Oracle® Fusion Middleware

Installing Oracle JDeveloper

14*c* (14.1.2.0.0)
F85515-01
December 2024

Oracle Fusion Middleware Installing Oracle JDeveloper, 14*c* (14.1.2.0.0)

F85515-01

Contributing Authors: Oracle Corporation

Contents

4 Uninstalling Oracle JDeveloper

A Updating the JDK After Installing and Configuring an Oracle Fusion Middleware Product

Abstract

Documentation for installers and system administrators that describes how to install and configure Oracle JDeveloper.

Preface

This document describes how to install and configure Oracle JDeveloper.

Audience

This guide is intended for system administrators or application developers who are installing and configuring Oracle JDeveloper. It is assumed that readers are familiar with web technologies and have a general understanding of Linux and Windows platforms.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://support.oracle.com/portal/ or visit Oracle Accessibility Learning and Support if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

Refer to the Oracle Fusion Middleware Library for additional information.

- For Oracle JDeveloper information, see Oracle JDeveloper documentation.
- For installation information, see Fusion Middleware installation documentation.
- For upgrade information, see Fusion Middleware upgrade documentation.
- For administration-related information, see Fusion Middleware administration documentation.
- For release-related information, see Fusion Middleware release notes.

Conventions

This document uses the following text conventions:

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

Prepare for Oracle JDeveloper installation by verifying that your system meets the basic requirements, then obtain the correct installation software.

About Oracle JDeveloper

Oracle JDeveloper provides a productive development environment that covers the full development life cycle. JDeveloper integrates features that address many technologies used for building standards-based enterprise applications.

Some of the important features of Oracle JDeveloper are:

- 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 provides a simpler way to define the components that construct an application, simplifying or eliminating tedious coding.
- Integrates the full set of features needed to develop composite applications with features that address technologies such as Java, Database, web page design, web services and XML, Fusion Middleware, and others.

> **Note:**
>
> This release of JDeveloper requires Java SE Development Kit (JDK) 17.0 or later.

Road Map for Installing Oracle JDeveloper

This road map provides all the steps required to install Oracle JDeveloper.

Table 1-1 Road Map for Installing Oracle JDeveloper

Task	Description	Documentation
Verify your system environment	Before you begin the installation, verify that the minimum system and network requirements are met.	See Verifying the Oracle JDeveloper Certification Information and Verifying Your System Requirements for Installing Oracle JDeveloper
Obtain the appropriate software distribution	To install Oracle JDeveloper on your system, obtain an Oracle JDeveloper distribution.	See Understanding and Obtaining the Oracle JDeveloper Distributions

Table 1-1 (Cont.) Road Map for Installing Oracle JDeveloper

Task	Description	Documentation
Install the software	Run the Oracle Universal Installer to install Oracle JDeveloper. Installing JDeveloper transfers the software to your system and creates the Oracle home directory.	See Installing Oracle JDeveloper
Customize your Oracle JDeveloper on Startup	To use Oracle JDeveloper, customize the settings on your system.	See Customizing Oracle JDeveloper on Startup

Verifying the Oracle JDeveloper Certification Information

Before you install Oracle JDeveloper, you should understand compatibility with other certified software.

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) web page.

> **Note:**
>
> This version of Oracle JDeveloper ships with Apache ANT 1.10.14.0.0. Higher versions of ANT may work with this release, but they are not supported, and they may cause unexpected behavior.

Verifying System Requirements

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.

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

For information about the certified JDK for your installation, see About JDK Requirements for an Oracle Fusion Middleware Installation.

Understanding and Obtaining the Oracle JDeveloper Distributions

Oracle JDeveloper is available in two distinct distributions: Oracle JDeveloper Studio and Oracle JDeveloper Java.

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 14c (14.1.2.0.0) 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 About JDK Requirements for an Oracle Fusion Middleware Installation. Oracle JDeveloper Studio can also be installed with 64-bit platform-specific installers for Linux or Windows. See Installing Oracle JDeveloper Studio
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. You only need to unzip the file. See Installing the JDeveloper Java Distribution.

Download either an Oracle JDeveloper Studio installer or the Java distribution .zip file from the Oracle JDeveloper downloads page.

Deploying and Testing Applications Developed in Oracle JDeveloper

With 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.

See Using Oracle WebLogic Server with Oracle JDeveloper.

To deploy Java Enterprise Edition (Java EE) applications from Oracle JDeveloper, install and configure the Oracle WebLogic Server and Coherence distribution. See Planning the Oracle WebLogic Server Installation in the *Oracle Fusion Middleware 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. See Planning the Oracle Fusion Middleware Infrastructure Installation in the *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

2

Installing the Oracle JDeveloper Software

Follow the steps pertinent to your operating system to install the Oracle JDeveloper software.

> **Note:**
>
> The Oracle JDeveloper software installation described here does not enable you to develop SOA or Business Process Management applications.
> To install a version of JDeveloper preconfigured for SOA Suite or Business Process Management Suite applications, see Introducing the Quick Start Distributions in *Oracle Fusion Middleware Installing Oracle SOA Suite and Business Process Management Suite Quick Start for Developers*.

Understanding the Oracle JDeveloper Installation Types

The installation steps depend on the Oracle JDeveloper distribution and installer you are using.

> **Note:**
>
> The installation types described here do not support SOA extensions. To install a version of JDeveloper preconfigured for SOA Suite or Business Process Management Suite applications, see Introducing the Quick Start Distributions in *Oracle Fusion Middleware Installing Oracle SOA Suite and Business Process Management Suite Quick Start for Developers*.

You must obtain the appropriate installer for your operating system as described in Understanding and Obtaining the Oracle JDeveloper Distributions. Depending on the distribution and installer you are using, the type of installation you perform differs:

- If you are installing the Oracle JDeveloper Java distribution, all you need to do is unzip the files. This process is covered in Installing the JDeveloper Java Distribution.
- 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 plan to run the generic installer for Oracle JDeveloper Studio, ensure that the following prerequisites are met:
  - The required JDK must be installed.

    Identify what JDK versions are certified for JDeveloper 14c (14.1.2.0.0) on your operating system by using this site: `http://www.oracle.com/technetwork/developer-tools/jdev/documentation/index.html`

    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 the installation process, the JDK that you wish to use with the product must be the one that you use to start the installer program. Make sure that the desired JDK is the first one in your systems *PATH* environment variable, or reference it explicitly on the command line.

Installing Oracle JDeveloper Studio

You can install Oracle JDeveloper Studio on Linux, UNIX, Mac OS X, and Windows operating systems.

> **Note:**
>
> The Oracle JDeveloper Studio installation described here cannot be used to develop SOA Suite or Business Process Management Suite applications. See Understanding the Oracle JDeveloper Installation Types for details.

Starting the Oracle JDeveloper Studio Installation Program

The steps to start the Oracle JDeveloper Studio installation program depend on the type of installer you are using.

When the installation program appears, you are ready to begin the installation. For a description of each installation program screen, see Navigating the Installation Screens.

You can use one of the following procedures to install Oracle JDeveloper Studio:

Using the Generic Installer

To start the generic (platform-independent) installation program:

1. Sign in to the host system.

   Be sure that the user account you use for installing Oracle JDeveloper has the required privileges. See Selecting an Installation User in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.

2. Verify that a certified JDK exists on your system by running `java -version` from the command line. At the time of publication, the certified JDK for 14c (14.1.2.0.0) was 17.0.12. See the appropriate certification document on the *Oracle Fusion Middleware Supported System Configurations* page.

3. At the command line, navigate to the directory where you downloaded the generic installation program.

4. Start the installation program by running the `java` executable file from the JDK directory on your system:

   - (Linux, UNIX, and Mac OS X) `/home/Oracle/Java/jdk17.0.12/bin/java -jar V1045138-01.jar`
   - (Windows) `C:\home\Oracle\Java\jdk17.0.12\bin\java -jar V1045138-01.jar`

Replace the JDK location in these examples with the actual JDK location on your system.

> **Note:**
>
> You can also start the installer in silent mode by using a saved response file instead of using the installer screens. See Using the Generic Installer in Silent Mode.

Using the Platform-Specific Installer

To start the platform-specific installation program:

1. Sign in to the host system.

   Be sure that the user account you use for installing Oracle JDeveloper has the required privileges. See Selecting an Installation User in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.

2. At the command line, navigate to the directory where you downloaded the platform-specific installation program.

   > **Note:**
   >
   > After you have downloaded the platform-specific installer, ensure that you rename the file before you start the installation program, as given below:
   >
   > - For Linux X86-64, rename the file from `V998596-01.zip` to `V998595-01-2.zip`.
   > - For Windows, rename the file from `V998594-01.zip` to `V998593-01-2.zip`.

3. Start the installer by running the following command:
   - (Linux, UNIX, and Mac OS X) `./V998595-01.bin`
   - (Windows) `V998593-01.exe`

Using the Generic Installer in Silent Mode

To start the generic installation program in silent mode:

1. Create a Response file.
   a. Run your product graphical installer, as described in the product installation guide.
   b. For product installation, on the Installation Summary screen, click **Save Response File** to save the installation parameters to a response file.

      This sample shows the information that you can specify in a response file for a silent installation of JDeveloper.

```
[ENGINE]

#DO NOT CHANGE THIS.
Response File Version=1.0.0.0.0

[GENERIC]

#Set this to true if custom installation must be done, all other
```

```
required variables need to be provided. If this is set to true then
variable "TYPICAL_TYPE" must be set to false as the variables are
mutually exclusive.
INSTALL_TYPE=Typical

#Set this to true if you wish to skip software updates
DECLINE_AUTO_UPDATES=true

#The oracle home location. This can be an existing Oracle Home or a new
Oracle Home
ORACLE_HOME=
```

To see other sample response files that can be used for silent installation, see Sample Response Files for Silent Installation and Deinstallation in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

2. Set the *ORACLE_HOME* variable in the response file to the location of your Oracle home directory.
3. At the command line, navigate to the directory where you downloaded the generic installation program.

> **Note:**
>
> After you have downloaded the generic installer, ensure that you rename the `V998598-01.jar` file to `V998597-012.jar` before you start the installation program.

4. Start the installation program in silent mode using the saved response file by running the java executable file from the JDK directory on your system.
   - (Linux, UNIX, and Mac OS X) `/home/Oracle/Java/jdk17.0.12/bin/java -jar V998597-01.jar -silent -responseFile <FULL_PATH>/<FILE_NAME>`
   - (Windows) `C:\home\Oracle\Java\jdk17.0.12\bin\java -jar V998597-01.jar -silent -responseFile <FULL_PATH>/<FILE_NAME>`
   - Replace the JDK location in these examples with the actual JDK location on your system.
   - Replace <*FULL_PATH*> with the full path to the location of the response file.
   - Replace <*FILE_NAME*> with the name of the response file.

For more information about silent installation, see Using the Oracle Universal Installer in Silent Mode in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

Navigating the Installation Screens

The installer shows a series of screens where you verify or enter information.

The following table lists the order in which installer screens appear. If you need additional help with an installation screen, click **Help**.

Table 2-1 Oracle JDeveloper Install Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen opens 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*. This screen does not appear on Windows operating systems.
Welcome	This screen introduces you to the product installer.
Installation Location	Use this screen to specify the location of your Oracle home directory. Install the JDeveloper component in a new Oracle home. The only component in the Oracle installer that can be installed into an existing Oracle home is the Application Development Framework (ADF), if ADF has not already been installed by other Oracle Fusion Middleware products. Ensure that the name of the directory does not contain spaces. For example, do not use "`C:\Program Files`" as the Oracle home. For more about Oracle Fusion Middleware directory structure, see Understanding Directories for Installation and Configuration in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.
Prerequisite Checks	This screen verifies that your system meets the minimum requirements. If there are any warning or error messages, refer to Verifying Your System Requirements for Installing Oracle JDeveloper.
Installation Summary	Use this screen to verify the installation options that 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 for a silent installation. For more information about silent installation, see Using the Oracle Universal Installer in Silent Mode in *Installing Software with the Oracle Universal Installer*. Click **Install** to begin the installation.
Installation Progress	This screen shows the installation progress. When the progress bar reaches 100%, click **Finish** to dismiss the installer, or click **Next** to see a summary.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen. For more information about customizing Oracle JDeveloper using the Installation Completed screen, see Customizing Oracle JDeveloper on Startup.

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

Customizing Oracle JDeveloper on Startup

You can customize Oracle JDeveloper Studio on startup using options on the Installation Complete screen of the installer.

The following table describes the options in the **Next Steps** section of the Installation Complete screen:

Table 2-2 Options to Customize Oracle JDeveloper on Startup

Option	Description and Action
Start JDeveloper with Default Settings	Select if you want to start Oracle JDeveloper using the default settings. All your Oracle JDeveloper projects are saved in the default location.
Customize Settings and Then Start JDeveloper	Select 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 option to start JDeveloper after quitting the installation wizard is checked, uncheck it. Click **Finish** to quit the installation wizard without starting Oracle JDeveloper.

Note:

- During installation, a file called `product.conf` is created in your home directory. This file contains your JDeveloper preferences. Old preferences carry over to new installations of JDeveloper unless you delete this file.
- On Windows systems, if you are uninstalling your current version of Oracle JDeveloper to install a newer version, Oracle recommends that you restart your system after you uninstall the software and before you begin the new installation.

Verifying the Oracle JDeveloper Studio Installation

After you complete the installation of Oracle JDeveloper Studio, verify that the installation was successful.

To verify that Oracle JDeveloper Studio installed successfully, you can compare the directory structure created on your expected directory structure shown in the following table.

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.
`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 in this file is updated.
`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.

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

Directory or File	Description
`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`	This file contains information about the inventory location of the installation. You may not see this file on a Windows installation.
`oui`	This directory contains files used by the Oracle Universal Installer, including the uninstaller 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 directory called `cfgtoollogs`. This directory contains the log files for your installation and configuration transactions.

Installing the JDeveloper Java Distribution

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, at: `http://www.info-zip.org/`.

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

To install Oracle JDeveloper from `jdev_java_141200.zip`:

1. Obtain the required JDK.

   Use the following site to identify what JDK versions are certified for this release of JDeveloper on your operating system: `http://www.oracle.com/technetwork/developer-tools/jdev/documentation/index.html`

   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_141200.zip` to the target directory.

> **Note:**
>
> In this guide, 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_141200.zip` into `C:\jdev_install\`, then *`JDEV_HOME`* refers to `C:\jdev_install`.

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 example:

- (Linux, UNIX, and Mac OS X) `/home/Oracle/Java/jdk17.0.12/bin`.
- (Windows) `C:\home\Oracle\Java\jdk17.0.12/bin`.

3

Next Steps After Installing Oracle JDeveloper Studio

After installing Oracle JDeveloper, you can start JDeveloper and perform related tasks.

Preparing to Start Oracle JDeveloper

You can start Oracle JDeveloper on Linux, UNIX, Mac OS X, Windows, and multiuser environments.

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

Before you start Oracle JDeveloper, you can specify settings for Linux, UNIX, and Mac OS X environments.

Setting the System Resource Limit

The minimum recommended system resource 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 as follows:

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.

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

You can permanently redefine the location of your user home directory 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

also contains 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 are located in different areas on your computer.

- The default location for the `system` subdirectory is `$HOME/.jdeveloper/system14.1.2.0.0.XX.XX.XX`, where `XX.XX.XX` is the unique number of the product build.
- The default location for user-generated content is `$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 set the user home directory for all instances of JDeveloper on your system.

Use either of the following methods to set the user home directory:

- Editing product.conf
- Setting the JDEV_USER_DIR environment variable, which requires you to restart your terminal session

Editing product.conf

With 14c (14.1.2.0.0), 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 exists from a previous installation. JDeveloper uses the settings stored in `product.conf` even if they are from a previous installation.

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

1. Find the `product.conf` file. It should be located at `$HOME/.jdeveloper/14.1.2.0.0`.
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 use a directory that contains spaces as the home directory. 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 that you have defined.

   When you start Oracle JDeveloper for the first time, you can verify that the user home directory has been set to your preferred directory by clicking **About** in the Help menu, toggling the **Properties** tab, and finding the definition for `ide.user.dir`.

Setting the JDEV_USER_DIR environment variable

You can set the `JDEV_USER_DIR` environment variable to your preferred home directory path. The following examples and syntax 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 use a directory that contains spaces as the home directory. 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. To confirm the change, at the command line, use the following command to display the environment variable:

   ```
   echo $JDEV_USER_DIR
   ```

   The output is the directory you specified. For this example, the command outputs the following:

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

When you start Oracle JDeveloper for the first time, you can verify that the user home directory has been set to your preferred directory by clicking **About** in the Help menu, toggling the **Properties** tab, and finding the definition for `ide.user.dir`.

Changing the JDK Location on Linux, UNIX, and Mac OS X

You can permanently change the location of your JDK, if the location has changed since the first time you started JDeveloper.

To change a JDK location:

1. Find the `product.conf` file. The file is located at `$HOME/.jdeveloper/14.1.2.0.0`.
2. Open `product.conf` in an editor. Find the following line:

   ```
   SetJavaHome /path/jdk
   ```

3. Uncomment this line, and replace `/path/jdk` with the path to your JDK directory.

   For example, if the location of your JDK is `/usr/local/java`, your definition looks like the following:

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

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

If you uninstall JDeveloper but do not delete the `product.conf` file, all settings stored in this file, including the JDK path, are preserved for future installations. If you reinstall the same

version of JDeveloper as your previous installation, your new installation automatically reads that JDK location again.

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 as follows:

   a. Navigate to the following location on your system:

      JDEV_HOME/jdeveloper/jdev/bin/clear_cursors.tar

   b. To extract the replacement cursors from the tar file, run the following command:

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

Optimizing Oracle JDeveloper on Windows

Before starting Oracle JDeveloper, you can specify settings for a Windows environment.

Setting the User Home Directory on Windows

You can permanently redefine the location of your user home directory in a Windows environment.

When you define a 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 also contains 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 are located in different locations.

- The default location for the `system` subdirectory is `%USERPROFILE%\Application Data\JDeveloper\system14.1.2.0.XX.XX.XX` on Windows platforms.

  In these locations, `XX.XX.XX` is the unique number of the product build.

- The default location of the user-generated content is `C:\Documents and Settings\My Documents\JDeveloper\mywork` on 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 set the user home directory for all instances of JDeveloper on your system.

Use either of the following methods to set the user home directory:

- Editing product.conf
- Setting the environment variable JDEV_USER_DIR

Editing product.conf

With 14c (14.1.2.0.0), 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 uses the settings stored in `product.conf` even if they are from a previous installation.

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

1. Find the `product.conf` file. It should be located at `%USERPROFILE%\AppData\Roaming\JDeveloper\14.1.2.0.0`.
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 `N:\users\jdoe`, your definition should look like this:

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

   > **Caution:**
   >
   > Do not use a directory that contains spaces as the home directory. 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` also overrides any environment variable you have defined.

   When you start Oracle JDeveloper for the first time, you can verify that the user home directory has been set to your preferred directory by clicking **About** in the Help menu, toggling the **Properties** tab, and finding the definition for `ide.user.dir`.

Setting the environment variable JDEV_USER_DIR

You can set the environment variable `JDEV_USER_DIR` to your preferred home directory path on a Windows system, including individual users of Oracle JDeveloper on a multiuser system.

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 use a directory that contains spaces as the home directory. 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 lists all your defined variables. Verify that your newly defined variable appears similar to the following:

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

When you start Oracle JDeveloper for the first time, you can verify that the user home directory has been set to your preferred directory by clicking **About** in the Help menu, toggling the **Properties** tab, and finding the definition for `ide.user.dir`.

Changing the JDK location on Windows

You can permanently change the location of your JDK, if the location has changed since the first time you started JDeveloper.

To change a JDK location:

1. Find the `product.conf` file. The file is located at `%USERPROFILE%\AppData\Roaming\JDeveloper\14.1.2.0.0`.
2. Open `product.conf` in an editor. Find the following line:

   ```
   SetJavaHome \path\jdk
   ```

3. Uncomment this line, and replace `\path\jdk` with the path to your JDK directory.

   For example, if the location of your JDK is jdk17.0.12 on your `D:` drive, your definition looks like the following:

   ```
   SetJavaHome D:\jdk17.0.12
   ```

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

If you uninstall JDeveloper but do not delete the `product.conf` file, all settings stored in this file, including the JDK path, are preserved. If you reinstall the same version of JDeveloper as your previous installation, your new installation automatically reads that JDK location again.

Optimizing Oracle JDeveloper in a Multiuser Environment

When you install and configure Oracle JDeveloper for a multiuser environment, you need to account the number of users and the power of the server so that you can deliver optimal performance for JDeveloper and your users.

Multiuser environments include Microsoft Terminal Server, Citrix MetaFrame and Citrix MetaFrame XP for Windows, and Citrix MetaFrame 1.1 for UNIX. These environments allow many clients to access one installation of Oracle JDeveloper. In all cases, users can save their projects locally.

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

With administrative privileges, you can install Oracle JDeveloper on a Citrix MetaFrame server or a Microsoft Terminal Server.

Run the Oracle JDeveloper installer on your server as detailed in Installing Oracle JDeveloper. Do not start Oracle JDeveloper at the end of installation, because you must first configure the user home directories and terminal server clients as described in the following sections.

Configuring Terminal Server Clients to Run Oracle JDeveloper

After you install a Citrix MetaFrame or Microsoft Terminal Server client locally and JDeveloper is installed and configured, you can configure a Terminal Server client to run Oracle JDeveloper.

1. Verify that the color resolution of the Terminal Server client has been set to at least 256 colors.
2. Sign in to the 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. Start Oracle JDeveloper.
5. Oracle JDeveloper prompts you to confirm that the user home directory needs to be created. Click **Yes**.
6. Verify that the user home directory has been set to your preferred directory:
   - From the **Help** menu, select **About**.
   - Toggle the **Properties** tab, and find the definition for `ide.user.dir`.

Troubleshooting a System DLL ole32.dll Memory Error

You may encounter the following error when you run Oracle JDeveloper in a multiuser environment:

```
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 product.conf file in `$HOME/.jdeveloper/14.1.2.0.0` 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:

1. In 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.

Starting Oracle JDeveloper

To start Oracle JDeveloper, follow the steps for the operating system you are using, or start it from the command line.

On Linux, UNIX, and Mac OS X operating systems:

1. Go to the `bin` directory.

   JDEV_HOME`/jdeveloper/jdev/bin/`

2. Start JDeveloper:

   `./jdev`

On Windows operating systems:

- From the Start Menu, select **All Programs**, select **Oracle Fusion Middleware 14.1.2.0.0**, and then select **JDeveloper Studio 14.1.2.0.0**.

On the command line, enter 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)

Migrating Oracle JDeveloper From a Previous Version

Oracle JDeveloper 14c (14.1.2.0.0) supports migration from version 12.2.1.4.0.

The supported migration paths to 14.1.2.0.0 are from 12c (12.2.1.4.0) and 14c (14.1.1.0.0). If you are currently not running a supported version, you should migrate your application to one of those releases before migrating to 14.1.2.0.0.

Migrating User Settings

When you start Oracle JDeveloper for the first time (and each time that 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 box appears, 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 appears as a tooltip.

Alternatively, you can click **Find a previous installation manually** (with the image of a magnifying glass) at the top right of the installation list to browse for an installation manually. Clicking **Yes** in 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 you start Oracle JDeveloper from the command line. For example, use `jdev -migrate`.

> **Note:**
>
> If you migrate to 14c (14.1.2.0.0) from another version of Oracle JDeveloper, reinstall the extensions that you want to use. Some extensions for older versions might no longer exist.
>
> For more information on installing extensions, see Enabling Oracle JDeveloper Extensions.

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 14*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 you migrate them. When you accept the initial prompts to migrate your projects, JDeveloper updates the format for crucial XML files but does not rewrite or update specific project code. After migration, retest your applications to verify that they work.

See the Oracle JDeveloper page on Oracle Technology Network (OTN) at Oracle JDeveloper for more information about migrating specific types of projects to 14*c*.

Migrating JSF and JSTL Projects

This release of Oracle JDeveloper requires that you migrate all projects with JavaServer Faces and JSTL to the latest versions of the technologies. In addition, any `web.xml` deployment descriptors must also be migrated.

For the most current information on migration, see the Oracle JDeveloper documentation page on the Oracle Technology Network (OTN): Oracle JDeveloper .

Using Headless Migration

You can also migrate files silently from the command line using the headless migration tool. This tool can be found at `jdeveloper/jdev/bin/ojmigrate`.

Understanding How to Use ojmigrate

The `ojmigrate` command has the following structure:

```
ojmigrate [option]... file...|@file
```

The `option` parameter is optional. The `file` or the `@file` parameter value is required.

Using the `option` Parameter

You can use the following flags for the `option` parameter:

- `-ade` to connect to the current ADE view
- `-dry` to initiate a dry run and skip calling migrators
- `-failFast` to stop the migration after the first failure

`-generateDefaults` to let migrator helpers generate a migration.prperties file, alongside the `.jws` file, that contains the defaults for migration options.

To view these values and their use cases, invoke the help file for `ojmigrate` at the command line.

Using the `file` Parameter

To migrate one application, specify the path to the `.jws` file you want to migrate as the value for `file`.

For example, to migrate `example.jws` with no option flag, the command is the following:

```
ojmigrate example.jws
```

To migrate more than one application, use the `@file` parameter described in the next section.

Using the `@file` Parameter

To migrate more than one application at a time, create a file that contains all the paths to the applications that you want to migrate, and use the `@file` parameter to point to this file.

To autogenerate this file, use a script like the following:

```
find . -name "*.jws" -printf "%P\n" > workspaces.txt
```

After this file is generated, check the file to make sure that all the applications you want migrated are included. You can also create the file manually.

Once you have created and saved this file, specify its path as the value for `@file` in the example command structure. The @ symbol must be preppended to your filename when you specify this parameter.

For example, if you have listed the location of multiple `.jws` files in `example.txt`, your command would be the following:

```
ojmigrate @example.txt
```

Troubleshooting ojmigrate

If you encounter errors during migration, you can try to troubleshoot them by using the strategies in this section. To isolate the cause of your errors, run `ojmigrate` with the `-failFast` flag. This stops `ojmigrate` after its first unsuccessful migration and analyzes the issue.

Migrating Applications With `-ade`

If you see messages about being unable to check out read-only files, migrate these applications in ADE.

Make sure that you are in ADE view and that you have a transaction started. Because the ADE extension is not part of the JDeveloper non-debug builds, you should either use a JDeveloper debug build or manually install the ADE extension.

Once you are in ADE view, invoke `ojmigrate` with the `-ade` flag.

The commands that are mentioned in this section resemble the following:

```
ade useview VIEW_NAME
ade begintrans TXN_NAME
ojmigrate -ade @workspaces.txt
```

Using a Two-Pass Migration

Migration may have failed because some migrators require extra data normally provided in the migration wizard, or because the defaults that the migrators are using are not suitable for your application. In these cases, do a two-pass migration.

For the first pass, run the `ojmigrate` command with the `-generateDefaults` flag. The command resembles the following:

```
ojmigrate -genarateDefaults @workspaces.txt
```

This command generates a properties file that contains a list of defaults that are formatted as name/value pairs with comments. The key/value pair is per application and per NodeMigratorHelper, meaning that there is one set of key pairs for every application, and the key is prefixed with the class name of the corresponding NodeMigratorHelper.

The properties file is generated beside the `.jws` application that you are trying to migrate. Its title is formatted as *workspacename*.migration.properties.

If the file is generated, open it, change the appropriate values, and save your changes. If you do not see the file after you run the command, no migrators generated defaults.

On the second pass, run the command without the `-generateDefaults` flag.

```
ojmigrate @workspaces.txt
```

If you still encounter errors, then some migrators may not be headless-friendly. A developer should prepare your migrator for headless migration before you attempt the two-pass migration again.

Preparing a NodeMigratorHelper for Headless Migration

If you need data from the user before migration, do the following:

1. Override the `NodeMigratorHelper.generateDefaults(MigrationInf[], TraversableContext)` method. This method is called on your helper on the first run of the two-pass migration.

   Then, use the `putDefault(String key, String value, String comment)` method to store the keys and the default values that you retrieve on second run of the two-pass migration.

2. In your helper's migration method, call `isMigrationHeadless()` on the helper to test whether the migration is headless. If the migration is headless, use to `getDefault(your_key)` method to ask for your data values. Read the data and act accordingly during the migration.

Installing Oracle JDeveloper JavaFX Extensions

Oracle JDeveloper JavaFX extension is available internally through JDeveloper updates. If you do not manually install the JavaFX extension with JDeveloper 14c, then you will not be able to visually edit your JSP or JSF files.

> **Note:**
>
> JDeveloper no longer includes the JSF Visual Editor as a built-in feature, since JavaFX is no longer part of the JDK. To use the visual JSF editor with JDeveloper 14c, you must manually download the JavaFX extension.

To locate and install extensions using Oracle JDeveloper:

1. Start JDeveloper.
2. Click **Help** and select **Check for Updates** to start an Update wizard.
3. On the Updates screen of the Update wizard, select **JavaFX Extension (21.0.1)** and click **Next**.
4. The operating system-specific JavaFX Extension will appear. For instance, if your operating system is Linux, then JavaFX Extension for Linux option will be shown. Click **Next**.
5. On the **License Agreements** screen, agree to the license terms for the extensions that you selected on the previous screen. Click **Next** to begin the download.
6. When the extensions have finished downloading, the wizard displays a summary of the installations or upgrades that you installed. Click **Finish** to exit the wizard.
7. In order to integrate the new JavaFX extension, JDeveloper must be restarted. When prompted, click **Yes** to restart JDeveloper.

Understanding Oracle JDeveloper Accessibility Information

You can use a screen reader with Oracle JDeveloper and become familiar with assistive technologies in Oracle products and Oracle JDeveloper.

Using a Screen Reader and Java Access Bridge with Oracle JDeveloper

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

- Microsoft Windows (32-bit and 64-bit)
- Java Access Bridge - Integrated with Java SE 7 Update 6 and later
- JAWS version 2022 (build 2022.2204.20 ILM)
- Microsoft Edge 79+ or later
- Mozilla Firefox 79+ or later

To set up a screen reader and enable the Java Access Bridge, follow these steps.

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

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

   For more information, refer to the documentation for your screen reader.
2. Install Oracle JDeveloper.
3. Enable Java Access Bridge:
   a. Open the Control Panel in Microsoft Windows and click **Appearance and Personalization**.
   b. Click **Ease of Access Center**, and then click the **Make the computer easier to see** link.
   c. Ensure that the following check boxes are selected:
      - JAWS 2022
      - Enable Java Access Bridge
4. Confirm that the following files are installed in the *JDEV_HOME*`\jdk\jre\bin` directory:
   - `JavaAccessBridge-64.dll`
   - `JAWTAccessBridge-64.dll`
   - `WindowsAccessBridge-64.dll`

   > **Note:**
   >
   > If you are installing Java Access Bridge for Windows 32-bit, check for `JavaAccessBridge-32.dll`, `JAWTAccessBridge-32.dll`, and `WindowsAccessBridge-32.dll`, respectively.
5. Confirm that the following files have been installed in the *JDEV_HOME*`\jdk\jre\lib\ext` directory.
   - `access-bridge-64.jar` (`access-bridge-32.jar` for Windows 32-bit).
   - `jaccess.jar`
6. Confirm that the `accessibility.properties` file has been installed in the *JDEV_HOME*`\jdk\jre\lib` directory.
7. Start your screen reader.
8. Start Oracle JDeveloper by running the `jdev.exe` file located in the *JDEV_HOME*`\jdeveloper\jdev\bin` folder.

A console window that contains error information (if any) appears 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.

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. See also Oracle JDeveloper Accessibility Information in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*.

Using Oracle WebLogic Server with Oracle JDeveloper

Installing Oracle JDeveloper Studio 14c (14.1.2.0.0) also automatically installs Oracle WebLogic Server 14c (14.1.2.0.0).

> **Note:**
>
> The information in this section is not applicable to the Java edition of Oracle JDeveloper.

Oracle JDeveloper uses the preconfigured Oracle WebLogic Server installation as the Integrated Oracle WebLogic Server and JDeveloper managed server for testing and debugging your applications from within the IDE. After you install 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 Deploying Applications in *Oracle Fusion Middleware Developing Applications with Oracle JDeveloper*.

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.

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

> **Note:**
>
> - The Oracle WebLogic Server domain that is created for you during installation, default domain, 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. See Preparing the Standalone Application Server for Deployment in *Oracle Fusion Middleware Administering Oracle ADF Applications*.
> - Ensure that the user account you use for domain creation in Windows has the required administrator privileges.

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

Adding Server Certificates to Support Secure Mode in JDev

When running Oracle WebLogic in secure mode, you must manually add your server certificates before connecting through the Create Application Server Connection wizard.

This process applies only when using secure production mode. If you do not manually add the server certificate, the server connection fails.

Before starting the Create Application Server Connection wizard, follow these steps to add a new server certificate in JDeveloper.

1. Open the **Tools** menu and select **Preferences**.
2. Select **Keystores** and add the keystore password.

   ```
   Keystore Password = "DemoTrustKeyStorePassPhrase"
   ```

3. Provide the certificate location. The `cert.pem` certificate should be located in the following directory:

   ```
   [jdk_path]/lib/security/cacerts
   ```

4. Provide the key alias that was created for the certificate.

   ```
   Key Alias= KeyAlias
   ```

Additional Resources

Useful Oracle resources related to Oracle JDeveloper are available on the Web.

The following table lists some helpful sites:

Table 3-1 Oracle Resources on the Web

Description	URL
Oracle JDeveloper Home Pages	https://www.oracle.com/application-development/technologies/jdeveloper.html
Oracle JDeveloper Discussion Forum	https://community.oracle.com/tech/developers/categories/jdeveloper_and_adf
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

Setting the Language for Log Messages Generated by the JDeveloper Integrated Server

If you are installing Oracle JDeveloper in a non-English environment, the messages written to the JDeveloper Integrated Server log files will be written by using the locale of the host operating system.

Specifically, on Windows systems, if you want the messages in the log files to be in a specific language, ensure to configure the operating system to use the language of choice. This is true of most Java applications.

Complete the following steps to change the language setting on a Windows system:

1. Complete the instructions specified in How do I view and change the system locale settings to use my language of choice? on the Java.com Web site.
2. Change the language format:
   a. Go to the **Control Panel** > **Region and Language** > **Formats** tab.
   b. In the Formats tab, select **English (United States)** from the Format drop-down list.

4
Uninstalling Oracle JDeveloper

Follow the instructions in this section to uninstall Oracle JDeveloper.

Preparing to Uninstall Oracle JDeveloper

Before uninstalling Oracle Fusion Middleware software components, stop all Oracle JDeveloper processes.

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

Uninstalling Oracle JDeveloper

When you run the Oracle JDeveloper uninstaller, it removes everything in the Oracle home from which the uninstaller is started.

Be sure that no system components are using the Oracle home (referred to in this guide as `JDEV_HOME`) that you want to remove.

Starting the Uninstaller

To uninstall Oracle JDeveloper Studio:

1. Navigate to the following location:
   - (Windows) *`JDEV_HOME`*`\oui\bin\`
   - (Linux, UNIX, or Mac OS X) *`JDEV_HOME`*`/oui/bin/`
2. Start the uninstaller by running the following command:
   - (Windows) `install.exe -deinstall`
   - (Linux, UNIX, or Mac OS X) `./deinstall.sh`

Manually Removing Artifacts

If you selected **No** on the warning screen during uninstallation, you must manually remove your *`JDEV_HOME`* directory and all subdirectories.

To do this, use the following methods:

- (Linux, UNIX, or Max OS X)

  If your *`JDEV_HOME`* directory is `/home/Oracle/`, enter the following commands:

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

- (Windows)

If your *JDEV_HOME* directory is `C:\Oracle\`, 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 to perform a clean reinstallation of JDeveloper, 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` is located at `%USERPROFILE%\AppData\Roaming\JDeveloper\14.1.2.0.0` in Windows and `$HOME/.jdeveloper/14.1.2.0.0` in Linux.

A

Updating the JDK After Installing and Configuring an Oracle Fusion Middleware Product

Consider that you have an unsupported JDK version installed on your machine. When you install and configure an Oracle Fusion Middleware product, the utilities, such as Configuration Wizard (`config.sh|exe`), OPatch, or RCU point to a default JDK. The supported JDK version for this release is jdk17.0.12 and it carries security enhancements and bug fixes. You can upgrade the existing JDK to a newer version, and can have the complete product stack point to the newer version of the JDK.

You can maintain multiple versions of JDK and switch to the required version on need basis.

About Updating the JDK Location After Installing an Oracle Fusion Middleware Product

The binaries and other metadata and utility scripts in the Oracle home and Domain home, such as RCU or Configuration Wizard, use a JDK version that was used while installing the software and continue to refer to the same version of the JDK. The JDK path is stored in a variable called JAVA_HOME which is centrally located in .globalEnv.properties file inside the *ORACLE_HOME*`/oui` directory.

The utility scripts such as config.sh|cmd, launch.sh, or opatch reside in the *ORACLE_HOME*, and when you invoke them, they refer to the JAVA_HOME variable located in .globalEnv.properties file. To point these scripts and utilities to the newer version of JDK, you must update the value of the JAVA_HOME variable in the .globalEnv.properties file by following the directions listed in Updating the JDK Location in an Existing Oracle Home .

To make the scripts and files in your Domain home directory point to the newer version of the JDK, you can follow one of the following approaches:

- Specify the path to the newer JDK on the Domain Mode and JDK screen while running the Configuration Wizard.

  For example, consider that you installed Oracle Fusion Middleware Infrastructure with the JDK version 8u191. So, while configuring the WebLogic domain with the Configuration Assistant, you can select the path to the newer JDK on the Domain Mode and JDK screen of the Configuration Wizard. Example: `/scratch/jdk/jdk17.0.12`.

- Manually locate the files that have references to the JDK using `grep` (Linux) or `findstr` (WINDOWS) commands and update each reference.

  See Updating the JDK Location in an Existing Oracle Home .

> **Note:**
>
> If you install the newer version of the JDK in the same location as the existing JDK by overwriting the files, then you don't need to take any action.

Updating the JDK Location in an Existing Oracle Home

The `getProperty.sh|cmd` script displays the value of a variable, such as JAVA_HOME, from the .globalEnv.properties file. The `setProperty.sh|cmd` script is used to set the value of variables, such as OLD_JAVA_HOME or JAVA_HOME that contain the locations of old and new JDKs in the .globalEnv.properties file.

The `getProperty.sh|cmd` and `setProperty.sh|cmd` scripts are located in the following location:

(Linux) *ORACLE_HOME*`/oui/bin`

(Windows) *ORACLE_HOME*`\oui\bin`

Where, *ORACLE_HOME* is the directory that contains the products using the current version of the JDK, such as jdk17.0.12.

To update the JDK location in the .globalEnv.properties file:

1. Use the `getProperty.sh|cmd` script to display the path of the current JDK from the JAVA_HOME variable. For example:

   (Linux) *ORACLE_HOME*`/oui/bin/getProperty.sh JAVA_HOME`

   (Windows) *ORACLE_HOME*`\oui\bin\getProperty.cmd JAVA_HOME`

   ```
   echo JAVA_HOME
   ```

   Where JAVA_HOME is the variable in the .globalEnv.properties file that contains the location of the JDK.

2. Back up the path of the current JDK to another variable such as OLD_JAVA_HOME in the .globalEnv.properties file by entering the following commands:

   (Linux) *ORACLE_HOME*`/oui/bin/setProperty.sh -name OLD_JAVA_HOME -value` *specify_the_path_of_current_JDK*

   (Windows) *ORACLE_HOME*`\oui\bin\setProperty.cmd -name OLD_JAVA_HOME -value` *specify_the_path_of_current_JDK*

   This command creates a new variable called OLD_JAVA_HOME in the .globalEnv.properties file, with a value that you have specified.

3. Set the new location of the JDK in the JAVA_HOME variable of the .globalEnv.properties file, by entering the following commands:

   (Linux) *ORACLE_HOME*`/oui/bin/setProperty.sh -name JAVA_HOME -value` *specify_the_location_of_new_JDK*

   (Windows) *ORACLE_HOME*`\oui\bin\setProperty.cmd -name JAVA_HOME -value` *specify_the_location_of_new_JDK*

   After you run this command, the JAVA_HOME variable in the .globalEnv.properties file now contains the path to the new JDK, such as jdk17.0.12.

Updating the JDK Location in an Existing Domain Home

You must search the references to the current JDK, for example 1.8.0_191 manually, and replace those instances with the location of the new JDK.

You can use the `grep` or `findstr` commands to search for the JDK-related references.

You'll likely be required to update the location of JDK in the following three files:

(Linux) *DOMAIN_HOME*`/bin/setNMJavaHome.sh`

(Windows) *DOMAIN_HOME*`\bin\setNMJavaHome.cmd`

(Linux) *DOMAIN_HOME*`/nodemanager/nodemanager.properties`

(Windows)*DOMAIN_HOME*`\nodemanager\nodemanager.properties`

(Linux) Start `bash` and then run *DOMAIN_HOME*`/bin>source setDomainEnv.sh`

(Windows)*DOMAIN_HOME*`\bin\setDomainEnv.cmd`