Oracle® Fusion Middleware

Installing Oracle JDeveloper

12*c* (12.1.3)

E48738-03

January 2015

This book describes how to install Oracle JDeveloper Studio (12.1.3) on various operating systems.

ORACLE®

Oracle Fusion Middleware Installing Oracle JDeveloper, 12*c* (12.1.3)

Primary Author: Hannah Cheng

Contributing Author: Peter Laquerre, Rekha M. Kamath, Brian Fry, Jacob Danner, Dominique Poudret

Contents

Preface

This document contains information about installing Oracle JDeveloper Studio.

Audience

This document is intended for application developers who develop Java applications. It is assumed that readers are familiar with Web technologies and have a general understanding of Windows and UNIX platforms.

Related Documents

For more information, see the following documents:

- *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware Guide*

- *Oracle Fusion Middleware System Requirements and Specifications Guide*

- *Oracle Fusion Middleware Installing and Configuring Oracle WebLogic Server and Coherence Guide*

- *Oracle Fusion Middleware Developing Fusion Web Applications with Oracle Application Development Framework*

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.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
italic	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 in examples, text that appears on the screen, or text that you enter.

1

Preparing to Install Oracle JDeveloper

This chapter helps you prepare for your Oracle JDeveloper installation.

This chapter contains the following sections:

- What's New in 12.1.3
- What is Oracle JDeveloper?
- Roadmap for Installing Oracle JDeveloper
- Verifying the Oracle JDeveloper Certification Information
- Verifying Your System Requirements for Installing Oracle JDeveloper
- Understanding and Obtaining the Oracle JDeveloper Distributions
- Deploying and Testing Applications Developed in Oracle JDeveloper

1.1 What's New in 12.1.3

This section describes a few changes that have taken place since the last release. If you are already familiar with JDeveloper, you should be aware of the following changes:

- This release of JDeveloper requires Java SE Development Kit (JDK) 7 Update 15 or later.

- You cannot download 12*c* (12.1.3) SOA IDE extensions for the JDeveloper distributions covered in this book. To develop SOA or Business Process Management applications with JDeveloper, you need to use Quick Start for SOA Suite or Business Process Management Suite.

  The Quick Start distributions allow you to install a version of JDeveloper that is already pre-configured for SOA Suite or Business Process Management Suite applications. You can also use the Quick Start versions of JDeveloper to develop Oracle Event Processing applications. See "Introducing the Quick Start Distributions" in *Installing SOA Suite and Business Process Management Suite Quick Start for Developers* to get started.

- JDeveloper used to store many configuration options in a file called `jdev.conf`. In 12*c* (12.1.3), `jdev.conf` has been renamed to `product.conf`. You should now edit `product.conf` for post-installation tasks, such as redefining your user home directory or JDK path.

- In past versions of this book, Mac OS X had separate installation sections because of minor configuration differences. This book now combines Mac OS X installation instructions with Linux and UNIX.

1.2 What is Oracle JDeveloper?

Oracle JDeveloper provides an extremely productive development environment that covers the full development life cycle and integrates features addressing the multitude of technologies used for building standard-based enterprise applications.

Following are some of the salient features of Oracle JDeveloper:

- Integrates development features for Java, SOA, Web 2.0, Database, XML and Web services into a single development tool.

- Covers the full development life cycle from initial design and analysis, through the coding and testing phases, all the way to deployment.

- Focuses on increasing developers productivity by offering a visual and declarative approach to application design that provide a simpler way to define the components that construct an application, thereby simplifying and eliminating tedious coding.

- Integrates the full set of features needed for developing composite applications with specific features that address technologies such as Java, Database, web page designing, Web Services and XML, Fusion Middleware, and others.

1.3 Roadmap for Installing Oracle JDeveloper

This guide provides all the steps required to install Oracle JDeveloper. Within the procedures, the guide also provides references to additional information you refer to. Table 1–1 shows the steps that you need to do to install Oracle JDeveloper.

Table 1–1   Roadmap for Installing Oracle JDeveloper

Task	Description	Documentation
Verify your system environment	Before beginning the installation, verify that the minimum system and network requirements are met.	See Section 1.4 and Section 1.5
Obtain the appropriate software distribution	To install Oracle JDeveloper on your system, obtain an Oracle JDeveloper distribution.	See Section 1.6
Install the software	Run the installation program to install the software. This transfers the software to your system and creates the Oracle home directory for the software that you have installed.	See Chapter 2
Customize your Oracle JDeveloper on Startup	To proceed to use Oracle JDeveloper, customize the settings on your system.	See Section 2.2.3

1.4 Verifying the Oracle JDeveloper Certification Information

For the most current information on software that is certified for use with Oracle JDeveloper, see the Oracle JDeveloper documentation information on the Oracle Technology Network (OTN):

http://www.oracle.com/technetwork/developer-tools/jdev/documentation/12130
0-cert-2164864.html

You can find other Oracle JDeveloper documentation listed on this page:
http://www.oracle.com/technetwork/developer-tools/jdev/documentation/index
.html

> **Note:**   This version of Oracle JDeveloper ships with Apache ANT
> 1.7.1.0.0. Higher versions of ANT might work with this release, but
> they are not supported, and might cause unexpected behavior.

1.5 Verifying Your System Requirements for Installing Oracle JDeveloper

This release of Oracle JDeveloper has been tested and certified on Windows, Linux,
and Mac OS X. It has not been certified for porting platform like Solaris, AIX, or HPI,
but Oracle provides support if you are running Oracle JDeveloper on any platform
with Java SE Development Kit (JDK) 7 Update 15 or later.

For information about recommended CPU, memory, display, and hard drive
configurations, see "Oracle JDeveloper Requirements" in *Oracle Fusion Middleware
System Requirements and Specifications*

For information about the certified JDK for your installation, see "Installing a JDK" in
Planning an Installation of Oracle Fusion Middleware

1.6 Understanding and Obtaining the Oracle JDeveloper Distributions

Oracle JDeveloper is available in two distinct distributions as described in Table 1–2.

Table 1–2   Oracle JDeveloper Distributions

Distribution	Description	Installer Types
Oracle JDeveloper Studio	Includes not only the Oracle JDeveloper Integrated Development Environment, but also the embedded Oracle Fusion Middleware Infrastructure software, which includes Oracle WebLogic Server and Oracle Application Development Framework for developing and testing Java and Oracle ADF applications.	The 12*c* (12.1.3) Oracle JDeveloper Studio installer is available in a generic form that requires you to install a certified JDK before you can run it. For information about JDK versions, see "Installing a JDK" in *Planning an Installation of Oracle Fusion Middleware*. Oracle JDeveloper Studio can also be installed with 64-bit platform-specific installers for Linux or Windows.
Oracle JDeveloper Java	A smaller version of Oracle JDeveloper in ZIP file format for developers who want to develop pure Java applications without the Oracle WebLogic Server or Oracle ADF. You do not need to run an installer to begin developing with this version.	Oracle JDeveloper Java is contained in a ZIP file. There is no executable installer for the Java distribution.

Download either an Oracle JDeveloper Studio installer or the Java distribution ZIP
from the Oracle Technology Network (OTN) web site:

http://www.oracle.com/technetwork/developer-tools/jdev/downloads/index.htm
l

1.7 Deploying and Testing Applications Developed in Oracle JDeveloper

If you use the Oracle JDeveloper Studio distribution, you can test your applications locally in an embedded version of Oracle WebLogic Server that contains the required libraries and Oracle ADF runtime software. For more information, see Section 3.5.

To deploy Java Enterprise Edition (Java EE) applications from Oracle JDeveloper, install and configure the Oracle WebLogic Server and Coherence distribution. For more information, see *Installing and Configuring Oracle WebLogic Server and Coherence*.

To deploy Oracle ADF applications from Oracle JDeveloper, install and configure the Oracle Fusion Middleware Infrastructure distribution. For more information, see *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

2

Installing Oracle JDeveloper

This chapter describes how to install the Oracle JDeveloper software. You will not be able to use the software described in this chapter to develop SOA or Business Process Management applications.

To install a version of JDeveloper pre-configured for SOA Suite or Business Process Management Suite applications, see "Introducing the Quick Start Distributions" in *Installing SOA Suite and Business Process Management Suite Quick Start for Developers*.

This chapter contains the following sections:

- Understanding the Oracle JDeveloper Installation Types
- Installing Oracle JDeveloper Studio on Various Platforms
- Verifying Your Installation
- Installing the JDeveloper Java Distribution

2.1 Understanding the Oracle JDeveloper Installation Types

> **Note:** The installation types described in this chapter do not support SOA extensions. To install a version of JDeveloper pre-configured for SOA Suite or Business Process Management Suite applications, see "Introducing the Quick Start Distributions" in *Installing SOA Suite and Business Process Management Suite Quick Start for Developers*.

You should have obtained the appropriate installer for your operating system as described in Section 1.6. Depending on what distribution and installer you are using, you should take note of the following JDK prerequisites.

If you are installing the Oracle JDeveloper Java distribution, all you need to do is unzip the files. This process is covered in Section 2.4.

If you are using a platform-specific installer for Oracle JDeveloper Studio, the JDK version specific to this Oracle JDeveloper installation is installed automatically.

If you are going to run the generic installer for Oracle JDeveloper Studio, ensure that the following prerequisites are met:

- You must have the required JDK installed on your system.

  Identify what JDK versions are certified for JDeveloper 12*c* (12.1.3) on your operating system using this site:
  http://www.oracle.com/technetwork/developer-tools/jdev/documentation/121300-cert-2164864.html#OperatingSystemsandJavaSE

You can then obtain a certified version of the JDK from the following location on OTN:

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

- To streamline your installation process, the JDK you wish to use with the product should be the one you use to start the installer program. In that case, be sure the desired JDK is the first one in your system *PATH*, or reference it explicitly on the command line.

2.2 Installing Oracle JDeveloper Studio on Various Platforms

This section describes how to install Oracle JDeveloper Studio on various platforms.

> **Warning:** The Oracle JDeveloper Studio installations described in this section cannot be used to develop SOA Suite or Business Process Management Suite applications. See Section 2.1 for details.

It contains the following topics:

- Starting the Installation Program
- Navigating the Installation Screens to Install Oracle JDeveloper
- Customizing Oracle JDeveloper on Startup

2.2.1 Starting the Installation Program

Depending on the type of installer you are using, you can launch the Oracle JDeveloper Studio installation program in one of the following ways:

- Launching the Generic Installer on Linux, UNIX, and Mac OS X
- Launching the Generic Installer on Windows
- Launching the Platform-specific Installer for Linux, UNIX, and Mac OS X
- Launching the Windows-specific Installer
- Launching the Installer in Silent Mode

Launching the Generic Installer on Linux, UNIX, and Mac OS X

To launch the platform-independent installer on your hosted Linux, UNIX, or Mac OS X machine, follow these steps:

1. Log into the host computer where you want to install Oracle JDeveloper.

   Be sure the user account you use for installing Oracle JDeveloper has the required privileges. For more information, see "Selecting an Installation User" in *Planning an Installation of Oracle Fusion Middleware*.

2. Ensure that a certified JDK exists on the system on which you are about to install Oracle JDeveloper. If you checked the prerequisites for generic installers as described in Section 2.1, you should have already obtained and installed a certified JDK on your system.

   You can double-check your JDK version by running `java -version` from the command line. This command will return the JDK version that you have installed on your machine. Make sure the outputted version is certified for 12*c* (12.1.3) JDeveloper by checking here:

```
http://www.oracle.com/technetwork/developer-tools/jdev/documentation/12
1300-cert-2164864.html#OperatingSystemsandJavaSE
```

3. Go to the directory where you downloaded the installation program.

4. Launch the installation program by invoking `java -jar` from the JDK directory on your system, as shown in the example below:

```
/home/Oracle/jdk7_15/jdk1.7.0_15/bin/java -jar jdev_suite_121300.jar
```

Be sure to replace `/home/Oracle/jdk7_15/jdk1.7.0_15` in this example with the location of the JDK on your system.

When the installation program appears, you are ready to begin the installation. See Section 2.2.2 for a description of each installation program screen.

Launching the Generic Installer on Windows

To launch the platform-independent installer on your Windows machine, do the following:

1. Log into the host computer where you want to install Oracle JDeveloper.

   Be sure the user account you use for installing Oracle JDeveloper has the required privileges. For more information, see "Selecting an Installation User" in *Planning an Installation of Oracle Fusion Middleware*.

2. Ensure that a certified JDK exists on the system on which you are about to install Oracle JDeveloper. If you checked the prerequisites for generic installers as described in Section 2.1, you should have already obtained and installed a certified JDK on your system.

   You can double-check your JDK version using one of the methods described here: `https://www.java.com/en/download/help/version_manual.xml`. Make sure your JDK version is certified for 12*c* (12.1.3) JDeveloper by checking here: `http://www.oracle.com/technetwork/developer-tools/jdev/documentation/12 1300-cert-2164864.html#OperatingSystemsandJavaSE`

3. Navigate to the directory where you downloaded the installation program.

4. Launch the installation program by invoking `java -jar` from the JDK directory on your system, as shown in the example below:

```
\home\Oracle\jdk7_15\jdk1.7.0_15\bin\java -jar jdev_suite_121300.jar
```

Be sure to replace `\home\Oracle\jdk7_15\jdk1.7.0_15` in this example with the location of the JDK on your system.

You can also navigate to the folder where you downloaded the Oracle JDeveloper installer and double-click on the icon to launch the installation wizard.

When the installation program appears, you are ready to begin the installation. See Section 2.2.2 for a description of each installation program screen.

Launching the Platform-specific Installer for Linux, UNIX, and Mac OS X

To launch the platform-specific installer on Linux, UNIX, or Mac OS X, follow these steps:

1. Log into the host computer where you want to install Oracle JDeveloper.

   Be sure the user account you use for installing Oracle JDeveloper has the required privileges. For more information, see "Selecting an Installation User" in *Planning an Installation of Oracle Fusion Middleware*.

2. Navigate to the location where the platform-specific installer is located.

3. Launch the installer by running the following command:

    ```
    ./jdev_suite_121300_linux64.bin
    ```

You can also navigate to the folder where you downloaded the Oracle JDeveloper installer and double-click on the icon to launch the installation wizard.

When the installation program appears, you are ready to begin the installation. See Section 2.2.2 for a description of each installation program screen.

Launching the Windows-specific Installer

To launch the platform-specific installer on Windows, follow these steps:

1. Log into the host computer where you want to install Oracle JDeveloper.

    Be sure the user account you use for installing Oracle JDeveloper has the required privileges. For more information, see "Selecting an Installation User" in *Planning an Installation of Oracle Fusion Middleware*.

2. Navigate to the location where the platform-specific installer is located.

3. Launch the installer. You launch it from the command line by running the following command:

    ```
    jdev_suite_121300_win64.exe
    ```

You can also navigate to the folder where you downloaded the Oracle JDeveloper installer and double-click on the icon to launch the installation wizard.

When the installation program appears, you are ready to begin the installation. See Section 2.2.2 for a description of each installation program screen.

Launching the Installer in Silent Mode

You can also launch the installer in silent mode to ensure that no configuration options are displayed during the installation process. For more information, see Section A.1.

2.2.2 Navigating the Installation Screens to Install Oracle JDeveloper

The installation program for the Oracle JDeveloper Studio edition displays a series of screens, in the order listed in Table 2–1.

If you need additional help with any of the installation screens, click the screen name.

Table 2–1   Oracle JDeveloper Install Screens

Screen	Description
Installation Inventory Setup	This screen appears if this is the first time you are installing any Oracle product on this host. Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location.
	For more information about the central inventory, see "Understanding the Oracle Central Inventory" in *Installing Software with the Oracle Universal Installer*.
Welcome	This screen introduces you to the product installer.

Table 2–1   (Cont.) Oracle JDeveloper Install Screens

Screen	Description
Installation Location	Use this screen to specify the location of your Oracle home directory. Ensure that the name of the directory that you choose does not contain spaces.
	For more information about Oracle Fusion Middleware directory structure, see "Selecting Directories for Installation and Configuration" in *Planning an Installation of Oracle Fusion Middleware*.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements.
	If there are any warning or error messages, refer to Section 1.5.
Installation Summary	Use this screen to verify the installation options you selected. If you want to save these options to a response file, click **Save** and provide the location and name of the response file. Response files can be used later in a silent installation situation.
	For more information about silent or command line installation, see "Using the Oracle Universal Installer in Silent Mode" in *Installing Software with the Oracle Universal Installer*.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Completed	This screen appears when the installation is complete. Review the information on this screen.
	You can use this screen to customize Oracle JDevleoper. For more information about customizing Oracle JDeveloper using the Installation Completed screen, see Section 2.2.3.

For more information about the installation screens that appear while installing Oracle Jdeveloper, see "Oracle Fusion Middleware Infrastructure Installation Screens" in *Installing with Oracle Universal Installer*.

2.2.3 Customizing Oracle JDeveloper on Startup

On completing the Oracle JDeveloper Studio installation, the installer displays a list of options that you can use to proceed with the product that you have installed. You can customize Oracle JDeveloper using the options listed on the Installation Complete screen of the installer. See Figure 2–1.

Figure 2–1   Options to Customize Oracle JDeveloper Installer
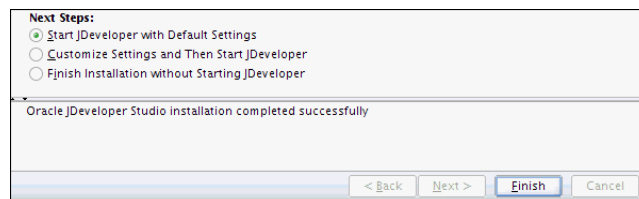
To customize Oracle JDeveloper using the installation wizard, select one of the options listed in the Next Steps section of the Installation Complete screen. Table 2–2 describes the options that you see on the Installation Complete screen of the wizard:

Table 2–2   Options to Customize Oracle JDeveloper on Startup

Option	Description and Action
Start JDeveloper with Default Settings	Select this option if you want to start Oracle JDeveloper using the default settings. All your Oracle JDeveloper projects will be saved in this default location.
Customize Settings and Then Start JDeveloper	Select this option to do the following: ■ Change the location where Oracle JDeveloper projects are saved ■ Change the location where Oracle JDeveloper user preferences are saved ■ Identify the file types to be associated with Oracle JDeveloper (Windows only)
Finish Installation without Starting JDeveloper	If the box to launch JDeveloper after quitting the installation wizard is checked, uncheck it. Press **Finish** to quit the installation wizard without starting Oracle JDeveloper.

Note:

■ During installation, a file called `product.conf` is created in your home directory by default. This will contain your JDeveloper preferences. Old preferences will carry over to new installations of JDeveloper unless you delete this file.

■ On Windows systems, if you are deinstalling your current version of Oracle JDeveloper to install a newer version, Oracle recommends that you reboot your system after you deinstall the software and before you begin a new installation of the software.

2.3 Verifying Your Installation

To verify that Oracle JDeveloper Studio installed successfully, you can compare your directory structure to the one described in Table 2–3.

Table 2–3 describes the directory structure that will be created on your system at the end of your Oracle JDeveloper installation.

Table 2–3   Directory Description for Oracle JDeveloper Home

Directory or File	Description
JDeveloper Home	This directory is the Oracle home that contains the binary files and configuration files that are required to run Oracle JDeveloper.
coherence	This directory contains the Coherence product files.
em	This directory contains files used by Oracle Enterprise Manager Fusion Middleware Control.
inventory	This directory contains information about the components, feature sets, and patches installed in this Oracle JDeveloper home directory.

Table 2–3 (Cont.) Directory Description for Oracle JDeveloper Home

Directory or File	Description
registry.xml	This file is located in the `inventory` directory. It contains the location and versions of all components currently registered with this Oracle Jdeveloper installation. Whenever you add a new plug-in or add-on, the information gets updated in this file.
jdeveloper	This directory contains the files that are needed to run the Oracle JDeveloper application. It also contains files that can be used to modify the application settings.
OPatch	This directory contains OPatch and supported files. OPatch is a tool used to patch Oracle Fusion Middleware software.
oracle_common	This directory contains the binary and library files required for Oracle Enterprise Manager Fusion Middleware Control and Java Required Files (JRF).
oraInst.loc	You may not see this file on an Windows installation. This file contains information about the inventory location of the installation.
oui	This directory contains files used by the Oracle Universal Installer, including the deinstaller program. If you need to run the Oracle Universal Installer again after the product is installed, you should do so from this directory.
wlserver	This directory contains the WebLogic Server product files.

For Windows distributions with the MAF extension installed, you may also see a folder called `cfgtoollogs`. This directory contains the log files for your installation and configuration transactions.

2.4 Installing the JDeveloper Java Distribution

> **Warning:** The Oracle JDeveloper Studio installations described in this section cannot be used to develop SOA Suite or Business Process Management Suite applications. See Section 2.1 for details.

JDeveloper Java does not require an installer. To install the Oracle JDeveloper Java distribution, all you need is an unzip tool. You can download a free, cross-platform unzip tool, Info-Zip, available at: http://www.info-zip.org/.

> **Caution:** Ensure that the directory that you install Oracle JDeveloper in does not contain spaces. For example, do not use `C:\Program Files` as the installation directory.

To install JDeveloper from `jdev-java-121300.zip`:

1. Obtain the required JDK.

   Identify what JDK versions are certified for 12*c* (12.1.3) JDeveloper on your operating system using this site:
   http://www.oracle.com/technetwork/developer-tools/jdev/documentation/121300-cert-2164864.html#OperatingSystemsandJavaSE

   You can then obtain a certified version of JDK from the following location on OTN:
   http://www.oracle.com/technetwork/java/javase/downloads/index.html

2. Unzip `jdev-java-121300.zip` to the target directory.

> **Note:** In this document, in the context of the Java Edition, *JDEV_HOME* is used to represent the directory of the installation. For example, if you unzipped `jdev-java-121300.zip` into `C:\jdev_install\`, then *JDEV_HOME* refers to `C:\jdev_install`.

2.4.1 Specifying the JDK Location for JDeveloper Java

When you start the Java edition of Oracle JDeveloper for the first time, the `jdev` script prompts you to provide the location of your JDK installation if it cannot locate it. When prompted, enter the path to the Java executable file.

For Windows, your path definition may look like this: `D:\jdk1.7.0_15\bin`

For Linux, UNIX, and Mac OS X, your path may look like this: `/home/jdk1.7.0_15/bin`.

3

Next Steps After Installing Oracle JDeveloper Studio

This chapter describes the steps that you need to follow after installing Oracle JDeveloper.

This chapter contains the following sections:

- Preparing to Launch Oracle JDeveloper
- Launching Oracle JDeveloper
- Migrating to Oracle JDeveloper 12.1.3 From a Previous Version
- Enabling Oracle JDeveloper Extensions
- Using Oracle WebLogic Server with Oracle JDeveloper
- Understanding Oracle JDeveloper Accessibility Information
- Oracle on the Web

3.1 Preparing to Launch Oracle JDeveloper

This section describes tasks that you may want to perform after installing Oracle JDeveloper. This section describes the tasks for Linux, Windows, and multiuser environments.

The section is divided into the following topics:

- Optimizing Oracle JDeveloper on Linux, UNIX, and Mac OS X
- Optimizing Oracle JDeveloper on Windows
- Optimizing Oracle JDeveloper in a Multiuser Environment

3.1.1 Optimizing Oracle JDeveloper on Linux, UNIX, and Mac OS X

This section describes tasks that you may want to perform after installing Oracle JDeveloper on Linux, UNIX, or Mac OS X.

It contains the following topics:

- Setting the System Resource Limit
- Setting the User Home Directory on Linux, UNIX, and Mac OS X
- Changing the JDK Location in Linux, UNIX, and Mac OS X
- Changing System Cursors on Linux, UNIX, and Mac OS X

3.1.1.1 Setting the System Resource Limit

The minimum recommended system resource limit for Oracle JDeveloper on Linux systems is 4096.

To determine the resource limit configuration for your system, enter the following command:

```
/bin/sh -c 'ulimit -n'
```

If the value returned is less than `4096`, set the system resource limit by following these steps:

1. Open the `limits.conf` file, which is located at `/etc/security/`.

2. Find the following parameters:

   ```
   soft nofile value_of_the_parameter
   hard nofile value_of_the_parameter
   ```

3. Change the value of these parameters to `4096`:

   ```
   soft nofile 4096
   hard nofile 4096
   ```

> **Note:** If the parameters listed in Step 2 do not exist, add the parameters with their values as listed in Step 3 to the `limits.conf` file.

3.1.1.2 Setting the User Home Directory on Linux, UNIX, and Mac OS X

This section describes how to permanently redefine the location of your user home directory using the home environment variable in a Linux, UNIX, or Mac OS X environment.

When you define the user home directory, it will contain a `system` subdirectory that stores the user's preferences for JDeveloper, also known as the domain home. The user home directory will also contain a separate subdirectory for user-generated content and other configuration files that are specific to a given user, also known as the application home.

If you do not define a user home directory, these subdirectories will be located in different parts of your computer.

- The default location for the `system` subdirectory is `$HOME/.jdeveloper/system12.1.3.XX.XX.XX`, where `XX.XX.XX` is the unique number of the product build.

- The default location for user-generated content is `$HOME/jdeveloper/mywork`.

There are two ways to set your user home directory if you do not want your JDeveloper files to be stored at `$HOME`. Both methods will set the user home directory for all instances of JDeveloper on your system.

You only need to use one of the following methods to set the user home directory:

- Editing `product.conf`

- Setting environment variable `JDEV_USER_DIR`, which will require you to restart your terminal session

Editing `product.conf`

Starting in 12*c* (12.1.3), many JDeveloper settings, including the location of your JDK, are stored in `product.conf`. This file is created by JDeveloper on first startup unless

the file already exists from a previous installation. JDeveloper will use the settings stored in `product.conf` even if they are from a previous installation.

> **Note:** In previous versions of JDeveloper, `product.conf` was named `jdev.conf`.

1. Find the `product.conf` file. It should be located at *$HOME*`/.jdeveloper/12.1.3.0.0`.

2. Open `product.conf` in an editor. Add a line to set `AddVMOption -Dide.user.dir` to your preferred directory path.

   For example, if your preferred directory is `$HOME/mydocs/jdevfiles`, your definition should look like this:

   `AddVMOption -Dide.user.dir=$HOME/mydocs/jdevfiles`

   > **Caution:** Do not set the home environment variable to a directory that contains spaces. For example, do not specify `$HOME/my projects` as the home directory.

3. Save your changes. The changes should take effect immediately when you start JDeveloper. The changes made to `product.conf` will also override any environment variable you have defined.

   When you launch Oracle JDeveloper for the first time, you can verify that the user home directory has been set to your preferred directory by accessing the **Help** menu, selecting **About** from the drop-down menu, toggling the **Properties** tab, and finding the definition for `ide.user.dir`.

Setting environment variable `JDEV_USER_DIR`

This is another way of setting the user home directory. You can set the environment variable `JDEV_USER_DIR` to your preferred directory path. The examples and syntax provided below are for the bash shell on Linux systems.

1. Open your startup configuration file with an editor. For the bash shell, this file is named `.bashrc`. The configuration file should be located in your home directory. If the appropriate file for your shell does not exist at this location, create a new file.

2. Set the environment variable `JDEV_USER_DIR` to your preferred directory.

   For example, if your preferred directory is `$HOME/mydocs/jdevfiles`, your definition may look like this:

   `export JDEV_USER_DIR=$HOME/mydocs/jdevfiles`

   > **Caution:** Do not set the home environment variable to a directory that contains spaces. For example, do not specify `$HOME/my project folder` as the home directory.

3. Save your file and exit the editor. Start a new terminal session.

4. In the command line, use the following command to display the environment variable to confirm the change:

```
echo $JDEV_USER_DIR
```

This should output the directory you specified. For this example, the command will output the following:

```
$HOME/mydocs/jdevfiles
```

When you launch Oracle JDeveloper for the first time, you can verify that the user home directory has been set to your preferred directory by accessing the **Help** menu, selecting **About** from the drop-down menu, toggling the **Properties** tab, and finding the definition for ide.user.dir.

3.1.1.3 Changing the JDK Location in Linux, UNIX, and Mac OS X

This section describes how to permanently change the location of your JDK if it has changed since the first time you launched JDeveloper.

To change a JDK location, do the following steps:

1. Find the product.conf file. It should be located at
   $HOME/.jdeveloper/12.1.3.0.0.

2. Open product.conf in an editor. Find the line that reads like the following:

   ```
   SetJavaHome /path/jdk
   ```
   Uncomment this line. Replace /path/jdk with the path to your JDK directory.

   For example, if the location of your JDK is /usr/local/java, your definition will look like this:

   ```
   SetJavaHome /usr/local/java
   ```

This will set the JDK path for all installations of JDeveloper on your system.

If you deinstall JDeveloper but do not delete the product.conf file, all settings stored in this file, including the JDK path, will be preserved for future installations. If you re-install JDeveloper with the same numerical version as your previous installation, your new installation will automatically read that JDK location again.

3.1.1.4 Changing System Cursors on Linux, UNIX, and Mac OS X

On Linux platforms, the Java cursors might display as large and opaque, creating difficulties when used for drag and drop operations. To address this problem, Oracle JDeveloper provides a set of cursors to replace the default set. You must have write-access to the JDK to replace the cursors.

To replace the cursors:

1. Create a backup copy of the default cursors located in the JDK directory at:

   jdk_install/jre/lib/images/cursors

2. Extract the replacement cursors from the .tar file by following these steps:

   a. Navigate to the following location on your system:

      JDEV_HOME/jdeveloper/jdev/bin/clear_cursors.tar

   b. Run the following command to extract the replacement cursors from the tar file:

      ```
      tar -xvf clear_cursors.tar
      ```

3.1.2 Optimizing Oracle JDeveloper on Windows

This section describes tasks that you may want to perform after installing Oracle JDeveloper.

It contains the following topics:

- Setting the User Home Directory on Windows
- Changing the JDK location on Windows

3.1.2.1 Setting the User Home Directory on Windows

This section describes how to permanently redefine the location of your user home directory using the home environment variable in a Windows environment.

When you define a user home directory for JDeveloper, it will contain a `system` subdirectory that stores the user's preferences for JDeveloper. It will also contain a separate subdirectory for user-generated content and other configuration files that are specific to a given user.

If you do not define a user home directory, these subdirectories will be located in different parts of your computer

- The default location for `the system` subdirectory, or domain home, is:
  - `%USERPROFILE%\AppData\Roaming\JDeveloper\system12.1.3.XX.XX.XX` on Windows 7 systems. `XX.XX.XX` is the unique number of the product build.
  - `%USERPROFILE%\Application Data\JDeveloper\system12.1.3.XX.XX.XX` on all other Windows platforms. `XX.XX.XX` is the unique number of the product build.

- The default location of the user-generated content, or application home, is:
  - `C:\JDeveloper\mywork` on Windows 7 systems.
  - `C:\Documents and Settings\My Documents\JDeveloper\mywork` on all other Windows platforms.

There are two ways to set your user home directory if you do not want your JDeveloper files to be stored at the default location. Both methods will set the user home directory for all instances of JDeveloper on your system.

You only need to use one of the following methods to set the user home directory:

- Editing `product.conf`
- Setting the environment variable `JDEV_USER_DIR`

Editing `product.conf`

Starting in 12c (12.1.3), many JDeveloper settings, including the location of your JDK, are stored in `product.conf`. This file is created by JDeveloper on first startup unless the file already exists from a previous installation. JDeveloper will use the settings stored in `product.conf` even if they are from a previous installation.

> **Note:** In previous versions of JDeveloper, `product.conf` was named `jdev.conf`.

1. Find the `product.conf` file. It should be located at
   `%USERPROFILE%\AppData\Roaming\JDeveloper\12.1.3`.

2. Open `product.conf` in an editor. Add a line to set `AddVMOption -Dide.user.dir` to your preferred directory path.

   For example, if your preferred directory is `N:\users\jdoe`, your definition should look like this:

   ```
   AddVMOption -Dide.user.dir=N:\users\jdoe
   ```

   > **Caution:** Do not set the home environment variable to a directory that contains spaces. For example, do not specify `C:\My Projects` as the home directory.

3. Save your changes. The changes should take effect immediately when you start JDeveloper. The changes made to `product.conf` will also override any environment variable you have defined.

   When you launch Oracle JDeveloper for the first time, you can verify that the user home directory has been set to your preferred directory by accessing the **Help** menu, selecting **About** from the drop-down menu, toggling the **Properties** tab, and finding the definition for `ide.user.dir`.

Setting the environment variable `JDEV_USER_DIR`

This is another way of setting the user home directory. To set the home environment variable on a Windows systems, including individual users of Oracle JDeveloper on a multiuser system, you need to edit the value of `JDEV_USER_DIR`.

Do the following:

1. From the Windows **Start** menu, select **Control Panel**, and then select **System**.

2. Click **Advanced system settings**, then click **Environment Variables**.

3. In the **User Variables** section, click **New**.

4. Add `JDEV_USER_DIR` as a user variable.

5. Set the value of this variable to your home directory (for example, `N:\users\jdoe`), and click **OK**.

   > **Caution:** Do not set the home environment variable to a directory that contains spaces. For example, do not specify `C:\My Projects` as the home directory.

6. To check your variable setting, open a command shell and enter the following command:

   ```
   set
   ```

   This will list all of your defined variables. Find and verify that your newly defined variable appears similar to the following:

   ```
   JDEV_USER_DIR=N:\users\jdoe
   ```

When you launch Oracle JDeveloper for the first time, you can verify that the user home directory has been set to your preferred directory by accessing the **Help** menu, selecting **About** from the drop-down menu, toggling the **Properties** tab, and finding the definition for `ide.user.dir`.

3.1.2.2 Changing the JDK location on Windows

This section describes how to permanently change the location of your JDK if it has changed since the first time you launched JDeveloper.

Starting in 12*c* (12.1.3), most JDeveloper settings, including the location of your JDK, are stored in `product.conf`. This file is created by JDeveloper on first startup unless the file already exists from a previous installation. Your new installation of JDeveloper will use the settings stored in `product.conf` even if they are from a previous installation.

> **Note:** In previous versions of JDeveloper, `product.conf` was named `jdev.conf`.

To change a JDK location, do the following steps:

1. Find the `product.conf` file. It should be located at `%USERPROFILE%\AppData\Roaming\JDeveloper\12.1.3.0.0`.

2. Open `product.conf` in an editor. Find the line that reads like the following:

    ```
    SetJavaHome \path\jdk
    ```
    Uncomment this line. Replace `\path\jdk` with the path to your JDK directory.

    For example, if the location of your JDK is `jdk1.7.0_15` on your `D:` drive, your definition will look like this:

    ```
    SetJavaHome D:\jdk1.7.0_15
    ```

If you deinstall JDeveloper but do not delete the `product.conf` file, all settings stored in this file, including the JDK path, will be preserved. If you re-install JDeveloper with the same numerical version as your previous installation, your new installation will automatically read that JDK location again.

3.1.3 Optimizing Oracle JDeveloper in a Multiuser Environment

This section contains information about how you can install JDeveloper in Microsoft Terminal Server, Citrix MetaFrame and MetaFrame XP (for Windows), and MetaFrame 1.1 for UNIX environments. These environments allow many clients to access one installation of Oracle JDeveloper. In all cases, users can save their projects locally.

When installing and configuring Oracle JDeveloper for a multiuser environment, you will need to account for resource planning, such as number of users and power of the server to deliver optimal performance for JDeveloper and your users.

This section contains the following sections:

- Installing Oracle JDeveloper on a Citrix MetaFrame Server or a Microsoft Terminal Server

- Configuring Terminal Server Clients for Running Oracle JDeveloper

- Troubleshooting system DLL ole32.dll Memory Error

3.1.3.1 Installing Oracle JDeveloper on a Citrix MetaFrame Server or a Microsoft Terminal Server

You need to have administrative privileges to install Oracle JDeveloper on a Citrix Metaframe Server or a Microsoft Terminal Server.

Run the Oracle JDeveloper installer on your Citrix MetaFrame or Microsoft Terminal Server as detailed in Chapter 2. Do not launch Oracle JDeveloper at the end of installation, as you should configure the user home directories and terminal server clients as described in the following sections.

3.1.3.2 Configuring Terminal Server Clients for Running Oracle JDeveloper

This section assumes that you have already installed a Citrix MetaFrame or Microsoft Terminal Server client locally and that JDeveloper has been installed and configured by the system administrator.

To configure a terminal server client for running Oracle JDeveloper, do the following:

1. Verify that the color resolution of the terminal server client has been set to a minimum of 256 colors. This minimum resolution is a prerequisite for installing Java JDK 7.0.

2. Log in to your terminal server.

3. Verify that the user home environment variable has been defined. Confirm the naming convention that is used on your system. The default variable is `JDEV_USER_DIR`.

4. Launch Oracle JDeveloper.

5. Oracle JDeveloper prompts you to confirm if the user home directory needs to be created. Click **Yes**.

6. Verify that the user home directory has been set to your preferred directory by accessing the **Help** menu, selecting **About** from the drop-down menu, toggling the **Properties** tab, and finding the definition for `ide.user.dir`.

3.1.3.3 Troubleshooting system DLL ole32.dll Memory Error

If you run Oracle JDeveloper in a multiuser environment, you might encounter the following error:

```
The system DLL ole32.dll was relocated in memory. The application will not
run properly. The relocation occurred because the DLL Dynamically
Allocated Memory occupied an address range reserved for Windows NT system
DLL's. The vendor supplying the DLL should be contacted for a new DLL.
```

If you see this error, update the `HOME/.jdeveloper/12.1.3.0.0/product.conf` file by uncommenting the following parameter:

```
AddVMOption -Xheapbase100000000
```

In addition, each user must modify the default project to apply this setting. To specify this value in the default project settings, follow these steps:

1. From the **Application** menu, select **Default Project Properties**.

2. In the Default Project Properties dialog, click **Run/Debug/Profile**, and then click **Edit**.

3. Click the **Launch Settings** node.

4. On the Launch Settings page, enter `-Xheapbase100000000` in the **Java Options** field.

3.2 Launching Oracle JDeveloper

To start Oracle JDeveloper, do the following:

On Linux, UNIX, and Mac OS X:

1. Navigate to the following location on your system:

   JDEV_HOME/jdeveloper/jdev/bin/

2. Run the following command:

   ./jdev

On Windows:

From your Start Menu, select **All Programs**, then select **Oracle Fusion Middleware 12.1.3**, and then select **JDeveloper Studio 12.1.3**.

You can also start Oracle JDeveloper from the command line by running one of the following commands:

- *JDEV_HOME*\jdeveloper\jdeveloper.exe

- *JDEV_HOME*\jdeveloper\jdev\bin\jdevw.exe

- *JDEV_HOME*\jdeveloper\jdev\bin\jdev.exe (to display a console window for internal diagnostic information)

3.3 Migrating to Oracle JDeveloper 12.1.3 From a Previous Version

Oracle JDeveloper 12*c* (12.1.3) supports migration from Oracle JDeveloper 11.1.1.7.0 or 11.1.2.4.0. Oracle recommends migrating to Oracle JDeveloper 11.1.1.7.0 or 11.1.2.4.0 from all other earlier versions before migrating to this release of Oracle JDeveloper.

This section provides more information about migrating from a previous version to Oracle JDeveloper 12*c* (12.1.3). It contains the following topics:

- Migrating User Settings

- Migrating Projects

3.3.1 Migrating User Settings

When you start Oracle JDeveloper for the first time (and each time you add a new extension or upgrade to a newer version), JDeveloper displays the **Confirm Import Preferences** dialog to confirm whether to import your preferences and settings from a previous installation.

When the **Confirm Import Preferences** dialog is displayed, click **Show All Installations** to view a list of all the previous installations. From this list, choose the installation that you want to import preferences and settings from. When you hover the mouse over an item in list, the path to the installation is displayed as a tooltip.

Alternatively, you can click **Find a previous installation manually** (with the image of a magnifying glass) on the top right of the installation list to browse for an installation manually. Clicking **Yes** on the Confirm Import Preferences dialog imports user preferences and the state of the IDE from the previous installation.

To force Oracle JDeveloper to display the **Confirm Import Preferences** dialog upon subsequent startups, use the -migrate flag when starting Oracle JDeveloper from the command line, for example, jdev -migrate.

> **Note:** If you migrate to 12*c* (12.1.3)from another version of Oracle JDeveloper, you will need to reinstall the extensions that you want to use. Some extensions for older versions may no longer exist.
>
> For more information on installing extensions, see Section 3.4, "Enabling Oracle JDeveloper Extensions".

3.3.2 Migrating Projects

When you open an application or project that was created in a previous release, Oracle JDeveloper prompts you to migrate the project to Oracle JDeveloper 12*c*. Depending on the content of the projects, Oracle JDeveloper might display additional prompts to migrate some specific source files as well.

Oracle recommends that you create a backup copy of your projects before migrating them. Accepting the initial prompts to migrate your projects means that JDeveloper will update the format for crucial XML files, but migration does not rewrite or update specific project code. After migration, you should re-test your applications to completely verify that they work.

See the Oracle JDeveloper page on OTN for more information about migrating specific types of projects to 12*c*.

3.3.2.1 Migrating JSF and JSTL Projects

This version of JDeveloper requires that all projects with JavaServer Faces and JSTL be migrated to the latest versions of the technologies (version 2.0 and 1.2 respectively). Additionally, any `web.xml` deployment descriptors will be migrated to version 2.5.

For the most current information on migration, see the Oracle JDeveloper documentation page on OTN:
http://www.oracle.com/technetwork/developer-tools/jdev/documentation/1212-cert-1964670.html#Abrams-SupportInformation-MigrationSupport.

3.4 Enabling Oracle JDeveloper Extensions

Before you can use an extension in Oracle JDeveloper, you first need to download the extension from the Oracle Technology Network (OTN). An example of an available Oracle JDeveloper Extension is JUnit.

You can find and install extensions from inside Oracle JDeveloper. To do so, follow these steps:

1. Launch JDeveloper.

2. Click on **Help** in the ribbon menu. Choose **Check for Updates** from the drop-down menu. This will launch an Update wizard.

3. The first screen of the Update wizard allows you to select sources to check for extensions. **Oracle Fusion Middleware Products** and **Official Oracle Extensions and Updates** are selected by default. You can check the other two boxes if you want more options.

> **Note:** Starting in 12*c* (12.1.3), you can no longer download SOA extensions. You need to use a Quick Start distribution to obtain a version of JDeveloper pre-configured for either SOA Suite or Business Process Management Suite. See "Introducing the Quick Start Distributions" in *Installing SOA Suite and Business Process Management Suite Quick Start for Developers* for more information.

Press **Next** when you have identified and selected your sources.

4. The **Updates** screen allows you to select specific extensions, such as JUnit. Check the extensions that you want to install and press **Next**.

5. On the **License Agreements** screen, you need to agree to the license terms for the extensions you selected on the previous screen. Press **Next** to begin the download.

6. When the extensions have finished downloading, the wizard will display a summary of the installations or upgrades that you installed. Click **Finish** to exit the wizard.

7. You will usually need to restart JDeveloper to finish installing updates. A dialogue box will pop up after you exit the wizard that gives you this option. If you have unsaved work, press **No** before restarting JDeveloper.

Alternatively, you can manually download and install an Oracle JDeveloper extension from OTN. To do so, follow these steps:

1. Go to the following link:
   http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/1311
   67.xml.

2. Select an Oracle JDeveloper Extension. Make sure that you are selecting extensions for 12*c* (12.1.3), as there may be different versions of each extension for different versions of JDeveloper.

3. Follow the instructions to download the zip file.

4. Verify if there are additional installation instructions in the extension archive.

5. From the **Help** menu, select **Check for Updates**.

6. In Step 1 of the wizard, select **Install from a Local File** and navigate to the ZIP file on your system.

7. Finish the wizard and restart Oracle JDeveloper. After you restart Oracle JDeveloper, you will be able to use the extension.

3.5 Using Oracle WebLogic Server with Oracle JDeveloper

> **Note:** This section is not applicable to the Java edition of Oracle JDeveloper.

Installing Oracle JDeveloper Studio 12*c* (12.1.3) also automatically installs Oracle WebLogic Server (12.1.2). Oracle JDeveloper uses this preconfigured installation as the Integrated Oracle WebLogic Server and JDeveloper managed server for testing and debugging your applications from within the IDE. After installing Oracle JDeveloper, all the applications that you need to begin developing, testing and debugging are installed and configured.

For additional information about using a standalone Oracle WebLogic Server instance with JDeveloper, see the Deploying Applications chapter in the *Oracle Fusion Middleware User's Guide for Oracle JDeveloper*.

3.5.1 Using the Integrated Oracle WebLogic Server

Oracle JDeveloper is bundled with an integrated application server called Integrated WebLogic Server, and a default instance called `IntegratedWebLogicServer` is defined for it.

All applications are bound, by default, to `IntegratedWebLogicServer`. Oracle JDeveloper manages the Integrated WebLogic Server lifecycle for testing your application. The first time Integrated WebLogic Server is needed, Oracle JDeveloper creates the `DefaultDomain` and prompts you to provide the administrative username and password. The location of configuration files for the default domain is the `DefaultDomain` directory located in the Oracle JDeveloper system directory.

> **Note:** The Oracle WebLogic Server domain that is created for you during installation, `DefaultDomain`, is not intended for use outside of the IDE. To deploy ADF applications to a standalone Oracle WebLogic Server, the server must be configured to run ADF applications.
>
> For more information, see "Preparing the Standalone Application Server for Deployment" in *Administering Oracle ADF Applications*.

The `IntegratedWebLogicServer`'s `DefaultDomain` uses Java DB. If the `IntegratedWebLogicServer` fails to create the `DefaultDomain`, you should search for any preexisting instances of Java DB or Derby Client running in the background and stop them.

3.6 Understanding Oracle JDeveloper Accessibility Information

This section contains the following topics:

- Using a Screen Reader and Java Access Bridge with Oracle JDeveloper
- Finding Accessibility Information

3.6.1 Using a Screen Reader and Java Access Bridge with Oracle JDeveloper

To make the best use of our accessibility features, Oracle Corporation recommends the following minimum configuration:

- Windows XP, Windows Vista
- Java J2SE 1.7.0_15
- Java Access Bridge 2.0.1
- JAWS 12.0.522
- Microsoft Internet Explorer 7.0 or higher
- Mozilla Firefox 3.5 or higher

To set up a screen reader and Java Access Bridge, follow these steps:

> **Note:**
>
> - These steps apply to machines that have the Windows operating system.
>
> - You must use a screen reader that is compatible with Windows.

1. Install the screen reader if it is not already installed.

   For more information about installation, refer to the documentation for your screen reader.

2. Install Oracle JDeveloper.

   For information about performing a silent installation, see Section A.1, "Installing Oracle JDeveloper Studio in Silent Mode".

3. Download Java Access Bridge for Windows version 2.0.1. The latest version of the file is available at the following location: http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136191.html.

   For more information about Java Access Bridge, refer to the Java Access Bridge documentation available on the web site.

4. Extract the contents of the zip file to a folder, `accessbridge_home`.

5. Install Java Access Bridge by running the `install.exe` file from the `accessbridge_home\installer` folder.

   The installer first checks the JDK version for compatibility. Then, the Available Java virtual machines dialog displays.

6. Click **Search Disks**. Then customize your search for only the drive that contain the Oracle JDeveloper build and the JDK version in the program files directory (if it exists).

   The search process can take a long time if the disk that has many instances of JDK or Oracle JDeveloper, or when searching multiple disks. However, unless you complete an exhaustive search of your disk, Access Bridge will not be configured optimally, and will not be correctly installed to all of the Java virtual machines on your system. After selecting the disk to search, click **Search**.

7. Confirm that you want to install the Java Access Bridge into each of the Java virtual machines displayed in the dialog, by clicking **Install in All**.

8. Click **OK** when you see the Installation Completed message.

9. Confirm that the following files have been installed in the `Winnt\System32` directory (or the equivalent Windows XP or Vista directory), or copy them from `accessbridge_home\installerfiles`, as they must be in the system path to work with Oracle JDeveloper:

   ```
   JavaAccessBridge.dll
   JAWTAccessBridge.dll
   WindowsAccessBridge.dll
   ```

   Note that the system directory is required in the `PATH` system variable.

10. Confirm that the following files have been installed in the `JDEV_HOME\jdk\jre\lib\ext` directory, or copy them from `accessbridge_home\installerfiles`:

```
access-bridge.jar
jaccess-1_4.jar
```

11. Confirm that the file `accessibility.properties` has been installed in the *jdev_home*`\jdk\jre\lib` directory, or copy it from `\installerfiles`.

12. Start your screen reader.

13. Start Oracle JDeveloper by running the file `jdev.exe` located in the folder *JDEV_HOME*`\jdeveloper\jdev\bin`.

A console window that contains error information (if any) is displayed first . The Oracle JDeveloper window appears when Oracle JDeveloper starts up. Any error or warning messages that appear do not affect the functionality of Oracle JDeveloper.

3.6.2 Finding Accessibility Information

For the latest configuration information or for information about addressing accessibility and assistive technology issues, see the Oracle Accessibility FAQ at http://www.oracle.com/us/corporate/accessibility/faqs/index.htm. Also, see the Oracle JDeveloper Accessibility Information chapter in the *Oracle Fusion Middleware User's Guide for Oracle JDeveloper*.

3.7 Oracle on the Web

Oracle provides a number of resources on the Web. Some sites you might find helpful are listed in Table 3–1, " Oracle on the Web":

Table 3–1   Oracle on the Web

Description	URL
Oracle JDeveloper Home Pages	http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html
Oracle JDeveloper Discussion Forum	http://forums.oracle.com/forums/forum.jspa?forumID=83
Corporate Site	http://www.oracle.com/
Oracle Technology Network	http://www.oracle.com/technetwork/index.html
Oracle Accessibility Site	http://www.oracle.com/us/corporate/accessibility/index.html

4

Deinstalling Oracle JDeveloper

This chapter describes how to remove Oracle JDeveloper products from your system.

It contains the following sections:

- Performing Pre-Deinstallation Tasks
- Deinstalling Oracle JDeveloper

4.1 Performing Pre-Deinstallation Tasks

Before deinstalling Oracle Fusion Middleware software components, you should stop all Oracle JDeveloper processes.

For more information about starting and stopping Oracle Fusion Middleware, refer to "Starting and Stopping Oracle Fusion Middleware" in *Administering Oracle Fusion Middleware*.

4.2 Deinstalling Oracle JDeveloper

This section contains information and instructions for removing Oracle JDeveloper components. When you run the Oracle JDeveloper deinstaller, it removes everything under the Oracle home (referred to in this guide as `JDEV_HOME`) from which the deinstaller is started. Be sure that no system components are using the Oracle home you want to remove.

This section contains the following topics:

- Starting the Deinstaller
- Manually Removing Artifacts

4.2.1 Starting the Deinstaller

To deinstall Oracle JDeveloper Studio, do the following:

1. Go to the following location in your JDEV_HOME:

   - On your Windows system:

     `JDEV_HOME\oui\bin\`

   - On your Linux, UNIX, and Mac OS X systems:

     `JDEV_HOME/oui/bin/`

2. Start the deinstaller by running the following command:

- On your Windows system:

  ```
  install.exe -deinstall
  ```

- On your Linux, UNIX, Mac OS X systems:

  ```
  ./deinstall.sh
  ```

4.2.2 Manually Removing Artifacts

If you selected **No** on the warning screen during deinstallation, you must manually remove your *JDEV_HOME* directory and all sub-directories. For example, if your *JDEV_HOME* directory is /home/Oracle/, then on a Linux or UNIX operating system run the following command:

```
cd /home/Oracle/
rm -rf JDEV_HOME
```

On a Windows operating system, if your *JDEV_HOME* directory is C:\Oracle*JDEV_HOME*, use a file manager window and navigate to the C:\Oracle\ directory, then right-click on the *JDEV_HOME* folder and select **Delete**.

If you plan on doing a clean reinstall of JDeveloper in the future, you should also find and delete the product.conf file that contains JDeveloper settings from previous installations.

If you did not redefine your user home directory, product.conf will be located at %USERPROFILE%\AppData\Roaming\JDeveloper\system12.1.3.xx.xx.xx for Windows and *$HOME*/.jdeveloper/12.1.3.0.0/ for Linux.

A

Silent Installation and Deinstallation of Oracle JDeveloper

Silent-mode installation and deinstallation is a way of setting installation or deinstallation configurations only once to ensure that no configuration options are displayed during the installation or deinstallation process. During installation in silent mode, the installation program reads the settings from a response file that you create before beginning the installation.

This appendix describes how to perform a silent installation and deinstallation of Oracle JDeveloper.

It contains the following sections:

- Installing Oracle JDeveloper Studio in Silent Mode
- Deinstalling Oracle JDeveloper in Silent Mode

A.1 Installing Oracle JDeveloper Studio in Silent Mode

For more information, see "Running the Oracle Universal Installer in Silent Mode" in *Installing with the Oracle Universal Installer*.

A.2 Deinstalling Oracle JDeveloper in Silent Mode

For information about deinstalling Oracle JDeveloper in silent mode, see "Running the Oracle Universal Installer for Silent Deinstallation" in *Installing with the Oracle Universal Installer*.