD Edwards EnterpriseOne instance back to a runnable state. If a failure occurs during a restore action, SnapShot attempts to resave the JD Edwards EnterpriseOne snapshot so you can correct the problem and then retry the action.

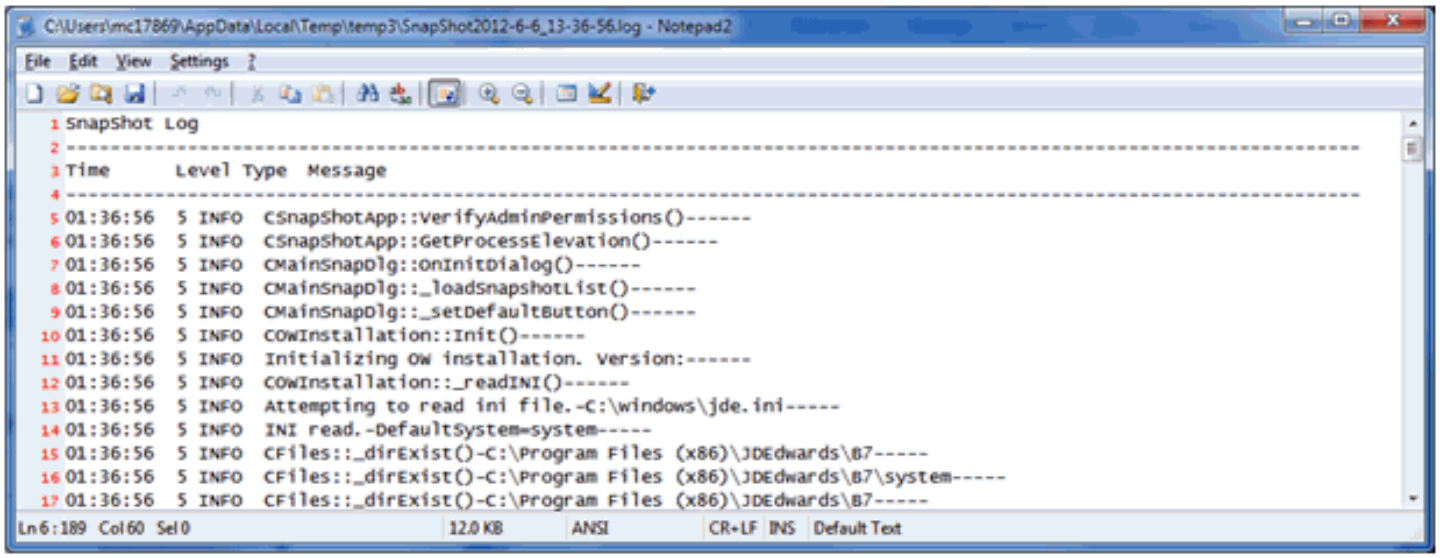
CAUTION: If a second error occurs when SnapShot is attempting to rollback changes, the error will likely result in either a non-runnable instance of JD Edwards EnterpriseOne or a saved snapshot that is corrupt. In either case, you will probably need to reinstall JD Edwards EnterpriseOne.

This section describes these topics:

- *Examining the Log File*
- *Remedial Actions*

Examining the Log File

When an error is encountered in SnapShot, the first thing that you should do is to carefully examine any on screen error messages for an indication of what went wrong and why. If you are not able to determine the cause and possible resolution of an error by examining the error message on the screen, examine the log file. As shown in the preceding screen example, the location and name of the log file is specified in the **Log File** section near the bottom of the main SnapShot window.



Above is an example of a SnapShot log file. When troubleshooting within the SnapShot log file, you should scan the **Type** column for a status of **ERR**, which indicates an error. If the error message itself is insufficient to isolate the cause of the problem, examine the messages immediately before and after the error for hints as to the root cause.

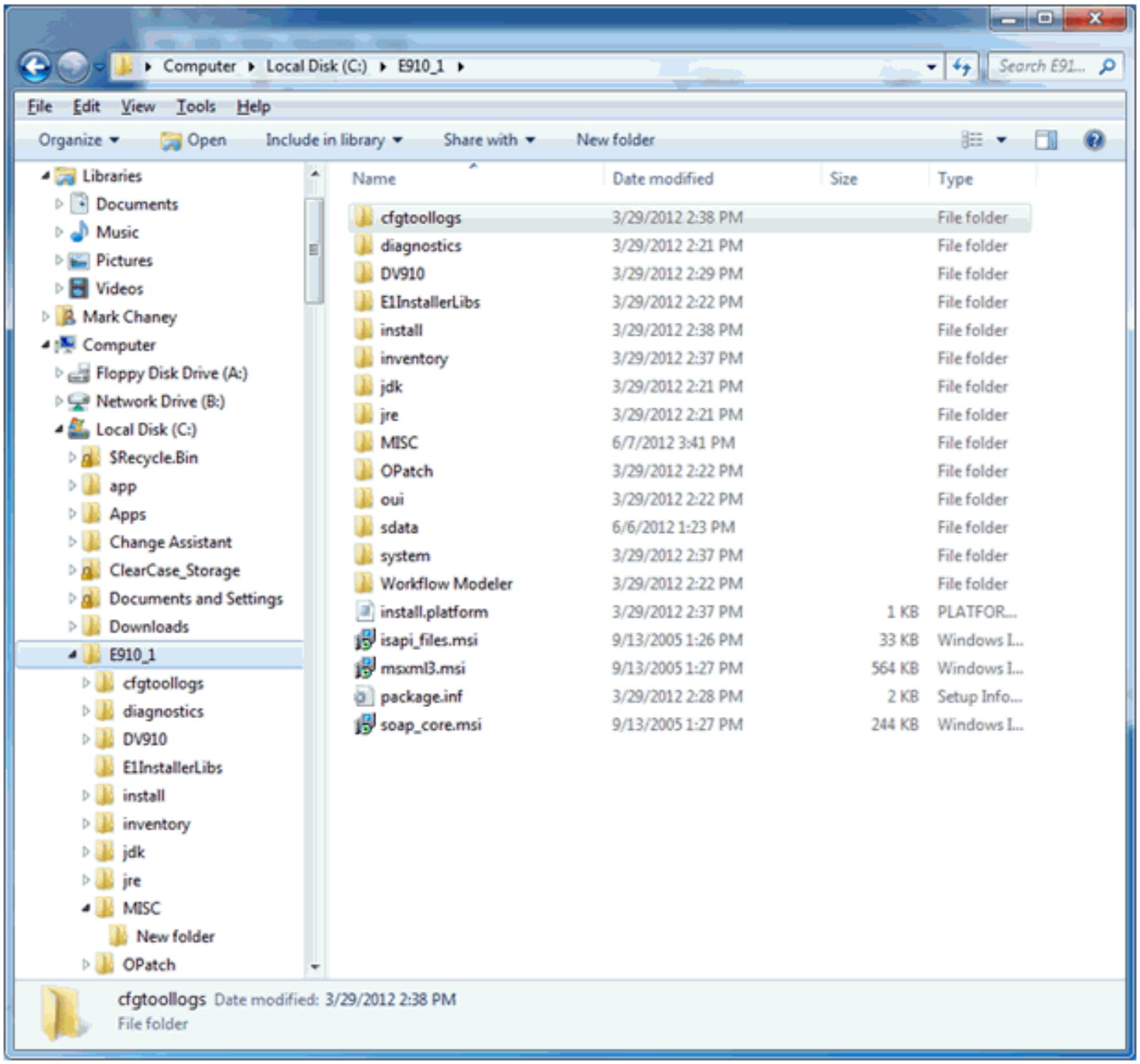
Remedial Actions

This section describes these topics:

- *Examples of Healthy Environments*
- *Simple Fixes*

Examples of Healthy Environments

This section illustrates a healthy JD Edwards EnterpriseOne installation and saved snapshot directories and registry settings.



The preceding shows the installation directory of a healthy JD Edwards EnterpriseOne instance.

In the preceding example screen, note that when you save a snapshot of the JD Edwards EnterpriseOne installation, the highlighted files and subdirectory shown above are created. Here are brief descriptions of the new files and subdirectory.

- *STARTMENU*.**

  The subdirectory that is prefixed with `STARTMENU` is the folder in the Start menu for the original installation.

- *DESKTOP*.**

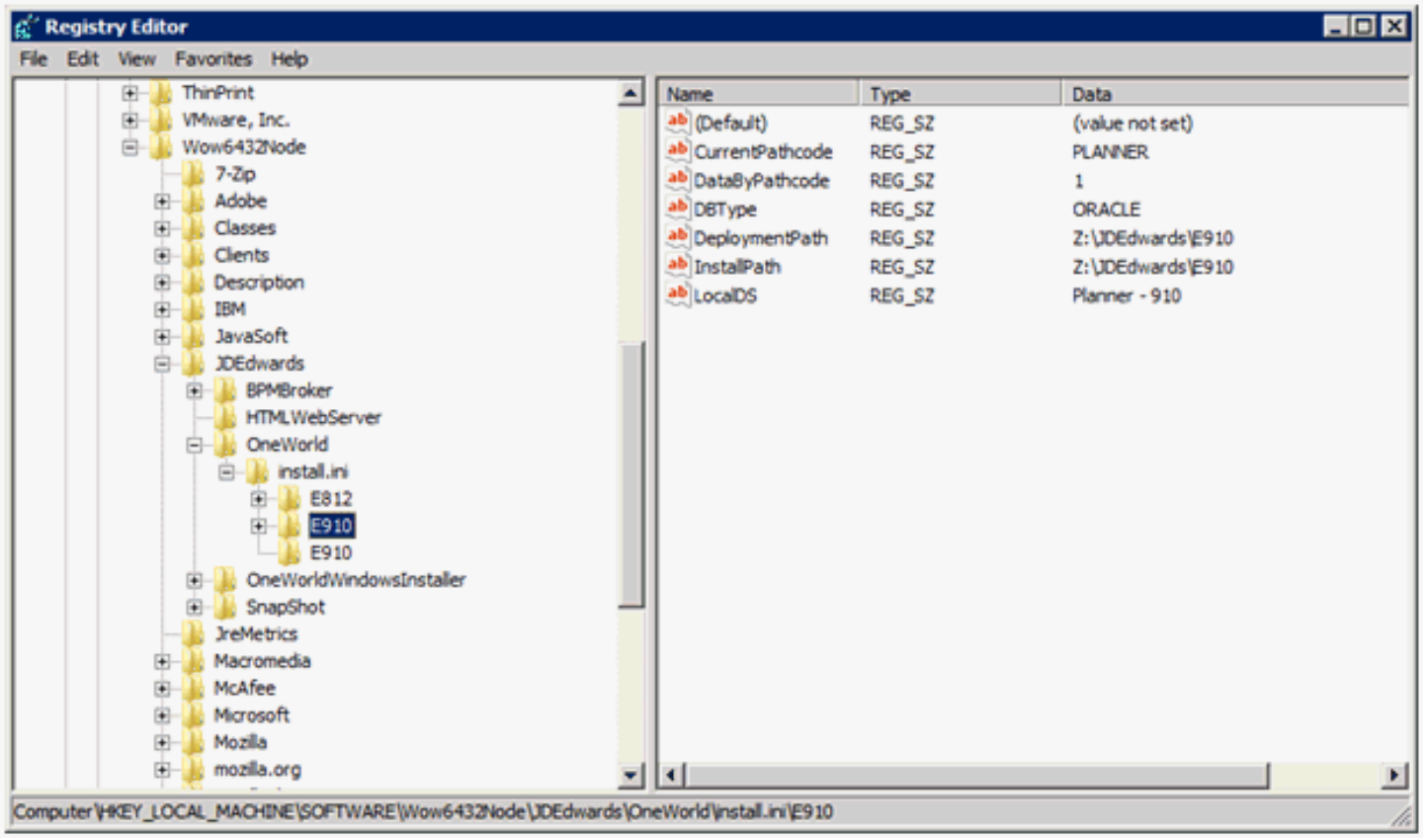
  The file that starts with `DESKTOP` is the shortcut to JD Edwards EnterpriseOne that was on the desktop of the workstation.

- *jde.ini*

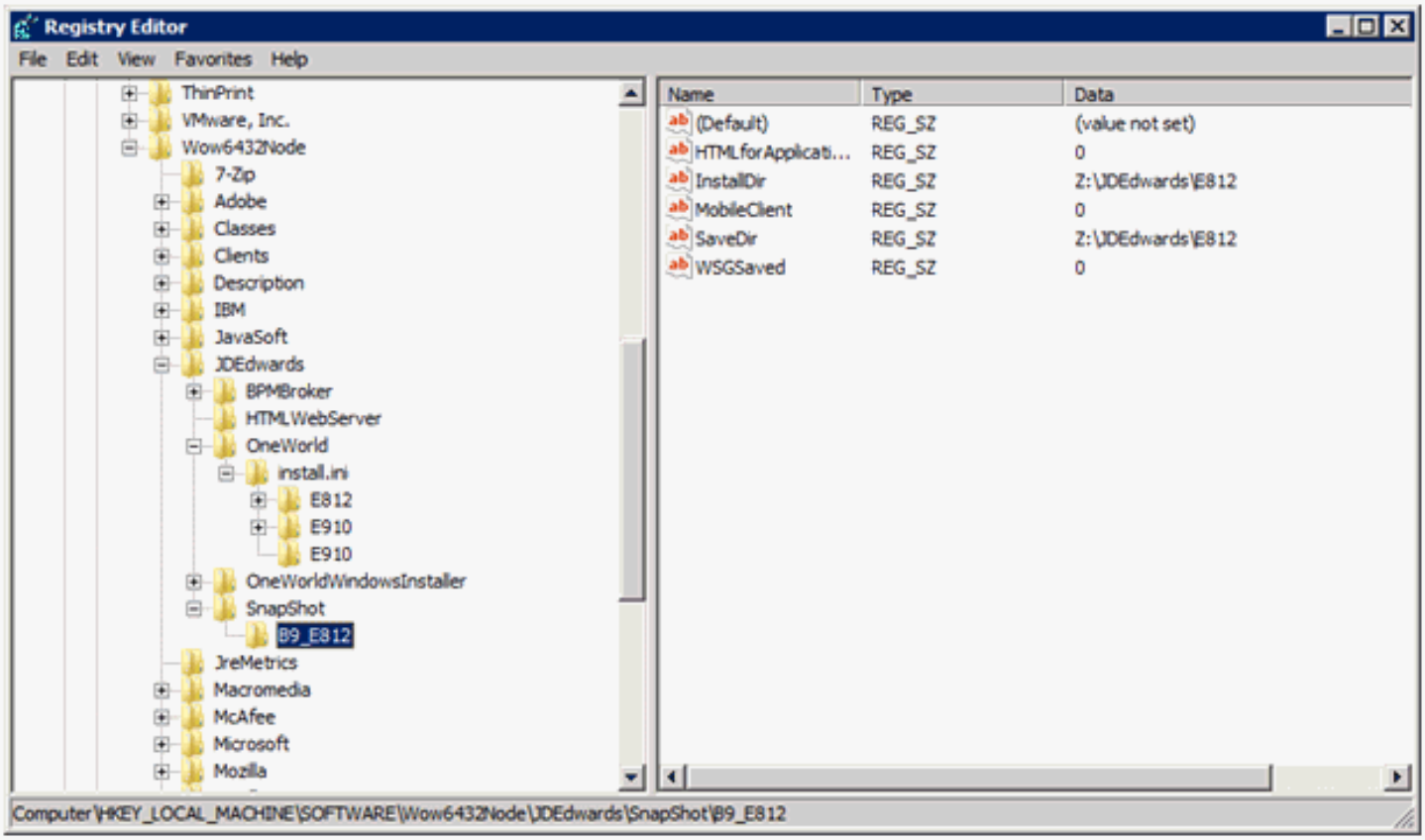
  The `jde.ini` file is the same file that was in the `c:\Windows` directory.

- *RegKeys*.**

  The files that are prefixed with `RegKeys` are binary files that contain the saved registry entries.



The preceding image shows the registry entries for a healthy JD Edwards EnterpriseOne Deployment Server installation.



The preceding image shows the registry keys for a saved JD Edwards EnterpriseOne snapshot.

Simple Fixes

Not all issues can be resolved by simple steps. However this section describes a few simple fixes for these conditions:

- *Missing Version Information*
- *Missing Saved Version*

Missing Version Information

If the **Existing Version** fields are empty when you run `SnapShot.exe` and you are certain that a JD Edwards EnterpriseOne instance is active, it is likely that this file is missing:

`c:\Windows\jde.ini`

If you happened to save a copy of the `jde.ini` file when you last saved a snapshot of this installation, you can copy that `jde.ini` file into the `c:\Windows` directory and rerun `SnapShot.exe`.

Missing Saved Version

If a saved snapshot is not listed in the **Saved Versions** field and you know that the saved snapshot exists, some registry entries are probably missing or pointing to the wrong directory.

To resolve this issue, perform these steps:

1. Edit the registry with `regedit.exe`.
2. Create a subkey under this path:

   ```
   \HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\JDEdwards\SnapShot
   ```

   The name that you give for the subkey should be the name of the saved snapshot.
3. Click on the subkey that you just created.
4. Create a string value called **SaveDir**.
5. Set the value data for this to the directory where the saved snapshot is stored.
6. Create a second string value called **InstallDir**.
7. Set the value data for this to the installation directory (that is, the directory in which the saved snapshot will be restored).

The following image illustrates properly configured registry entries.

5 Rebuilding Business Functions for Vertex Header Files (for Vertex users only)

Rebuilding Business Functions for Vertex Header Files (for Vertex users only)

This applies to customers who use the Vertex Quantum Payroll Tax and Vertex Sales Tax Q Series (formerly Quantum for Sales and Use Tax) applications with *JD Edwards EnterpriseOne* .

These are the business functions that you may need to rebuild after applying *Tools Release 9.2* .

Business Function	Description
B0700058	Establish connection to Vertex
B0000182	Initialize environment for vendor business functions
B0000183	Free environment for vendor business function
B7300004	Retrieve Quantum GeoCodes
B7300002	Validate Quantum GeoCodes
B7300012	Retrieve Quantum software information
X00TAX	Calculate and edit tax amounts

For more information about using Vertex Quantum applications with ERP, see *Configuring JD Edwards EnterpriseOne for Use with Vertex Quantum Tax Applications* in the *JD Edwards EnterpriseOne Applications Release Installation Guide* or *JD Edwards EnterpriseOne Applications Release Upgrade Guide*, for the applicable JD Edwards EnterpriseOne Applications Release.

Note: Refer to the *Understanding Certifications* to verify the supported versions of Vertex which vary according to platform.

6 Building Application Business Functions

Building Application Business Functions

If you are upgrading your *IBM i* to *JD Edwards EnterpriseOne Release 9.2* with *Tools Release 9.2* , you must make modifications to the **jde.ini** file. These changes are required to enable successful builds of applications business functions on the *IBM i* machine.

You must add the following *bold* sections to your existing **jde.ini** setting on your *IBM i JD Edwards EnterpriseOne* Enterprise Server.

```
[BSFN BUILD]
LinkLibraries=JDEUNICODE JDELIB JDEKRNL OWVER V_VERIFY XERCESWRAP
```

Note: In your **jde.ini** file, you should enter this command line for **CRTSRVPGM**.

```
[AS400]
CRTSRVPGM=CRTSRVPGM SRVPGM(%ls/%ls) MODULE(%ls/*ALL)
BNDSRVPGM(JDEUNICODE JDELIB JDEKRNL OWVER V_VERIFY
XERCESWRAP) EXPORT(*ALL) OPTION(*DUPPROC *DUPVAR *UNRSLVREF)
ALWLIBUPD(*YES) TGTRLS(*CURRENT)
```

Note: You should enter the above command line as a single contiguous string.

7 Re-Linking Business Functions

Re-Linking Business Functions

After you have used Server Manager to install your Tools Release to an *IBM i* -based Enterprise Server, as in previous *JD Edwards EnterpriseOne* Tools Releases, you must re-link the Business Functions.

For each path code, re-link the business functions located in the associated *IBM i* library. The program LINKBSFN is delivered with the installation of each Tools Release to simplify this process. The following commands must be executed against each full package for a specified path code.

1. Run LINKBSFN against the full package of the path code.

   ```
   LINKBSFN
   ```

   After you have entered the LINKBSFN command, press F4 to display the system prompt. Enter the package library name for the first parameter and the path code for the second parameter.
2. Wait until the previous step completes successfully before continuing to the next step. Repeat this step for every package and path code you are updating.

8 Appendix A - Working With Multiple Tools Release Foundations

As a best practice, you should always set up multiple tools release foundations, including a test environment for applying fixes to JD Edwards EnterpriseOne tools releases. This helps ensure that a stable environment exists before you promote fixes to your production environment. You can also use this same principle and procedure to set up multiple foundations of major tools releases, such as Tools Release 9.1 and Tools Release 9.2.

The procedures described in this document enable customers to target specific environments when they install a tools release. For example, customers can keep the production environment running on the existing foundation and apply the new tools release to all other environments. This configuration enables them to perform tests on the new foundation code without impacting the production environment. When they complete the testing, they can then apply the new tools release to production.

By completing the tasks below, you can configure your *JD Edwards EnterpriseOne* systems to handle many foundations on a single enterprise server. These tasks require that you reconfigure pathcodes and `jde.ini` files to set up multiple foundations, but once the initial tasks are completed, the new configuration requires very little maintenance.

Before you begin the installation, consider these factors:

- The multiple foundation setup uses one security server process per port.
- If you have multiple Enterprise Servers at different tools release levels, make sure each of them is pointing to the correct security data sources.
- If you have multiple foundations or multiple Enterprise Servers at different tools release levels, a dual maintenance of users and passwords is required.
- Before beginning installation of the *Tools Release 9.2* , check the associated `readme.txt` file for late-breaking information and requirements.

This appendix consists of the following tasks:

- *Understanding Multiple Foundation Code on the Enterprise Server*
- *Understanding Multiple Foundation Code on the Deployment Server*
- *Working with Packages for Multiple Foundations*
- *Working with the Development Client*
- *Editing the Development Client jde.ini File*
- *Promoting Foundation Code*
- *Setting up the Multiple jde.ini Files on the Deployment Server*

Understanding Multiple Foundation Code on the Enterprise Server

To install multiple foundations, you must first install and configure new system code on the Enterprise Server. This section includes these topics:

- *Working With Tools Releases on the Enterprise Server*
- *Registering an Existing Enterprise Server with Server Manager*
- *Understanding PORTTEST*

Note: If your *JD Edwards EnterpriseOne* configuration includes multiple servers running *JD Edwards EnterpriseOne* host code (for example, additional application servers or data servers), then you will need to apply the procedures for installing multiple foundations to each server. For this type of configuration to work successfully:

- Make sure that the port settings for the `[JDENET]` section in the `jde.ini` match between all servers running the same foundation.
- Ensure the new services are running on all servers.
- Run `PORTTEST` successfully on all servers for all foundation releases that you have installed.

Working With Tools Releases on the Enterprise Server

Refer to the *JD Edwards EnterpriseOne Tools Server Manager Guide* as described in the section of this guide entitled: *Understanding Server Manager and This Guide*.

This section discusses these topics:

- *Creating a New User and Group*
- *Creating a New Host Code Directory*
- *Creating a New Host Code Directory*
- *Creating an IFS Directory*
- *Editing the jde.ini File*
- *Editing the jde.ini File*
- *Editing the JDE.INI*
- *Deleting the Subsystem Entry*
- *Creating a Subsystem for the Service Pack*
- *Verifying that the Library List is Set Up Correctly*
- *Linking Business Functions to Service Packs*

Creating a New User and Group

The section discusses how to create a new UNIX user and group, which varies by platform.

- *Creating Users and Groups on Solaris Platforms*
- *Creating Users and Groups on RS/6000 Platforms*
- *Copying .oneworld and .profile from an Existing User to a New User*

Creating Users and Groups on Solaris Platforms

This section discusses how to create users and groups on a Solaris Enterprise Server.

- *Personnel*

  System administrator
- *Logon status*

  On the Solaris Enterprise Server, logged on as root.
- *Prerequisites*

  None
- *Concurrent Tasks*

  None

Use the following procedure to create a new UNIX administrator user and group on a Solaris Enterprise Server.

1. Log on to the UNIX Enterprise Server as root (super user). You can log on directly from a UNIX workstation or use `telnet` from the Microsoft Windows Deployment Server.
   - Solaris

     Run the `admintool` program and from the `Browse` menu, select `Groups`.
   - Solaris

     From the `Edit` menu, select `Add`. Enter the following value in the Group Name field:

     `pssprelease_number`

     where *release_number* is the Service Pack base release being installed. For example, for Service Pack 1 you would enter:

     `pssp1`
2. Click OK to display a list of groups.
3. Verify that the `pssprelease_number` group is listed.
   - Solaris

     From the `Browse` menu, select `Users`.
4. Select to Add users by following this procedure:
   - Solaris

     From the `Edit` menu, select `Add`.
5. Complete these fields:
   - Login Name

     Enter the value:

     `pssprelease_number`

     where *release_number* is the Service Pack base release being installed. For example, for Service Pack 1 you would enter:

```
pssp1
```

- Home Directory

  Enter the value:

  ```
  /home/pssprelease_number
  ```

  where *home* is the name of the home directory of the *JD Edwards EnterpriseOne* user you just created. The system may fill this field automatically.
- Primary Group

  Enter the same value as you entered for the Login Name field. For example:

  ```
  pssprelease_number
  ```

  where *release_number* is the Service Pack base release being installed. For example, for Service Pack 1 you would enter:

  ```
  pssp1
  ```
- Start-up Program

  Applicable to HP9000 only

  Enter this value:

  ```
  /usr/bin/ksh
  ```

  Alternatively, you can select from a list of available shells.
- Login Shell

  Applicable to Solaris only

  Enter this value:

  ```
  /usr/bin/ksh
  ```

  Alternatively, you can select from a list of available shells.

6. Click OK.
7. Specify password.
   - Solaris

     Select `Normal Password` from the drop down menu, enter a password, verify the password, and click OK.
8. Click OK.
   - Solaris

     Exit the `admintool`.

Creating Users and Groups on RS/6000 Platforms

This section discusses how to create users and groups on a RS/6000 Enterprise Server.

- *Personnel*

  System administrator

- *Logon status*

  On the RS/6000 Enterprise Server, logged on as root.

- *Prerequisites*

  None

- *Concurrent Tasks*

  None

Use the following procedure to create a new UNIX user and group on an RS/6000 Enterprise Server.

1. Log on to the RS/6000 as root (super user), or use `telnet` from the Microsoft Windows deployment server.
2. Run the `smit` program.
3. Select `Security and Users`.
4. Select `Group` and `Add a Group`.
5. Add a group called:

   `pssprelease_number`

   where *release_number* is the Service Pack base release being installed. For example, for Service Pack 1 you would enter:

   `pssp1`

   Use the default values for the remaining fields.
6. To return to the Security menu, press `F3` three times.
7. Select `Users` and `Add a User` and complete these fields:

   - User Name

     Enter the value:

     `pssprelease_number`

     where *release_number* is the Service Pack base release being installed. For example, for Service Pack 1 you would enter:

     `pssp1`
   - Primary Group

     Enter the value:

     `/home/pssprelease_number`

     where *home* is the name of the home directory of the *JD Edwards EnterpriseOne* user you just created. The system may fill this field automatically.
   - Primary Group

     Enter the same value as you entered for the Login Name field. For example:

     `pssprelease_number`

     where *release_number* is the Service Pack base release being installed. For example, for Service Pack 1 you would enter:

pssp1

- Home Directory

  Enter the value:

  /home/pssprelease_number

  where *home* is the name of the home directory of the *JD Edwards EnterpriseOne* user you just created. The system may fill this field automatically.
- Initial Program

  Enter this value:

  /bin/ksh

8. Press **Enter** to accept the values.
9. To exit, press **F3**.
10. To assign a password for **pssprelease_number**, enter this command:

    passwd pssprelease_number

Copying .oneworld and .profile from an Existing User to a New User

1. Go to the home directory of an existing UNIX user. For example, your user might be *psft900* .
2. Using this command, copy the **.oneworld** file to the home directory of the new user that you created in *Creating a New User and Group*:

   cp .oneworld .profile /home/user

   where *user* is the name of the new user you created in *Creating a New User and Group*.

Creating a New Host Code Directory

This section discusses how to create a host code directory on UNIX operating systems.

1. On the UNIX-based Enterprise Server, navigate to the directory where the *JD Edwards EnterpriseOne* host code is installed. For example:

   mount_point/JDEdwards/
2. Create a new directory for the service pack you want to test. For example, your directory name might be **E *910* SPTEST** or **E920_DVJDPY**.

   mkdir host_code_dir

   where *host_code_dir* is the name of the new host code directory.

   The new directory will be used to store both the new service pack and the path codes that will access the service pack.

Granting Ownership and Authority

Enter these commands to grant ownership and authority to the directory you created in *Creating a New User and Group*:

chmod 777 host_code_dir

chown user host_code_dir

chgrp group host_code_dir

Editing the .oneworld File

This section discusses how to change the `.oneworld` file.

1. Navigate to the home directory of the new user created above.
2. Edit the `.oneworld` file to update the `EVRHOME` path with the new host code directory. For example:

   ```
   export EVRHOME=/mount_point/JDEdwards/host_code_dir
   ```
3. If no `.oneworld` script exists check the `.profile` file for a call to the `enterpriseone.sh` script. Edit the `enterpriseone.sh` file to update the `EVRHOME` path with the new host code directory. For example:

   ```
   export EVRHOME=/mount_point/JDEdwards/host_code_dir
   ```

Moving the Path Codes to the New Directory

1. From the `mount_point/JDEdwards/E920` directory, determine the path codes that you want to access the new Tools Release. Move the selected path codes into the newly created directory using this command:

   ```
   mv path_code /mount_point/JDEdwards/host_code_dir
   ```

   where *path_code* is the path code selected to be moved where a valid value for path code might be PD920.

   For example, if you wanted to test the new Tools Release with all environments except Production (PROD, PD), move the Development (DEV,DV), Pristine (PRST920, PS) and Prototype (CRP, PY) path codes.

   Do not copy the path codes. Instead you should always move them. You should not associate more than one Tools Release or Service Pack with an environment at the same time.
2. Enter the following commands to grant ownership and authority to the path codes you moved:

   ```
   chmod 775 /mount_point/JDEdwards/host_code_dir/path_code
   chown user /mount_point/JDEdwards/host_code_dir/path_code
   chgrp group /mount_point/JDEdwards/host_code_dir/path_code
   ```

Creating Remaining Directories

This section discusses creating the remaining directories necessary to complete the install.

1. Logon to the Enterprise server as the new user created above.
2. Navigate to the new directory created above. For example:

   ```
   /mount_point/JDEdwards/host_code_dir
   ```

   In the new directory, create the following subdirectories with these exact case-sensitive names:
   - `ini`
   - `packages`
   - `PrintQueue`

Copying the jde.ini to the New Directory

1. Navigate to this directory:

   ```
   /mount_point/JDEdwards/e920/ini
   ```
2. Use the following command to copy the `jde.ini` to the host code directory:

   ```
   cp JDE.INI /mount_point/JDEdwards/host_code_dir/ini
   ```
3. Enter the following command to change the permissions of the file:

```
chmod 666 /mount_point/JDEdwards/host_code_dir/ini/JDE.INI
```

Creating a New Host Code Directory

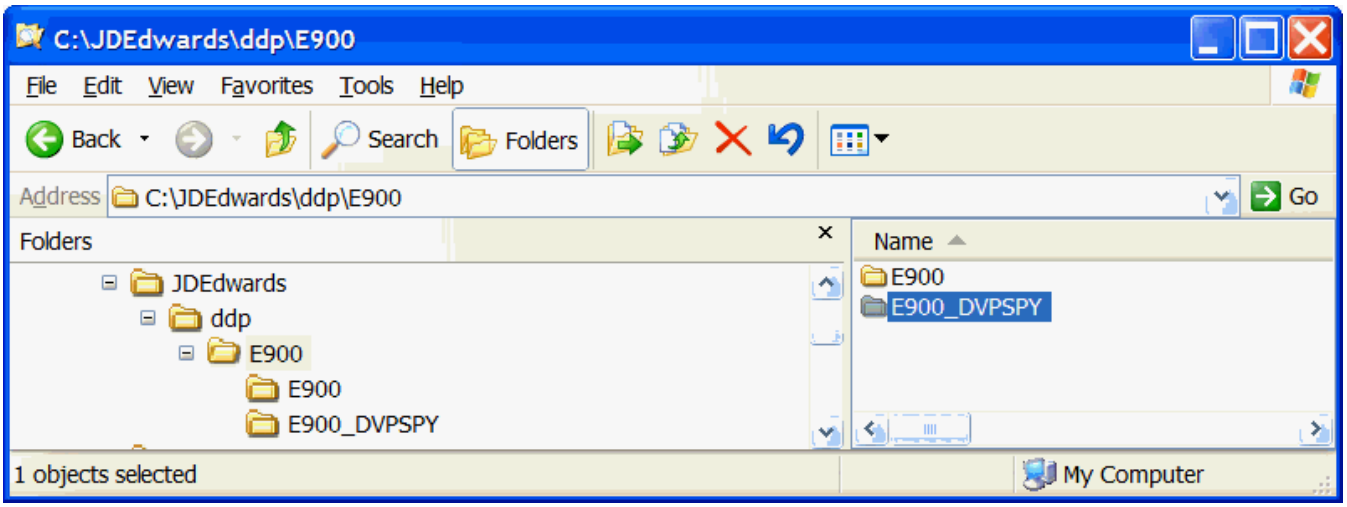
This section discusses how to create a new host code directory for an Enterprise Server running the Microsoft Windows operating system.

1. On the Enterprise Server, navigate to the drive and directory that contain the *JD Edwards EnterpriseOne* host code. For example:

   ```
   cd z:\JDEdwards\ddp
   ```

   where *z* is the drive on which the *JD Edwards EnterpriseOne* host code is installed.
2. Create a new directory for the tools release you want to test. For example, `E 910 TRTEST` or `E 910 _DVPTRY`.

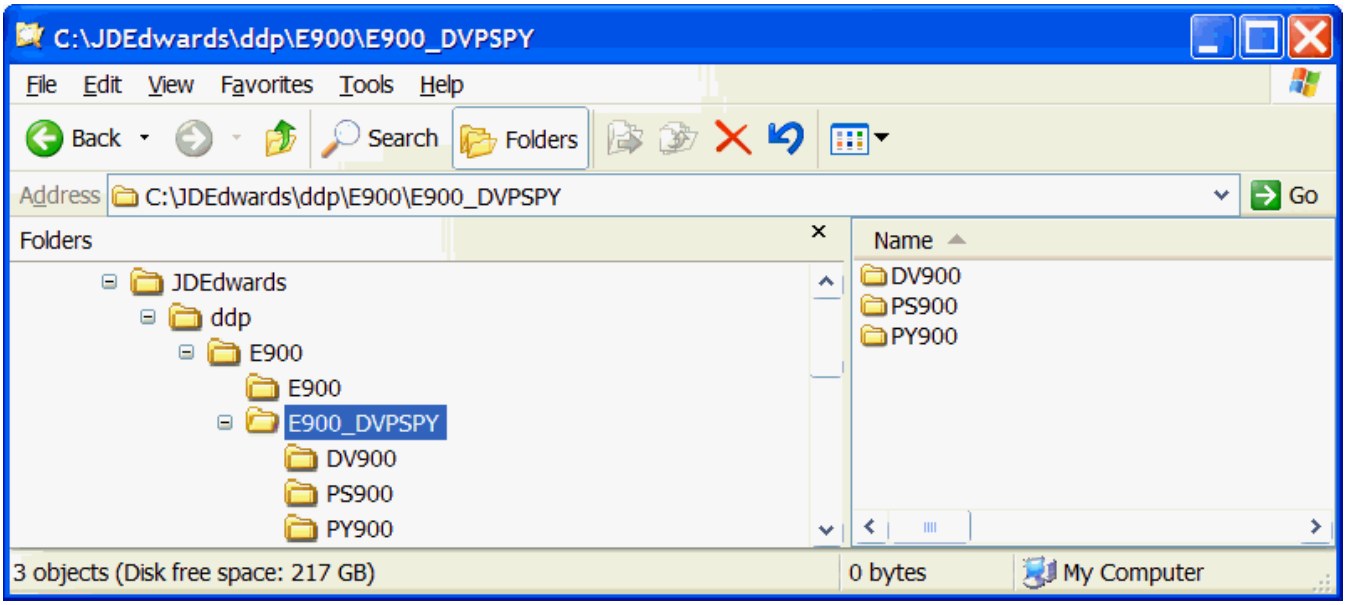
   The new directory will be used to store both the new tools release and the path codes that will access the tools release.



3. From the `\JDEdwards\ddp\E920` directory, select the path codes that you want to access the new tools release. Move the selected path codes into the newly created directory.

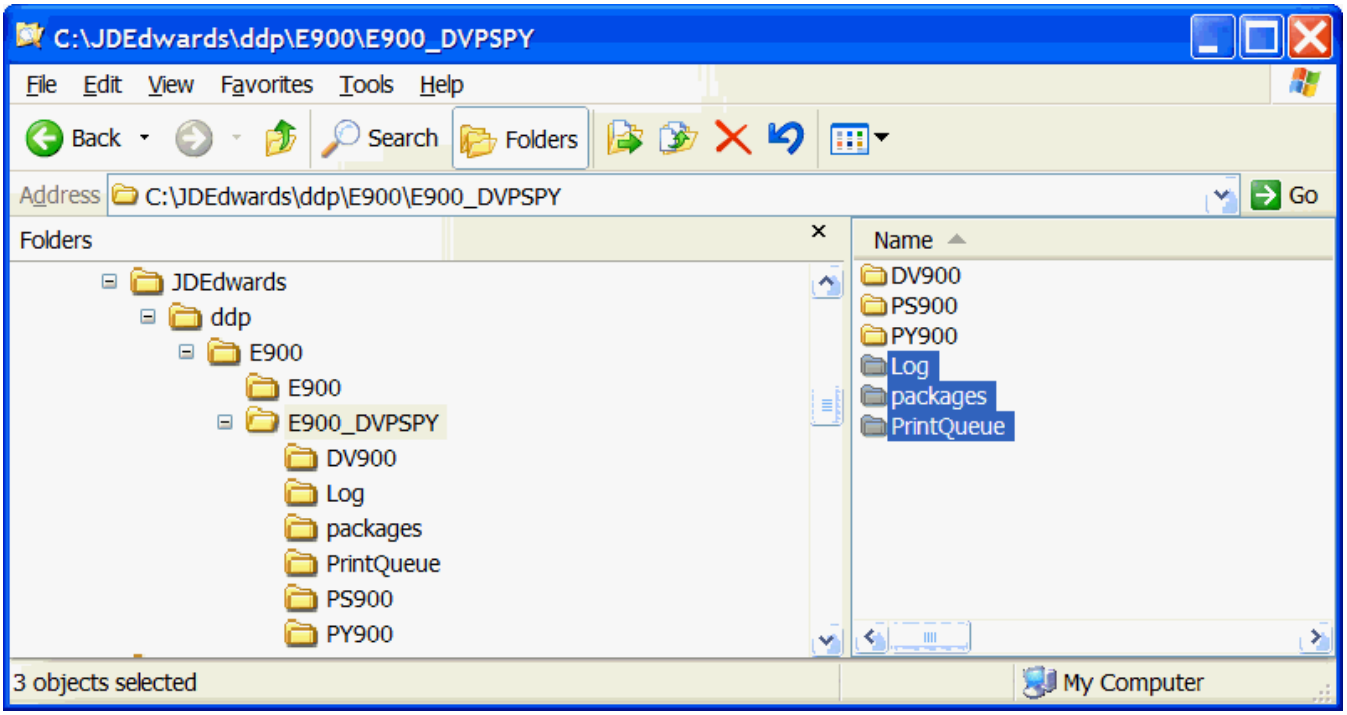
   For example, if you wanted to test the new tools release with all environments except Production (PROD, PD), move the Development (DEV,DV), Pristine (PRIST, PS) and Prototype (CRP, PY) path codes.

   Do not copy the path codes. Instead you should always move them. You should not associate more than one tools release with an environment at the same time.



4. In the new directory, create the following subdirectories:

   - **log**
   - **packages**
   - **PrintQueue**



Creating an IFS Directory

Create a new IFS directory for storing the JDE logs for the new service pack by entering the following command:

```
MD DIR(logdirectory)
```

where *logdirectory* is the name of the IFS directory for storing the JDE logs. For example, your directory name might be **PSE *910* SP**.

Editing the jde.ini File

This section discusses how to make modifications to the **jde.ini** file that is located on a UNIX Enterprise Server.

1. On the Enterprise Server, navigate to the **/ini** directory under the new host code directory. The file is typically located in this directory:
   /mount_point/JDEdwards/host_code_dir/ini
2. Edit the **jde.ini** file to change the paths in each of the entries below to point to the new host code directory:

   Note: Replace *host_code_dir* in the entries below with the name of the new host code directory.

   ```
   [DEBUG]
   DebugFile=mount_point/JDEdwards/host_code_dir/log/jdedebug.log
   JobFile=mount_point/JDEdwards/host_code_dir/log/jde.log
   JDETSFile=mount_point/JDEdwards/host_code_dir/log/JDETS.log

   [INSTALL]
   B9=mount_point/JDEdwards/host_code_dir


   [BSFN BUILD]
   BuildArea=mount_point/JDEdwards/host_code_dir/packages
   ```
3. Locate the **[JDENET]** section and find the port settings for **serviceNameList** and **serviceNameConnect**. For each entry, add 100 to the number. For example, if the settings are both 6013, change them to 6113.
   ```
   [JDENET]
   serviceNameList=6113
   serviceNameConnect=6113
   ```
4. In the **[JDEIPC]** section, add 1000 to the **startIPCKeyValue** entry. For example:
   ```
   [JDEIPC]
   startIPCKeyValue=7000
   ```

   Note: If the **startIPCKeyValue** value is commented out, delete the comment mark to enable the setting.
5. In the **[SECURITY]** section, either clear security or edit **DefaultEnvironment** to reference a valid environment that accesses the new service pack.
6. Replace all references to the production environment (for example, PD920) with a valid environment that will access the new service pack (for example, PY920).

Editing the jde.ini File

This section discusses how to make the appropriate changes to the **jde.ini** file on a Microsoft Windows system.

1. Navigate to the **\system\bin32** directory under the new host code directory, and edit the **jde.ini** file.
2. Locate the **[DEBUG]**, **[INSTALL]**, and **[BSFN BUILD]** sections in the **jde.ini** file, and change the paths in each of these entries to point to the new host code directory:

   Note: Replace *host_code_dir* in the entries below with the name of the new host code directory.

   ```
   [DEBUG]
   DebugFile=d:\JDEdwards\ddp\host_code_dir\log\jdedebug.log
   JobFile=d:\JDEdwards\ddp\host_code_dir\log\jde.log
   JDETSFile=d:\JDEdwards\ddp\host_code_dir\log\JDETS.log

   [INSTALL]
   E920=d:\JDEdwards\ddp\host_code_dir
   ```

```
StartServicePrefix=JDEdwards Update x.x
```

where *x.x* is the tools release number. For example, 9.2.

```
[BSFN BUILD]
BuildArea=d:\JDEdwards\ddp\host_code_dir\packages
```

3. Locate the **[JDENET]** section, and find the port settings for **serviceNameList** and **serviceNameConnect**. For each entry, add 100 to the number. For example, if the settings are both 6013, change them to 6113.

```
[JDENET]
serviceNameList=6113
serviceNameConnect=6113
```

4. In the **[JDEIPC]** section, add 1000 to the **startIPCKeyValue** entry as shown below:

```
[JDEIPC]
startIPCKeyValue=7000
```

> **Note:** If the **startIPCKeyValue** value is commented out, delete the comment mark.

5. In the **[SECURITY]** section, either clear security, or edit **DefaultEnvironment** to reference a valid environment that accesses the new tools release.
6. Replace all references to the production environment (for example, PD812) with a valid environment that will access the new tools release (for example, PY920).

Editing the JDE.INI

This section discusses editing the **JDE.INI** on an *IBM i* server.

> **Note:** Do not use the A98OWMNU option to access the **JDE.INI**. This option takes you to the Production **JDE.INI**, not the service pack **JDE.INI**.

1. Edit the **JDE.INI** you just copied by entering the following command:

```
WRKMBRPDM FILE(syslib/INI)
```

where *syslib* is the name of the new system library. For example, B9SPSYS.

2. Enter Option 2 to Edit the file.
3. Locate the **[DEBUG]** and **[INSTALL]** sections in the **JDE.INI**, and change the paths in each of the entries below to point to the new IFS directory and Library respectively:

```
[DEBUG]
DebugFile=logdirectory/JDEDEBUG
JobFile=logdirectory/JDE.LOG
JDETSFile=logdirectory/JDETS.LOG

[INSTALL]
DefaultSystem=syslib
```

4. Locate the **[JDEIPC]** section and change the **startIPCKeyValue**. This value should be an unused start key not within the IPC range of another instance. For example, if the IPC value of another instance is 5000, then you should set the **startIPCKeyValue** to 7000.

```
[JDEIPC]
startIPCKeyValue=7000
```

5. Locate the **[JDENET]** section, and find the port settings for **serviceNameList** and **serviceNameConnect**. For each entry, add 100 to the number. For example, if the settings are both 6013, change them to 6113.

   ```
   [JDENET]
   serviceNameListen=6113
   serviceNameConnect=6113
   ```

6. In the **[SECURITY]** section, either clear security or edit **DefaultEnvironment** to reference a valid environment that accesses the new Tools Release.
7. Replace all references to the production environment (for example, PD9) with a valid environment that will access the new service pack (for example, PD812).

Deleting the Subsystem Entry

The service pack installation creates an entry for the new service pack under the existing subsystem. This is not necessary, as a new subsystem will be created for the new service pack.

1. To delete the entry, type the following command:

   ```
   WRKSBSD SBSD(syslib/*ALL)
   ```

2. Enter Option 4 next to the entry that includes the new *syslib* and delete the entry.

Creating a Subsystem for the Service Pack

1. Ensure you are signed on as QSECOFR before performing these steps.
2. Add the **SYSTEM** library to your library list.

   ```
   ADDLIBLE syslib
   ```

3. Enter these commands, depending on your Tools Release:

   ```
   CHGCMD CMD(CRTOWSBS) PGM(*LIBL/CRTOWSBS)

   CRTOWSBS SUBSYSTEM(subsystemname) SYSLIB(syslib)
   ```

   where *subsystemname* is the name of the new subsystem for the newly installed Tools Release. For example, **PSE *910* 1SP**.

   > **Note:** The **CALL PGM** command requires four parameters, and the last three parameters should be the new **SYSTEM** library.

Verifying that the Library List is Set Up Correctly

Sign onto your *IBM i* as ONEWORLD. To ensure the library list is set up correctly, verify that it contains the newly created **SYSTEM** library. For example, E *910* PSYS.

Linking Business Functions to Service Packs

For each *JD Edwards EnterpriseOne* path code you want associated with the new service pack, re-link the business functions located in the associated *IBM i* library. The following commands must be executed against each full package for a specified path code:

1. Run **LINKBSFN** against the pathcode's full package. Type **LINKBSFN**, then press **F4** to display the system prompt. Enter the package library name for the first parameter and the pathcode for the second parameter.
2. Wait until the previous step completes successfully before continuing to the next step.
3. Repeat this step for every package and pathcode you are updating.

Registering an Existing Enterprise Server with Server Manager

Any Enterprise Server running Tools Release 8.97 and greater should be registered with Server Manager. For details, refer to the section entitled: "Register or Create a JD Edwards Enterprise Server as a New Managed Instance" in the *JD Edwards EnterpriseOne Tools Server Manager Guide* .

Understanding PORTTEST

To ensure that the new tools release is properly installed on the Enterprise Server, complete the following task to run PORTTEST on all environments accessing the new tools release.

Note: If Security Server is on, then *JD Edwards EnterpriseOne* services must be running before PORTTEST can be run.

This section contains the following topics:

- *Running PORTTEST*
- *Running the PORTTEST*
- *Running the PORTTEST*
- *Starting the Services*
- *Starting the Services*
- *Starting the Services*

Running PORTTEST

This section discusses how to run PORTTEST on a Microsoft Windows Enterprise Server.

1. On the Enterprise Server, open a command prompt window and change to the drive that contains the *JD Edwards EnterpriseOne* host code.
2. Enter these commands:

   ```
   cd \JDEdwards\ddp\host_code_dir\System\bin32

   PORTTEST userid password environment
   ```

   where *userid* is a valid *Tools Release 9.2* user, *password* is the password for that user, and *environment* is the environment you are verifying. These parameters are case-sensitive.
3. Repeat the PORTTEST command for each environment accessing the new tools release.

Running the PORTTEST

This section discusses the procedure to run the PORTTEST command on a UNIX Enterprise server.

1. Log on the Enterprise Server as the new user that you created in *Creating a New User and Group*.
2. Enter these commands:

   ```
   cd mount_point/JDEdwards/host_code_dir/system/bin32

   PORTTEST userid password environment
   ```

   where *userid* is a valid *Release 9.2* user, *password* is the password for that user, and *environment* is the environment you are verifying. These parameters are case-sensitive.
3. Repeat the PORTTEST command for each environment accessing the new service pack.

Running the PORTTEST

This section discussing running the PORTTEST application on an *IBM i* Enterprise Server.

1. Sign on to the *IBM i* as ONEWORLD.
2. On Current *Release 9.2* Versions, select the newly installed instance of *JD Edwards EnterpriseOne* .
3. Go to the *Release 9.2* Menu by entering the following commands:

   ```
   ADDLIBLE JDEOW
   GO JDEOW/A98OWMNU
   ```
4. Choose option 4 to run PORTTEST.
5. Repeat the PORTTEST command for each environment accessing the new service pack

Starting the Services

This section discusses how to manage **Services** on a Microsoft Windows Enterprise Server.

1. On the Enterprise Server, open a command prompt window and change to the drive that contains the *JD Edwards EnterpriseOne* host code.
2. Enter the following commands:

   ```
   cd \JDEdwards\ddp\host_code_dir\system\bin32
   jdesnet -i
   exit
   ```
3. From the Microsoft Windows Control Panel, open **Services**.
4. Select **Tools Release Update x.xx E920 Network** services.
5. Click **Startup**.
6. Select **Manual**.
7. Click **This Account**.
8. If your default printer is a local printer, enter a local user name and password. If your default printer is a network printer, enter a network user name and password.
9. Click OK.
10. Select **Tools Release Update x.xx E920 Network Services**, and click Start.

Starting the Services

This section discusses how to manage **Services** on a UNIX Enterprise Server.

1. Navigate to the drive and directory that contains the *JD Edwards EnterpriseOne* host code. For example:

   ```
   cd /mount_point/JDEdwards/host_code_dir/system/bin32
   ```
2. Enter the following command:

```
./RunOneWorld.sh
```

Starting the Services

This section discusses how to manage **Services** on an *IBM i* Enterprise Server.

1. Sign on to the *IBM i* as OneWorld.
2. On Current *Release 9.2* Versions, select the newly installed instance of *JD Edwards EnterpriseOne* .
3. Enter these commands:

   ```
   ADDLIBLE JDEOW
   GO JDEOW/A98OWMNU
   ```

4. Choose option 6 to end JDE server.
5. Choose option 7 to clear the IPC.
6. Start the Enterprise Server using these options:
   a. Choose option 5 to start the JDE server.
   b. Choose option 8 to display active jobs.
7. Verify that the entry NETWORK is running with a PGM-MONITOR in SELW status.
8. Verify that the entry SENTINEL is running with a PGM-MONITOR in SIGW status.

Note: Until you perform a net request, the CPU is 0 (zero).

Understanding Multiple Foundation Code on the Deployment Server

This chapter discusses:

- *Installing Multiple Tools Releases on the Deployment Server*
- *Adding a New Machine Record for the New Tools Release*
- *Modifying the Existing Machine Record to Remove Environments*

Installing Multiple Tools Releases on the Deployment Server

1. Sign on to the Deployment server using the JDE user ID.
2. Make sure that all *JD Edwards EnterpriseOne* processes (for example, Solution Explorer) are not running.
3. On the Deployment server, navigate to this directory:

   `z:\JDEdwards\E920`

   where *z:* is the drive on which the *JD Edwards EnterpriseOne* host code is installed.
4. In the `E920` directory, create a directory called

   `System_TR_Original`

   where *TR_Original* is the release number of the tools release currently installed. For example, `System_920`.
5. From the `\JDEdwards\E920` directory, copy (**not** move) the following directories to the new `System` directory (`System_TR_Original`) that was just created. This will serve as backup or rollback location for the previous release.

   `System`

   `SystemComp`

   `OneWorld Client Install`

   CAUTION: Ensure you do not move these directories; they are required in both locations. This is because the installer needs to save a copy of the `jdbj.ini` and `jas.ini` files so you can relocate them in the new system directory.

   Note: If you get a warning message, it indicates files are locked. You need to ensure that *JD Edwards EnterpriseOne* or any other application is not running and causing a lock on any file.

6. Install a Management Agent on the Deployment Server. This should be the Tools Release 9.2 version of the Microsoft Windows version of the JD Edwards EnterpriseOne Management Agent.

   Refer to this chapter in the Server Manager Guide:

   Install a Management Agent

   > **Note:** The installer for the Management Agent requires a 32-bit JRE. You can use the same JRE that you used to install the Deployment Server using the OUI installer.

7. Register the Deployment Server to the Server Manage Console.

   Refer to this chapter in the Server Manager Guide:

   Register a JD Edwards Deployment Server as a New Managed Instance

8. Obtain and deploy the Tools Release software component for the Deployment Server.

   Refer to this chapter in the Server Manager Guide:

   Managed Software Components

   > **Note: Changing Tools Releases on the Deployment Server.** When you use Server Manager to install the Software Component for a Tools Release prior to 9.2 on a Deployment Server, these two directories (which are the entire contents of the Tools Release portion of the Deployment Server) are backed up in a `.jar` file:
   >
   > - `\System`
   > - `\OneWorld Client Install`
   >
   > Therefore, when you use Server Manager to change a Tools Release prior to 9.2 on the Deployment Server, only those directories are restored. For Tools Release 9.2 and greater, the Server Manager install of the Tools Release performs differently because the contents are already stored in a `.jar` file where they can be extracted. In all cases, whenever Server Manager installs or changes a Tools Release, additional special files such as certain `.ini` and `.html` files are backed up. To fully preserve a Deployment Server, you should manually perform a full backup of the machine.

9. In the `E920` directory, create another new directory called `System_TR_New` where `TR_New` is the release number of the tools release being installed. For example, `System_91_A1`.
10. Move the newly installed `System`, `SystemComp`, and `OneWorld Client Install` files and directories into the new directory:

    ```
    \JDEdwards\E920\System_TR_New
    ```

    For example, `\JDEdwards\E920\System_91_A1`.
11. Restore the original directories by copying the `System`, `SystemComp`, and `OneWorld Client Install` directories from `System_TR_original` to the base installation directory `x:\JDEdwards\E920`.
12. Locate the `jde.ini` in this location:

    ```
    \JDEdwards\E920\System_TR_New\OneWorld Client Install\Misc
    ```
13. Modify the above-located `jde.ini` to change the port to that on which you wish to run the multiple foundation. For example, valid ports might be 6014, 6015, 6016, 6017, etc.
14. Tools Releases 9.2.9 and later: In the same `jde.ini` change the [LOCALWEB] webhostname and webport number. Ensure that they are different for multiple foundations.

Adding a New Machine Record for the New Tools Release

1. On the Deployment Server, log on to the deployment environment.
2. From the Fast Path menu, run P9654A.
3. On Work with Locations and Machines, click Find and expand the location tree.
4. Click Enterprise Servers and then click Add.
5. On Enterprise Server Revisions, complete the following fields:

Field	Description
Machine Name	Enter the name of your existing primary Enterprise Server.
Description	Enter a description for the machine.
Release	Enter E920.
Host Type	Click the visual assist button and select the type of Enterprise Server.
Port Number	This should match the port number previously set in the new tools release `jde.ini`
Database Type	Click the visual assist button and select the type of database being used on the Enterprise Server.
Installation Path	Enter the installation directory path in which *Release 9.2* is installed on the Enterprise Server.
Installation Path	Enter the installation library in which *Release 9.2* is installed on the Enterprise Server. For example: `SYSLIB`

6. Click OK.

> **Note:** For Oracle databases, the program prompts for a connect string. Enter or verify the connect string and click OK.

7. On Work with Locations and Machines, click Find and expand the tree.
8. Expand the Enterprise Server, click the newly added machine and click Select.
9. On Enterprise Server Revisions, click the Environments form exit.
10. On Machine Environments Revision, add the environments which you want to access with the new tools release. You can add the environments either by typing their names into the grid or by selecting them using the visual assist.
11. Click OK twice to return to Work with Locations and Machines.

Modifying the Existing Machine Record to Remove Environments

1. On the Deployment Server, log on to the deployment environment.
2. From the fast path menu, run P9654A.
3. On Work with Locations and Machines, click Find and expand the location tree.
4. Expand Enterprise Servers, click the initial Enterprise Server machine record and click Select.
5. On Enterprise Server Revisions form, click the Environments form exit.
6. Delete all the environments that you added in the previous section.

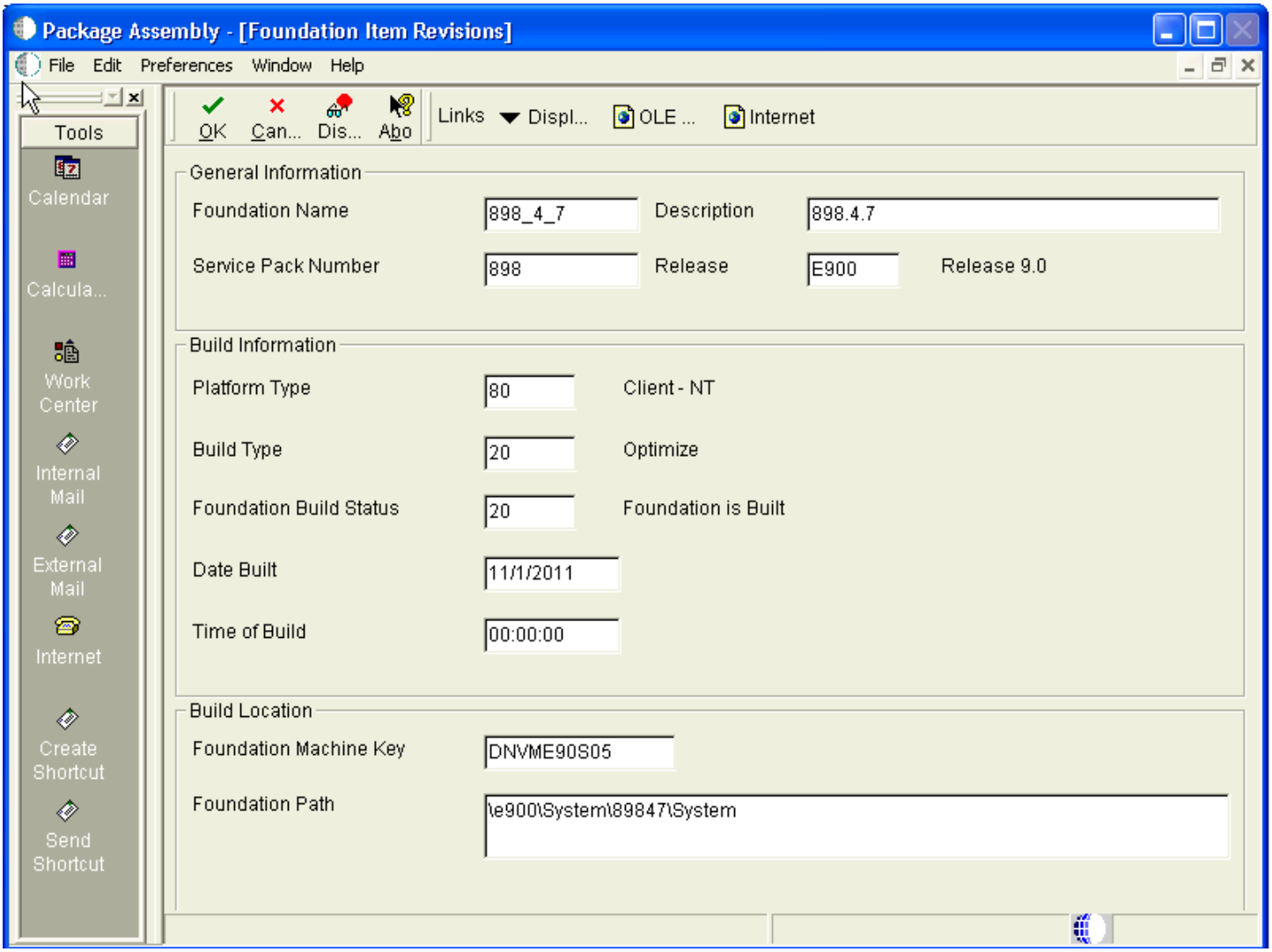
   This should leave you with only those environments that you have defined to access the production tools release (for example, PD920).
7. Click OK twice.

Working with Packages for Multiple Foundations

Deploying the new tools release is easily done using the Package Build application by using a non-default foundation. Use these steps to create and include a new foundation for the multi-foundation tools release.

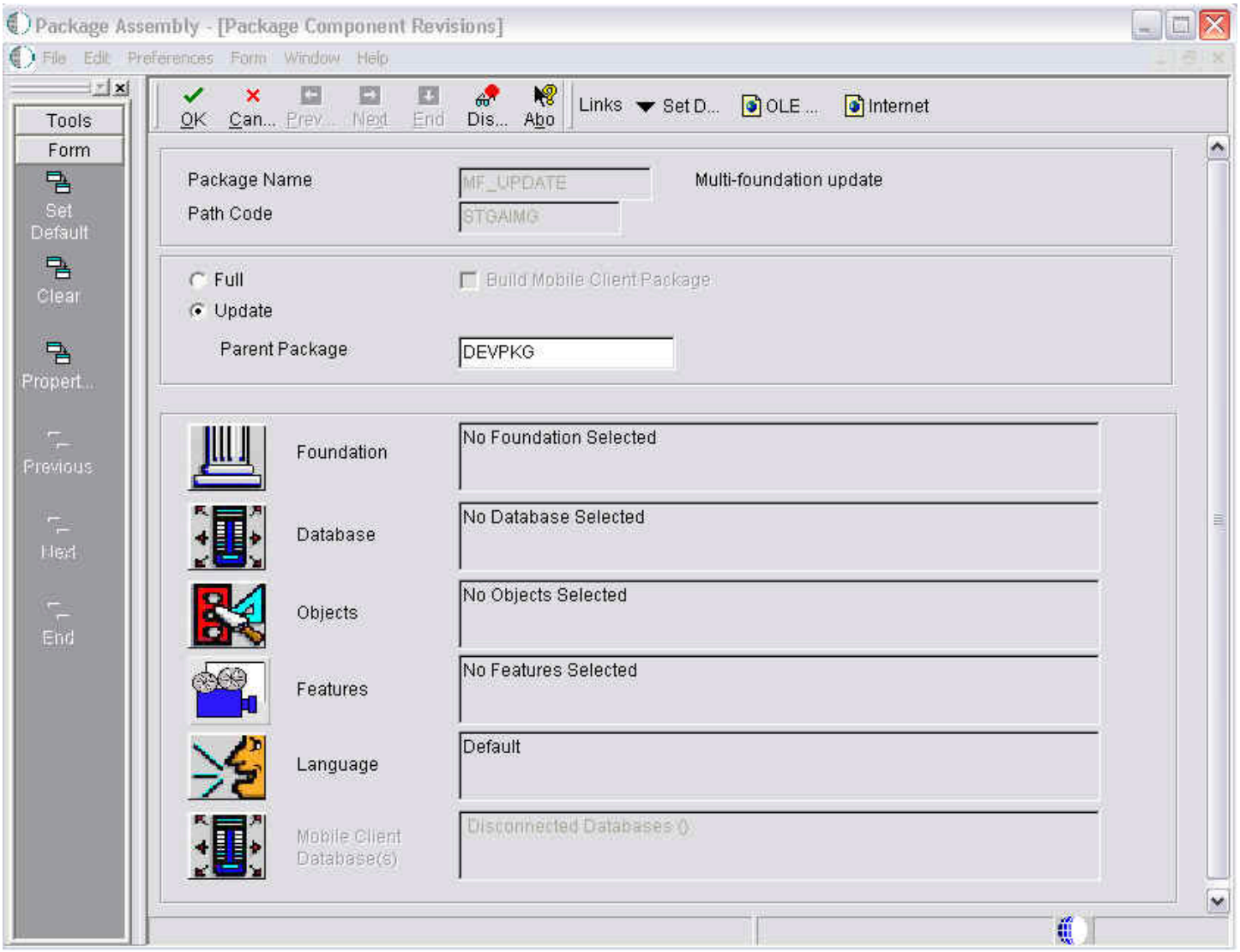
1. Sign on to the Deployment Server using the JDE user ID.
2. Sign on to EnterpriseOne using the JDE user ID into the Deployment (DEPXXX) environment.
3. Launch the Package Assembly application from menu GH9083.

4. Go to Form/Foundation and click the **Add** button on the Foundation Component Selection Screen



5. On Foundation Item Revisions, click the **OK** button to save the Foundation Item Revision information.

6. Click the **Find** button to display the new Foundation item list.
7. On the Package Component Revision, click the **Add** button to add an update package.
   Complete these fields:
   - *Name*
   - *Description*
   - *Path Code*
8. Click the **Next** button.



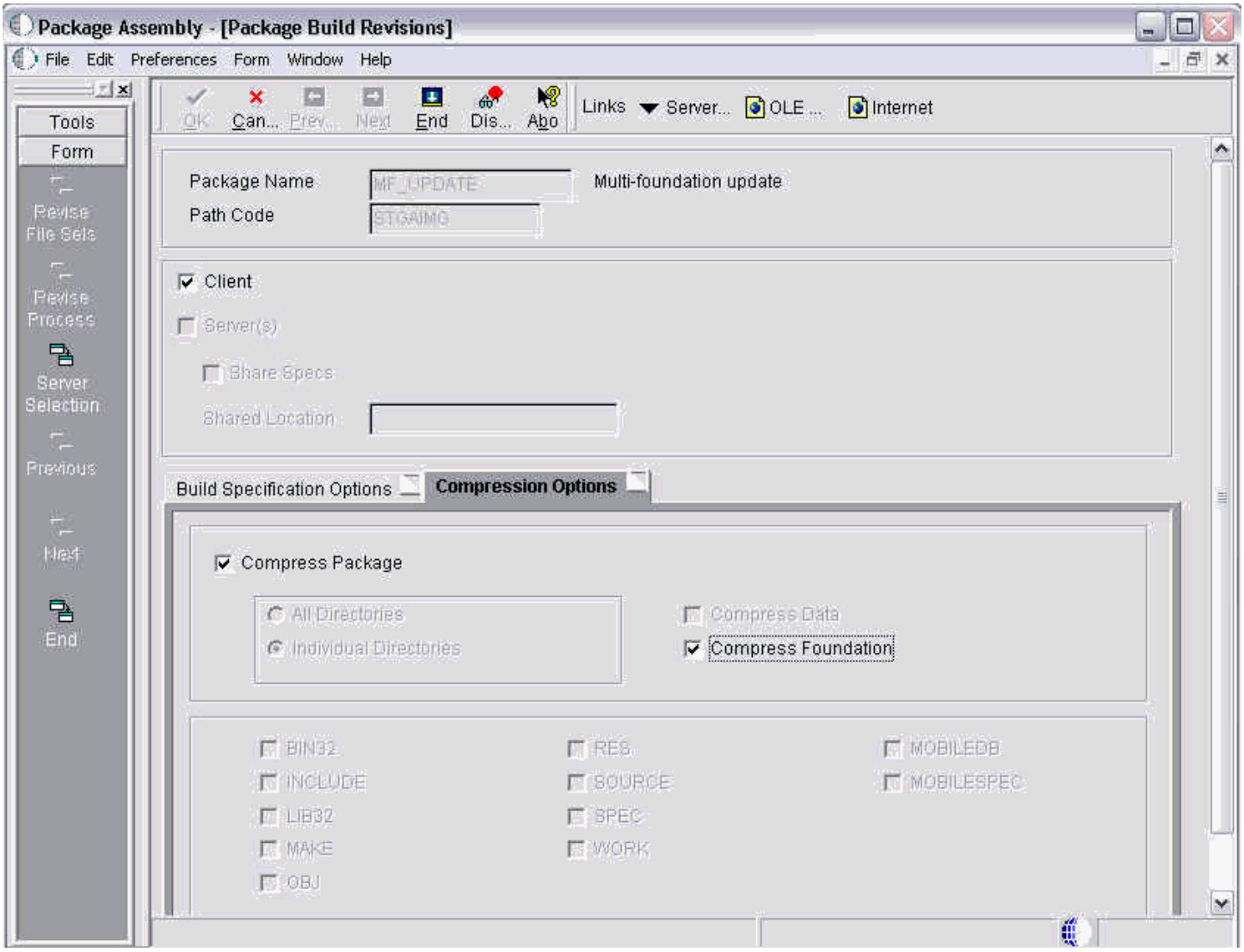
9. On the Package Component Revisions, enter a valid value for the Parent Package and select the **Foundation** icon.

10. On Foundation Item Revisions, select the **Find** button.
11. Select the Multi-foundation for the Update Package by clicking the first column of the selected row.

    > **Note:** Ensure that the check mark displays in the first column of the selected row record.

12. Click the **Close** button.
13. Click the **End** button to complete the Package Assembly process.
14. Click the **Row > Active/Inactive** icon.

15. Click the **Define Build** button and continue through the subsequent screens.


Package Assembly - [Package Build Revisions]
File Edit Preferences Form Window Help
Tools
Form
Revise File Sets
Revise Process
Server Selection
Previous
Next
End
OK Can... Prev... Next End Dis... Abo
Links Server... OLE ... Internet
Package Name
MF_UPDATE
Multi-foundation update
Path Code
ST9AIMO
Client
Server(s)
Share Specs
Shared Location
Build Specification Options
Compression Options
Compress Package
All Directories
Individual Directories
Compress Data
Compress Foundation
BIN32
INCLUDE
LIB32
MAKE
OBJ
RES
SOURCE
SPEC
WORK
MOBILEDB
MOBILESPEC

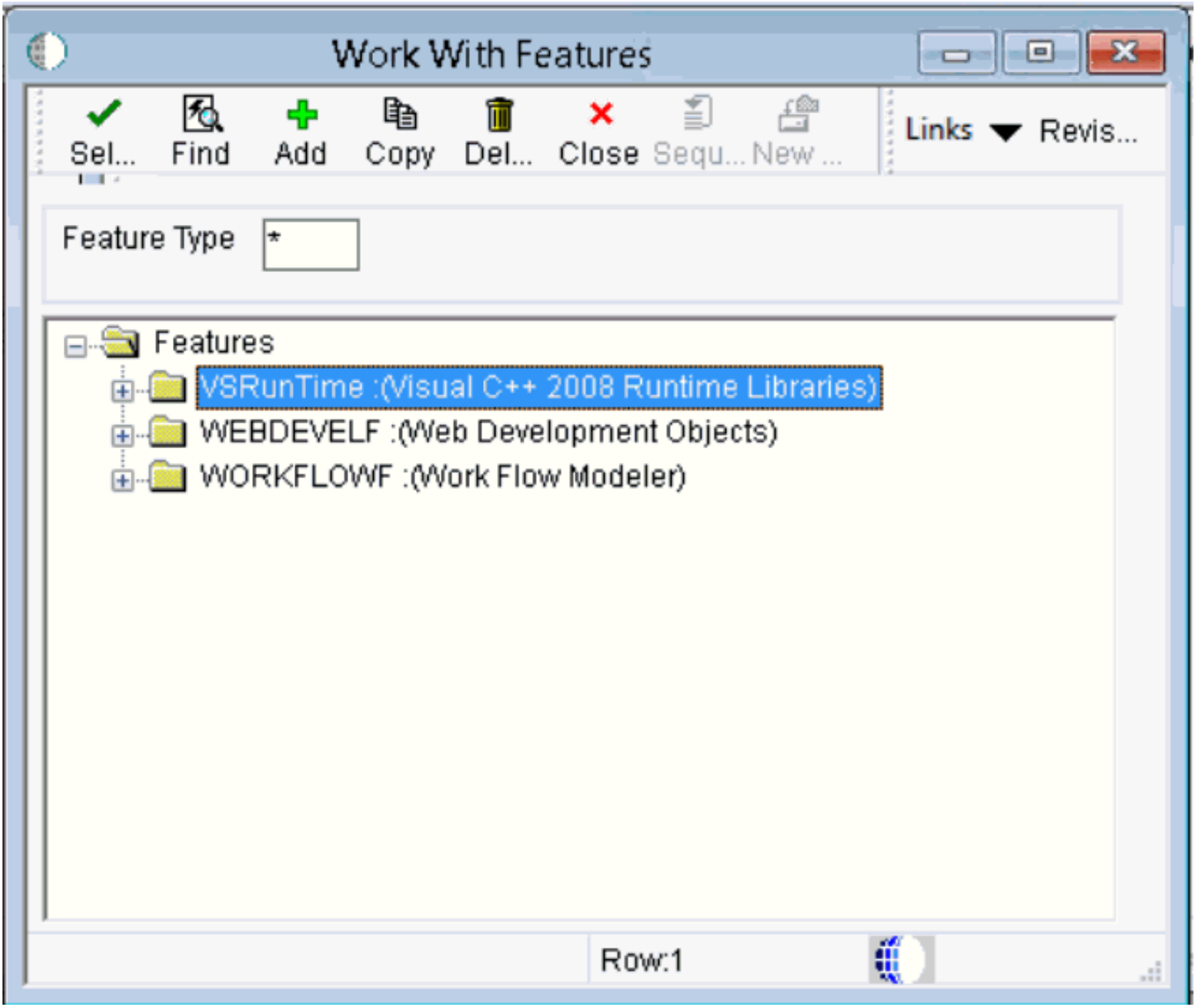
16. On Package Build Revisions, select the **Compression Options** tab and select your compression options.
17. Proceed to submit the build and deploy the package.

Working with the Development Client

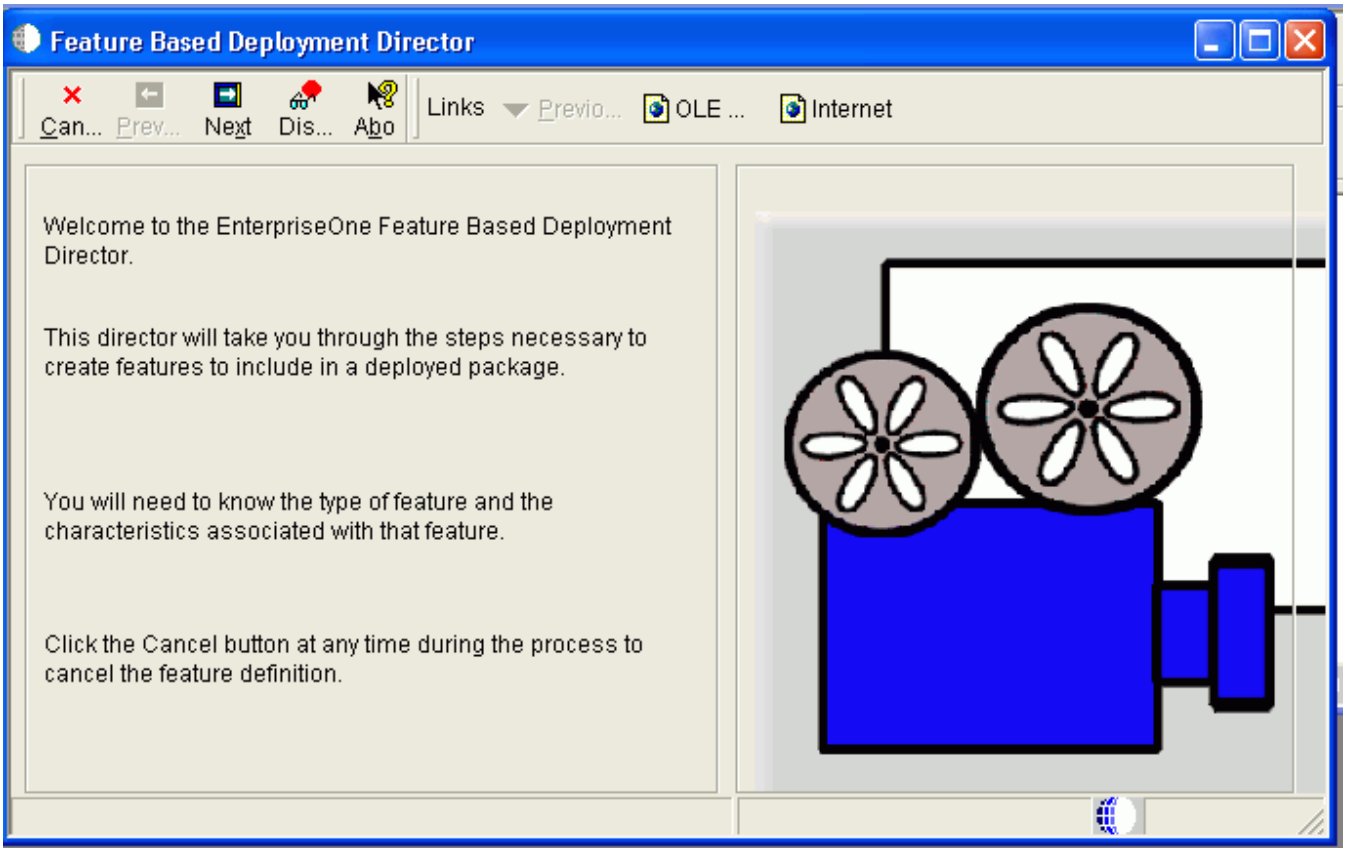
Use the procedure in this section to set up your Development Client.

1. On the GH9083 menu, launch the Package Build application.

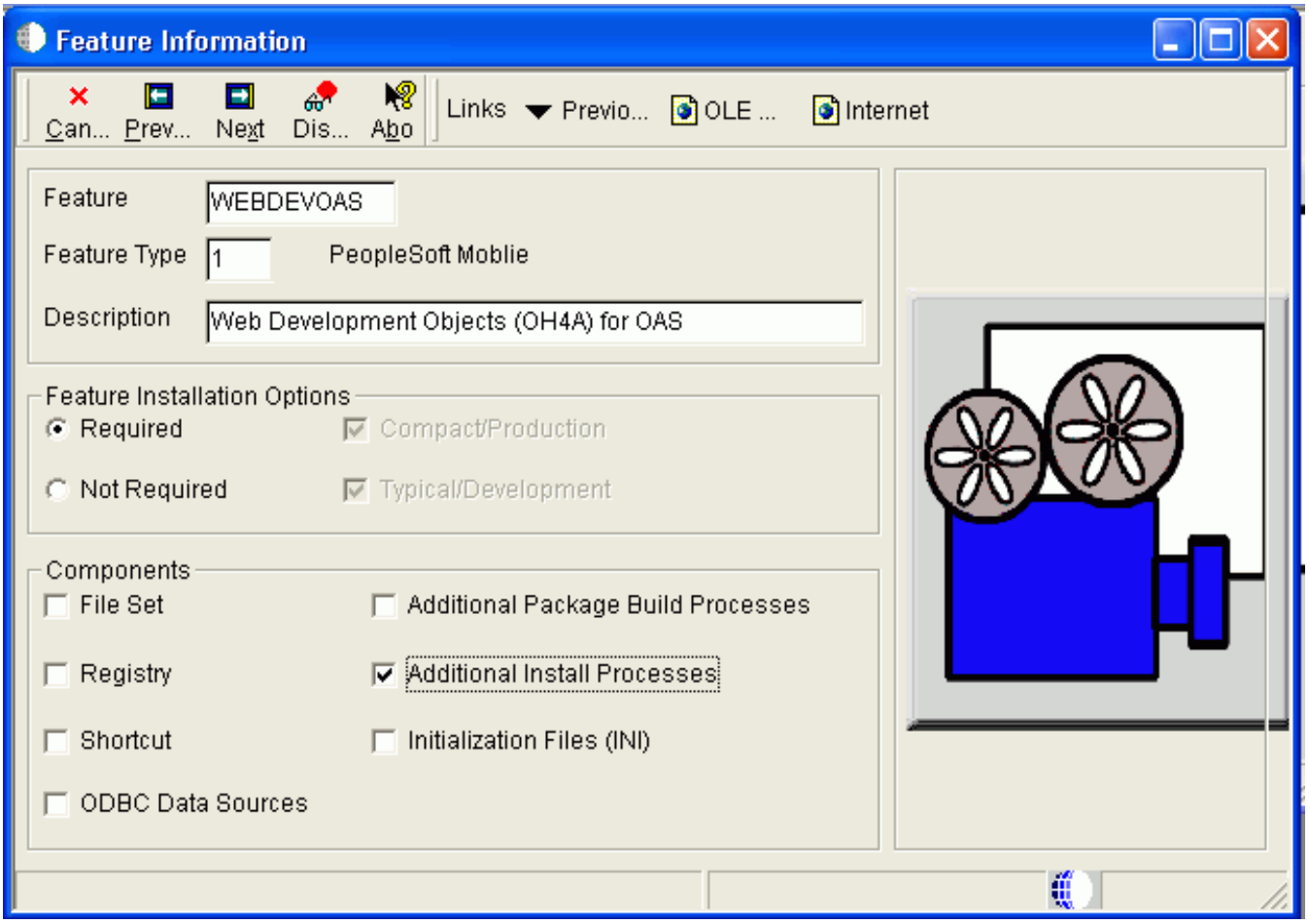
2. Create an update package by selecting Package Assembly and then Add.



3. On Work With Features, select Add.



4. On Feature Based Deployment Director, click Next.

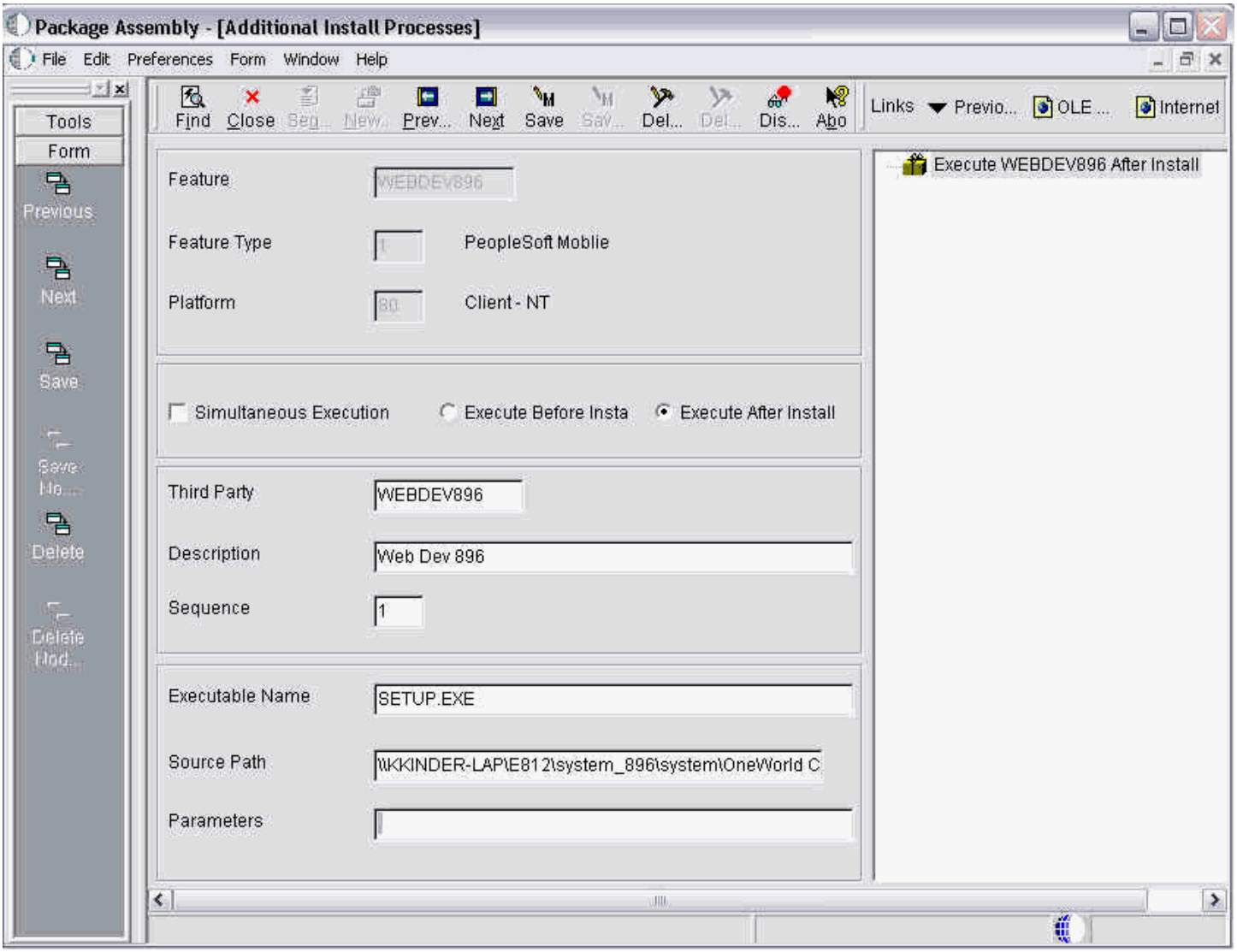


5. On Feature Information, complete these fields:

Field	Description
Feature Type	Enter a valid value for this feature type.
Description	Enter a description for the feature.
Feature Installation Options	Select your feature installation options by checking the appropriate check boxes.

Field	Description
Components	Verify this check box is selected: **Additional Install Processes**

6. Click Next.


Package Assembly - [Additional Install Processes]
File Edit Preferences Form Window Help
Tools
Form
Previous
Next
Save
Delete
Find Close Prev... Next Save Del... Dis... Abo
Links Previo... OLE ... Internet
Execute WEBDEV896 After Install
Feature
WEBDEV896
Feature Type
PeopleSoft Moblie
Platform
Client - NT
Simultaneous Execution
Execute Before Insta
Execute After Install
Third Party
WEBDEV896
Description
Web Dev 896
Sequence
1
Executable Name
SETUP.EXE
Source Path
\\KKINDER-LAP\E812\system_896\system\OneWorld C
Parameters

7. On Package Assembly, Additional Install Processes, complete these fields:
   - *Feature*

     Enter a meaningful name for the feature. For example, if you are using WebLogic:

     WEBDEVWLS

     If you are using WAS, an example would be

     WEBDEVWAS

     You can specify any name for the feature.
   - *Feature Type*

     Enter the value **1**.
   - *Description*

     Enter a description. For example:

     Web Development Objects (WLSH4A) for WLS

     or

     Web Development Objects (WASH4A) for WebSphere

     You can specify any name for the description.
   - *Required*

     Ensure this option is selected.
   - *Additional Install Processes*

     Ensure this option is selected.
8. Click Save.
9. Click Next.
10. Continue with the rest of the package assembly; there is no need to select any objects.
11. Define the package and, when prompted, make sure to select:

    `Build Feature Inf`
12. Submit the package, which creates the new feature `.inf` for Web Development.

    The newly created feature .inf file is located in the `package.inf` directory.

> **Note:** For any subsequent package builds for the non-default foundation for which you want to include webdev feature, you can use the package that you created with this procedure.

Editing the Development Client jde.ini File

> **Note:** If you are using SnapShot, you do not need to manually edit the `jde.ini` file to manage multiple clients for multiple foundations, as described in this procedure.

This section describes editing the **jde.ini** file for a Development Client.

In order for the *Release 9.2* clients to access the new tools release, you must edit the **jde.ini** file on each client to point to the new directory.

Complete the following task on all *Release 9.2* clients that have path codes that will access the new tools release.

> **Note:** You will be unable to access multiple tools releases with a single client. To access multiple tools releases, you will need to install a different client for each tools release. This will require using the SnapShot application to save each client before installing the next one.

1. On the *Release 9.2* workstation, open Microsoft Windows Explorer, and navigate to your Microsoft Windows directory. For example:

   cd c:\WINDOWS
2. Open the **jde.ini** file.
3. Locate the **[JDENET]** section, and edit the port settings for **serviceNameListen** and **serviceNameConnect**. Change these entries to match those of the server **jde.ini** file for the new tools release.

```
[JDENET]
serviceNameList=portNumber
serviceNameConnect=portNumber
```

where *portNumber* is the same port number you entered in the **jde.ini** file on the Enterprise Server.

Promoting Foundation Code

This section discusses how to promote foundation code into the production environment when using a Microsoft Windows-based Enterprise Server.

1. On the Enterprise Server, stop the *Release 9.2* services for the following environments:
   - The environment that you want to promote to production
   - The production environment
2. Navigate to this directory:

   z:\JDEdwards\ddp\E920
3. Create a directory called **System Backup**, and move the **System** directory into the **System Backup** directory.
4. Navigate to this directory:

   z:\JDEdwards\ddp\foundation_code_dir
5. Copy the **System** directory to this directory:

   z:\JDEdwards\ddp\E920
6. Navigate to this directory and copy the **jde.ini** file.

   z:\JDEdwards\ddp\E920\System Backup\System\bin32
7. Paste the **jde.ini** file into this directory:

   z:\JDEdwards\ddp\E920\System\bin32

The program asks if you want to replace the existing jde.ini file. Click OK to indicate you want to replace the existing `jde.ini` file.

8. Start the *JD Edwards EnterpriseOne* services.
9. On the Deployment Server, create a `System Backup` subdirectory under the *JD Edwards EnterpriseOne* host code directory. For example

   ```
   z:\JDEdwards\E920\System Backup
   ```
10. Copy the `System` and `SystemComp` directories into the `System Backup` directory.
11. Navigate to the directory containing the system code being promoted. For example:

    ```
    z:\JDEdwards\System x.xx
    ```

    where x.xx is the release number of the tools release
12. Move the `System` and `SystemComp` directories to this directory:

    ```
    z:\JDEdwards\E920
    ```
13. Using P9654A, edit the machine records to include the environments that you want to access the promoted tools release.

Setting up the Multiple jde.ini Files on the Deployment Server

You may need to set up multiple `jde.ini` files on the Deployment Server to address the requirement for the Development Client to access different pathcodes and ports for different application releases of JD Edwards EnterpriseOne. For details, refer to the appendix of this guide entitled: *Data by Pathcode*.

9 Appendix B - Manual Cleanup of an Uninstalled Oracle Database

Manual Cleanup of an Uninstalled Oracle Database

If the deinstallation of the OEE database fails for some reason, to completely remove the old installation before reinstalling the database, you need to perform manual steps that are listed in the appendix of the *JD Edwards EnterpriseOne Development Client Installation Guide* entitled *Manual Cleanup of an Uninstalled Oracle Database*.

10 Appendix C - Data by Pathcode

Data by Pathcode

Note: Beginning with Tools Release 9.2.5.0 and the removal of the E1Local database, the contents of this appendix are obsolete. If you are installing Tools Release 9.2.5.0 or higher, you can ignore these topics.

When a Development Client is installed, it may be configured so that each pathcode on the client has its own local database or for all pathcodes to share the same local database. Initially, JD Edwards EnterpriseOne is set up for each pathcode on a Development Client to have its own database. This section describes how to configure these scenarios.

This appendix discusses these topics:

- *Datasource Name Generation*
- *Package Build Considerations*
- *Additional Spec Datasource Settings in the jdbj.ini*
- *Summary of Possible Administrator Changes*

Datasource Name Generation

The Development Client installation creates entries in the registry based on values in the Development Client installer **oraparam.ini** file located at:

\\<deployment server>\<release>\OneWorld Client Install\install\oraparam.ini

Assuming you have edited the oraparam.ini file as described in the section of this guide entitled: *Update the oraparam.ini File*, these entries are in the **oraparam.ini** file:

```
[Attributes]
DataByPathCode=0|1
LocalDS=Local (this can be any string; the default value is "Local")
```

The registry entries on the Development Client machine after the Development Client is installed are located under:

HKEY_LOCAL_MACHINE\Software\Wow6432Node\Jdedwards\OneWorld\install.ini\<releasenumber>

In the above location, assuming the **oraparam.ini** file contains the requisite entries, these registry key values are assigned:

```
DataByPathCode=0|1LocalDS=some_string
```

Using these values, the WLSH4A and WASH4A installers create these datasource entries that point to the EnterpriseOne specs stored in a local database:

1. Entry in the **jdbj.ini** used by the web client:

   ```
   [JDBj - SPEC DATA SOURCE]
   name=
   ```

2. Entry in the **jde.ini** used by the Development Client:

```
[LOCALWEB]
Spec Datasource=
```

For **WLSH4A**, the **jdbj.ini** file is located in this directory on the Development Client machine:

<installpath>\system\JAS\EA_JAS_80.ear\webclient\WEB-INF\classes\jdbj.ini

For **WASH4A**, the **jdbj.ini** is located in this directory on the client machine:

<installpath>\JAS\EA_JAS_80.ear\webclient.war\WEB-INF\jdbj.ini

The rules for generating the spec datasource name in the **jdbj.ini** and **jde.ini** are based on the values of the **DataByPathCode** and **LocalDS** in the **[Attributes]** section of **oraparam.ini**, which is located in this directory of the Development Client installer:

OneWorld Client Install\install

In hierarchical order, the rules are:

1. **DataByPathCode is 0 or missing**

   This condition indicates a shared local database. The datasource name will be "OneWorld Local" regardless of the value of LocalDS.

   The following table shows the parameters used to create a datasource name that points to a local database:

ini File	Settings
oraparam.ini	[Attributes] DataByPathcode=0 or missing LocalDS is missing
jdbj.ini	[JDBj - SPEC DATA SOURCE] name=OneWorld Local
jde.ini (Development Client machine)	[LOCALWEB] Spec Datasource=OneWorld Local

2. **DataByPathCode is 1 and LocalDS is Missing**

   This condition indicates separate databases and LocalDS is missing. The datasource name will be "Local" followed by a hyphen and then the pathcode.

ini File	Settings
oraparam.ini	[Attributes] DataByPathcode=1 LocalDS is missing
jdbj.ini	[JDBj - SPEC DATA SOURCE] name=Local - <pathcode>

ini File	Settings
jde.ini (Development Client machine)	[LOCALWEB] Spec Datasource=Local - <pathcode>

3. **DataByPathCode is 1 and LocalDS is Set**

   This condition indicates separate databases and LocalDS is set. The datasource name will be the value of LocalDS followed by a hyphen and then the pathcode.

ini File	Settings
oraparam.ini	[Attributes] DataByPathcode=1 LocalDS=Local
jdbj.ini	[JDBj - SPEC DATA SOURCE] name=Local - <pathcode>
jde.ini (Development Client machine)	[LOCALWEB] Spec Datasource=Local - <pathcode>

CAUTION: The WLSH4A and WASH4A installers will update these strings in the ini files on the Development Client.

Package Build Considerations

The package build process uses a value in the **c:\Windows\jde.ini** on the build machine when building a full package to determine the directory structure that will be created on target workstations when the package is installed. This key and setting are:

```
[INSTALL]
DataByPathCode=0|1
```

The directory structure that is formed by the package build based on this setting is stored in the **inf** file for the package.

The following table shows what the directory structure would look like when the built package is installed on a workstation using these example values:

- *Installation Directory*

  c:\E920

- *Pathcodes*

  DV920

  PY920

Note: These values are only examples; additional pathcodes could be included.

DataByPathCode	Directory Structure	Comment
DataByPathCode=0	`c:\E920\JDEData`	Contains database shared among all pathcodes
	`c:\E920\DV920`	
	`c:\E920\PY920`	
DataByPathCode=1 This is the default value for package build.	`c:\E920\DV920\JDEData`	Contains database for DV920
	`c:\E920\PY920\JDEData`	Contains database for PY920

CAUTION: The value of the `DataByPathCode` key in the `[INSTALL]` section of the `jde.ini` on the machine(s) on which any packages are built MUST match the value of the `DataByPathCode` key in the `[Attributes]` section of the `oraparam.ini` in `\OneWorld client Install\install` directory for the Development Client installer.

Additional Spec Datasource Settings in the jdbj.ini

As noted in the preceding sections of this appendix, the name of the Spec Data Source stored in the `jdbj.ini` is determined based on the values of the DataByPathCode and LocalDS keys in the `oraparam.ini` file. Additional entries in the `jdbj.ini` complete the definition of the datasource. These entries are copied to the `jdbj.ini` by the WLSH4A and WASH4A installer and come from the `[DB SYSTEM SETTINGS - SECONDARY]` section of the Development Client `jde.ini`.

The following table shows the mapping between Entries in the `jde.ini` and `jdbj.ini`.

jde.ini	jdbj.ini
DB SYSTEM SETTINGS - SECONDARY	JDBj - SPEC DATA SOURCE
Type	databaseType
DatabaseInstance	databaseInstance

During the installation of the Development Client, the **jde.ini** file for all the Development Clients is created according to specific rules. Refer to the *JD Edwards EnterpriseOne Development Client Installation Guide* for details.

Summary of Possible Administrator Changes

The following table describes the only changes that the administrator might need to make to affect the definition of the datasource for the local database.

File	Setting	Comment
\\<deployment server name>\<release>\OneWorld Client Install\misc\jde.ini	[DB SYSTEM SETTINGS - SECONDARY] Type DatabaseInstance=	Type defines the type of the local database. DatabaseInstance defines the instance name of the local database.
	[INSTALL]DataByPathCode	0 = share database among pathcodes 1 = separate databases for each pathcode This setting is only used when building a full package.
\\<deployment server name>\<release>\OneWorld Client Install\misc\<pathcode>.ini	[DB SYSTEM SETTINGS - SECONDARY] Type DatabaseInstance	Any values in this file override their respective values in this file: **\\<deployment server name>\<release>\OneWorld Client Install\misc\jde.ini** Type defines the type of the local database. DatabaseInstance defines the instance name of the local database.
\\<deployment server name>\<release>\OneWorld Client Install\install\oraparam.ini	[Attributes]DataByPathCod LocalDS	0 = share database among pathcodes 1 = separate databases for each pathcode This o ptional datasource prefix is only used if DataByPathCode=1.

11 Appendix D - Updating the JRE/JDK Used by the Installers

Updating the JRE/JDK Used by the Installers

The installation of a Development Client places two JRE's under the installation directory. One is provided by OUI itself and is used only by OUI during deinstallation. The second is in the system directory and is used by EnterpriseOne at runtime.

If you need to replace the JRE used at runtime (for example, to avoid security vulnerabilities), replace the one on the Deployment Server under the system directory (such as, `\\<deployment server name>\<release>\system\jre`) and build and deploy a package containing that foundation/system.

If you need to replace the JRE used during deinstallation, follow these steps after installing a package onto the Development Client:

1. Drill down to `<E1_install_dir>\jre\1.8.0`
2. Copy the contents of the JRE you want into that 1.8.0 directory. That directory should contain, among other things, bin and lib subdirectories.

Note: It is very important that you do not rename the 1.8.0 directory and that the directory contains the new JRE files and subdirectories.

Note: When a package is installed, that `jre\1.8.0` directory will be replaced. You will need to replace the JRE after each package is installed.