UNISYS | Securing Your Tomorrow™

ClearPath Application Development Solutions

Clearpath OS 2200 IDE for Eclipse™ Installation Guide

OXYGEN (v4.7.0)

October 2017 4729 2107-012

Contents

Tables

Section 1
Introduction to the ClearPath OS 2200 IDE for Eclipse

The Eclipse platform is one of the most useful integrated development tools available to a developer. Its built-in functionality is generic, open, and extensible using plug-ins that support development activities including the design, development, debugging, and deployment of simple and multi-tiered applications. Using appropriate plug-ins, you can use the Eclipse platform to develop applications in Java and many other programming languages. ClearPath OS 2200 IDE for Eclipse supports the IPv6 network. Refer to http://www.eclipse.org/eclipse/ for more information.

This section describes how to install and configure the OS 2200 IDE for Eclipse.

1.1. Documentation Updates

This document contains all the information that was available at the time of publication. Changes identified after release of this document are included in problem list entry (PLE) 19200841. To obtain a copy of the PLE, contact your Unisys representative or access the current PLE from the Unisys Product Support website:

http://www.support.unisys.com/all/ple/19200841

Note: *If you are not logged into the Product Support site, you will be asked to do so.*

1.2. Prerequisites

This document assists site administrators, installers, and programmers to prepare, install, and verify OS 2200 IDE for Eclipse software and associated software.

This document assumes that you are familiar with:

- ClearPath OS 2200 and application development concepts
- Java and Java 2 Platform, Enterprise Edition (J2EE) concepts

1.3. Notation Conventions

This document uses the following notation conventions.

Convention	Description
Italic	Names of variables to which values must be assigned appear in italic font.
Bold	Items such as the names of screen objects, key names, and command options are emphasized in text.
Monospace	Examples and system output, such as prompt signs and responses to commands, appear in monospace font.
>	Screen examples often use one greater-than (>) symbol generically to represent any kind of command line prompt.

1.4. What's New?

New or Revised Information	Location
Updated the Eclipse, Java and Windows versions.	Throughout the document.
Removed references to Application Development Guide as it is no longer supported.	Throughout the document.
Deleted information on Install XDoclet and JBoss Enterprise Application Platform.	Section 2, Get Started
Changed folder name from install to install46. Added dot(.) at the end of the ECL commands.	2.4 Install the PADS Library
Deleted information on Configure Eclipse Web Tools Project.	Section 3, Install and Start OS 2200 IDE for Eclipse

Section 2
Get Started

This section describes how to install and configure the OS 2200 Integrated Development Environment (IDE) for Eclipse and its related and required products.

2.1. Software Component Levels

This document refers to specific levels of components that were current at the time of the publication and are known to perform together. If a newer level of a component exists, you can choose to use it. However, we recommend that you get familiar with the Eclipse environment using the specified levels before moving to the updated levels.

Table 2–1. Software Component levels

Components	Vendor
Eclipse IDE for Java EE Developers 4.7.0	Eclipse Foundation
Eclipse C/C++ Development Tools 9.2.1	Eclipse Foundation
Log Viewer 0.9.8.8	Eclipse Market Place
EGit 4.4.0	Eclipse Foundation
Unisys Composite Application Development Environment (CA) 4.7.0 (includes COBOL editor)	Unisys website (www.unisys.com)
JBoss Enterprise Application Platform for ClearPath OS 2200 4.3A	Unisys website (www.unisys.com)
JBoss Enterprise Application Platform 4.3	Unisys website (www.unisys.com)
CIFS 8R4	Unisys website (www.unisys.com)
Photran (Fortran Editor) 9.1.0	Eclipse Foundation
Scintilla wScite 3.3.4	Scintilla (www.scintilla.org)

2.2. PC Hardware and Software Requirements

The following hardware system configuration and software requirements for using Eclipse software are recommended:

- Hardware
  - Minimum 2 GB RAM
  - Processor speed of at least 2 GHz
  - Minimum 20 GB mass storage
  - Minimum resolution of 1024 x 768
- Software
  - OS 2200 Eclipse 4.7.0 is qualified on the following operating systems.

Operating System	OS Architecture	Java/Eclipse Architecture	J2SDK
Windows 7/8/8.1/10	32	32	J2SE8
Windows 7/8/8.1/10	64	32/64	J2SE8

  - An unzip utility (this document assumes that WinZip is available)

Microsoft Windows XP Operating System will be discontinued by Microsoft starting April 2014. As a result, Unisys will no longer support Windows XP. Unisys therefore strongly recommends you to use a qualified version of Windows Operating System in order to use ClearPath OS 2200 IDE for Eclipse.

Note: *Unless you have an explicit written contractual agreement with Unisys, Microsoft Windows XP will not be supported beyond April 2014. You must contact Unisys Support if you wish to enter into an agreement with Unisys to get the support for Windows XP.*

2.3. Download Installation Files

You can download the Eclipse IDE software and the related documentation from the www.unisys.com site. The Eclipse IDE software is available as a ZIP file.

2.3.1. Download Eclipse IDE Installation file (.zip format)

Perform the following steps to download Eclipse IDE installation file in the zipped format.

1. Using Internet Explorer, navigate to the Unisys website at

   www.unisys.com

2. From the **Home** page, point to **Offerings** > **High-End Servers**, and then click **ClearPath Forward Systems**.

The ClearPath Forward Systems page is displayed.

3. Go to **ClearPath OS 2200 Software** and click **ClearPath OS 2200 IDE for Eclipse**.

   The ClearPath OS 2200 IDE for Eclipse page is displayed.

4. Go to **Software Download** and click **Download the software now**.

   The ClearPath OS 2200 IDE for Eclipse™ Software Download page is displayed.

5. Enter your First name, Last name, Job Title, Company, and Email Address.

6. Select the **I agree to the terms of the software agreement** check box and click **Submit**.

   The software download email is sent to the email address specified in Step 5.

7. In the software download email, do any one of the following:

   - Click **All in One Download 4.7.0(32 Bit)** for 32-bit system.
   - Click **All in One Download 4.7.0(64 Bit)** for 64-bit system.

   The **eclipse-2200-4-7-0-<YYMMDD>.zip or eclipse-2200-4-7-0-<YYMMDD>-X64.zip** folder is downloaded, where YYMMDD is the build version representing year, month, and date.

   Note: *Do not delete the software download mail as it is required while downloading other relevant files.*

   You may download the following documentation files to your local system:

   - Read me file
   - Quick Start Guide
   - Installation and Configuration Guide

   To download the documents, do the following:

   a. Right-click the link **Read me file** and select **Save Target as**.

      The **Save As** window appears.

   b. Navigate to the location where you want to save the document and click **Save**.

      The *Readme.txt* file is now available in the selected location.

      Note: *Follow the same steps to download the other documents.*

8. Double-click **Quick_Start.rtf** on your local system.

   The *Quick Start Guide* opens.

9. Download the Eclipse 2200 debug library through the link provided in the *Quick Start Guide*.

   The folder should now contain the following files:

   - Readme.txt (*Read me file*)
   - Eclipse-2200-Installation-and-Configuration-Guide.pdf
   - Eclipse-2200-BestPractices-Limitations.pdf

- Eclipse-2200-Internalization-Support.pdf
- Quick_Start.rtf (*Quick Start Guide*)
- Third_Party_License_and_Attribution.pdf
- Eclipse2200PadsLib46.zip
- 32-bit/eclipse-2200-all-in-one-4-7-0-V.zip
- 64-bit/eclipse-2200-all-in-one-4-7-0-V-X64.zip
- eclipse-2200-ca-4-7-0-V-updatesite.jar

  where V represents the Interim Correction (IC) release number.

Notes:

- *Be sure to read the Readme.txt file for comments and instructions that occurred after this document was prepared.*
- *To permanently change the values or to configure the Java version required for Eclipse, see Appendix E, "Configuring Eclipse.ini File."*

2.3.2. Download and Install J2SE

Perform the following steps to download and install J2SE:

1. Using Internet Explorer, go to the following website:

   http://www.oracle.com/index.html

2. Under **Software Downloads**, click **Java for Developers.**

   The **Downloads** tab is displayed.

3. Under **JDK,** click **Download**.

4. Follow the instructions to download and install the latest version of JDK.

   Note: *Alternatively, under* ***Previous Releases****, click* ***Java SE 8*** *and follow the download instructions.*

5. Verify the installation by opening a command prompt window and entering the following command:

   ```
   java-version
   ```

Notes:

- *Refer to the Windows Installation (IFTW) and Java Update FAQ at the Oracle-Sun Microsystems website for more information.*
- *Refer to Appendix B, "Setting Up Workstation Environment Variables" and Appendix E, "Configuring Eclipse.ini File" to set up environment variables and configure the Java version required for Eclipse, respectively.*

2.4. Install the PADS Library

Perform the following steps to install the PADS library, eclipse2200*pads$lib46, on the OS 2200 from a zip file:

1. Right-click the Eclipse2200PadsLib46.zip folder, and then click **WinZip > Extract to here**.

   The Eclipse2200 folder is created containing the following two folders:

   - **install46**

     This file contains the elements necessary to perform a SOLAR install of the pads$lib46 file, provided the user has the proper privileges to do a SOLAR install.

   - **pads$lib46**

2. Map a drive to *sys-id*\os2200.

   where *sys-id* is the IP address or computer name of the target OS 2200 system.

3. Copy the folder Eclipse2200 to this drive.

   Note: *The copied folder is private, with the owner being the account under which the drive was mapped to the OS 2200.*

4. Establish a command session on the target OS 2200 and enter the following ECL command:

   ```
   @chg,p eclipse2200*pads$lib46.
   ```

5. On a fundamental security system, it is also advisable to make the library file read-only, using the following ECL command:

   ```
   @chg,v eclipse2200*pads$lib46.
   ```

   Note: *To use the debugging feature in this level of Eclipse OS 2200, use the* ***eclipse2200*pads$lib46.debugging*** *library. It is recommended to delete the previous level of the debugging library to avoid any compatibility issues.*

2.5. Product Support

For more information, visit the Unisys Product Support website at:

www.support.unisys.com.

Section 3
Install and Start OS 2200 IDE for Eclipse

Unisys provides both a prepackaged all-in-one version of OS 2200 IDE for Eclipse and the Eclipse OS 2200 Composite Application (CA) feature. Consider how you are currently using Eclipse software when selecting one for installation. If you currently do not have Eclipse software installed, or if you wish to install multiple versions of Eclipse software on your workstation, Unisys recommends installing the all-in-one package. If you already have Eclipse 4.7.0 software installed and would want to extend it with the Unisys CA capabilities, install the Unisys CA feature.

3.1. Install Eclipse OS 2200 All-In-One

Unisys prepackages and tests Eclipse software and a set of Eclipse plug-ins that are appropriate for developing J2EE Java composite applications. Unisys recommends that you start with this Eclipse environment and add other plug-ins to meet your specific needs.

The Eclipse OS 2200 all-in-one package contains the following components.

Table 3–1. Eclipse OS 2200 Components

Components	Vendor
Eclipse IDE for Java EE Developers 4.7.0	Eclipse Foundation
C/C++ Development Tools (CDT) 9.2.1	Eclipse Foundation
Unisys Composite Application (CA) 4.7.0	Unisys Corporation

Notes:

- *SQL Explorer is no longer included in the all-in-one software package. Data Explorer, which is part of the Data Tools Platform (DTP), is included.*
- *The Eclipse CDT does not include a C compiler. If you wish to compile C/C++ projects on your workstation instead of using an OS 2200 C compiler, you must install a compiler (refer to www.eclipse.org/cdt).*

3.1.1. Install Eclipse OS 2200 All-In-One Package from Zip Folder

Perform the following steps to install the Eclipse OS 2200 all-in-one package using the zip folder:

1. Extract the files from the **eclipse-2200-4-7-0-<yymmdd>.zip** folder where yymmdd is the build version representing year, month, and date.

   This extraction creates C:\eclipse-2200-4-7-0-<yymmdd> and several subfolders.

2. Create a shortcut on the desktop.

   Notes:

   - *To increase JVM memory use the **–Xmx** parameter on the Eclipse command line, as in the following example:*

     ```
     eclipse.exe -vmargs –Xms256m –Xmx512m –Xmn64m
     ```

   - *If there is not enough heap memory for JVM, the debugger might not function. If you have 2 GB of RAM on your workstation, you might want to increase the JVM memory by using the –Xmx parameter on the command line. For example:*

     ```
     eclipse.exe -vmargs -Xms256m -Xmx512m -Xmn64m
     ```

   - *To permanently change the values, see Appendix E, "Configuring Eclipse.ini File."*

3.2. Start Eclipse IDE

Perform the following steps to start Eclipse IDE:

1. To launch the Eclipse IDE, double-click eclipse.exe shortcut on your desktop.

   A dialog box that prompts for a location to hold the workspace folder is displayed.

   Note: *If a Java Runtime Environment is not installed, Eclipse IDE displays an error message and does not start. To install JDK and JRE, see Section 1.5.*

2. Use the generated path that is displayed or select another path, and click **OK**.

   The first time you launch Eclipse IDE a **Unisys License Agreement** dialog box appears.

3. Click **OK** to accept the license agreement.

   The **Welcome** window appears.

4. Click **Workbench** icon in the top right-hand corner to display the Eclipse workbench window.

   The Workbench window appears.

Note: *When Eclipse is running for a very long period of time, it sometimes does not handle the files and projects properly. Double-clicking on eclipse-clean.bat cleans all the metadata related to Eclipse plugin and opens Eclipse in clean mode.*

3.3. Install the Eclipse OS 2200 CA Feature

If you have installed the Eclipse OS 2200 all-in-one package, do not install the Eclipse OS 2200 Composite Application (CA) feature; instead, continue with steps in the Section, "Set the Java Compliance Level."

The Eclipse OS 2200 Composite Application (CA) feature contains Eclipse plug-ins that are needed to work with OS 2200 environment. You can install the CA feature in Eclipse Java EE.

The following table lists the composite application components:

Table 3–2. Composite Application Components

CA Components	Description
TDE	Unisys Traditional Development Environment
Telnet	Telnet plug-in for ClearPath OS 2200, ClearPath MCP, UNIX, Linux, and other Telnet-compliant systems
Eclipse COBOL	Version of Eclipse COBOL plug-in feature modified by Unisys
JAI	Java Application Integrator, which simplifies the use of TIP/HVTIP annotations and the DMS and BIS resource adapters by automating the generation of connection code
COBOL Language Def.	Eclipse COBOL feature extended for OS 2200 COBOL Dialect
PLUS Editor	Editor for OS 2200 PLUS programs

Note: *The CA feature installation would not be successful if the latest versions of CDT are not available in Eclipse. Update your Eclipse with the latest versions of CDT, JST, and WST from the following Eclipse website:www.eclipse.org*

3.3.1. Update Site Installation

Perform the following steps to install CA feature from the Unisys remote site:

1. In the software download email, click **CA Package 4-7-0** to download .jar file.
2. On Eclipse **Help** menu, click **Install NewSoftware**.

   The installation wizard appears.
3. Click the **Available Software Sites** link.

   The **Preference** dialog box appears with the list of updated sites.
4. Clear all check boxes and click **OK**.

   Note: *The Unisys CA feature will fail to install if the checked sites fail to update the Eclipse. It is, therefore, recommended to clear all the checkboxes listing the sites in* ***Available Software Sites***.
5. Click **Add**.

The **Add Repository** dialog box appears.

6. Type a name for the remote site.

   For example, **Unisys CA update site**.

7. Click **Archive** and specify the location of the .jar file downloaded in Step 1.
8. Click **OK**.

   The Unisys CA update site is selected in the **Work with** list.

9. Ensure the following check boxes are clear:
   - **Show only the latest version of the available software**

     The Unisys CA update site by default has the latest version of the software.
   - **Group items by category**

     Helps identify the features being installed.
   - **Contact all update sites during install to find required software**

     For details, see Note in Step 4.
10. Click **Select All**.

    The listed features are selected.

11. Click **Next**.

    The Install Details window appears.

12. Click **Next**.

    The Review Licenses window appears.

13. Read the license information and click **I accept the terms of the license agreement**.
14. Click **Finish**.

    The **Installing Software** progress bar appears.

15. The Security Warning window appears. Click **OK** to continue installation.

    The Software Updates window appears.

16. Click **Restart Now**.

    The Eclipse IDE restarts. This completes the installation.

17. To verify the Eclipse IDE installation, from **Window** menu, point to **Open Perspective**, and click **Other**.

    The Open Perspective page appears.

18. Select **OS 2200** and click **OK**.

    The OS 2200 perspective is loaded on Eclipse.

3.4. Set the Java Compliance Level

The Java compliance level for your Eclipse projects must match the level of Java running on your OS 2200 target partition. By default, Eclipse 4.7.0 software generates projects with a compliance level of Java 1.8.

If you want Eclipse 4.7.0 to be compatible with Java level on the ClearPath OS 2200 JProcessor, perform the following steps:

1. From Eclipse Window menu, click **Preferences**.

   The **Preferences** dialog box appears.

2. In the left pane, expand **Java** and select **Compiler**.
3. Under **JDK Compliance**, set **Compiler compliance level** to the version compatible with the Java level on the ClearPath OS 2200 JProcessor.
4. Click **Apply** and **OK**.

3.5. Access Libraries for Resource Adapters

Java projects that use resource adapters for OS 2200 assets must reference certain Java libraries (jar files) at compile time and run time, as specified in the Table 2-3. The CA feature provides wizards that reference the libraries automatically and assist the developer in generating code that uses resource adapters for OS 2200 assets.

Table 3–3. Java Libraries

Purpose	Libraries Needed
Tip/HVTIP resource adapter	OS2200.jar
BIS resource adapter	bisra.jar ra-util.jar
DMS resource adapter	dmsra.jar ra-util.jar
Building a project to generate DMS class files	classbuilder.jar

The CA feature includes the versions of the jar files that were current at release time. To use different library files, install them on the workstation and then indicate the location of the new file when using the CA feature.

Section 4
Update and Uninstall OS 2200 IDE for Eclipse

The Eclipse files on the Unisys Product Support website (www.unisys.com) are tested for proper operation and compatibility with one another. Before using other versions of these plug-ins or plug-ins obtained from other sources, you should test them thoroughly in your own Eclipse environment.

New releases of OS 2200 IDE for Eclipse are announced on the Unisys Product Support website. Corrections to the Unisys CA feature are distributed using the Eclipse Update Manager.

4.1. Manage the Eclipse OS 2200 CA Feature

Eclipse IDE lets you update, disable, enable, and uninstall individual Eclipse plug-in features, such as the Eclipse OS 2200 CA feature.

To read details on the topic:

1. On the **Help** menu, point to **Help Contents**, and click **Workbench User Guide**.

   The **Workbench User Guide** opens.
2. Expand **Tasks** and click **Updating and installing software**.

To get the latest updates for the CA feature

1. On the **Help** menu, click **Check for Updates**.

   The installation wizard appears.
2. Select the desired features, click **Next**, and follow the instructions.

   The most up-to-date levels of your currently installed Eclipse plug-ins are downloaded and installed.

4.2. Third-Party Plug-Ins

Third-party plug-ins are software products that are made available by individuals and organizations outside Unisys. The support for those products is provided by them.

Before installing any third-party plug-in, examine the release notes or other supporting documentation to see what changes, if any, you might have to make to your existing installation or configuration.

The procedure for downloading and installing updated plug-ins is generally the same as the procedure for the initial download and installation described in this guide. You should review those instructions before proceeding.

4.3. First Use of Plug-Ins

Eclipse software normally detects the presence of new or updated plug-ins with no special actions required on your part.

Occasionally, some plug-ins may require a one-time change in the Eclipse startup process to properly integrate the plug-in with the Eclipse environment. For example, you may have to start Eclipse IDE with the "-clean" option to reinitialize its internal memory. This kind of special initialization procedure should be included in the documentation of any plug-in that requires it.

4.4. Uninstall OS 2200 IDE for Eclipse

The following subsections illustrate the steps to repair, modify and uninstall the OS 2200 IDE for Eclipse installed using the zip format and the .exe format. You now have an upgraded version of the OS 2200 IDE for Eclipse.

4.4.1. Uninstall the OS 2200 IDE for Eclipse

To uninstall the Eclipse software installed using the .zip folder, perform the following steps:

To uninstall a specific level of Eclipse software:

1. Delete from your system the folder containing the code for the Eclipse level you want to uninstall.
2. Delete any short-cut references to that level of Eclipse software.

To uninstall a specific level of Eclipse software (.exe format), delete the folder where Eclipse is extracted.

Note: *Ensure Eclipse is not running during deletion.*

4.4.2. Uninstalling the Eclipse OS 2200 CA Feature

To uninstall only the Eclipse OS 2200 CA feature, use Eclipse Update Manager. For more information, refer to the Eclipse Help system.

Section 5
JVM Settings

You can configure Eclipse to enhance how it works for you while working with the large and complex elements to prevent memory errors.

This section describes how you can improve Eclipse performance by changing the heap size for Java Virtual Machine (JVM).

5.1. Java Heap Memory

When you start Eclipse, JVM gets memory from the operating system. JVM has a heap, the run-time data area from which memory for all class instances and arrays are allocated. The data area is created as you start JVM.

If Eclipse handles huge files and lots of objects are being created, it is recommended that you increase the heap size to its optimal to prevent memory errors. You can increase the Java heap memory based on the Eclipse need.

5.1.1. JVM Settings

The heap size in Eclipse cannot be changed dynamically. You need to provide the following Java heap size parameters while starting JVM:

- **-Xms<size>** sets the initial Java heap size.
- **-Xmx<size>** sets the maximum Java heap size.
- **-Xmn<size>** denotes new generation of Java Heap space.

On a 32-bit system, the default size of heap space in Eclipse is **-Xms=40M** and **-Xmx=256M**.

When Eclipse starts, the JVM heap space is equal to the initial Java heap size. As application progresses, the heap space is expanded to its maximum size to accommodate new objects. If there is no more memory left to create a new object in the Java heap, JVM throws Java.lang.OutOfMemoryError and the Eclipse application ends.

To reclaim memory back from dead objects, JVM runs the garbage collector periodically, a process that removes dead objects from the Java heap space and returns back to the Java heap.

A Java heap is divided in the following three categories:

- **New Generation:** Part of the Java heap memory where newly created objects are stored.
- **Old or Tenured Generation:** During the course of Eclipse application, many objects got created and died, but objects that remained live moved to this category.
- **Perm Space:** Part of the Java heap memory where JVM stores Meta data about classes and methods, String pool, and Class-level details.

If you are using a 32-bit operating system and have 2 GB of RAM on your workstation, you might want to increase the JVM memory by using the -Xmx parameter on the command line. For example:

```
eclipse.exe -vmargs -Xms256m -Xmx512m -Xmn64m
```

Notes:

- *To permanently change the values, see Appendix E, "Configuring Eclipse.ini File."*
- *On a 64-bit operating system, depending upon the RAM available, you can increase the maximum heap size to a larger value.*

Appendix A
Java Installation

A.1. Build Applications for an earlier version of Java Targets Using the Java 8.0 JDK

You can use Eclipse IDE with a Java 8.0 compiler and library to build applications for target systems that use an earlier version of Java, for example 1.5 runtime environment, by following these suggestions:

To set the compiler compliance level to 1.5, do the following:

1. On the Eclipse **Window** menu, click **Preferences**.

   The **Preferences** dialog box appears.
2. In the left pane, expand **Java** and select **Compiler**.
3. Set **Compiler Compliance Level** to **1.5** under **JDK Compliance**.
4. Select the **Use default compliance settings** check box.
5. Click **OK**.

Note: *Do not use any Java 8.0 classes or methods that are not supported in Java 1.5.*

The compiler compliance level causes the compiler to detect the use of Java 8.0 language constructs and ensures that the output class files are compatible with the Java 1.5 runtime environment. However, it does not detect the use of classes or methods that do not exist in the Java 1.5 environment.

Why You May Want to use an earlier version of Java

The OS 2200 runtime environment may use a different version of Java compared to Java 8 that is used by OS 2200 Eclipse IDE.

You can install multiple version of Java alongside the Java 8.0 JDK and an earlier version of Java libraries when building your OS 2200 Java applications. Doing so allows the Java compiler and the Eclipse IDE to ensure that your Java programs conform to the selected version of Java.

In the Eclipse IDE, you can switch between different versions of Java using the steps described "A.1. Build Applications for an earlier version of Java Targets Using the Java 8.0 JDK".

If you use another programming environment in addition to the Eclipse IDE, ensure to select the libraries and other environment parameters for the desired Java level using the methods required by your chosen programming tools.

A.2. Manage Windows PATH for Multiple Versions of Java

When you install multiple versions of Java or JDK, the installation procedure places copies of several Java run-time executable programs in the Windows\System32 directory, in addition to the copies that are in the standard Java installation directory. Normally, the latest version of Java is used as the default Java environment. If you want to change the default Java environment, see Appendix E, "Configuring Eclipse.ini File."

A.3. Manage Custom Installations of Java JRE

You can configure Eclipse IDE to use the correct version of Java when multiple versions of Java are installed on a machine.

To configure the correct version of JRE, perform the following:

1. On the Eclipse **Window** menu, click **Preferences**.

   The **Preferences** dialog box appears.

2. In the left pane, expand **Java** and select **Installed JREs**.

3. Click **Add**.

   The **Add JRE** dialog box appears.

4. Select **Standard VM** from list of installed JRE types and click **Next**.

5. To browse to the location of the JRE, click **Directory.**

   The Browse for Folder window appears.

6. Select the desired folder, and click **OK.**

   The **JRE Home** and **JRE Name** fields are populated with default values. You can change the **JRE Name** (optional).

   ***Note** : Alternatively, you can type the full path to the directory containing the JRE in the **JRE home directory** box.*

   For example, the default path for the Java 6 JRE is:

   C:\Program Files\Java\jre6.

   To set JRE parameters, enter the parameters in the **Default VM Arguments** field or click **Variables** to choose the parameters.

7. Click **Finish** to close the **Add JRE** dialog box.

8. Click **OK** to close the **Preferences** dialog box.

Note: *To allow the automatic switching to find your JREs, you must add these entries even if you already configured the same JRE with a different name. Alternatively, you can change the name of an already configured JRE to match the required name in the procedure.*

Appendix B
Workstation Environment Variables

Procedures for configuring components sometimes require you to either verify or set environment variables. This appendix describes how to accomplish this task.

Set the named Windows environment variables to the following settings.

Table B–1. Windows Environment Variables

Variable	Setting
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_*xx*
JBOSS_HOME	C:*folder*
JBOSS_DIST	C:*folder*

where *xx* represents the current update version of the JDK and *folder* is the directory in which JBoss EAP is installed.

Note: *If you did not install the J2SE and JBoss in the default locations, change the settings to the location where they are installed.*

Perform the following steps to set environment variables:

1. Right-click the **My Computer** icon on the desktop and click **Properties**.

   The **System Properties** dialog box appears.
2. Click the **Advanced** tab, and then click **Environment Variables**.

   The **Environment Variables** dialog box appears.
3. Click **New** under **System variables**.

   The **New System Variable** dialog box appears.
4. Type the name (for example, **JAVA_HOME**) in the **Variable Name** box.
5. Type the value (for example, **C:\Program Files\Java\jdk1.8.0_*xx***, where *xx* is the updated release number you have installed) in the **Variable Value** box.
6. Click **OK** to close the **New System Variable** dialog box.
7. Click **OK** to close the **Environment Variables** dialog box.
8. Click **OK** to close the **System Properties** dialog box.

Perform the same procedure for each environment variable.

Appendix C
OS 2200 Environment Requirements

The ClearPath OS 2200 environment has certain requirements for OS 2200 IDE for Eclipse. The minimum OS 2200 requirements are

- Telnet server
- CIFS/SMB server, where the developer has a user-id that enables remote access to the CIFS/SMB server. The developer's Telnet login must enable access to cifsut on the OS 2200.

The recommended OS 2200 requirements are

- CPComm
- 100-megabit connection

The following options are available for specifying the work file for an OS 2200 project:

- Use standard OS 2200 share (os2200).

  A share is mapped to the os2200 folder on the ClearPath OS 2200 system and the share name is "os2200" (case sensitive).
- Use a nonstandard name for os2200 share.

  A share is mapped to the os2200 folder, but it has a different name.
- Cannot use os2200 share.

  A share is mapped directly to the work file and the os2200 folder is not used.

Appendix D
Install and Configure XDoclet

Eclipse Web Tools Platform (WTP) uses XDoclet to build EJB2 projects. The EJB3 projects do not require XDoclet.

D.1. Installing XDoclet

Perform the following steps to install Xdoclet.

1. Using Internet Explorer, navigate to the Source Forge website at

   http://sourceforge.net/projects/xdoclet
2. Click **Browse All Files** link.

   The **Home** page containing links to **xjavadoc** and **xdoclet** is displayed.
3. Click **xdoclet**.

   The list of all the versions of the xdoclet is displayed.
4. Click **1.2.3** version of xdoclet.
5. From the list of file formats, click **xdoclet-bin-1.2.3.zip** .

   The **File Download** dialog box appears.
6. Click **Save**.

   The **Save As** dialog box appears.
7. Save the zip folder in the C: drive.
8. Extract the file to C: drive.

   The xdoclet-1.2.3 folder is created.

D.2. Configuring XDoclet

After all components are installed, you must configure XDoclet in Web Tools Platform (WTP) and configure WTP for a specific application server. Perform these procedures each time you create a new Eclipse workspace.

D.2.1. Configure XDoclet in WTP

XDoclet is required by the Eclipse Web Tools Platform (WTP) feature for building EJB2 projects (it is not required for EJB3 projects).

Perform the following steps to configure XDoclet in WTP:

1. On the Eclipse **Window** menu, click **Preferences.**

   The **Preferences** dialog box appears.

2. In the left pane, expand Java EE, and then click **XDoclet**.
3. Type or browse to the location of the XDoclet installation folder in the **XDoclet Home** box.
4. Select the XDoclet version number in the **Version** list.

   Note: *The version number must match the version of the xdoclet downloaded. Selecting the right version clears the missing library error message in the dialog box.*

5. Expand Xdoclet in the left pane to view **ejbdoclet** and **webdoclet**.
6. Click **ejbdoclet** and select **JBoss** in the right pane.
7. Click **Apply**.
8. Click **webdoclet** and select **JBoss** in the right pane.
9. Click **Apply**, and then click **OK.**

Note: *If you do not wish to configure the application server, click **OK** to close the dialog box.*

Appendix E
Configuring Eclipse.ini file

To configure the Java version to be used by Eclipse, open eclipse.ini file and include –vm with the value of Java as shown in the example below. For more information on the variables in Eclipse.ini file, refer to http://wiki.eclipse.org/Eclipse.ini.

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.200.v20160318-1642.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_1.1.400.v20160518-1444
-product
org.eclipse.epp.package.jee.product
--launcher.defaultAction
openFile
-showsplash
org.eclipse.platform
--launcher.defaultAction
openFile
-vm
C:\Program Files (x86)\Java\jdk1.8.0_31\bin\javaw.exe
--launcher.appendVmargs
-vmargs
-Dosgi.requiredJavaVersion=1.8
-XX:+UseG1GC
-XX:+UseStringDeduplication
-Dosgi.requiredJavaVersion=1.8
-Xms256m
-Xmx512m
-Xmx1024m
```

Notes:

- *The above example displays the Java configuration for 32-bit Eclipse on a 64-bit operating system.*
- *To modify the heap size, change the values of –Xms and –Xmx.*

4729 2107-012