JD Edwards EnterpriseOne

Implementing 64-bit Processing with JD Edwards EnterpriseOne Learning Path

1.0

JD Edwards EnterpriseOne
Implementing 64-bit Processing with JD Edwards EnterpriseOne Learning Path

1.0

Part Number: G45779-01

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

Overview

This learning path guides you through the steps required to update your existing JD Edwards EnterpriseOne 9.2 32-bit install to Tools Release 9.2.6 or later with 64-bit processing.

Before you begin the tasks included in this learning path, ensure that your EnterpriseOne install is at:

- Applications Release 9.2.

Verify that a full package has been built for the path codes implementing 64-bit.

We recommend that you migrate one path code to 64-bit ensuring that your business functions migrate properly. This process is described in more detail in the multi-foundation content included in this learning path. Specification merge for UBEs, business views, and table indexes is deprecated with the 64-bit Deployment Server update. Compare tools are available for these objects. Please see the *Software Updates* guide for more information regarding the compare tools.

Tools Release 9.2.6 and later provides support for upgrading Deployment Server, Enterprise Servers, and Development Clients to 64-bit. If any path code or server remains at 32-bit, the source of truth will be 32-bit. A conversion tool from 32-bit to 64-bit is provided, but the ability to reverse engineer from 64-bit back to 32-bit is not supported.

If you are using a 64-bit development client, business function development is always performed in 32-bit. When you check-in or build through Busbuild the 32-bit version is run through the 64-bit conversion and both the 32-bit and the 64-bit versions are written to the repository in a single record. Any subsequent par that is generated via Configuration Assistant will contain both the 32-bit and 64-bit versions and can be deployed to either a 32-bit or 64-bit path code.

After all path codes and servers have been converted to 64-bit, and the flag in the Release Master has been set to indicate the system is completely 64-bit, the source of truth will be 64-bit.

JD Edwards compare and merge tools enable you to compare two versions of JD Edwards EnterpriseOne objects and non-code objects, such as versions, UDCs, and data dictionary items. Tools also includes a compare tool for RDA and merge tools for business views and tables. If you have made custom modifications to business views, forms, or event rules, the compare and merge tools can help you retrofit your changes and verify that they were carried forward to a new release. Furthermore, you can use these tools anytime you want to compare items between two environments.

See *JD Edwards EnterpriseOne Tools Software Updates Guide*

For more information regarding the following:

- UDO Table Architecture
- Table conversion Schedule
- ASI Validations
- Errors and Warnings

See the appendix in *JD Edwards EnterpriseOne Tools Upgrade Guide*

Important: Review the *Visual Studio 2017* requirements for JD Edwards EnterpriseOne on My Oracle Support.

A *64-bit Processing FAQ* is available on My Oracle Support for your reference.

Please review the *JD Edwards EnterpriseOne Licensing Information User Manual*.

Please review the *Known Issues* document on My Oracle Support for any issues still outstanding on the Tools 9.2.6.x releases.

JD Edwards EnterpriseOne Tools Release 9.2.6 or later (64-bit) supports the following platforms:

- Linux
- Solaris
- Microsoft Windows
- IBM iSeries
- HP-UX
- IBM AIX

Before You Begin

It is important to review and understand the following documents before implementing 64-bit processing.

Accessing Certifications

When updating JD Edwards EnterpriseOne Tools Releases, it is critical to review and meet the minimum technical requirements for essential third-party components. You can locate the JD Edwards EnterpriseOne Tools Release 9.2.6 Certifications (Minimum Technical Requirements) from My Oracle Support.

To access JD Edwards EnterpriseOne Tools Release 9.2.6 Certifications:

1. Enter `https://support.oracle.com` in your web browser to navigate to My Oracle Support.
2. Click the Certifications tab.
3. Search for JD Edwards EnterpriseOne Tools Release 9.2.6.

In particular for the 64-bit implementation of JD Edwards EnterpriseOne, ensure that the requisite JDK on the Enterprise Server and the Deployment Server meets the version requirements as specified in Oracle Certifications. If you need to upgrade the JDK, you should do so before you use Server Manager to deploy the new JD Edwards EnterpriseOne Tools Release. Verify the version requirements on the following:

- JDK
- Database Client
- Visual Studio Runtime

See My Oracle Support (MOS) Document ID: *2415818.1* regarding installation of the Visual Studio 2017 Runtime Redistributable files.

Required Components

See the following My Oracle Support document for more details:

Required Components for JD Edwards EnterpriseOne Release 9.2.x (9.2.5.0 and above) (*Doc ID 2054020.1*)

Find the Tools Release you are upgrading to and click the link for:

Required Components for Installing JD Edwards EnterpriseOne Tools 9.2.x with Applications 9.2 (Already 64-bit) to open the pdf containing the information.

Upgrading Third-Party Components

Implementing 64-bit with JD Edwards EnterpriseOne requires upgrading third-party components to the latest supported version. This task must be performed on the Deployment Server, as well as all Enterprise Servers and Development Clients. EnterpriseOne web servers are not affected. The following table indicates the components that must be updated to the latest supported version on each platform.

Product	Deployment Servers (Windows Only)	Enterprise Servers (Windows, Linux, or Unix)	Development Client (Windows Only)
Java Runtime Environments (JREs)	Yes	Yes	Yes
Java Development Kit (JDK)	Yes	No	Yes
Database clients and drivers	Yes	Yes	Yes
Java Database Connectivity (JDBC) drivers	Yes	Yes	Yes
Microsoft Visual Studio Runtime libraries	Yes	Yes (Windows servers only)	Yes
Microsoft Visual Studio	Yes	Yes (Windows servers only)	Yes
Unix Compilers	No	Yes	No
Database	Yes	No	Yes
WebLogic Application Server (WLS)	No	No	Yes
WebSphere Application Server (WAS)	No	No	Yes

Java Runtime Environment (JRE) and Java Development Kit (JDK)

A JDK provides tools to compile Java source files and to generate JAR files containing Java class files. Java-based programs run within a JRE. A JRE provides files that are necessary at runtime such as dynamic-linked Libraries (DLLs on Windows, shared libraries on UNIX platforms, and JAR files that contain compiled Java class files. If a JDK is installed, a JRE does not need to be installed separately because a JDK already contains a JRE. A WLS application server requires a JDK to be installed prior to the installation of WLS. A JDK is provided with a WAS installation.

The EnterpriseOne foundation included in a client package contains a JRE that is used on the Development Client. The bitness of the JRE should match the bitness of the foundation. To include a new JRE in a package, replace the JRE in the `system/jre folder` on the Deployment Server before building the package; be sure that the package includes the proper foundation.

Downloading

Click the following URL to download the JDK:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

1. Under Java Platform, Standard Edition, you can download the JDK or JRE, depending on the particular EnterpriseOne component (Server Manager Console or Agent, Development Client, or Enterprise Server) for which the JDK or JRE is to be used. Each JDK includes a JRE. Some EnterpriseOne components require a JDK, but others require a JRE only. Do not use any non-supported version or edition of the Java JRE or JDK.
2. Follow the navigation aids to download the JDK or JRE that is appropriate for your operating system.
3. Follow the installation instructions that are provided with the JDK or JRE.

Verifying on UNIX

After installing the JDK or JRE, run the following test to verify that the version of Java is recognized by the system:

1. Access the command line of the operating system and enter:

   ```
   cd <JDK_or_JRE_install_dir>/bin java -version
   ```
2. Verify that the output shows the version that you downloaded.

Database Client and Drivers

The database client is used by EnterpriseOne on the Deployment Server, Enterprise Servers, and Development Clients to connect to databases. The database client must be at the same or lower release level as the database to which it is connecting. In addition, the database client must be of the same bitness as the EnterpriseOne foundation on the Deployment Server, Enterprise Server, or Development Client.

Download and install the database client of the proper bitness from the database vendor onto the proper EnterpriseOne platform.

If both 64-bit and 32-bit Oracle database clients exist on either a Deployment Server or a Development Client, the Windows operating system will find the proper database client DLLs based on the bitness of the EnterpriseOne system or foundation that is running.

ODBC Data Source Drivers

The ODBC database driver (64-bit or 32-bit) is used to connect to databases from each of the database vendors that EnterpriseOne supports: Oracle, SQL Server, and UDB. If ODBC data sources are used, their drivers and definitions may need to be updated to allow EnterpriseOne Deployment Server and Development Clients to connect to databases. The main reason for updating the ODBC data sources after they have been created is to change the bitness of the EnterpriseOne software.

The Development Client installer automatically handles the task of setting up ODBC data sources when a package is installed.

> **Note:** Before installing a 64-bit package on a Development Client, you must ensure that the 64-bit version of the database vendor's ODBC driver exists. For example: `c:\Windows\System32\cwbodbc.dll` or `c:\Windows\System32\sqlsrv32.dll`

JDBC Database Drivers

JDBC database drivers are copied to Development Clients when web clients (WLSH4A for WebLogic Server or WASH4A for WebSphere Application Servers) are installed on those Development Clients. These JDBC drivers must be updated to reflect the bitness of EnterpriseOne foundation on the target Development Client.

By default, the JDBC database drivers are copied from one of these directories, depending on the H4A installed to be run (WebLogic Server or WebSphere Application Server):

WebLogic Server

```
<EnterpriseOne_dep_svr_install_directory>\OneWorld
 Client Install\WebDevFeature\WLSH4A\JDBC
```

WebSphere Application Server

```
<EnterpriseOne_dep_svr_install_directory>\OneWorld Client
Install\WebDevFeature\WASH4A\JDBC
```

Microsoft Windows Development Tools and Libraries

Either Visual Studio (VS) or Visual Studio Runtime libraries must be installed on each Microsoft Windows platform. VS must be installed if you want to build (compile and link) EnterpriseOne business functions on the Windows platform. If you do not want to build EnterpriseOne business functions on a given Windows platform, instead of VS, install the Runtime libraries.

Microsoft Visual C Runtime libraries

Visual C++ Runtime libraries are dynamic-link libraries (DLLs) provided by Microsoft and are required by EnterpriseOne programs to run on Windows operating systems. The major release number of the runtime libraries that are installed on a Windows computer must match the major release number with which the EnterpriseOne software was built. The EnterpriseOne system or foundation and the business functions for 9.2.3.x were built with Visual C++ 2017 Runtime libraries. Runtime libraries for 2017 Visual C++ 2017 Runtime libraries must be installed on the computer system for EnterpriseOne to run.

Additionally, the bitness of the runtime libraries must match the bitness of the EnterpriseOne software that is running. Even though a Windows operating system may be 64-bit, programs of both 64-bit and 32-bit can run on the computer. You cannot mix 64-bit and 32-bit executables and DLLs in a running process space for a single program.

Windows operating systems come with certain releases of Visual C++ Runtime libraries (64-bit and 32-bit). If a given release of Windows does not include the Visual C++ Runtime libraries necessary by EnterpriseOne Deployment Server, Enterprise Server, or Development Client (either foundation or business functions), the proper Visual C++ Runtime libraries must be installed before installing EnterpriseOne.

The installers for the Visual C++ Runtime libraries are both release and bitness specific. These installers can be downloaded from the Microsoft Download Center and installed on EnterpriseOne Deployment Servers, Windows Enterprise Servers, and Development Clients. The EnterpriseOne Development Client installer requires the same release and bitness of Visual C++ Runtime libraries that the EnterpriseOne foundation uses. The Visual C++ Runtime libraries of the correct release and bitness must be installed prior to running the Development Client installer. However, the Development Client installer can install Visual C++ Runtime libraries of a different release but the same bitness as the EnterpriseOne business functions. This installation can be performed by adding a feature to the installable EnterpriseOne package that runs the Visual C++ Runtime installer.

> **Note:** You can download and install the Microsoft Visual Studio C++ 2017 Redistributable Packages (x86 or x64) from the Microsoft Download Center.

Microsoft Visual C Runtime Libraries

Microsoft Visual Studio is used to build and debug EnterpriseOne business functions. Building can be performed on either a Deployment Server or Development Client. Debugging of business functions, however, is supported only on Development Clients.

To ensure that the proper instance of Visual Studio is used during an EnterpriseOne build or a debugging session, follow these steps:

1. Install the version of Visual Studio specified in Oracle Certification.
2. Ensure that the following `jde.ini` setting indicates the proper version of Visual Studio:

```
[JDE_CG]
VisualStudioVersion=2017
```

Note: The `jde.ini` file referred to above is in the `c:\Windows` directory of the Deployment Server or Development Client in which the build or debugging will be performed.

UNIX and Linux Development Tools and Libraries

See: *JD Edwards EnterpriseOne 9.2.X Platform and Compiler* Support Statement (Doc ID 2059885.1)

Application Server

During the development of an EnterpriseOne application, an EnterpriseOne developer can run and test the application in a web browser. An application server is used to produce the EnterpriseOne application objects (forms, buttons, and so on) that are displayed in the browser. The application server can be either WLS (from Oracle) or WAS (from IBM); however, as a best practice, all Development Clients installed from a given Deployment Server should use the same type of application server (WLS or WAS).

WebLogic Server

WebLogic Server is a Java Application Server developed by Oracle. It can be installed on an EnterpriseOne Development Client and used to process requests from a Development Client's web client. WebLogic Server needs to be installed on a Deployment Server only if EnterpriseOne Server Manager Console is to be installed in the Deployment Server.

See the Oracle Certifications information to determine which JDK to use for WebLogic Server. You must install the JDK before installing WebLogic Server.

WebSphere Server

WebSphere Server is not used internally for EnterpriseOne development.

WebSphere Server is a Java Application Server developed by IBM. It can be installed on an EnterpriseOne Development Client and used to process requests from a Development Client's web client. WebSphere Server needs to be installed on a Deployment Server only if EnterpriseOne Server Manager Console will be installed there as well.

See the Oracle Certifications information to determine which JDK to use for WebSphere Server. You must install the JDK before installing WebSphere Server.

Understanding Multi-foundation

Multi-foundation is a JD Edwards EnterpriseOne architecture that allows for contrasting tools release foundations to reside on different path codes. For implementation details, refer to the appendix "Working With Multiple Tools Release Foundations" in one of these platform dependency documents as applicable:

- *Deployment Server Reference Guide for UNIX*
- *Deployment Server Reference Guide for Microsoft Windows*

- *Deployment Server Reference Guide for IBMi*

Before you upgrade to a new tools release foundation, it is a good practice to test and validate the functionality prior to upgrading the production (PD) path code. You can keep the production path code running on the existing tools release foundation and apply the new tools release foundation to all other path codes. This architecture enables you to perform tests on the new tools release foundation without impacting the PD path code. When you complete the validation, you upgrade the PD path code to the new tools release foundation.

The following are the common server configurations:

- Deployment Server with multiple physical Enterprise Servers
- Deployment Server with one physical server with multiple ports
- Deployment Server with one physical server with one port

For additional information, refer to this MOS Document, which provides a listing of all documents for understanding and implementing multi-foundation with JD Edwards EnterpriseOne:

E1: TR: Primary Note of Documents Explaining the Concepts and Implementation of Tools Release Multi-Foundation (Doc ID 1307646.1)

Deployment Server with Multiple Physical Enterprise Servers

In the above graphic, the PD path code resides on Enterprise Server 1 with a Tools Release 9.2.1 foundation. The prototype (PY) and development (DV) path codes share Enterprise Server 2 with a Tools Release 9.2.1 foundation. This is a multi-foundation architecture where different path codes on different Enterprise Servers have separate tools release foundations.

Deployment Server with One Physical Enterprise Server with Multiple Ports

In the above graphic, the PD path code uses Port 6017 with the Tools Release 9.2.1 foundation. The PY and DV path codes use Port 6117 with the Tools Release 9.2.1 foundation. This architecture illustrates multi-foundation. This is a multi-foundation architecture where the path codes are on the same Enterprise Server but on different ports. Each port can have a different tools release foundation.

Deployment Server with One Physical Enterprise Server with One Port

In the above graphic, the PD, PY, and DV path codes use Port 6017 and the Tools Release 9.2.1 foundation. This architecture is not multi-foundation because the path codes use the same tools release foundation on the same server and port.

Tools Release Upgrade Example

Initial State

In the above architecture, the Deployment Server and both the Enterprise Servers use the Tools Release 9.2.1 foundation.

Upgrade State

To upgrade a tools release foundation:

1. Manually back up the current tools release foundation on the Deployment Server.
2. Apply the new tools release foundation to the Deployment Server as the default system.
3. Apply the new tools release foundation to Enterprise Server 2.
4. Run validations on Enterprise Server 2.
5. After the system validations are performed, apply the new foundation to Enterprise Server 1.

Final State

The Deployment Server and both the Enterprise Servers are now running on the Tools Release 9.2.2 foundation.

2 Upgrading from Prior Releases of EnterpriseOne

Upgrade Considerations

- If you are planning an Applications Upgrade to a One-Click installation of JD Edwards EnterpriseOne, you should be aware of important technical considerations.

  These considerations are described in the Applications Upgrade Guide which describes an upgrade to the base Applications Release 9.2.

  Refer to the section titled Technical Considerations for Applications Upgrade to a 64-bit Version of 9.2 for your platform:

  Linux with Oracle Database

  Microsoft Windows with Oracle Database

  Microsoft Windows with SQL Server Database

  IBM i with DB2 for IBM i

3 Preparing for Your Tools Release Update

Backing Up JD Edwards EnterpriseOne Directories and Tables

When the tools release is changed on the Deployment Server, Server Manager replaces the System directory and its subdirectories. Therefore, before you begin you should make a backup of the existing directory structure.

On the Deployment Server, make a copy of these directories:

- /System
- /Systemcomp
- /OneWorld Client Install
- /E920/Planner/bin32

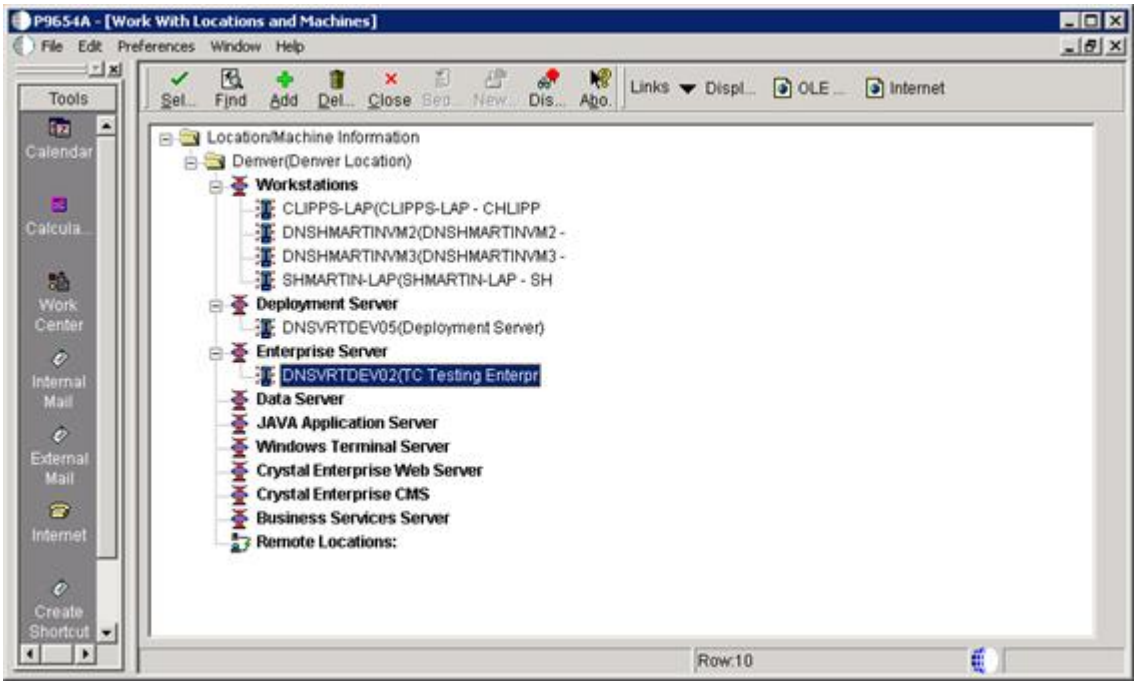
It is also recommend to back up the JD Edwards EntertpriseOne tables. Follow the JD Edwards EnterpriseOne backup procedures. See "*Backing Up JD Edwards EnterpriseOne Tables*" in the JD Edwards EnterpriseOne Administration Guide.

Associating Environments with the Enterprise Server

You associate environments with the Enterprise Server to automate the server map Object Configuration Manager (OCM) generation during the upgrade process.

1. Log in to the JDEPLAN environment on the Deployment Server.

2. Run the Work with Locations and Machines (P9654A) application.



3. Expand the Enterprise Server node and select your Enterprise Server.
4. Click Environments on the Form menu.
5. Add all environments associated with the path code you are upgrading on the Enterprise Server.
6. Click OK and restart JD Edwards EnterpriseOne.

Installing Business Function ESUs

A number of JD Edwards EnterpriseOne business functions require modification to be successfully processed on both a 32-bit system and a 64-bit system. To obtain the required updates to these functions, perform these actions:

- Get code current by taking all the currently available ESUs.
- Apply ESUs identified with a new Change Assistant query.

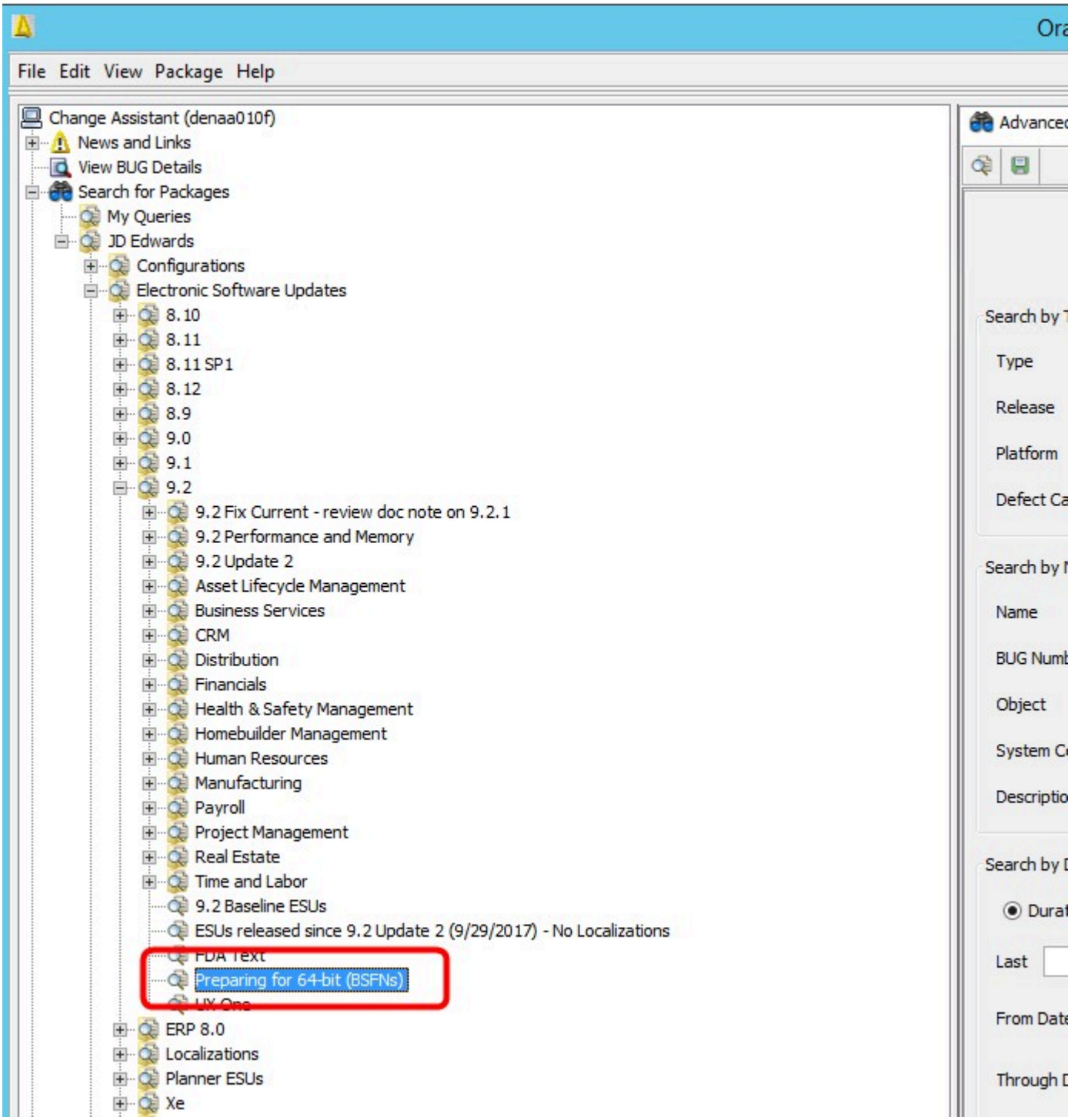
> **Note:** Applications Update UN3 includes all the original updates that were required for JD Edwards EnterpriseOne to run 64-bit.

Running New 64-bit Business Function Change Assistant Profile

Change Assistant can be run from any Microsoft Windows workstation or server, but it is best utilized when run from the JD Edwards EnterpriseOne Deployment Server.

1. Launch Change Assistant.
2. Expand the Search for Packages node.

3. Navigate to JD Edwards Electronic Software Updates 9.2 and select the profile, Preparing for 64-bit (BSFNs).

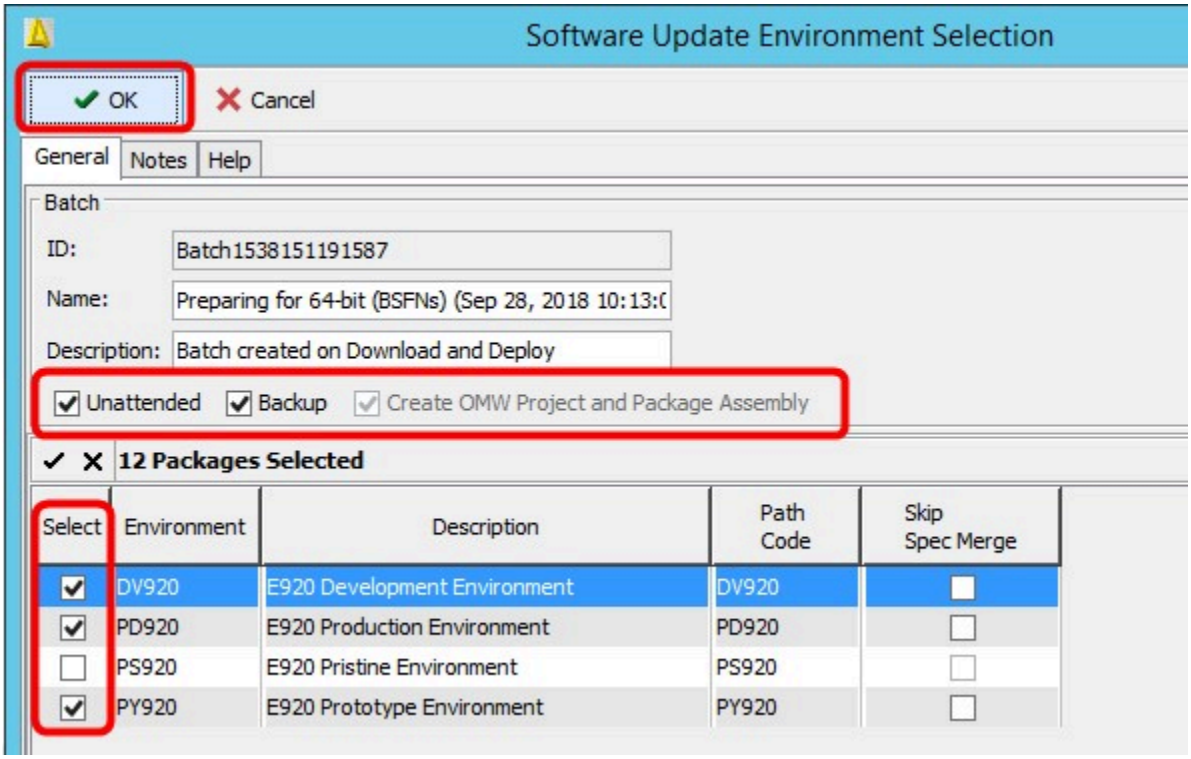


4. Click the **Search** button.

   Note: Change Assistant only displays the first 20 records retrieved.

5. Click the **>>** button in the upper-right side of the page to display all records.
6. Verify that all the ESUs were selected. If not, click the check mark above the grid to automatically select all the displayed ESUs.

7. To apply the ESUs to a target path code, click the **Download** button and select **Download To and Deploy**, <Folder to Download To>.
8. If Solution Explorer is not already running, you will be prompted to sign in to JD Edwards EnterpriseOne. Enter valid Planner credentials and click OK.
9. Select the environments to which you want to apply the ESUs. Optionally, select if the ESUs will:
    - Run in attended or unattended mode.
    - Create a backup of existing objects prior to updating the objects in each ESU package.
    - Create an OMW and package definition for each ESU package.



10. Click the **OK** button to begin the batch process to install the ESU packages.
11. Review the generated PDFs for success.
12. When all the ESU packages are installed, close Change Assistant.

4 Updating Your Tools Release to 9.2.4 and beyond (64-bit)

Deploying the Special Migration Tool (SMT)

The SMT deployment will create the 64-bit registry key, the bin64 directory containing 64-bit dynamic link libraries, and upgrade the ODBC data sources to 64-bit.

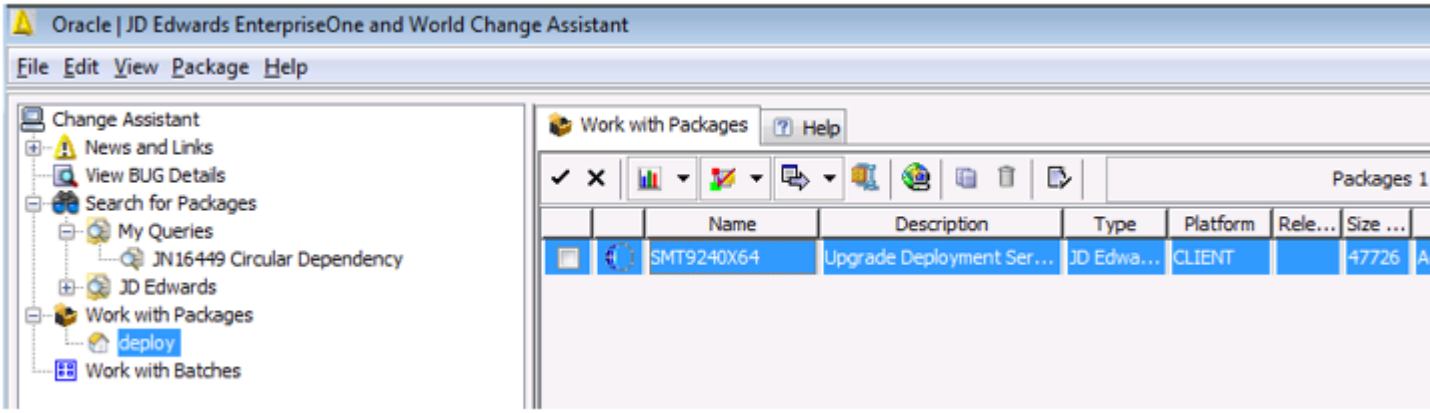
Note: Verify the following 64-bit versions of required third-party products are installed:

- Java Development Kit
- Database Client
- Visual Studio Runtime Libraries

Deploying the SMT

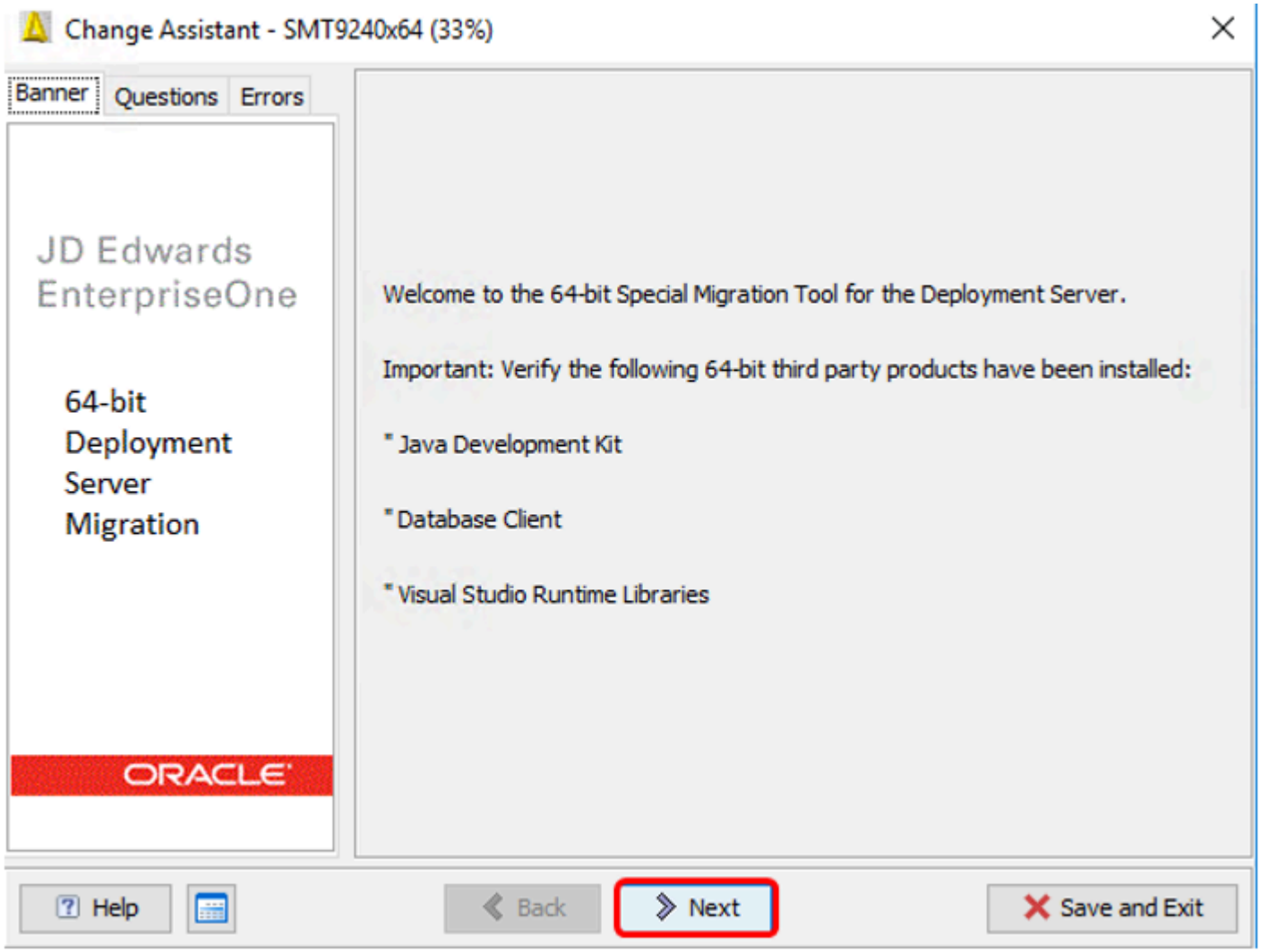
To deploy the SMT:

1. Launch Change Assistant as Administrator. (Right-click and Run as Administrator)
2. Select the folder that the SMT9240X64 PAR file was downloaded to under **Work with Packages**.
3. Select SMT9240X64-Upgrade Deployment Server 64-bit.

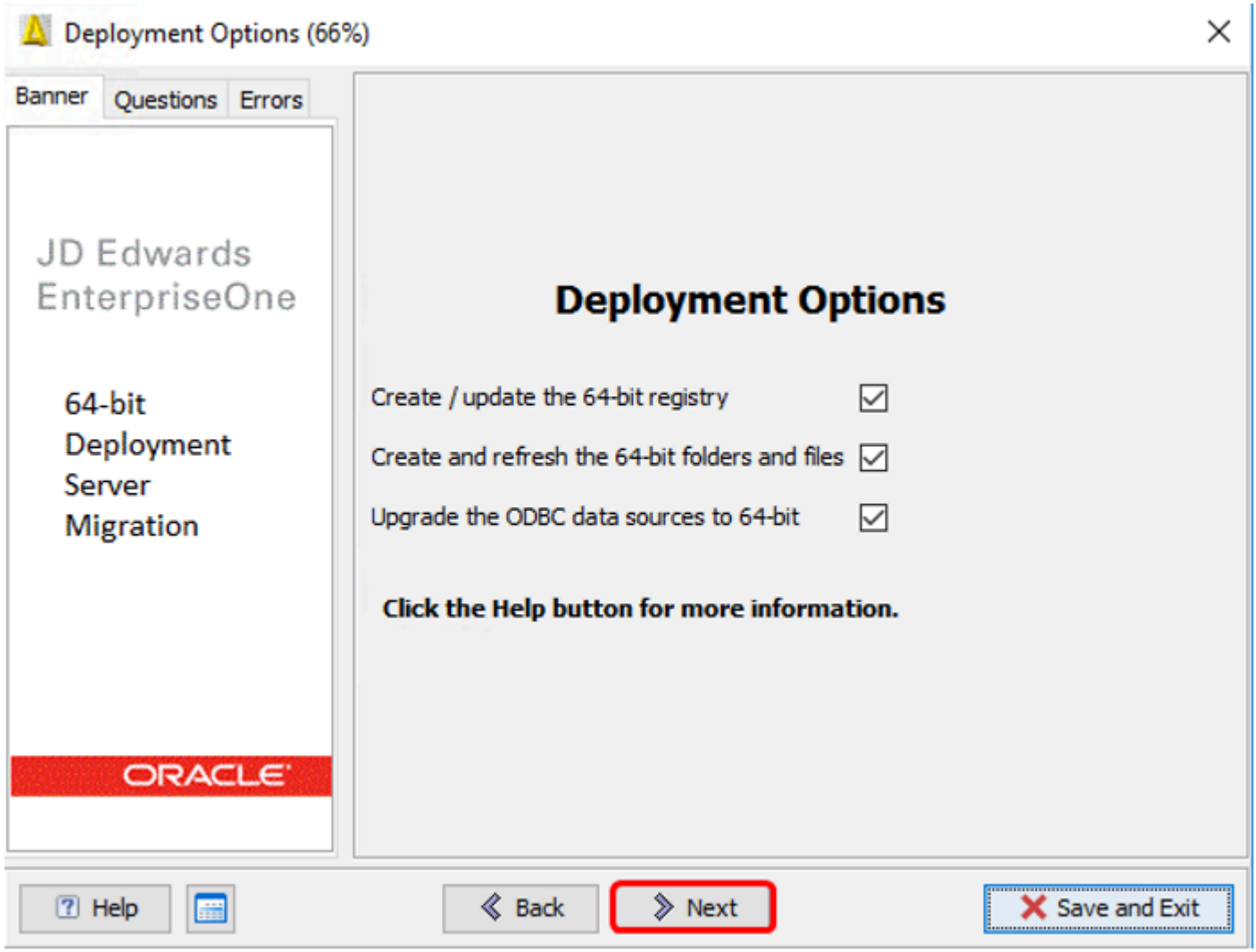


4. Click the **Deploy** button on the tool bar.

5. On the Welcome screen, click the **Next**button.



6. On the Deployment Options, check the following three options:



- **Create / update the 64-bit registry**

  This option will create the 64-bit registry key from the 32-bit registry key. If the 64-bit registry exists, it is updated from the current package. The 64-bit registry is required in order to use the Planner environments.

  Registry key locations:

  32-bit

  [HKEY_LOCAL_MACHINE\Software\Wow6432Node\JDEdwards\]

  64-bit

  [HKEY_LOCAL_MACHINE\Software\JDEdwards\]

- **Create and refresh the 64-bit folders and files**

  This option will create a bin64 folder containing 64-bit dynamic link libraries.

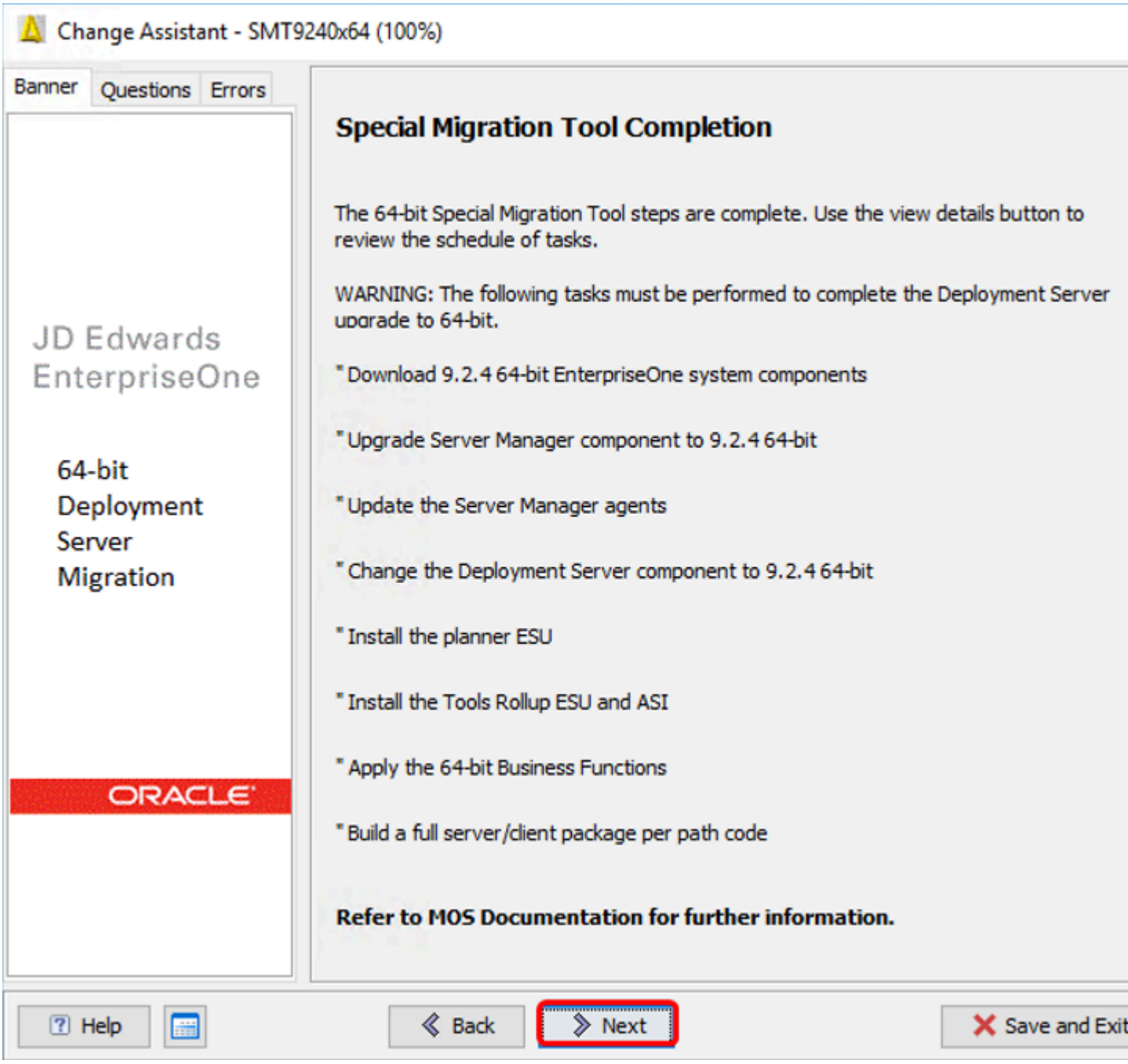
- **Upgrade the ODBC data sources to 64-bit**

  This option creates / upgrades the necessary ODBC data sources to 64-bit versions in the 64-bit registry key. The 64-bit versions of the third-party ODBC database software must be installed.

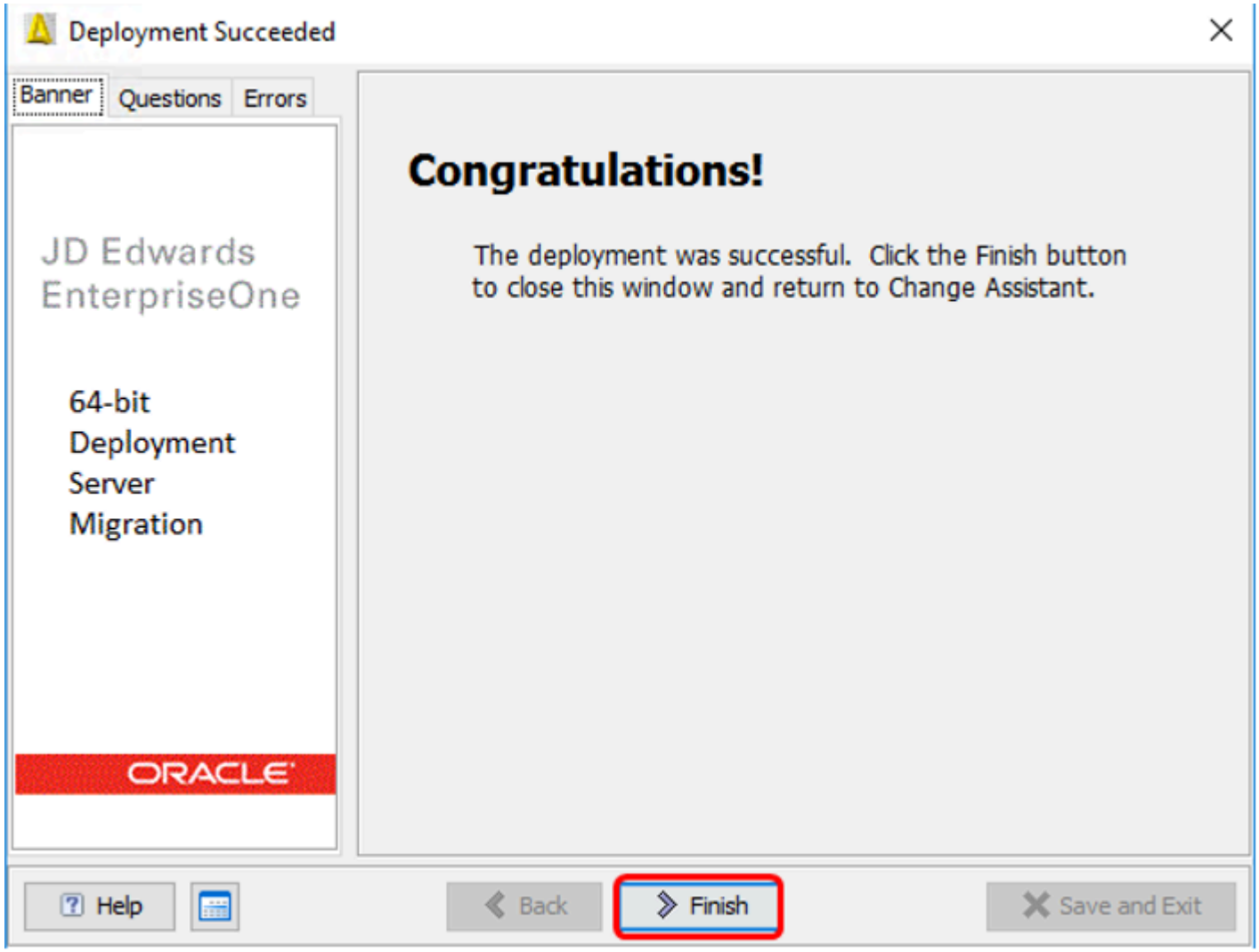
  To view the ODBC configuration, navigate to the Deployment Server location \client\ODBCDataSource.inf on the Deployment Server.

  This ODBC option does not apply when using Oracle database since the tnsnames.ora file is used. Access the x:\jde_odbcx64.log for errors and warnings.

7. Click the **Next** button.



8. On the Deployment Succeeded screen, click the **Finish** button.



Updating Server Manager Console and Agents

When you update the Management Console to a new tools release, you must update the Management Agents to the same version. Otherwise, your Server Manager cannot communicate and operate with the target machines as expected.

> **Note:** You must have Microsoft Visual Studio runtime libraries (32-bit and 64-bit) installed on all Microsoft Windows platforms where a Server Manager Agent resides, before the Server Manager Agent is updated.

Updating Server Manager Console

1. Log in to Server Manager as the jde_admin user.
2. On the left side of the Managed Homes and Managed Instances page, under INSTALL, click Manage Software.

3. Select Browse to locate the downloaded `JAR` file, `E1_ServerManagerConsole_<release>_<date>.jar` and click the **Upload** button.
4. Select Management Dashboard in the upper left of the page to return to the Managed Homes and Managed Instances page.
5. On the Managed Homes and Managed Instances page, under the Managed Instances column, locate Management Console and click the Home icon next to it.
6. In the General section of the page, under Management Console: home, click the Change button.
7. On the Change Tools Release dialog box, select the Software Component option with the version which was downloaded and distributed. Click the Change Component button to change to that version.

The update process starts immediately. Upon completion, the Management Console application will automatically restart and will prompt you to sign on.

Updating Server Manager Agent

1. Clear the browser cache.
2. Log in to Server Manager as the jde_admin user.
3. On the Managed Homes and Managed Instances page, under the Managed Home Location column, the page should display “Agent Update Required” for the different Managed Homes.
4. Select each Managed Home, and select Update at the top of the column.
5. Click OK on the dialog box to immediately begin updating the selected Management Agent. After the update is complete, Server Manager will automatically restart the Management Agent.

Configuring Server Manager for Security Security Server

1. Log in to Server Manager as the jde_admin user.
2. On the Managed Homes and Managed Instances page, under CONFIGURE, click Server Manager Users on the left.
3. Under Server Manager User Authentication, verify that the following fields have valid values:

```
Primary Security Server
Outgoing JDENET
 Port
```

4. Click Save.

Configuring Server Manager for JDBJ Data Source

1. Log in to Server Manager as the jde_admin user.
2. On the Managed Homes and Managed Instances page, under the Managed Instances column, click the Home icon next to Management Console.
3. On the left of the page, under the Configuration Panel, click JDBJ Database Configuration.
4. The value in each of the following fields in the JDBj Bootstrap Datasource section should be the same as a valid EnterpriseOne HTML Server for this same section.
    - System Datasource Name
    - Database Type
    - Database Name
    - Database Server Name
    - Database TCP/IP Port
    - Physical Database

- Object Owner
- Support Large Objects (LOBS)
- Unicode Database

5. Click the Apply button.

Setting Up Server Manager TNSNAMES (Oracle Database)

1. Log in to Server Manager as the jde_admin user.
2. On the Managed Homes and Managed Instances page, under Managed Instances column, click the Home icon next to Management Console.
3. On the left of the page, under the Configuration Panel, click JDBJ Database Configuration.
4. Scroll down to Oracle Database Settings section.
5. If the EnterpriseOne database is Oracle, then fill out the File Contents with the tnsames.ora information to sign on to the database. This information can be found in the Oracle Client location `product \12.1.0\client_1\network\admin`.
6. Click the Apply button.

Updating Deployment Server (64-bit)

Uploading Software Component

1. Log on to Server Manager as the jde_admin user.
2. On Managed Homes and Managed Instances page, on left of page, under INSTALL, click on Manage Software.
3. Select Browse to browse to a downloaded PAR file, “<release>-deployment-Serve64 06 9964.PAR” and select upload. This will take a few minutes to complete. When complete, the downloaded file will display under the Managed EnterpriseOne Software Components Section as “EnterpriseOne Deployment Server <version> <bitness> <date>”, for example: “EnterpriseOne Deployment Server 9.2.6.0 X64 10-15-2019_08_57”.
4. Under Managed EnterpriseOne Software Components, check the downloaded jar file, “EnterpriseOne Deployment Server <version> <bitness> <date>” and click on Distribute.
5. On Managed Software Component page, check the box associated with the EnterpriseOne Deployment Server and click the Distribute button . This will distribute the component to the Server Manager Agent.
6. Select Management Dashboard in upper left to return to the Managed Homes and Managed Instances page.

Change Tools Release Software Component

1. Log into Server manager as the jde_admin user.
2. On Managed Homes and Managed Instances page, under the Managed Instances column, locate the EnterpriseOne Deployment server and select on the deployment server.
3. In the General section of the page, under EnterpriseOne Deployment server: <deployment server>, click the Change button.
4. On the Change Tools Release dialog box, select the Software Component radio button for the version that was downloaded.
5. Select the second radio button “Component Change and Backup Current System for Multi-foundation”. This will save the current system and OneWorld Client Install files to a “foundation_<version>” directory and copy in the new downloaded system and OneWorld Client Install to the directories.

6. A dialog box will prompt for the user name and password for the SYSTEM datasource. Enter the user name and password and click the Ok button (Server Manager may prompt for the database user and password).
7. Click the Ok button.
8. Enter the 64-bit JRE location and click the Ok button.
9. Server Manager wiwll display several dialog boxes indicating the step it is on. Be patient thru this process.
10. When complete, it will have backed up the previous system and OneWorld Client install into a "foundation_<version>" directory and copied down the new system and OneWorld Client install into the current install location. Also, it creates a new foundation item/record of the new system for package build.

View New Foundation Directory

1. Navigate to the <deployment server>\E920 directory.
2. Validate there is a "foundation_<version>" directory created by Server Manager. The contents of this directory is the previous artifacts of the deployment server. The directory will contain all of the proper artifacts required for the deployment server, which are:
    - System
    - SystemComp
    - OneWorld Client Install
    - CD Templates\ESU&ASU
3. Validate that these existing directories under <deployment server>\E920 were updated with the new version:
4. System - validate system\bin32\ptf.log is the correct downloaded version.
5. SystemComp - validate system\bin32\ptf.log is the correct downloaded version
6. OneWorld Client Install – validate the date and time stamps has changed
7. CD Templates\ESU&ASU -¬ validate the date and time has changed
8. Navigate to the `<deployment server>\E920\package_inf file` and open an existing package definition file, for example: DV920FA.inf. Notice the path for the SSYS value now contains the path to the version path that was created, keeping the match between the package and foundation it was built against, consistent.

Installing and Applying the Planner ESU and Special Instructions

The Planner ESU contains special features that update the specifications and dynamic-link libraries (DLL) in the Planner path code. The Planner ESU distributes enhancements and fixes to the software programs. Planner ESUs provide an HTML (JNxxxx.htm) file that contains the special instructions for the software update. You will find the HTML file in the extracted directory for the Planner ESU.

Downloading Planner ESU

1. Obtain the latest 64-bit Planner ESU from the Oracle JD Edwards EnterpriseOne Update Center.
2. Click the Electronic Software Updates link in the News and Links section on the left-hand side of the screen.

   On the home page for Electronic Software Updates, you will see a Planner ESUs section. Links to the Planner ESUs are listed by EnterpriseOne applications release.
3. Click the Get It Now link for the appropriate JD Edwards EnterpriseOne Applications Release (9.2) to access the 64-bit Planner ESU currently available for download.

4. Download to the Deployment Server.

Running Planner ESU

1. On the Deployment Server with no other JD Edwards EnterpriseOne processes running, execute the downloaded self-extracting executable.
2. Follow the on-screen instructions to install the 64-bit Planner ESU.
3. After the executable has completed, follow the special instructions. The special instructions for 64-bit Planner ESU are within an HTML file in the extracted Planner ESU directory.
4. On the Deployment Server, sign in to the JDEPLAN environment of EnterpriseOne.
5. Launch Software Updates from the menu GH9612.
6. Locate and select the new planner ESU in the list of the ESUs.
7. Select the target path code to which the 64-bit Planner ESU should be applied.

Note: It is recommended to apply the 64-bit Planner ESU to every path code.

Applying the Tools Application Enhancement Rollup ESU

The Tools Application Enhancement Rollup ESU distributes enhancements and updates to the software programs. Planner ESUs provide an HTML (JNxxxx.htm) file that contains the Special Instructions for the software update. You will find the HTML file in the extracted directory for the Planner ESU.

Downloading the Tools Application Rollup ESU

1. Log in to the Oracle JD Edwards EnterpriseOne *Update Center* using your Oracle SSO ID.
2. Click the Electronic Software Updates link in the News and Links section on the left-hand side of the screen.
3. Enter `24710277` in the **BUG** field under Search.
4. Check License Agreement.
5. Click Search.
6. Click Add **+** next to the ESU (JN******) on the right to add it to the Download Basket. You can ignore the other updates.
7. Under Download Basket, click **Items(1)** to access the download basket.
8. Click Download next to the ESU.

   Note: If the results also show'Dependency' ESUs and those have not been applied to the target pathcode, they should also be downloaded and applied before applying the Tools Application Enhancement Rollup ESU.

9. Save the ESU executable to a directory on the Deployment Server.

   Note: If you downloaded the ESU from a machine other than the Deployment Server, transfer the downloaded files to a location on the Deployment Server prior to running the executable.

Running the Tools Application Rollup ESU

1. On the Deployment Server, navigate to the folder where the downloaded ESU executable files are located.

2. Run the executable file for the Tools Application Enhancement Rollup ESU (for example, `JN*****.exe`).

   Tip: Oracle recommends running the installer executable 'As Administrator' and verify that Solution Explorer is not running.

   Note: If there were additional ESUs marked as 'Dependency' that have not been applied yet, they can be run at any time, but must be done prior to applying the ESUs that are listed below them.

3. Accept the defaults on installer and execute to completion.

Applying the Tools Application Rollup ESU

1. From the Deployment Server, launch the Solution Explorer, and sign in to the JDEPLAN environment using the JDE user name and password.
2. Run the Electronic Software Updates application from the GH9612 menu.
3. In the grid, locate the Tools Application Enhancement Rollup ESU to install. Select the record and click the **Next** button.

   Note: If you are unable to locate the ESU, enter the name (JN*****) into the QBE line above the grid and click the **Find** button.

4. Select the path codes to which you want to apply the Tools Application Enhancement Rollup ESU. Optionally, select if the ESU will:
   - Run in attended or unattended mode.
   - Create a backup of existing objects prior to updating the objects in each ESU package.
   - Create an OMW and Package definition for each ESU package.
5. Click the Next button to begin the batch process to install the ESU package.
6. Review and ensure the successful completion of each of the PDFs as they are produced by the workbenches that are running.

   Note: If the ESU is running in Attended Mode, click the Next button for each workbench until the process is complete.

Deploying the Automated Special Instructions (ASI)

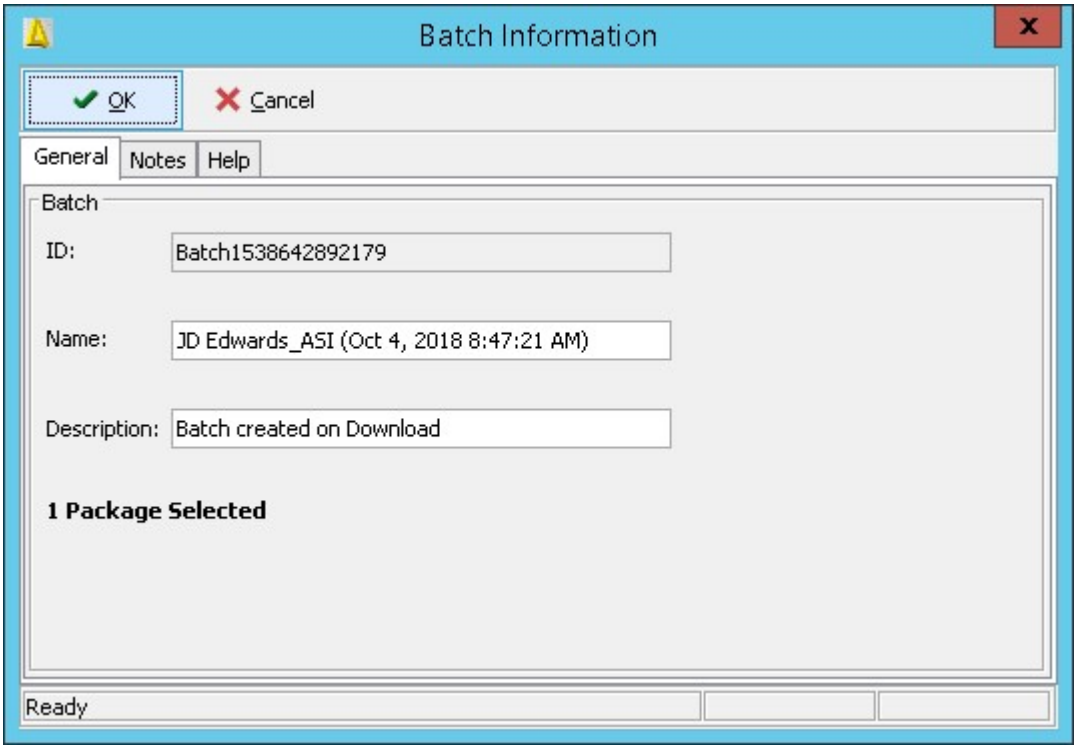
ASIs are special instructions for software updates that are executed automatically. They are downloaded separately from the software update you are installing, either using Update Center or Change Assistant. The JD Edwards EnterpriseOne Tools Release 9.2 ASI is a .par file and can be found using TL92* as the search criteria for the Update Name within the Update Center or Change Assistant.

Note: For Tools Release 9.2.8x and later, P96ATRC has replaced the ASI. See *E1: ESU: Running Automated Tools Release Configuration (ATRC) - Tools Release 9.2.7 and later (Doc ID 2991360.1)* for details.

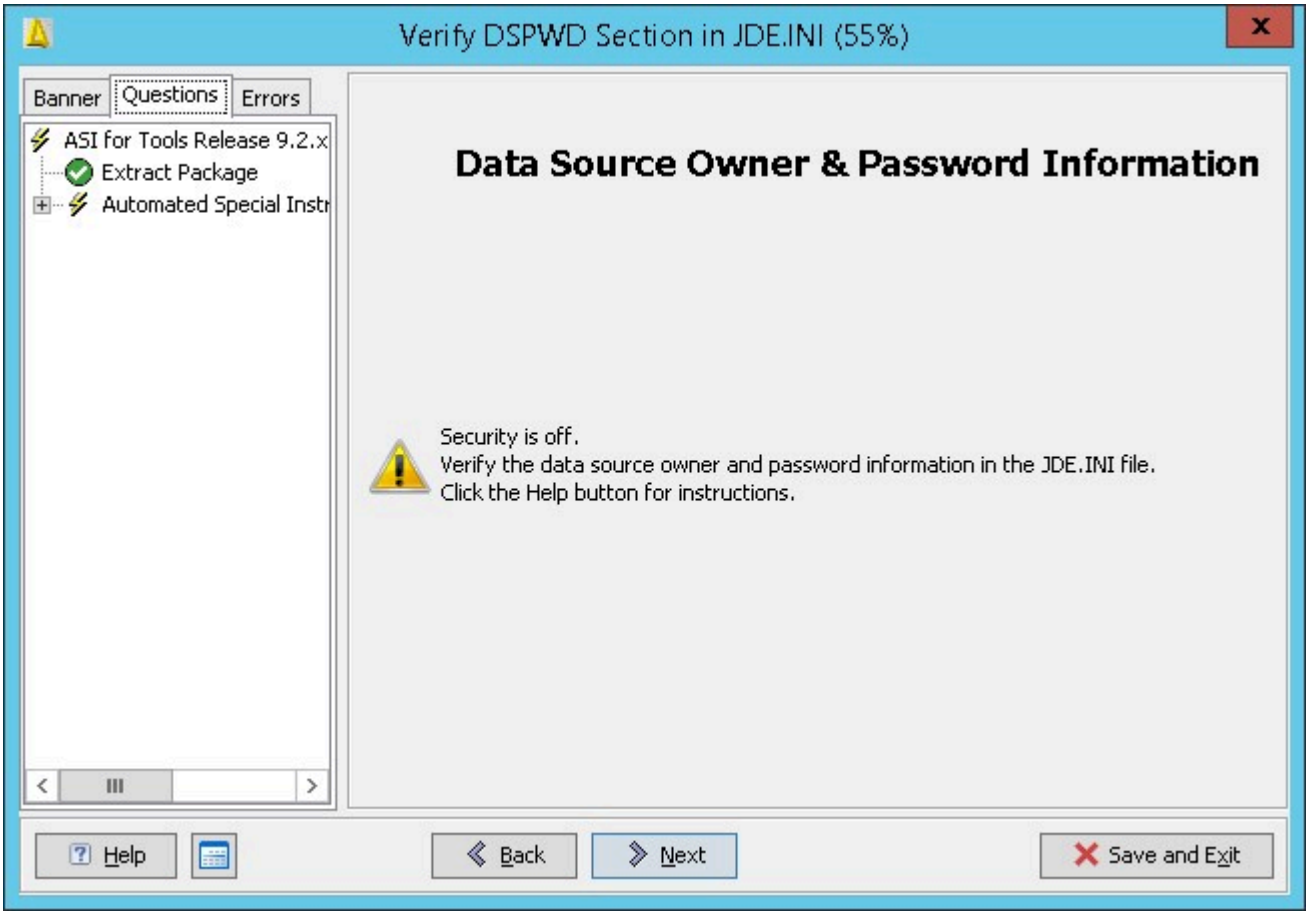
Deploying ASI

1. Obtain the latest TL92* ASI from the Oracle JD Edwards EnterpriseOne Update Center.

2. Download and access with Change Assistant.
3. Using Change Assistant, expand the Work with Packages node.
4. Navigate to the ASI package that you downloaded.
5. Select the ASI package by checking the box next to the package.
6. Click Deploy on the Work with Packages toolbar to start the ASI.

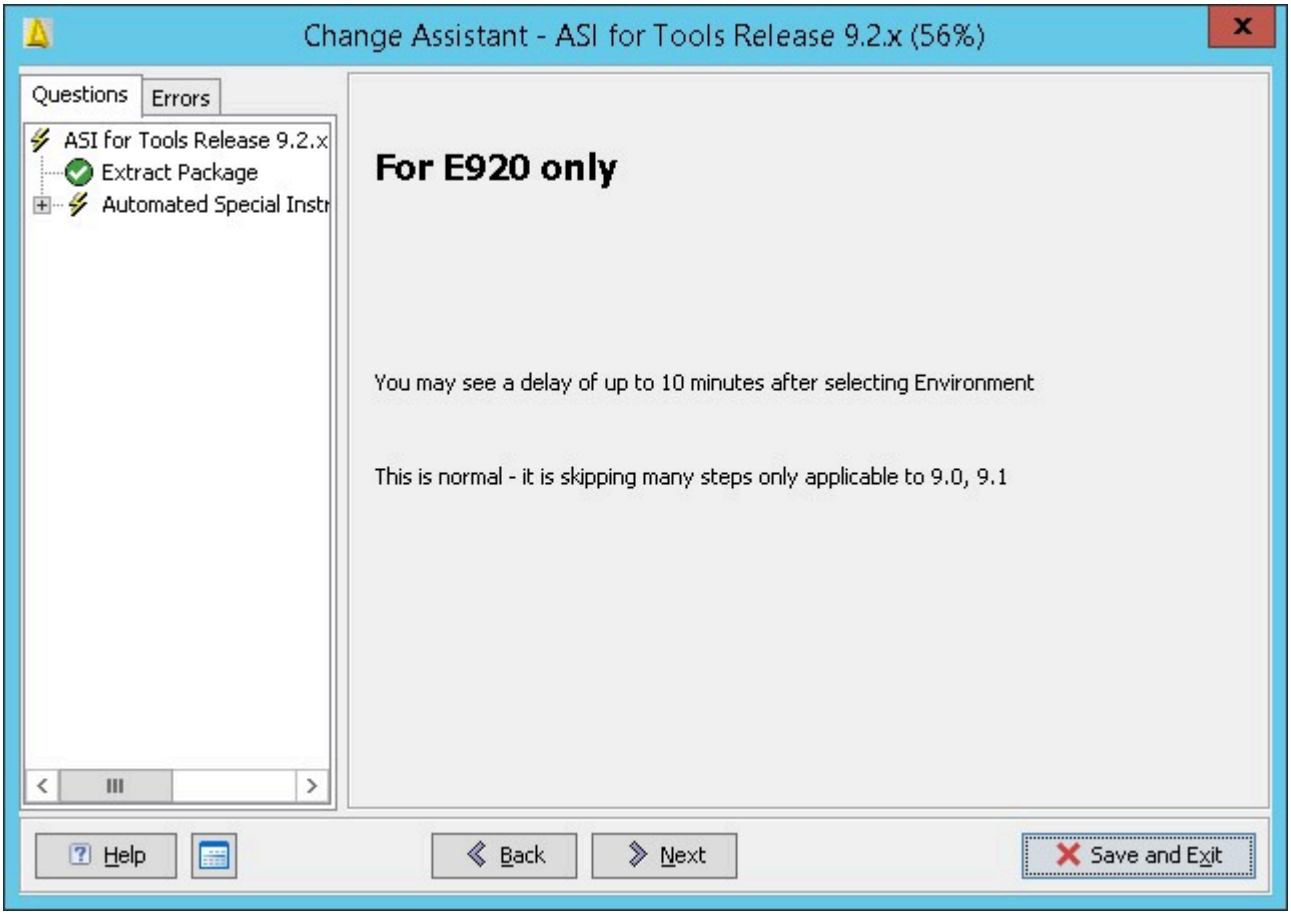


7. Click OK to continue.



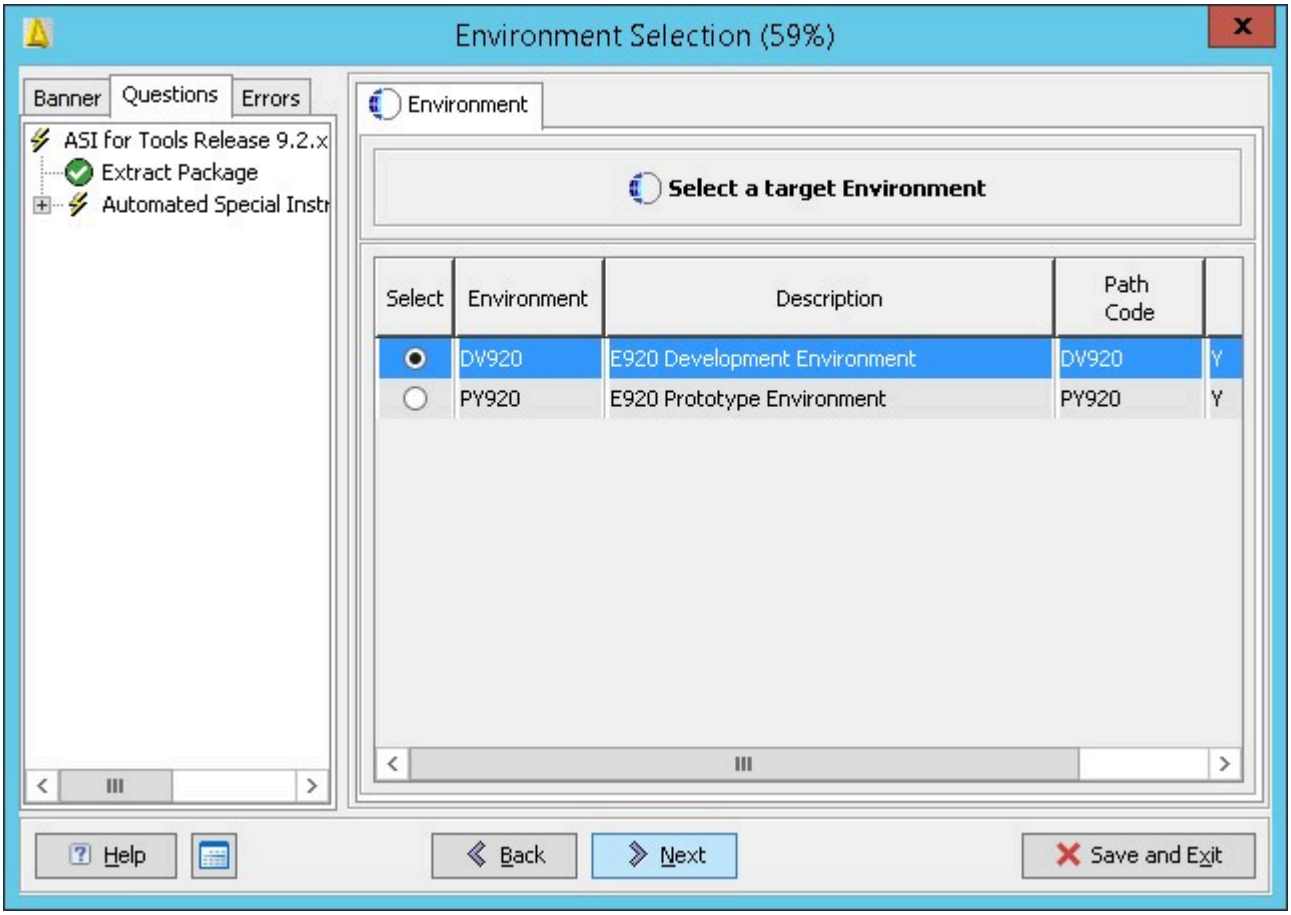
Note: If you have changed the password for any table owners created during the Platform Pack installation, they must be defined in the jde.ini file. The ASI assumes that the table owner's password is the same as the table owner, unless it finds overrides in the jde.ini file on the Deployment Server.

8. Click **Next** to continue deploying new tasks.



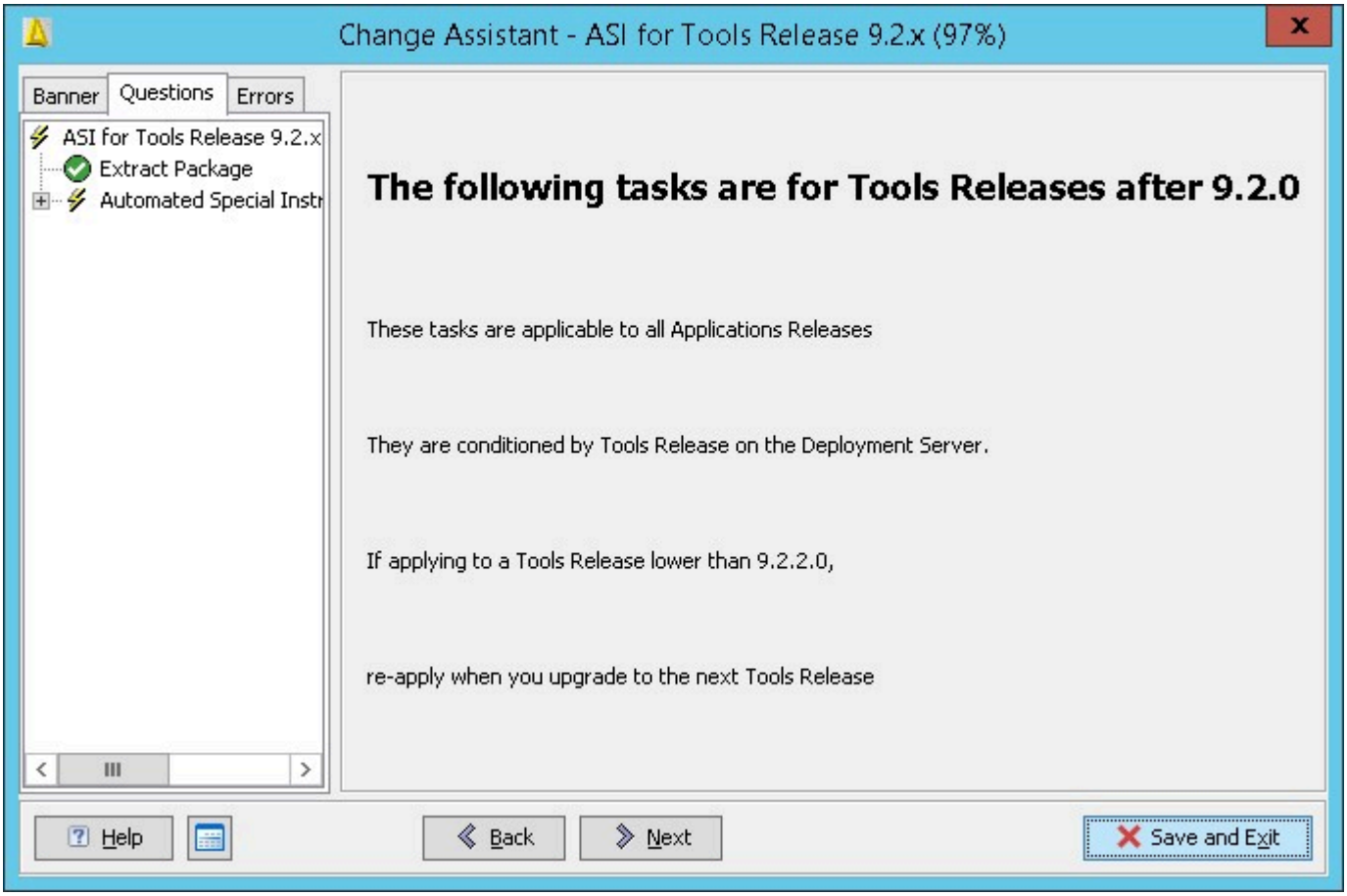
9. On Select Deployment Option, select **Install the ASI to the selected environment** and click **Next**.

10. Select the target environment and click **Next**.

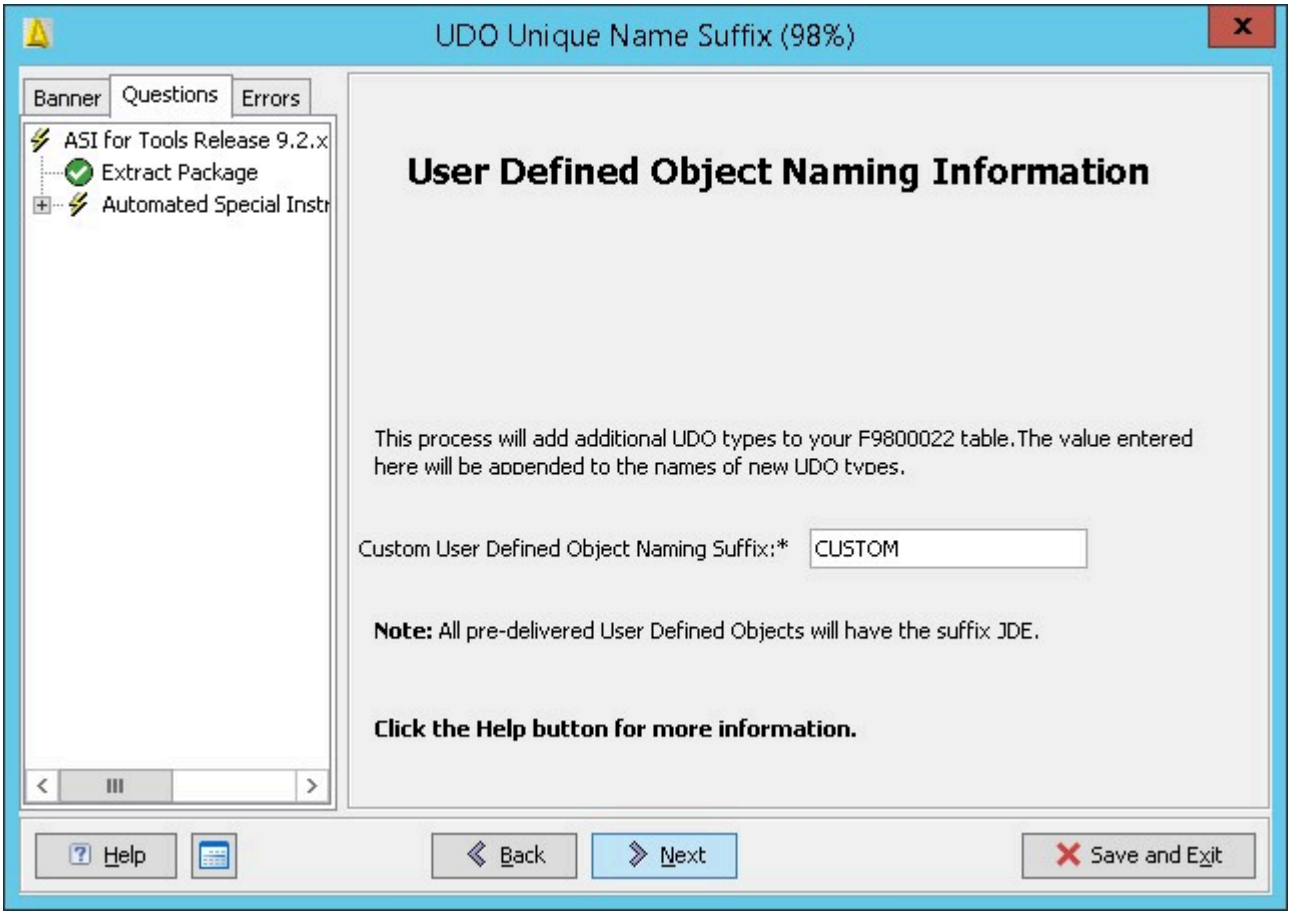


11. Click **Yes** on Warning - Unknown Release.
12. Click **Yes** on Warning - Transaction Data.

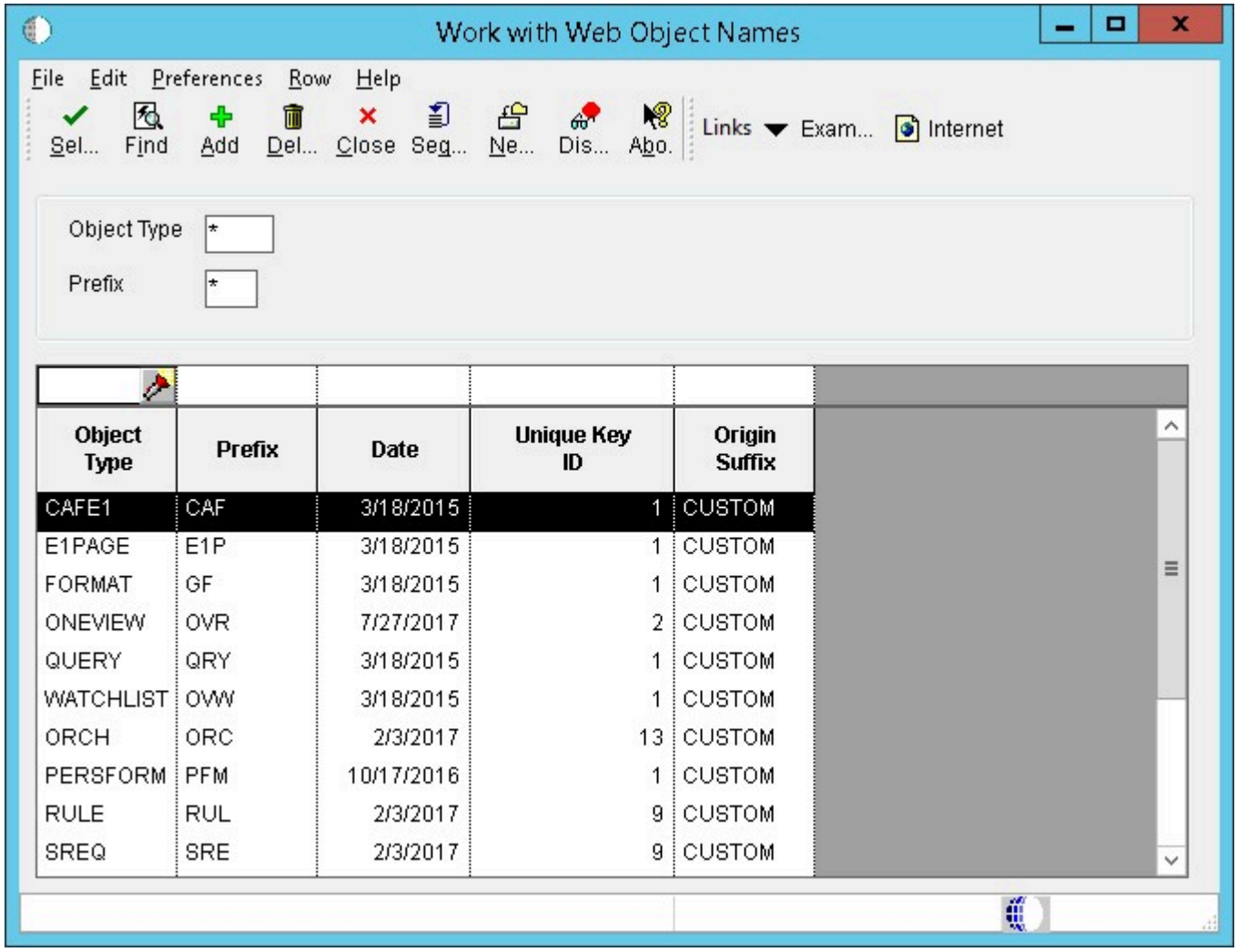
13. Click **Next**to continue.



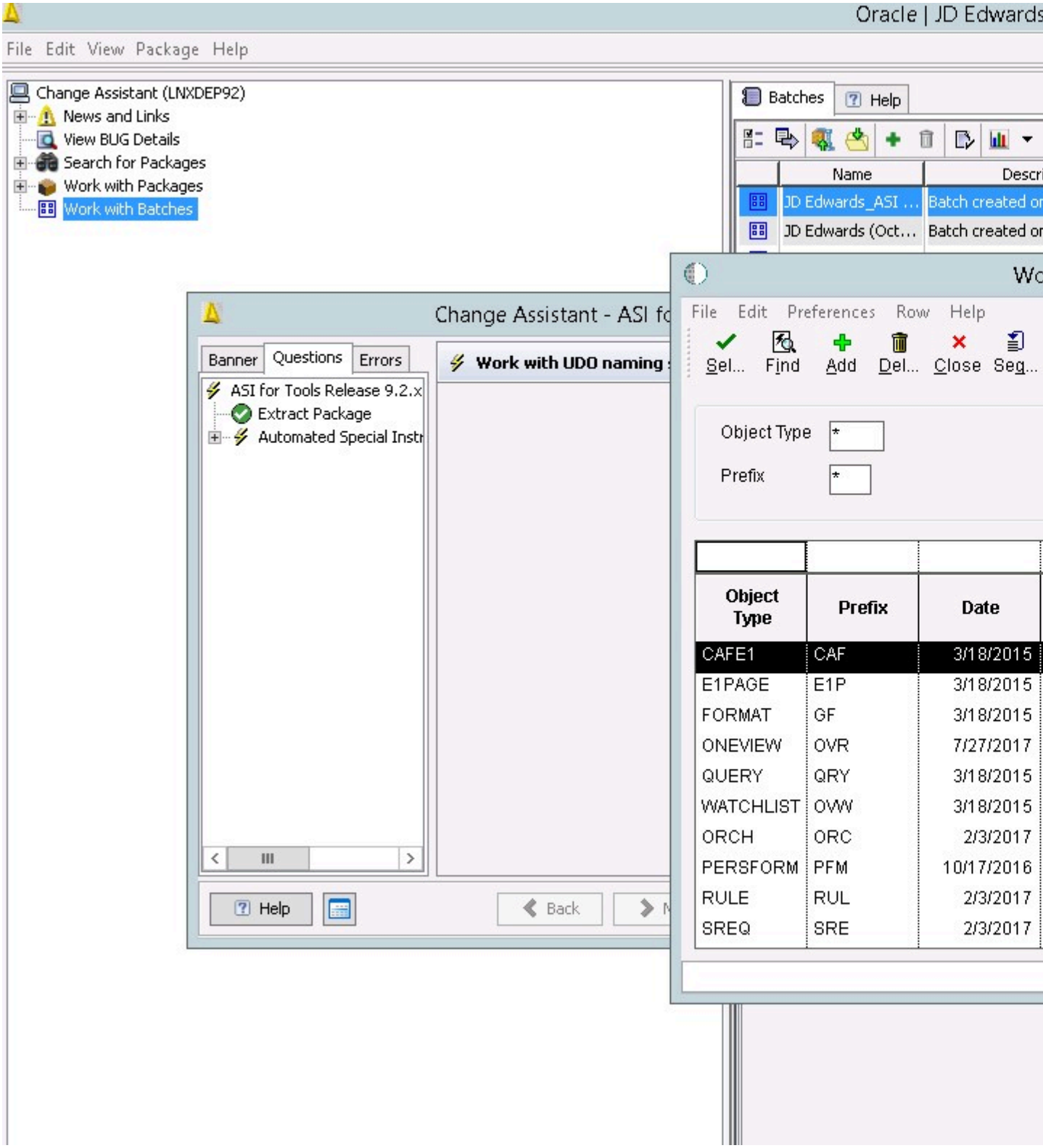
14. Enter a UDO suffix and click **Next**.



15. On the Work with Web Object Names select a new UDO type.

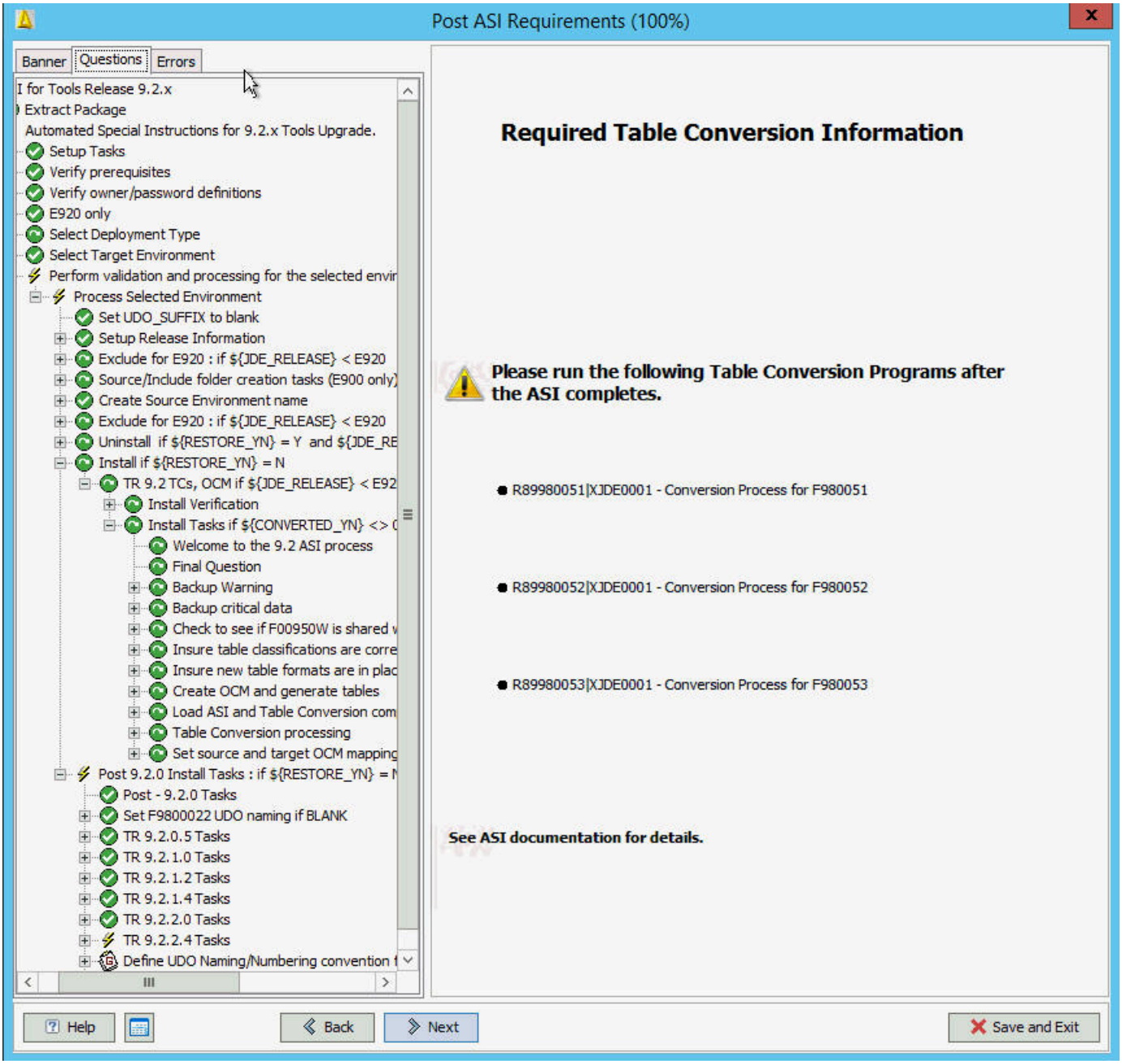


16. Update the Suffix and click **OK**.



17. Execute table conversions manually.

The ASI will only prompt you to run the table conversions if it has detected data in F980052 in the System - xxx data source and no data in F980052 in the associated Control Tables data source.



The dialog above appears only if F980051, F980052, and F980053 tables are in the incorrect format and have records in the F980052 table.

> **Note:** The following information assumes the ASI has been applied to target environment DV9X,X which has an associated WAN environment of JDV9XX. The associated WAN environment is to be used as the source environment for the table conversions.

Verify whether there are any records in your version of these three tables in System - xxx. You won't be able to use Universal Table Browser, you will need to use sqlplus / strsql or other database tool. If there are no records, no action is required. The ASI will have generated the tables for you into the Control Tables – xxx data source for your target environment.

The ASI applied the correct .par file to your Deployment Server.

The ASI reset the OCM for these 3 tables to point to Control Tables – xxx for the environment you applied the ASI to, plus any other environments that share the same path code.

Your source and target environment for the table conversion must be different. The ASI sets the OCM for these three tables to System - xxx for the WAN environment (JDV9XX) associated with the environment you ran the ASI against.

Sign in to JDEPLAN and run the table conversions one at a time and submit from Batch Versions. Click on Properties and change the source environment to JDV9XX and the target environment to DV9XX.

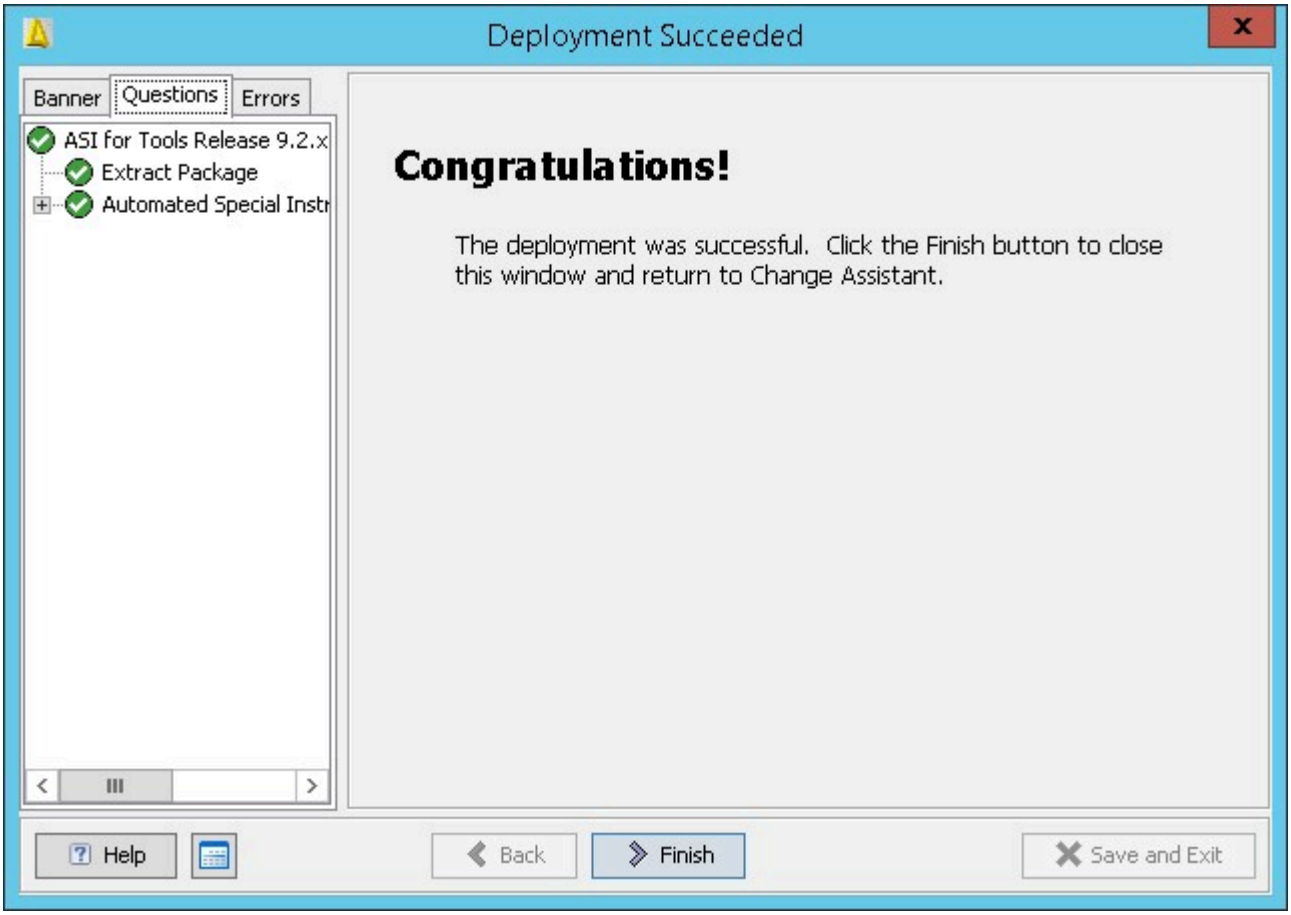
Once the table conversions have run, reset the OCM in JDEPLAN for the source environment to point to the Control Tables - xxx again.

If you have multiple environments with different Control Tables associated with this path code, copy the output table from the table conversion to each of the environments' Control Tables. For example:

- You have DV920 JDV920 DV920XX DV920YY and DV920XX where YY have their own unique Control Tables.
- Run the table conversion against JDV920 DV920, and then copy the F980051, 52, 53 from Control Tables - Test to the Control Tables for DV920XX and DV920YY.

18. In Tools Release 9.2.5 the E1Local database has been removed. The F986020 and F986030 have been moved to Business Data. You must generate the tables in Business Data for the target environment.
19. If building packages through the web package build application (Release 9.2.5), you must add new OCM mapping for all HTML environments.
20. If using Virtual Batch Queues (Release 9.2.5) run R986110V after applying the Tools Rollup ESU. This UBE sets values for the Virtual Batch Queue. This only needs to be run once and updates the F986110 in server maps.

21. Click **Finish**.



Verifying ASI Table Conversions

Verify the success of the ASI table conversion by verifying tables, security, and runtime.

Tables

UDO	Table	Verification
One View Reports and EnterpriseOne Page	F952400	Has the WOBNM column.
EnterpriseOne Watchlist	F952420	Has the WOBNM column.
Queries	F952430	Has the same record count as F98950.
Grid Formats	F952440	Has the same record count as F98950.
CAFE1	F952450	Has the same record count as F98950.

Security

Use (P00950) to verify that the UDOs have the same level of security as before the EnterpriseOne Tools Release upgrade.

Runtime

After the ASI runs, you should verify access to UDO is the same as before the EnterpriseOne Tools Release upgrade.

If the ASI halts or table conversions do not successfully run, it is recommended to fix the error and run the ASI uninstall program. The ASI uninstall program performs several clean-up operations, table clean-up, and restoring. This action resets your system and enables you to run the ASI as if it were the first time.

5 Implementing 64-bit Processing on the Enterprise Server

Applying 64-bit Component to Enterprise Server

Uploading the Software Component

1. Log in to Server Manager as the jde_admin user.
2. On the left of the Managed Homes and Managed Instances page, under INSTALL, click Manage Software.
3. Select Browse to browse to a downloaded PAR file, **`<release>-Enterprise-Server-<Platform>64_06_XX.PAR`** and select Upload. The upload will take a few minutes to complete. When complete, the downloaded file will display under the Managed EnterpriseOne Software Components section as **`EnterpriseOne Enterprise Server <version> <bitness> <date>`**, for example: EnterpriseOne Enterprise Server 9.2.6.0 X64 10-15-2019_08_57.
4. Under Managed EnterpriseOne Software Components, select the downloaded JAR file, **`EnterpriseOne Enterprise Server <version> <bitness> <date>`** and click Distribute.
5. On the Managed Software Component page, select the box associated with the EnterpriseOne Enterprise Server and click Distribute to distribute the component to the Server Manager Agent.
6. Select Management Dashboard in the upper left of the page to return to the Managed Homes and Managed Instances page.

Changing the Enterprise Server Component

1. Log in to Server Manager as the jde_admin user.
2. Click Manage Dashboard.
3. Click the Enterprise Server managed instance.
4. Click the Change button under Software Component Version.

   > **Note:** Do not select Change for E1 Application Component Version.

5. Select the … 9.2.6.0 x64 … component.
6. Click the Change Component button.

   > **Note:** A 64-bit component change in Server Manager requires database connection to validate the applications release is 9.2.

7. The system will prompt for a database user name and password. Enter the appropriate credentials.
8. Click Login.
9. Server Manager detects the new component is 64-bit and displays a message indicating the necessary third-party minimum technical requirements.
10. Click OK.
11. Click OK.
12. Enter the 64-bit Database Client Home Directory and click OK.

    > **Note:** Verify that the jde920 user running the Server Manager Agent has access to the 64-bit database client home directory.

13. Enter the path to the 64-bit JRE directory and click Submit.

> **Note:** Verify that the jde920 user running the Server Manager Agent has access to the 64-bit JRE home directory.

Server Manager will change the component on the Enterprise Server. Validate that the version has changed to 9.2.6 x64 under the Software Component Version section.

> **Note:** For Enterprise Servers that are Microsoft Windows-based, verify that the JDE service in services.msc is configured properly as it may have been changed.

The Enterprise Server is now ready for a 64-bit package build and deployment.

6 Building and Deploying a Full 64-bit Package

Understanding the Code Converter

This video describes the 64-bit code conversions for business functions.

Understanding the JD Edwards EnterpriseOne 64-bit Code Converter

Building a Full 64-bit Package

Assembling a Full Package

1. Access JD Edwards EnterpriseOne on the Deployment Server.
2. Sign in to the DEP920 path code.
3. Enter GH9083 in the Fast Path and press **Enter**.
4. From System Administration Tools, select the Package and Deployment Tools menu, Package Assembly.
5. On the Work with Packages form, click Add, and then on the Package Assembly Director form, click Next.
6. On the Package Information form, complete the Package Name, Description, and Path Code fields.

   > **Note:** The name of the package cannot be longer than eight characters.

7. Select the Director option, and click Next.
8. On Package Type Selection, select Full, and click Next:
9. On the Foundation Component form, verify you are selecting the 64-bit foundation.

   This could be the Default Foundation if the component change on the Deployment Server updated the default system.

   The system builds your package from the deployment data source that is associated with the default object path. Verify that the correct location appears on the form.
10. On the Object Component form, click Next.
11. On the Default Object Component form, click Next.
12. If desired, add a Feature, click Browse.
13. On the Feature Component Selection form, click Find to display a list of features, select one or more features, and then click Select to add the features that you want to include in your package.

    Typically, the WLSH4A (Weblogic H4A) Feature is included in the full package build for Development Clients.
14. Click Close to return to the Features Component form.
15. On the Feature Component form, click Next.
16. On the Language Component form select and include any installed languages and click Next.
17. Click End.
18. Click Activate on the Row menu.

19. Click Define Build from the Row menu.
20. Upon completion, review the package build PDFs and log files. In particular, examine the new logs for the 64-bit file conversion under the <package name>\work directory on the Deployment Server. Those files will be name *_64.log(ex: BSFN_64.log).

Building a Full Package

1. Access JD Edwards EnterpriseOne on the Deployment Server.
2. Sign in to the DEP920 pathcode.
3. Enter GH9083 in the Fast Path and press **Enter**.
4. Click Package Build on the right-side.
5. Find and select the defined package that you want to build.
6. If the package definition has a status of In Definition, you must change the status to Assembly-Definition Complete before you build the package. To change the status, select the package and select Activate from the Row menu.
7. On the Package Selection form, in the Express Option pane, select one of these options:
   - **Director** - Select this option if you want to configure the package build. Director enables you to navigate the package build definition forms.
   - **Express** - Select this option if you want to accept the default build parameters. Express enables you to accept the default options for the package build and skip the package build definition forms.
8. If you selected the Express option, skip to the Reviewing Package Selections task. If you selected the Director option, continue with the next task.
9. For Foundation, click the icon to the left.
10. On the Foundation Component Selection page, select the 64-bit foundation.
11. On the Package Build Location form, select one or both of these options:
    - **Client** - Select to indicate that the package is being built for installation on client workstations.
    - **Server(s)** - Selected by default to indicate that the package is being built for installation on one or more servers.
12. When building a server package, you can specify the Shared Location for the shared spec database and click **Next**.

    > **Note:** The default shared spec database is always the central objects data source for the package path code.

13. To select a server on the Server Selection form, double-click the row header for the server. A check mark indicates your selection. You can select multiple servers.

    > **Note:** Servers are automatically selected for an update package. They are selected based on the server selection of the parent package.

14. Click **Next**.
15. On the Build Specification Options form, select Build Options to take the package definition and copy and convert objects from the central data source to the replicated format used by workstations.
16. Complete the following fields and click **Next**:

Field	Description
All Specification Tables	Select this option if you want to build all specification tables into the package.
Individual Specification Tables	Select this option if you would like to select individual tables to include in the package. All of the tables listed on the Individual Specifications Selection form will be included in the package.

Stop Build Option	Indicate the point at which the system should stop the build. You can continue building on all errors, stop building on specification errors, stop building on business function errors, or avoid compressing when errors exist.
Replace jde.ini	For update packages, indicate if you want a new jde.ini file delivered with the package. Leave this unchecked unless the jde.ini file has changed. For example, the jde.ini may change when you perform upgrades or when you re-configure in release master.

17. If you chose to build individual specification tables, the Individual Specification Selection form appears.
18. To indicate that you do not want to build a specification table, clear the option.
19. Click **Next**.
20. Complete the following fields and click **Next**:

Field	Description
Build Mode	Specify the build mode, such as debug or optimize.
Stop-Build Option	Specify what action to take if errors occur while building business functions.
Build BSFN Documentation	Specify whether you want to build the documentation for the functions.
Clear Output Destination First	Indicate if you want the destination directory for the functions to be cleared before the build.
Generate NER	Generates the 64-bit NER source and include files.
Generate NER Headers for Opposite Bitness	Genertes teh 32-bit NER include files used if developing business functions.
Build Functions	Compiles and builds the included and source files into dlls.

21. On the Compression Options form, select Compress Client Options if you would like to compress the client package. Select this option to compress the applications included in the package, and to specify options for the compression process.
22. If you are compressing the client package, select from these options:

Option	Description
All Client Directories	Select to compress all of the directories listed on the Individual Directory Selection form. Individual Client Directories Select to compress only certain directories which you specify.
Compress Data	Indicate whether to compress the data in a package after the package is created. Compress Data compresses the Supported Local Database that is associated with this package.
Compress Foundation	Indicate whether to compress the foundation files in the package after the package is created. Compress Foundation compresses the foundation that is associated with the package.

23. Select Compress Server Options if you would like to compress the server package.
    - Select this option to compress the applications included in the package, and to specify options for the compression process.
    - You should select Compress Server Options if you plan to build the package on one enterprise server and deploy it to another enterprise server.

24. If you are compressing the server package, select from these options:

Option	Description
All Server Directories	Select to compress all of the directories listed on the Individual Directory Selection form.
Individual Server Directories	Select to compress only certain directories which you specify.

25. Click **Next**. If you chose to compress individual directories, the Individual Directory Selection form appears.
26. On the Individual Directory Selection form, indicate that you want to compress a directory for the client or server by clicking its option to select it and click **Next**.
27. If the package does not include features, skip to the next task. On the Build Features form, if you want to build a feature.inf file with the package, select the Build Feature INFs option. When you select this option, the Compress and Build options become available.
28. Click **Next**.
29. Review the package build selections and click **End**.

Deploying a Full 64-bit Package

After you define and successfully build a package, use the JD Edwards EnterpriseOne Deployment Director program (P9631) to schedule the package for deployment to individual workstations or enterprise servers.

1. Fast path to GH9083 menu.
2. On the right-pane click Package Deployment.
3. Click **Add** and **Next** to setup the deployment of the package.
4. On the Package Selection form, select the package that you want to deploy, and click **Next**.
5. On the Package Deployment Targets form, select a target to indicate the type of machines to which you want to deploy the package, and click **Next**.
6. On Package Deployment Attributes, complete these fields and click **Next**. These are both Optional fields to fill out:
    - Mandatory Installation
    - Date/Time
7. On the Deployment Client Workstations Selection form, select the client workstations to which you want to install the package, and then click **Next**.
8. Select a workstation by double-clicking in its row header. A check mark appears in the row header for each workstation that you select.
9. On the Enterprise Server Selection form, select the name of the server to deploy the package to. Click **Next** and **End**.
10. Under the Package, expand and then expand again under Enterprise Server.
11. Highlight the date and time.
12. Select **Row/ Deploy** and this will deploy the package to this Enterprise server.
13. When a user signs on to a client machine and then logs in they will be prompted to install the package.

Troubleshooting

Unable to Install Client Package at Sign-On

If you cannot install the package at sign-on, you may need to uninstall your existing E1 Client and then install the new package manually.

Unable to Install Web Dev Feature into an Existing WebLogic Instance

If you cannot install the Web Dev Feature into an existing WebLogic instance, you may need to uninstall WebLogic and install it again with a new 64-bit JDK and re-run the installer.

7 Configuring the Deployment Server for Development Client Install

Setting Up a Development Client Installer on the Deployment Server

See this link for information about setting up a Development Client Installer on the Deployment Server:

Understanding the JD Edwards Clients (Development and Web)

8 Validating Your 64-bit Implementation

Validating EnterpriseOne 64-bit Implementation

To verify that you have successfully implemented to Tools Release 9.2.6 64-bit navigate to the Server Manager Console and select the Enterprise Server instance. Under Software Component Version you will see:

- EnterpriseOne Enterprise Server 9.2.6.0 X64 <date stamp>

Verifying Bitness on IBMi

The tools processes print the bitness in their jde.logs and joblogs when they start. In addition, the copyright information has the bitness embedded in most of the EnterpriseOne IBMi programs and service programs at compile time. The bitness in the copyright details are visible when executing DSPPGM or DSPSRVPGM with the option “DETAIL(*COPYRIGHT)”.

The copyright detail for the Tools *SRVPGM, E920SYS64/JDEKRNL, on JDEOW1 includes:

- << 64Bit Time & other data types >>

The copyright detail for the Apps *SRVPGM, PD92064/CAEC, on JDEOW1 includes:

- << 32Bit Time & other data types >>

The bitness between the tools code and the business function code should match to function correctly. The tools/apps combination on JDEOW1 (E920SYS64/PD92064) needs to be updated.

9 Performing Post Upgrade Tasks

Updating the Release Master

If you have upgraded the Deployment Server to be 64-bit and enabled all pathcodes to be 64-bit, the Release Master (P00945) should indicate the release is 64-bit.

Tools Release 9.2.6 and beyond presents the ability to have a 64-bit only system. The Release Master application is used to indicate a 64-bit only system. When you have all the pathcodes, the Enterprise system, and the client foundation converted to 64-bit, and enabled 64-bit in the Release Master, the package will only contain `include64` and `source64`. The package will build the dlls with a 64-bit settings. The client installation will have `bin64`, `obj64`, `lib64`, `source64`, and `include64` files. The server will deploy the `bin64` directory. In Busbuild, you will be able to compile, link, and build in 64-bit only. After the 64-bit release is enabled, you cannot revert to 32-bit.

If you update all pathcodes to 64-bit and have enabled the Release flag for 64-bit, the package will only extract and build the 64-bit artifacts, resulting in the following:

- When assembling the package, you can only select 64-bit foundations.
- During the retrieval of the .c and .h files on the Enterprise Server, package build will only extract the 64-bit files, `include64` and `source64`.
- The business functions build process on the Enterprise Server will build the 64-bit artifacts for the 64-bit dlls or `.so` or `pgm` files.
- The process to copy the include file and the source file to the Development Client will only transfer the `checkin\include64` file and the `checkin\source64` file to the Development Client.
- The client Busbuild will only build and compile the 64-bit artifacts to create the `bin64`, `lib64`, and `obj64` files.
- Only the 64-bit directories will be compressed. These directories are `include64`, `source64`, `bin64`, `obj64`, and `lib64`.

Note: Once you update the Release Master to 64-bit, you cannot revert back to 32-bit.

Note: Beginning with Tools Release 9.2.6.x, when the Release 64-bit Flag is turned on in Release Master P00945, you should only build a 64-bit package to deploy to the Development Client. This Development Client must not have either the `\include` or `\source` folder under the path code. The ER compare uses the local `include64` and `source64` code to compare. Therefore, you should not change these JDE.INI settings:

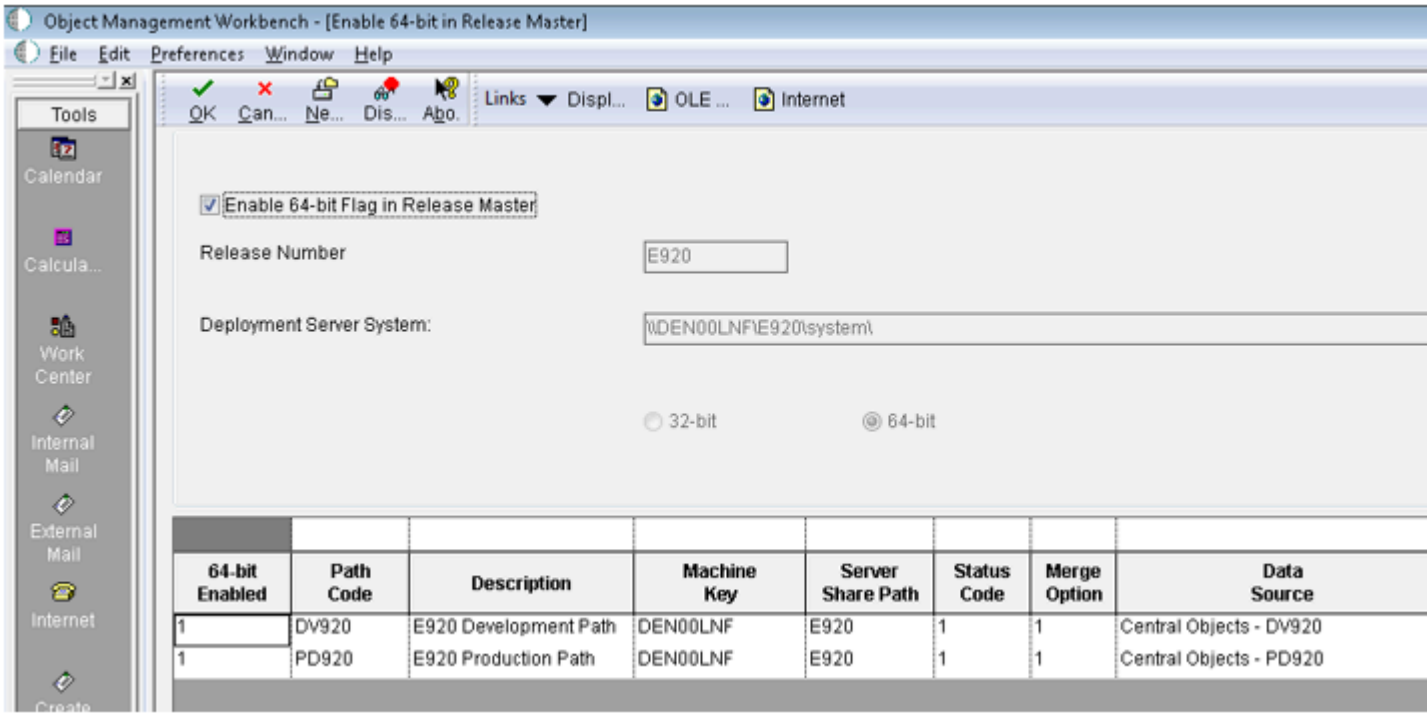
```
[SVR]

SourcePath=source

HeaderPath=include
```

Updating the Release Master

To update the Release Master:

1. Sign in to the Development Client on any standard environment.
2. Enter P00945 in the Fast Path to access the Work with Release Master form.
3. Select the release you want to update.

4. Select Enable 64-bit from the Row menu items.
5. Select the option Enable 64-bit flag in Release Master.



6. Click the OK button.
7. Click the OK button on the Confirm 64-bit Enabled dialog box.

Purging 32-bit Objects

When you have converted all pathcodes to 64-bit the previous 32-bit objects still exist in central objects, development clients, and the Deployment Server. It is advised to run Purge 32-bit Artifacts from Object Repository (R98780P) on each pathcode to remove the 32-bit objects.

The UBE will add each object to your default project, retrieve all artifacts from the repository table, repackage the artifacts into a PAR file to contain only the 64-bit versions and then remove the object from your default project. Only the last inserted record into the Repository and History tables will contain 64-bit only objects.

> **Note:** The Release Master must be set to 64-bit release before you run Purge 32-bit Artifacts from Object Repository (R98780P).

Purging 32-bit Objects

To remove 32-bit objects from a pathcode:

1. Log into the development client and the desired pathcode.
2. Launch Batch Versions.
3. Search for R98780P - Purge 32Bit Artifacts from Object Repository.
4. Run the current version.

5. Update the processing options to your preference.
6. Execute R98780P - Purge 32Bit Artifacts from Object Repository.