UNISYS

ClearPath Application Development Solutions

ClearPath OS 2200 IDE
for Eclipse™

Installation Guide

ClearPath OS 2200 Release 16.0

February 2015

4729 2107–010

Contents

Figures

Tables

Section 1
Introduction to the ClearPath OS 2200 IDE for Eclipse Software

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 www.eclipse.org/platform 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) 19036910. 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/19036910

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.

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 4.3.2	Eclipse Foundation
Eclipse C/C++ Development Tools 8.4.0	Eclipse Foundation
Log Viewer 0.9.88	Eclipse Market Place
EGit 3.4.1	Eclipse Foundation
Eclipse Java 8 Support	Eclipse Foundation
Unisys Composite Application Development Environment (CA) 4.3.2 (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 7R4	Unisys website (www.unisys.com)
Photran (Fortran Editor) 8.2	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 4.3.2 is qualified on the following operating systems.

Operating System	OS Architecture	Java/Eclipse Architecture	J2SDK
Windows 7	32	32	J2SE7
Windows 7	64	32/64	J2SE7

  - OS 2200 Eclipse 3.7.2 is supported on the following operating systems.

Operating System	OS Architecture	Java/Eclipse Architecture	J2SDK
Windows 7	32/64	32	J2SE6

  - 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 click **ClearPath Systems**.

   The ClearPath 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 login page is displayed.

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

   The software download page is sent to your inbox.

6. On the software download page, click **eclipse-2200-4-3-2-<YYMMDD>** where YYMMDD is the build version representing year, month, and day.

   The eclipse-2200-4-3-2-<YYMMDD> page is displayed. Clicking **eclipse-2200-4-3-2-<YYMMDD>** downloads the software.

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

   - **Read me file**
   - **Quick Start Guide**
   - **Installation and Configuration Guide**
   - **Application Development 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 is displayed.

   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.

   ***Note**: The Quick Start Guide provides links to the software and installation packages.*

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*)
   – *Installation and Configuration Guide* – 4729 2107-010
   – *Application Development Guide* – 3839 3831-002
   – Quick_Start.rtf (*Quick Start Guide*)
   – Eclipse2200PadsLib37.zip
   – eclipse-2200-all-in-one-4-3-2.zip (eclipse-2200-all-in-one-4-3-2-X.zip)
   – Eclipse-2200-ca-4-3-2-updatesite.jar (Eclipse-2200-ca-4-3-2-X-updatesite.jar)

     where X represents the IC (Interim Correction) 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 7** and follow the instructions to download.*

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

   `java-version`

   A screen similar to the following must be displayed.

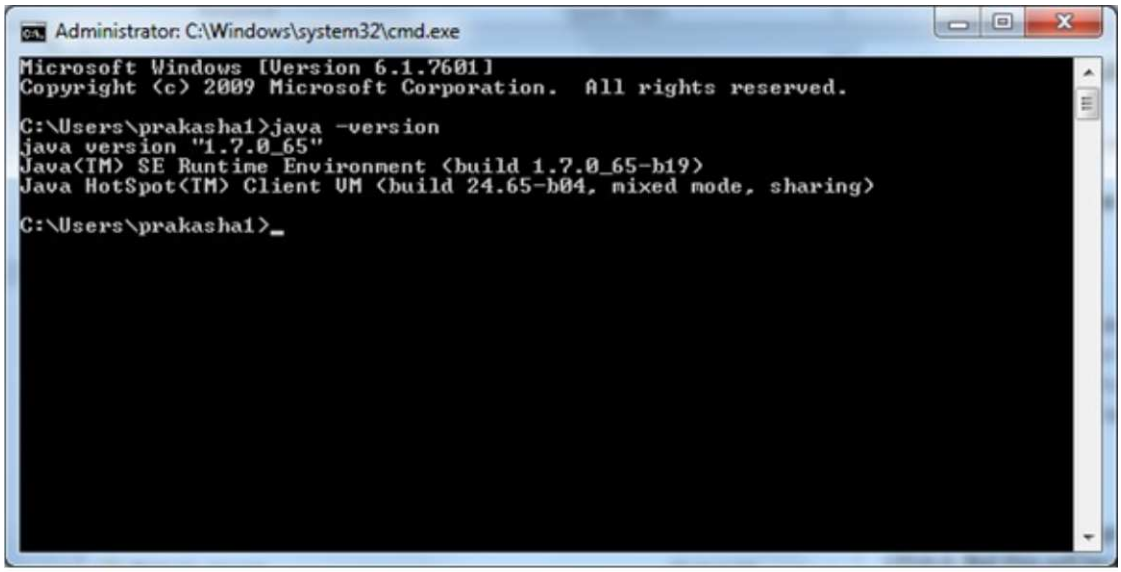

Figure 2–1. Java Installation Verification

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$lib37., on the OS 2200 from a zip file:

1. Right-click winzip > Extract to here > Eclipse2200PadsLib37.zip

   The Eclipse2200 folder is created containing the following two folders:

   – **install**

     This file contains the elements necessary to perform a SOLAR install of the pads$lib37 file, provided the user has the proper privileges to do a SOLAR install.

   – **pads$lib37**

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$lib37.
   ```

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$lib37.
   ```

   ***Note:** To use the debugging feature in this level of Eclipse OS 2200, use the eclipse2200*pads$lib37.debugging library. It is recommended to delete the previous level of the debugging library to avoid any compatibility issues.*

2.5. Install XDoclet

XDoclet is required by the Eclipse Web Tools Platform (WTP) feature for building EJB2 projects (It is not required for EJB3 projects). You can skip this installation if you want to develop EJB3 projects.

For information on installation and configuration of Xdoclet, see Appendix D.

2.6. Install JBoss Enterprise Application Platform

To install JBoss Enterprise Application Platform for development and testing purposes contact your Unisys representative.

2.7. 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 4.3.2 software installed and wish 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 2–1. Eclipse OS 2200 Components

Components	Vendor
Eclipse IDE for Java EE Developers 4.3.2	Eclipse Foundation
C/C++ Development Tools (CDT) 8.4.0	Eclipse Foundation
Unisys Composite Application (CA) 4.3.2	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 in the eclipse-2200-all-in-one-4-3-2.zip.

   This extraction creates C:\ eclipse-2200-all-in-one-4-3-2 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 2.4, "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.

Table 2-2 lists the composite application components.

Table 2–2. 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

Use the Eclipse Update Manager (refer to the Eclipse help system) to install the CA feature from either

- The Unisys remote site (recommended)

  Install the CA feature from the Unisys remote site so that you can easily update the CA feature whenever Unisys releases CA feature corrections.

- An archived site

  Use an archived site when you do not have or want direct client workstation access to the Internet, or if you wish to control the release of stability updates to workstations.

The following subsections provide the procedures for installing from each site.

3.3.1. Remote Site Installation

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

1. On Eclipse **Help** menu, click **Install New Software**.

   The installation wizard appears.

2. Click **Available Software Sites** link.

   The Available Software Sites window is displayed with the list of update sites.

3. Uncheck all.

   ***Note**: The Unisys CA feature will fail to install if the checked sites fail to update the Eclipse. It is, therefore, recommended to uncheck all the sites listed in **Available Software Sites**.*

4. Click **OK**.

   The installation wizard appears.

5. Click **Add**.

   The **Add Repository** dialog box appears.

6. Type a name for the remote site.

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

7. Based on the release level, set the URL to one of the following:

   ftp://ftp.support.unisys.com/pub/2200/IDE/Eclipse-2200-4-3-2/

   Note: The preceding link opens two folders, 32 and 64-bit. Select the folder based on your requirement.

8. Click **OK**.

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

9. Ensure the following fields 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 3.

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 dialog box 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.

3.3.2. Archived Site Installation

Perform the following steps to install CA feature from an archived site:

1. On the Eclipse **Help** menu, click **Install New Software**.

   The installation wizard appears.
2. Click **Add**.

   The **Add Repository** dialog box appears.
3. Click **Archive**.
4. Navigate to and select the eclipse-2200-ca.jar file in the C: drive (or the server location where you placed it), and click **Open**.

   The **Add Repository** dialog box appears.
5. Type a name for the archived site.

   For example, **Unisys CA update site**
6. Click **OK**.

   The Unisys CA update site is selected in the **Work with** list.
7. Ensure the following fields are unchecked:
   - **Show only the latest version of the available software**

     The Unisys CA update site will by default have 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**

     See Note under Step 3 in “Remote Site Installation”.

8. Click **Select All**.

   The listed features are selected.

9. Click **Next**.

   The Install Details window appears.

10. Click **Next**.

    The Review Licenses window appears.

11. Read the license information and click **I accept the terms of the license agreement**.

12. Click **Finish**.

    The **Installing Software** progress bar dialog box appears.

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

    The Software Updates window appears.

14. Click **Restart Now**.

    The Eclipse IDE restarts. This completes the installation.

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

    The **Open Perspective** page appears.

16. 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 4.3.2 software generates projects with a compliance level of Java 1.7. This level matches the JVM 1.7 released with ClearPath OS 2200 Release 16.0. If you wish Eclipse 4.3.2 to be compatible with the Java level on the Clearpath OS 2200 JProcessor, you can set the compliance level to 1.7.

Perform the following steps to set the compliance level:

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 1.7, if it is set to a different level.

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 2–3. 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.

3.6. Configure Eclipse Web Tools Projects

After all components are installed, you must configure WTP for a specific application server. Perform this procedure each time you create a new Eclipse workspace.

3.6.1. Configure the Application Server in WTP

To configure JBoss Enterprise Application Platform in WTP:

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

   The **Preferences** dialog box appears.

2. Expand **Server** in the left pane, select **Runtime Environments**, and click **Add**.

   The **New Server Runtime** wizard appears.

3. Select the appropriate JBoss 4.3 server and click **Next**.

   Refer to the *JBoss® Enterprise Application Platform for ClearPath® OS 2200 Installation, Administration, and Programming Guide* for more information.

4. Select **Default JRE** in the **JRE** list.

5. Type or browse to the location of the application server in the **Application Server Directory** box.

6. Click **Finish**.
7. Click **OK** to close the dialog box.

3.7. Use the Application Development Guide

The *ClearPath OS 2200 IDE for Eclipse™ Application Development Guide for Java EE Projects* provides step-by-step instructions for building J2EE components using Eclipse IDE. The guide is in the eclipse-2200-appl-dev-guide.zip file, which you downloaded from the Unisys.com website. Extract the guide to a folder on your workstation and use the procedures in the guide for the desired components.

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 opens.
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.

In 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 Java 1.5 Targets Using the Java 7.0 JDK

You can use Eclipse IDE with a Java 7.0 compiler and library to build applications for target systems that use the Java 1.5 runtime environment if you follow these suggestions:

- Set the compiler compliance level to 1.5, as follows:
  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**.
- Ensure not to use any Java 7.0 classes or methods that are not supported in Java 1.5.

The compiler compliance level causes the compiler to detect the use of Java 7.0 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 Install Java 1.5 JRE

The OS 2200 runtime environment currently supports Java version 1.5.

As long as your Java code conforms to the Java 1.5 standards and does not include references to any classes or methods that exist only in Java 7.0, you can develop your Java programs in the Eclipse IDE with only the Java 7.0 JDK installed and deploy your compiled classes to the OS 2200 Java environment without difficulty.

If you wish, you can install the Java 1.5 JRE alongside the Java 7.0 JDK and use the Java 1.5 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 Java 1.5 standards.

In the Eclipse IDE, you can switch between the Java 1.5 and Java 7.0 libraries simply by clicking a button on the Eclipse toolbar.

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. Because the Windows\System32 directory is at the head of the Windows PATH environment variable by default, it is usually not necessary to add any additional entries to the PATH to be sure that the basic Java executables are located automatically. Therefore, to ensure that the correct verion of Java is configured, see Appendix E, “Configuring Eclipse.ini File.”

A.3. Manage Custom Installations of Java JRE

If you used Oracle-Sun standard tools to install the Java JDKs and Java JREs, the ability to switch between Java 1.5 and Java 7.0 libraries for compilation is available with no further action on your part.

If you used nonstandard installation methods, or if you installed multiple levels of the Java 1.5 or Java 7.0 JRE and do not want always to use the latest version of one or both of these, you must manually add some configuration information to your Eclipse environment.

To configure a specific version of the Java 7.0 or Java 1.5 JRE on your system

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.

If you use these specific names, Eclipse IDE can automatically locate these JRE instances when you click **Java 1.5** or **Java 1.6** on the toolbar to select the appropriate run-time libraries.

***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.*

If you have additional JREs that you would like to select at compile time, enter them in this same **Installed JREs** setup, assigning different names to them. You can then switch among them manually by checking the box adjacent to the assigned name of the JRE.

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\jdk1.7.0_*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 (refer to Section 1).

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\jdk1.7.0_*xx***, where *xx* is the release update number you 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.*

   For information on how to configure the application server in WTP, refer to the Section 2.6.1, “Configure the Application Server in WTP”.

Appendix E
Configuring Eclipse.ini file

To configure the version of Java 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_1.3.0.v20130327-1440.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_1.1.200.v20140116-
2212
-product
org.eclipse.epp.package.jee.product
--launcher.defaultAction
openFile
--launcher.XXMaxPermSize
256M
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
--launcher.defaultAction
openFile
--launcher.appendVmargs
-vm
C:\Java\JDK\1.7\bin\javaw.exe
-vmargs
-Dosgi.requiredJavaVersion=1.6
-Xms256m
-Xmx512m
-XX:MaxPermSize=512m
-XX:+UseConcMarkSweepGC
-XX:+UseParNewGC
-XX:CMSInitiatingOccupancyFraction=20
-XX:+CMSPermGenSweepingEnabled
-XX:+CMSClassUnloadingEnabled
-Xverify:none
```

Note: *To modify the heap size, change the values of –Xms and –Xmx.*

4729 2107–010