# Eclipse Project Release Notes

Release 3.3.1.1

Last revised October 22, 2007

**This software is OSI Certified Open Source Software.**

**OSI Certified is a certification mark of the Open Source Initiative.**

[1. Target Operating Environments](#_gjdgxs)

[2. Compatibility with Previous Releases](#_30j0zll)

[3. Known Issues](#_1fob9te)

[4. Running Eclipse](#_44sinio)

[5. Upgrading a Workspace from a Previous Release](#_3j2qqm3)

[6. Interoperability with Previous Releases](#_1y810tw)

## 1. Target Operating Environments

In order to remain current, each Eclipse release targets reasonably current operating environments.

Most of the Eclipse SDK is "pure" Java code and has no direct dependence on the underlying operating system. The chief dependence is therefore on the Java Platform itself. Portions of the Eclipse SDK (including the RCP base, SWT, OSGi and JDT core plug-ins) are targeted to specific classes of operating environments, requiring their source code to only reference facilities available in particular class libraries (e.g. J2ME Foundation 1.0, J2SE 1.3 and 1.4, etc.).

In general, the 3.3 release of the Eclipse Project is developed on a mix of Java 1.4 and Java5 VMs. As such, the Eclipse Project SDK as a whole is targeted at both 1.4 and Java5 VMs, with full functionality available for 1.4 level development everywhere, and new Java5 specific capabilities available when running on a Java5 VM. Similarly, in cases where support has been added for Java6 specific features (e.g. JSR-199, JSR-269, etc.) Java6 VMs are required.

[Appendix 1](#_2xcytpi) contains a table that indicates the class library level required for each plug-in.

There are many different implementations of the Java Platform running atop a variety of operating systems. We focus Eclipse SDK testing on a handful of popular combinations of operating system and Java Platform; these are our *reference platforms*. Eclipse undoubtedly runs fine in many operating environments beyond the reference platforms we test, including those using Java6 VMs. However, since we do not systematically test them we cannot vouch for them. Problems encountered when running Eclipse on a non-reference platform that cannot be recreated on any reference platform will be given lower priority than problems with running Eclipse on a reference platform.

The Eclipse SDK 3.3 is tested and validated on the following reference platforms:

| **Reference Platforms** |
| --- |
| **Microsoft Windows Vista, x86-32, Win32** running (any of):   * Sun Java 2 Standard Edition 5.0 Update 11 for Microsoft Windows * IBM 32-bit SDK for Windows, Java 2 Technology Edition 5.0, SR4 (see caveat below) * BEA JRockit 5.0, for Microsoft Windows |
| **Microsoft Windows XP, x86-32, Win32** running (any of):   * Sun Java 2 Standard Edition 5.0 Update 11 for Microsoft Windows * IBM 32-bit SDK for Windows, Java 2 Technology Edition 5.0, SR4 * BEA JRockit 5.0, for Microsoft Windows * Sun Java 2 Standard Edition 1.4.2\_14 for Microsoft Windows * IBM 32-bit SDK for Windows, Java 2 Technology Edition 1.4.2 SR7 * BEA JRockit 1.4.2, for Microsoft Windows |
| **Red Hat Enterprise Linux 5.0, x86-32, GTK** running (any of):   * Sun Java 2 Standard Edition 5.0 Update 11 for Linux x86 * IBM 32-bit SDK for Linux on Intel architecture, Java 2 Technology Edition 5.0, SR4 * BEA JRockit 5.0, for Linux x86 * Sun Java 2 Standard Edition 1.4.2\_13 for Linux x86 * IBM 32-bit SDK for Linux on Intel architecture, Java 2 Technology Edition 1.4.2 SR7 * BEA JRockit 1.4.2, for Linux x86 |
| **SUSE Linux Enterprise Server 10, x86-32, GTK** running (any of):   * Sun Java 2 Standard Edition 5.0 Update 11 for Linux x86 * IBM 32-bit SDK for Linux on Intel architecture, Java 2 Technology Edition 5.0, SR4 |
| **Red Hat Enterprise Linux 4.0 update 2, x86-64, GTK** running:   * Sun Java 2 Standard Edition 5.0 Update 11 for Linux x86\_64 |
| **Sun Solaris 10, SPARC, GTK** running:   * Sun Java 2 Standard Edition 5.0 Update 11 for Solaris SPARC |
| **IBM AIX 5.3, Power, Motif 2.1** running:   * IBM 32-bit SDK, Java 2 Technology Edition 5.0, SR4 |
| **Red Hat Enterprise Linux 4.0 update 2, Power, GTK** running:   * IBM 32-bit SDK for Linux on pSeries architecture, Java 2 Technology Edition 1.4.2 service release 7 |
| **SUSE Linux Enterprise Server 10, Power, GTK** running:   * IBM 32-bit SDK for Linux on pSeries architecture, Java 2 Technology Edition 1.4.2 service release 7 |
| **Apple Mac OS X 10.4, Universal, Carbon** running:   * Apple Java 2 Platform Standard Edition (J2SE) 5, service release 4 for Tiger |

*Caveat: Using IBM 32-bit SDK for Windows, Java 2 Technology Edition 5.0, SR4 on Vista* Although we expect this to be fixed in IBM Java5 SR5, there is currently a conflict between the use of DirectDraw by the **AWT** libraries in IBM Java5 SR4 and the Windows Vista "Aero" theme. Although the Eclipse SDK itself does not use these libraries (and thus runs well on Windows Vista using that VM), other plug-ins that make use of the conflicting AWT capabilities may cause the advanced features of Aero to be disabled. As a workaround to avoid this problem, you can add -Dsun.java2d.noddraw=true to the VM arguments when launching Eclipse, which will prevent the AWT libraries from using DirectDraw.

Because Java 1.4.2 and Java5 based platforms are used for most Eclipse development, those platforms are listed here. Although there are teams doing some Java 6 based development we have not included specific Java6 VMs, since they have not yet received the general level of testing we require. *We expect that Eclipse will work fine on other current Java VMs running on window systems supported by SWT, but can not flag these as reference platforms without significant community support for testing them.*

Similarly, although untested, the Eclipse SDK should work fine on other OSes that support the same window system. For Win32: NT, 2000, and Server 2003; SWT HTML viewer requires Internet Explorer 5 (or higher). For GTK on other Linux systems: version 2.2.1 of the GTK+ widget toolkit and associated libraries (GLib, Pango); SWT HTML viewer requires Mozilla 1.4GTK2. For Motif on Linux systems: Open Motif 2.1 (included); SWT HTML viewer requires Mozilla 1.4GTK2.

SWT is also supported on the QNX Neutrino operating system, x86 processor, Photon window system, and IBM J9 VM version 2.0. Eclipse 3.3 on Windows or Linux can be used to cross-develop QNX applications. (Eclipse 3.3 is unavailable on QNX because there is currently no 1.5 J2SE for QNX.)

#### Internationalization

The Eclipse SDK is designed as the basis for internationalized products. The user interface elements provided by the Eclipse SDK components, including dialogs and error messages, are externalized. The English strings are provided as the default resource bundles.

Latin-1 locales are supported by the Eclipse SDK on all of the above operating environments; DBCS locales are supported by the Eclipse SDK on the Windows, GTK, and Motif window systems; BIDI locales are supported by the Eclipse SDK only on Windows operating environments.

The Eclipse SDK supports GB 18030 (level 1), the Chinese code page standard, on Windows XP and 2000, and Linux/GTK.

German and Japanese locales are tested.

#### BIDI support

SWT fully supports BIDI on Windows. On Linux GTK, SWT supports entering and displaying BIDI text. Within these limitations, the Eclipse SDK tools are BIDI enabled.

## 2. Compatibility with Previous Releases

### Compatibility of Release 3.3 with 3.2

Eclipse 3.3 is compatible with Eclipse 3.2.

**API Contract Compatibility:** Eclipse SDK 3.3 is upwards contract-compatible with Eclipse SDK 3.2 except in those areas noted in the  [*Eclipse 3.3 Plug-in Migration Guide*](http://dev.eclipse.org/viewcvs/index.cgi/%7Echeckout%7E/org.eclipse.platform.doc.isv/porting/eclipse_3_3_porting_guide.html) . Programs that use affected APIs and extension points will need to be ported to Eclipse SDK 3.3 APIs. Downward contract compatibility is not supported. There is no guarantee that compliance with Eclipse SDK 3.3 APIs would ensure compliance with Eclipse SDK 3.2 APIs. Refer to  [*Evolving Java-based APIs*](http://wiki.eclipse.org/index.php/Evolving_Java-based_APIs)  for a discussion of the kinds of API changes that maintain contract compatibility.

**Binary (plug-in) Compatibility:** Eclipse SDK 3.3 is upwards binary-compatible with Eclipse SDK 3.2 except in those areas noted in the  [*Eclipse 3.3 Plug-in Migration Guide*](http://dev.eclipse.org/viewcvs/index.cgi/%7Echeckout%7E/org.eclipse.platform.doc.isv/porting/eclipse_3_3_porting_guide.html) . Downward plug-in compatibility is not supported. Plug-ins for Eclipse SDK 3.3 will not be usable in Eclipse SDK 3.2. Refer to  [*Evolving Java-based APIs*](http://wiki.eclipse.org/index.php/Evolving_Java-based_APIs)  for a discussion of the kinds of API changes that maintain binary compatibility.

**Source Compatibility:** Eclipse SDK 3.3 is upwards source-compatible with Eclipse SDK 3.2 except in the areas noted in the  [*Eclipse 3.3 Plug-in Migration Guide*](http://dev.eclipse.org/viewcvs/index.cgi/%7Echeckout%7E/org.eclipse.platform.doc.isv/porting/eclipse_3_3_porting_guide.html) . This means that source files written to use Eclipse SDK 3.2 APIs might successfully compile and run against Eclipse SDK 3.3 APIs, although this is not guaranteed. Downward source compatibility is not supported. If source files use new Eclipse SDK APIs, they will not be usable with an earlier version of the Eclipse SDK.

**Workspace Compatibility:** Eclipse SDK 3.3 is upwards workspace-compatible with Eclipse SDK 3.2 unless noted. This means that workspaces and projects created with Eclipse SDK 3.2, 3.1 or 3.0 can be successfully opened by Eclipse SDK 3.3 and upgraded to a 3.3 workspace. This includes both hidden metadata, which is localized to a particular workspace, as well as metadata files found within a workspace project (e.g., the .project file), which may propagate between workspaces via file copying or team repositories. Individual plug-ins developed for Eclipse SDK 3.3 should provide similar upwards compatibility for their hidden and visible workspace metadata created by earlier versions; 3.3 plug-in developers are responsible for ensuring that their plug-ins recognize 3.2, 3.1, 3.0, 2.1, and 2.0 metadata and process it appropriately. User interface session state may be discarded when a workspace is upgraded. Downward workspace compatibility is not supported. A workspace created (or opened) by a product based on Eclipse 3.3 will be unusable with a product based an earlier version of Eclipse. Visible metadata files created (or overwritten) by Eclipse 3.3 will generally be unusable with earlier versions of Eclipse.

**Non-compliant usage of API's**: All non-API methods and classes, and certainly everything in a package with "internal" in its name, are considered implementation details which may vary between operating environment and are subject to change without notice. Client plug-ins that directly depend on anything other than what is specified in the Eclipse SDK API are inherently unsupportable and receive no guarantees about compatibility within a single release much less with earlier releases. Refer to  [*How to Use the Eclipse API*](http://www.eclipse.org/articles/Article-API%20use/eclipse-api-usage-rules.html)  for information about how to write compliant plug-ins.

### Compatibility of Release 3.3 with 3.2, 3.1, 3.0, 2.1 and 2.0

Since Eclipse 3.3 is compatible with Eclipse 3.2, 3.1, 3.0, 2.1 and 2.0 in most regards, and Eclipse 3.3 is compatible with 3.2, it follows that 3.3 is also compatible with 3.2, 3.1, 3.0, 2.1 and 2.0 in most aspects. If you are upgrading directly from 3.1, 3.0, 2.1 or 2.0, refer also to the  [*Eclipse 3.1 Plug-in Migration Guide*](http://dev.eclipse.org/viewcvs/index.cgi/%7Echeckout%7E/org.eclipse.platform.doc.isv/porting/eclipse_3_1_porting_guide.html) and the  [*Eclipse 3.2 Plug-in Migration Guide*](http://dev.eclipse.org/viewcvs/index.cgi/%7Echeckout%7E/org.eclipse.platform.doc.isv/porting/eclipse_3_2_porting_guide.html) for problems areas.

## 3. Known Issues

[3.1 Platform](#_3znysh7)

[3.1.1 Core](#_2et92p0)

[3.1.2 Ant](#_tyjcwt)

[3.1.3 User Assistance](#_3dy6vkm)

[3.1.4 UI](#_1t3h5sf)

[3.1.5 Text](#_4d34og8)

[3.1.6 SWT](#_2s8eyo1)

[3.1.7 Team and CVS](#_17dp8vu)

[3.1.8 Install/Update](#_3rdcrjn)

[3.1.9 Debug](#_26in1rg)

[3.1.10 Compare](#_lnxbz9)

[3.2 Java development tools (JDT)](#_35nkun2)

[3.3 Plug-in Development Environment (PDE)](#_1ksv4uv)

Note: Bug numbers refer to the Eclipse project bug database at [http://bugs.eclipse.org/bugs/](http://dev.eclipse.org/bugs/)

### 3.1 Platform

### 3.1.1 Platform - Core

#### Installation/Configuration issues that can cause Eclipse to fail start

Here are some common problems that can cause Eclipse not to start:

* As shown [above](#_gjdgxs), Eclipse 3.3 requires at least a 1.4.2 VM. Perhaps an older version of the VM is being found in your path. To explicitly specify which VM to run with, use the Eclipse -vm command-line argument. (See also the [Running Eclipse](#_44sinio) section below.)
* Running Eclipse on Gentoo Linux may result in the following error message:  
  \* run-java-tool is not available for sun-jdk-1.6 on i686  
  \* IMPORTANT: some Java tools are not available on some VMs on some architecturesIf this occurs, start Eclipse by specifying a -vm argument, either specify the path to a java vm or use: eclipse -vm `java-config --java` (bug [176021](https://bugs.eclipse.org/bugs/show_bug.cgi?id=176021))
* Eclipse must be installed to a clean directory and not installed over top of a previous installation. If you have done this then please re-install to a new directory. If your workspace is in a child directory of your old installation directory, then see the instructions below on "[Upgrading Workspace from a Previous Release"](#1ci93xb).
* Java sometimes has difficulty detecting whether a file system is writable. In particular, the method java.io.File.canWrite() appears to return true in unexpected cases (e.g., using Windows drive sharing where the share is a read-only Samba drive). The Eclipse runtime generally needs a writable configuration area and as a result of this problem, may erroneously detect the current configuration location as writable. The net result is that Eclipse will fail to start and depending on the circumstances, may fail to write a log file with any details. To work around this, we suggest users experiencing this problem set their configuration area explicitly using the -configuration command line argument. (bug [67719](https://bugs.eclipse.org/bugs/show_bug.cgi?id=67719))

#### Installing plug-ins by unzipping them into the plugins directory

If you have installed new plug-ins and they aren't showing up when you run, then perhaps you unzipped them into your "plugins" directory and your configuration might need to be refreshed. This can be accomplished by starting Eclipse with the -clean command line argument.

#### Hanging during class loading when out of permanent generation memory

The Sun VM may hang indefinitely during class loading if it runs out of permanent generation memory. This will cause CPU usage to stay at 100% until the process is ended. See the section [Running Eclipse](#_44sinio) for details on addressing this VM problem.

#### XML files with UTF-8 byte order mark fail to have content type detected

Eclipse will fail to detect the proper content type for XML files that have a UTF-8 byte order mark if Crimson is the XML parser (as it is on Sun 1.4 JREs, but not on Sun 1.5 JREs). This problem will prevent actions normally available when files of the affected content types are selected from being presented to the user. The workaround is to ensure the default XML parser supports UTF-8 BOMs (such as Xerces does). (bug [67048](https://bugs.eclipse.org/bugs/show_bug.cgi?id=67048))

#### No branding with old config.ini

If you have an old config.ini file and use it with a new Eclipse build, you may not get the correct product branding. This is because the id of the standard Eclipse product changed. Users in shared install scenarios may end up in this situation as previous builds of Eclipse automatically generated config.ini files in some cases. The work around is either to delete the local config.ini or update the eclipse.product line to read eclipse.product=org.eclipse.platform.ide.

#### **Invalid characters in install directory prevents Eclipse from starting**

Eclipse will fail to launch if installed in a directory whose path contains certain invalid characters, including :%#<>"!. The workaround is to install Eclipse in a directory whose path does not contain invalid characters. (bugs [3109](https://bugs.eclipse.org/bugs/show_bug.cgi?id=3109) and [17281](https://bugs.eclipse.org/bugs/show_bug.cgi?id=17281))

#### Problems with classloaders in created threads

There is a known issue with trying to load classes from a newly-created thread using a class loader different from the plug-in class loader. The result will be a ClassNotFoundException. As a workaround, do the following:

1. Create a thread in which to run your code.
2. Send yourThread.setContextClassLoader(yourClassLoader); // you can find your classloader by grabbing a class it loaded (YourPluginClass.class.getClassLoader())
3. Run your code in the newly created thread.

If you set the context class loader for the current thread, you are competing with other users of the thread (all of Eclipse), so the results will be unpredictable. However, there should be no problem in practice provided you reset the context class loader back to its original value when your use in the current thread is complete. (bug [8907](https://bugs.eclipse.org/bugs/show_bug.cgi?id=8907))

#### Deadlock creating executable extension in Plugin.startup

If Plugin.startup code is too complex and performs tasks such as creating an executable extension, a deadlock situation can be created. Only simple bookkeeping tasks should be performed in Plugin.startup code. (bug [5875](https://bugs.eclipse.org/bugs/show_bug.cgi?id=5875))

#### Potential Problems Converting Plug-in Manifests

If your plug-in ships with a plug-in manifest and not an OSGi bundle manifest, is shipped as a JAR file, and contains a nested JAR file then there may be problems in the automatic generation of the bundle manifest file. The packages defined in the nested JAR may not be exported correctly in the Export-packages bundle manifest header. To work around this you should ship your plug-in with a bundle manifest. (bug [97689](https://bugs.eclipse.org/bugs/show_bug.cgi?id=97689))

#### Location for Debug Options File on Mac OS

If you are running in debug mode on Mac OS, the default location for the .options file is inside the application bundle in the Eclipse.app/Contents/MacOS directory (like the eclipse.ini). (bug [88782](https://bugs.eclipse.org/bugs/show_bug.cgi?id=88782))

#### Configuration can become invalid when removing org.eclipse.update.configurator

When launching an Eclipse Application from within the Eclipse IDE it is possible to select the set of plug-ins that are included in the Eclipse Application. Removing the org.eclipse.update.configurator plug-in from the set of plug-ins to an existing configuration can cause the configuration to become invalid. This can result in extra plug-ins installed in the target application that are not resolved. To work around this, after the org.eclipse.update.configurator plug-in has been removed, the target configuration area should be cleared before launching. (bug [85835](https://bugs.eclipse.org/bugs/show_bug.cgi?id=85835))

#### Issues with JNI that use FindClass

There may be issues when using a JNI implementation that uses FindClass in a function where the JNIEnv pointer is not available, such as in a C callback (bug [125250](https://bugs.eclipse.org/bugs/show_bug.cgi?id=125250)). The reason is that FindClass, in this case, uses the application class loader to find the class. If the desired class is in the classpath used for the application classloader (e.g. defined by the VM argument -cp <classpath>), as it would typically be in a stand-alone application, there is no problem. However, under Eclipse, the application classloader does not have access to classes contained in plug-ins. Eclipse uses its own class loader to find classes contained in plug-ins.

The proper plug-in class loader is used by FindClass in JNI functions which are passed the JNIEnv pointer, but not when you have to use AttachCurrentThread to get the JNIEnv pointer. In this case the application classloader is used.

For example, the following will fail because AttachCurrentThread is used to get the JNIEnv pointer:

static JavaVM\* jvm; // Global variable  
  
void myCallback(void) {  
 JNIEnv\* env;  
 jvm->AttachCurrentThread((void\*\*)&env, NULL);  
 // Fails if some/class is not in the application classloader:  
 jclass cls = env->FindClass("some/class");  
 jmethodID methodID = env->GetMethodID(cls, "methodName",  
 "(Ljava/lang/String;)V or whatever signature");  
 env->CallVoidMethod(callback, methodID, ...);  
 jvm->DetachCurrentThread();  
 }  
}

A solution is to cache the method ID, for example:

static jmethodID mid; // Global variable  
  
JNIEXPORT jint JNICALL JNI\_OnLoad(JavaVM \*vm, void \*reserved) {  
...  
 // Store the JavaVM pointer  
 jvm = vm;  
  
 // Find the class and store the method ID  
 // Will use the class loader that loaded the JNI library  
 jclass cls = env->FindClass(className"some/class");  
 if(!cls) goto ERR;  
  
 mid = env->GetMethodID(cls, "methodName",  
 "(Ljava/lang/String;)V or whatever signature");  
 if(!mid) goto ERR;  
...  
}  
  
void myCallback(void) {  
 JNIEnv\* env;  
 jvm->AttachCurrentThread((void\*\*)&env, NULL);  
 env->CallVoidMethod(callback, mid, ...);  
 // Handle error ...  
 jvm->DetachCurrentThread();  
 }  
}

### 3.1.2 Platform - Ant

#### UTF-8 encoded buildfiles with Byte Order Mark

UTF-8 encoded buildfiles with byte order marks will fail to be parsed correctly depending on the XML parser being used for the build. Therefore a valid buildfile will fail to build with an error message similar to: "BUILD FAILED: C:\workspace\bom.xml:1: Document root element is missing.". To succeed in building with these files, ensure to include Xerces jars on the Ant runtime classpath so that the Xerces parser is used to parse the XML. As well the context menu for these files in the Navigator or Package Explorer will not have the run shortcuts for Ant builds. (bug [67048](https://bugs.eclipse.org/bugs/show_bug.cgi?id=67048))

#### Custom Ant tasks and Ant types must be separate from plug-in library JARs

Including the class files for custom Ant tasks or Ant types in the regular code JAR for your plug-in causes problems. These class files must be provided in a separate JAR that is contributed to the org.eclipse.ant.core.antTasks or antTypes extension point (and not declared as a library in the plug-in's manifest). This ensures that the Ant tasks and types are loaded by the special Ant class loader and not by a plug-in classloader. (bug [34466](https://bugs.eclipse.org/bugs/show_bug.cgi?id=34466)).

#### Concurrent Ant builds not supported

Eclipse can run Ant in the same JVM as the rest of Eclipse. Several aspects of Ant and its use of global Java resources (such as System.out and System.err), make it unsafe to run more than one Ant build concurrently in the same JVM. (bug [24129](https://bugs.eclipse.org/bugs/show_bug.cgi?id=24129)).

#### Running certain Ant tasks cause memory leakage

Certain Ant tasks are known to leak memory. Please see the bug report for details, patches, and possible workarounds. (bug [24448](https://bugs.eclipse.org/bugs/show_bug.cgi?id=24448))

#### Tasks that require input lock up workspace

As with using Ant from the command line, prompts for input from the console is not handled. This is not the same as making use of the <input> task, which works correctly within Eclipse. (bug [21748](https://bugs.eclipse.org/bugs/show_bug.cgi?id=21748))

#### "version" property is always set when running Ant in the same VM as Eclipse

The Xalan libraries set system properties including a version property. These get set as properties within the Ant build and therefore the "version" property cannot be set within an Ant buildfile due to the immutable nature of Ant properties. This property will always be set to "2.4.1" for Ant builds in the same VM as Eclipse. (bug [45717](https://bugs.eclipse.org/bugs/show_bug.cgi?id=45717))

#### XDoclet support from within Eclipse

Since there are differences when running Ant from the commandline and within Eclipse, some extra steps may be needed to have XDoclet support function correctly within Eclipse. Problems may occur creating XDoclet subtasks. The workarounds and full discussion can be found in bug report. (bug [37070](https://bugs.eclipse.org/bugs/show_bug.cgi?id=37070))

#### Ant Editor code completion based on Ant 1.6.1

Code completion provided by the Ant editor does not respect the user-specified version of org.eclipse.ant.core plug-in or ANT\_HOME. Code completion proposals are mostly based on Ant 1.6.1 with some updates to Ant 1.6.5 (bug [30886](https://bugs.eclipse.org/bugs/show_bug.cgi?id=30886))

#### Eclipse can hang due to implementation of the Ant <property> task (Windows 9X only)

On Windows 9X, using:<property environment="env"/> will cause Eclipse to hang if the build occurs in the same VM as Eclipse. Running the build in a separate VM will hang the build but not Eclipse. (bug [44196](https://bugs.eclipse.org/bugs/show_bug.cgi?id=44196))

#### Setting build loggers not supported when debugging Ant builds

When debugging Ant builds within Eclipse, setting -logger as a program argument will be ignored.

#### Renaming an External Tool builder set to run during auto-build will cause errors

If you rename an existing external tool builder that is configured to run during auto-builds, you will get the following error: Errors during build. Errors running builder "Integrated External Tool Builder" on project <PROJECT\_NAME>. The builder launch configuration could not be found. The workaround is to first disable the builder for auto-builds and then rename the builder. (bug [118294](https://bugs.eclipse.org/bugs/show_bug.cgi?id=118294))

#### Slow typing/saving of the Ant editor with imports that define numerous macrodefs

The Ant editor is slow on saving with buildfiles that have <import> declarations of buildfiles that have numerous <macrodef>s. See bugs [92640](https://bugs.eclipse.org/bugs/show_bug.cgi?id=92640) and [125117](https://bugs.eclipse.org/bugs/show_bug.cgi?id=125117) for possible workarounds

#### Failure to run Ant builds on non-Windows platforms if Eclipse installed in location with spaces in the path

Due to a bug in Ant 1.7.0, Ant builds will fail with an IllegalArgumentException if the Eclipse installation is in a location with spaces in the path. Embedded usage of Ant builds, such as plug-in export will also fail. See bug [187993](https://bugs.eclipse.org/bugs/show_bug.cgi?id=187993) for possible workarounds

### 3.1.3 Platform - User Assistance

#### Welcome page not displayed properly (Linux/Unix)

The default Welcome implementation is HTML-based and requires a supported browser in order to work. If no supported browser can be found, Welcome falls back to its Forms-based implementation, which has a different (simpler) appearance. Consult the [SWT FAQ](http://www.eclipse.org/swt/faq.php#browserplatforms) for supported browsers and setting up your browser to work with eclipse.

#### Help browser tool bar buttons do not work for some documents

The Help browser's Print, Synchronize, and Bookmark buttons do not work for pages that are not actually installed with the product. However, you can always use the print command in the browser's context menu to print the page you're reading. (bug [44216](https://bugs.eclipse.org/bugs/show_bug.cgi?id=44216))

#### Help documents not displayed in a browser or very slow document loading (Windows only)

If your LAN settings are not properly configured for local host access, your Help browser might open to a blank page or display an HTTP error instead of a help page, or you may experience long delays when loading help documents. Your system administrator can configure your LAN settings so that help documents can be accessed from the local help server.

1. In the Control Panel, open **Internet Options**, select the **Connections** tab and choose **LAN Settings**.
2. If your host was configured to use DHCP for IP assignment, make sure that the "Automatically detect settings" check box is cleared.
3. If you use a proxy server, ensure that the "Bypass proxy server for local addresses" is selected.
4. In "Advanced" settings for proxies, add "127.0.0.1;localhost" to the "Exceptions" if these addresses are not listed.
5. If you are using an automatic configuration script for proxy settings, and are not sure that the script is correct, clear the "Use automatic configuration script" check box.

If the above steps do not fix your problem, try changing the port and host properties on the **Help > Help Server** preference page. In general, setting host to localhost or 127.0.0.1 should work. Also, especially when running a firewall, you may want to specify port 80 or some other firewall-friendly value. (bugs [7036](https://bugs.eclipse.org/bugs/show_bug.cgi?id=7036), [9418](https://bugs.eclipse.org/bugs/show_bug.cgi?id=9418), [11394](https://bugs.eclipse.org/bugs/show_bug.cgi?id=11394))

#### Working disconnected from the network (Windows only)

If you are experiencing problems when not connected to the network, you must install the loopback adapter from the Windows installation CD. (bug [831](https://bugs.eclipse.org/bugs/show_bug.cgi?id=831))

#### Using Internet Explorer in offline mode (Windows only)

If you have been using Internet Explorer in Offline mode, when you access the help system you will get a message indicating that the web page you requested is not available offline or a blank page will display. Click **Connect** or deselect "Work Offline" in the Internet Explorer "File" menu to return the system behavior to normal.

#### Help topics not highlighted in High Contrast mode (Windows only)

Windows High Contrast settings are not consistently picked up by Internet Explorer when they are set from the Accessibility Options utility as opposed to when they are set using the predefined schemes. On Windows XP, it is recommended to set High Contrast as follows: Right click the desktop, chose properties, select Windows Classic style from the Windows and buttons drop down on the Appearance tab, and choose your scheme (for example High Contrast Black) from Color Scheme drop down. (bug [28609](https://bugs.eclipse.org/bugs/show_bug.cgi?id=28609))

#### Help browser displays a blank page

If you see a help launched with a blank page, and no errors displayed, it can be caused by a conflict between libraries in org.eclipse.tomcat plug-in and jars optionally installed in JRE jre/lib/ext directory. To fix the problem, ensure that the JRE used for running Eclipse does not contain any J2EE or Apache jars in the jre/lib/ext directory. (bug [63970](https://bugs.eclipse.org/bugs/show_bug.cgi?id=63970))

### 3.1.4 Platform - UI

#### High contrast settings

Eclipse was tested for High Contrast using 1152 x 864 resolution in Windows XP High Contrast mode. You can select this mode by selecting Accessibility Options > Display > Use High Contrast from the Windows XP Control Panel menu.

#### Default text file encoding may be detected incorrectly (Windows XP/2000 only)

**Note**: the bug report associated with this problem has been fixed. If you run Eclipse with JDK 1.5 or greater you should not have to use the workaround stated below any longer. However, the problem still exists when running Eclipse with JDK 1.4.x or lower, so in this case the workaround is still required .

The "Text file encoding" value displayed in the Preferences dialog under "Editors" may be wrong on platforms running Windows XP (or 2000) when the user locale and system locale differ.

Example of the manifestation of the bug: A Japanese user using Japanese Windows 2000 works in New York, United States. The user has selected English (United States) as the user locale. The "Text file encoding" value displayed by Eclipse is incorrect: "Cp1252" (English). It should display the system locale "MS932" (Japanese).

Workaround: The user can modify the user locale so that user locale and system locale are identical. In the example above, this means the user should set Japanese as the user locale. Then restart Eclipse. The "Text file encoding" value will then be correct: "MS932" (Japanese).

For Windows XP:

* To check the system locale: Open the Control Panel. Go to Regional and Language Options. Switch to the Advanced tab. The system locale is specified in "Language for non-Unicode programs".
* To change the user locale: Open the Control Panel. Go to Regional and Language Options. The user locale can be modified by changing the language in "Standards and formats".

For Windows 2000:

* To check the system locale: Open the Control Panel. Go to Regional Options. Look up the items in the General tab, inside the "Language settings for the system" group. The system locale is the item marked as (Default).
* To change the user locale: Open the Control Panel. Go to Regional Options. The user locale can be modified by changing the location in "Settings for the current user".

(bug [20641](https://bugs.eclipse.org/bugs/show_bug.cgi?id=20641))

#### Dirty state not tracked properly for OLE documents (Windows only)

The dirty state for an OLE document is not updated properly. This causes Eclipse to prompt to save the contents of the editor when the document is closed, even if the contents have already been saved. (bug [2564](https://bugs.eclipse.org/bugs/show_bug.cgi?id=2564))

#### OLE document crashes can cause Eclipse to also crash (Windows only)

If an OLE document crashes, Eclipse can crash, or the workbench menus can become inconsistent.

#### 2.1 Presentation based workspaces incorrectly get new Min/Max behavior

Workspaces that are currently using the Eclipse 2.1 Presentation will incorrectly 'inherit' the new min/max behavior when opened with 3.3.

Workaround:

1. Go to the 'Preferences -> Appearance' page, change the current presentation to 'Default' and select apply
2. Change it back to the 2.1 Presentation, select 'OK' and 'Yes' to the restart prompt

When the workbench re-opens the old min/max behaviour will be restored.

#### Toolbars only containing contributed controls exhibit display errors on Mac/Linux

Currently there is no way on the Max or Linux platforms to define the **height** for controls contributed to toolbars, nor will those platforms respect the size returned by the control's computeSize method. If you encounter this issue there is currently no truly viable workaround. (bug [183003](https://bugs.eclipse.org/bugs/show_bug.cgi?id=183003))

### 3.1.5 Platform - Text

None.

### 3.1.6 Platform - SWT

#### Eclipse plug-in based on the SWT Browser throws exception

The SWT Browser widget uses a platform-specific web browser to render HTML. The org.eclipse.swt.SWTError exception ("No more handles") is thrown on platforms that don't meet the requirements for running the Browser widget. Supported platforms and prerequisites are listed on the SWT FAQ item  ["Which platforms support the SWT Browser?"](http://www.eclipse.org/swt/faq.php#browserplatforms).

#### Crash when using the file dialog (Windows XP with SP2 only)

With some versions of Synergy from Telelogic, Eclipse will crash when you try to open a file dialog. This is due to a problem with the CMExplorer.dll. The workaround is to upgrade to Synergy 6.4 (or higher) or to run regsvr32 /u CMExplorer.dll and reboot (note that this will disable Active CM). (See bug [87798](https://bugs.eclipse.org/bugs/show_bug.cgi?id=87798) for details).

#### Opening File Dialog crashes eclipse (Vista only)

On Vista, launching eclipse using -vmargs -Xmx[any size] can crash eclipse when the FileDialog is opened. The workaround is to use the default heap size, i.e. do not use the -Xmx VM args. See [bug 188317](https://bugs.eclipse.org/bugs/show_bug.cgi?id=188317) for details.

#### Internet Explorer sometimes freezes on PDF documents with Acrobat Reader 6 (Windows only)

With Acrobat Reader 6 or 7, some users have experienced an unresponsive user interface for up to two minutes when closing a browser which is displaying a PDF document. The workaround is to disable displaying PDF in the browser. In Adobe Reader select Edit > Preferences... > Internet and uncheck 'Display PDF in browser'. (bug [56184](https://bugs.eclipse.org/bugs/show_bug.cgi?id=56184))

#### Crash while editing text (Windows XP with SP2 only)

Some users who have installed Service Pack 2 on Windows XP have experienced crashes while using editors in Eclipse. The workaround is to place a working version of Windows\System32\USP10.DLL in the Eclipse startup directory or uninstall Service Pack 2. (bug [56390](https://bugs.eclipse.org/bugs/show_bug.cgi?id=56390))

#### Input Method broken (Motif only)

Some versions of RedHat Linux such as Fedora Core 3 and Enterprise Linux WS release 4 use a new technology called IIIM (Intranet/Internet Input Method Framework) to replace the old XIM (X input method). When running on these new systems, Eclipse will crash if you attempt to enter any DBCS character. The workaround is to use a XIM based input method such as chinput. This problem may be fixed in newer releases of RedHat. (bug [89722](https://bugs.eclipse.org/bugs/show_bug.cgi?id=89722))

#### Eclipse does not start on Linux-Motif with Xinerama and a UTF-8 locale

The Linux-motif build of Eclipse does not launch properly when run on a computer with Xinerama (provides support for dual head monitors) and a UTF-8 locale. The workaround for this problem is to change the locale to a non-UTF-8 value, or to disable Xinerama. (bug [38843](https://bugs.eclipse.org/bugs/show_bug.cgi?id=38843))

#### Eclipse crashes on RedHat Linux 9 with Bluecurve (GTK+ only)

Users of the Bluecurve theme shipped with RedHat Linux 9 may experience problems running Eclipse. These problems may include crashes or reduced performance. We recommend changing to a different theme. (bugs [38305](https://bugs.eclipse.org/bugs/show_bug.cgi?id=38305), [67906](https://bugs.eclipse.org/bugs/show_bug.cgi?id=67906), [37683](https://bugs.eclipse.org/bugs/show_bug.cgi?id=37683), [58738](https://bugs.eclipse.org/bugs/show_bug.cgi?id=58738))

#### Eclipse hangs when pasting from an unresponsive application (GTK only)

If the application that is supplying the clipboard material is unresponsive, the paste operation hangs Eclipse for several minutes. This situation can be encountered when copying from an Eclipse target workbench, suspending the target workbench at a breakpoint and pasting into the hosting Eclipse workbench. (bug [44915](https://bugs.eclipse.org/bugs/show_bug.cgi?id=44915))

#### Unable to drag data between applications in simplified Chinese locale (Motif only)

When configured for the simplified Chinese locale, it is not possible to drag data between applications running on the Motif window system. This is a known limitation of the Open Motif library. (bug [29777](https://bugs.eclipse.org/bugs/show_bug.cgi?id=29777))

#### Crash when attempting to launch file browser (AIX Motif only)

There is a known AIX graphics bug affecting certain levels of AIX releases. Ensure that the AIX install includes the necessary service updates as described in the "Install notes/requirements for Eclipse on AIX" attachment to Eclipse bug report number [34524](https://bugs.eclipse.org/bugs/show_bug.cgi?id=34524)

#### Available colors on 8-bit Linux (Linux only)

Typically, in Gnome Linux installs running with 8-bit visuals (i.e. 256 color mode), before the Eclipse application is started there are no free colors. This may mean that Eclipse is unable to allocate the default widget background color, causing it to display a white background. The functionality, however, is otherwise unaffected.

#### List and ComboBox on Windows NT (Windows NT only)

On Windows NT only, you should avoid creating items in a List or ComboBox with strings longer than 1000 characters. Doing so may result in a General Protection Fault. This has been fixed in more recent versions of Windows.

#### IME-related crash (Linux Motif only)

When using Linux Motif and GB18030 IME "chinput", Eclipse can crash if the IME client window is left open when the parent window is disposed. (bug [32045](https://bugs.eclipse.org/bugs/show_bug.cgi?id=32045))

#### Using IBM J9 VM (Photon and AIX)

On QNX Photon and IBM AIX, the SWT library will not be found when running with an IBM J9 1.5 VM. This is a bug in the IBM J9 class library in version 1.5. You can workaround this problem by adding the SWT library directory to your LD\_LIBRARY\_PATH environment variable.

#### Missing permissions for SWT native libraries in workspace (HP-UX only)

When retrieving the SWT Motif fragment into an Eclipse workspace, the permissions of the native libraries are reset. This creates a problem on HP-UX because shared libraries need to have execute permission. Attempting to self-host with this fragment throws an UnsatisfiedLinkError...Permission Denied error. You must manually change the permissions to make these libraries accessible (assume the workspace is at /workspace):

cd /workspace/org.eclipse.swt.motif.hpux.PA\_RISC

chmod 555 \*.sl

#### gtk\_init\_check and X11 socket failure when using the IBM 1.4.2 JRE (GTK only)

Under RHEL 3.1 with the IBM 1.4.2 JRE and a large number of plugins, Eclipse may fail to launch with an exception from gtk\_init\_check along with this error:

\_X11TransSocketOpen: socket() failed for local  
\_X11TransSocketOpenCOTSClient: Unable to open socket for local

A workaround is to set the environment variable JAVA\_HIGH\_ZIPFDS to a value of 500 before starting Eclipse. (bug [106396](http://bugs.eclipse.org/bugs/show_bug.cgi?id=106396))

#### Key bindings can stop working on Debian (GTK+ only)

On some versions of Debian, Eclipse key bindings may stop working. In this context the only way to make the key bindings work again is to restart Eclipse.

The problem is that a focus issue exists in GTK+ 2.6.7 and earlier, for which SWT has a workaround. This workaround is incompatible with the GTK+ 2.6.7 fix, so a GTK+ version check is done at runtime to determine whether the workaround should be used or not. However, Debian backported the GTK+ focus fix into their libgtk+2.0 (2.6.4-2) package, so the SWT workaround and GTK+ fix are both incorrectly applied in this context.

To work around this problem, either get the Debian unstable version of GTK+, compile your own GTK+, or hack SWT's Shell.gtk\_realize(int) and change the version that it checks. See SWT bug [107013](https://bugs.eclipse.org/bugs/show_bug.cgi?id=107013) and GTK+ bug [109246](http://bugzilla.gnome.org/show_bug.cgi?id=109246) for more information.

#### Browser does not display applets (Windows and OS X)

The Browser widget cannot be used to display pages containing Java applets on Windows and OS X, as a result of crashes that occur when attempting to launch a second JVM for the applet that is in-process with the main process JVM. The workaround for clients wishing to display web pages with Java applets is to launch an external web browser to do so with org.eclipse.swt.program.Program (see bugs [59506](https://bugs.eclipse.org/bugs/show_bug.cgi?id=59506) and [100622](https://bugs.eclipse.org/bugs/show_bug.cgi?id=100622)).

#### Eclipse hangs with earlier versions of Quicktime (Intel Mac OS X only)

Some users reported encountering system hangs while using Eclipse on Intel-based Macs. These hangs were traced to a problem in some versions of QuickTime, which has now been fixed. Therefore, Eclipse users on Intel-based Macs should use Quicktime's update facilities to ensure that their Quicktime version is at least 7.1.1. (see bug [142892](https://bugs.eclipse.org/bugs/show_bug.cgi?id=142892)).

#### Typing in an editor crashes with IBM 1.5 VM (Linux GTK PPC only)

When running on the IBM Java 5.0 VM, Eclipse crashes while the user is typing in an editor. If using this VM you must disable the JIT with the -Xnojit vm argument to avoid the crashes (see bug [116730](https://bugs.eclipse.org/bugs/show_bug.cgi?id=116730)). The command line for launching Eclipse with this vm should be:

eclipse -vmargs -Dosgi.locking=none -Xnojit

#### Eclipse won't start (Linux GTK PPC only)

Eclipse fails to create a lock file with reason "No locks available". To launch eclipse you must disable file locking using the osgi.locking property. For example, you could launch eclipse as follows:

eclipse -vmargs -Dosgi.locking=none

#### SWT\_AWT bridge doesn't work (Mac OSX only)

In order to use the SWT\_AWT bridge on the Mac, OS X jre version 1.5.0 Release 5 (or greater) must be used.

#### Eclipse printing is disabled or Eclipse hangs when opening editor (GTK only)

In order to print from eclipse on GTK, you need to have GTK+ version 2.10 or later. In addition, at least two print backends must exist on the machine: file and lpr. Assuming a that GTK was installed in /usr, the installed backends can be viewed at /usr/lib/gtk-2.0/2.10.0/printbackends.

### 3.1.7 Platform - Team - CVS

The following are known problems with the CVS repository provider only, and do not apply to other repository providers. Additional information on how to use CVS from Eclipse can be found in the [Eclipse CVS FAQ](http://dev.eclipse.org/viewcvs/index.cgi/~checkout~/platform-vcm-home/docs/online/cvs_features2.0/cvs-faq.html).

#### CVS server compatibility

The CVS plug-in parses messages returned from the CVS server. If the format of these messages is not as expected, some of the plug-in's functionality may be missing. The CVS plug-in is compatible with all stable 1.11.X builds of the CVS server, and should be compatible with future releases in that stream unless text message formats change (the last tested server was 1.11.22). As for the 1.12.X feature releases of CVS, the Eclipse CVS client has been tested with builds up to 1.12.13. However, future releases could easily break the Eclipse CVS client. Basic functionality, such as Checkout, Commit, and Update, should always work, but there may be problems with more advanced commands such as Synchronizing and Browsing the repository.

#### SSH2 proxy settings lost upgrading to 3.3

CVS now uses the Platform proxy settings. As a result, any CVS proxy settings will be lost and must be re-entered on the General>Network Connections preference page.

#### Connection cannot be found after initially missing

If a connection initially fails due to a network problem, the connection may continue to fail even when the network problem is fixed. In order to establish the connection you must exit and restart Eclipse. (bug [9295](https://bugs.eclipse.org/bugs/show_bug.cgi?id=9295))

#### "Received broken pipe signal" error from server

Eclipse sometimes performs multiple commands within a single connection to the server. This may cause problems with CVS servers that are running server scripts in response to certain commands. (bugs [23575](https://bugs.eclipse.org/bugs/show_bug.cgi?id=23575) and [23581](https://bugs.eclipse.org/bugs/show_bug.cgi?id=23581))

#### "Terminated with fatal signal 10" error from server

There is a bug in the CVS server related to some compression levels. If you get this error, changing the compression level on the CVS preference page may help. (bug [15724](https://bugs.eclipse.org/bugs/show_bug.cgi?id=15724))

#### "Unknown response" error using ext connection method

There are a few situations that can result in an "Unknown response" error messages when using the ext connection method. One situation involves using an external communications client (e.g. rsh or ssh) that adds CRs to the communications channel (bug [21180](https://bugs.eclipse.org/bugs/show_bug.cgi?id=21180)). Another involves Eclipse not properly reading the stderr output of the external communications tool. (bug [11633](https://bugs.eclipse.org/bugs/show_bug.cgi?id=11633))

#### A disabled CVS capability may not be auto-enabled in existing workspaces

New in 3.0 is the ability to disable capabilities and the CVS support in Eclipse can be disabled. However, for backwards compatibility the CVS capability is auto-enabled in existing workspaces that already contain CVS projects. The auto-enabling function may not run if the team support plugin is not loaded at startup. (bug [66977](https://bugs.eclipse.org/bugs/show_bug.cgi?id=66977))

#### Builder output files may appear as changed

When folders containing build output are shared they may get improperly marked as dirty when build output is generated.

### 3.1.8 Platform - Install/Update

#### Manually installing features and plug-ins on a FAT file system (Windows only)

When features and plug-ins are manually installed on top of an Eclipse-based product install located on a FAT file system that has already been run at least once, the product must be explicitly restarted with -clean. That is,

eclipse.exe -clean

Then, open the Help > Software Updates > Manage Configuration dialog and toggle on the "Show disabled features" button in its toolbar. Select the newly "installed" feature and press on the "Enable feature" action on the right pane (or select the action from the feature's context menu). (bugs [52525](https://bugs.eclipse.org/bugs/show_bug.cgi?id=52525), [66120](https://bugs.eclipse.org/bugs/show_bug.cgi?id=66120), [67461](https://bugs.eclipse.org/bugs/show_bug.cgi?id=67461))

#### Non-responsive sites may use all free threads

Prior to Eclipse 2.1, if the connection to an update site did not respond (the site did not exist or was down), the workbench became non-responsive until the connection request timed out. Since 2.1, connections are made by a separate thread so that the UI stays responsive. Typically, unresponsive connections eventually time out and these threads terminate. In rare cases, servers accept the connection but never send a response, thereby keeping the connection thread live indefinitely. Update manager limits the number of active connection threads and will refuse to create more once the limit is reached. To work around the problem, exit and restart Eclipse. (bugs [18598](https://bugs.eclipse.org/bugs/show_bug.cgi?id=18598), [19775](https://bugs.eclipse.org/bugs/show_bug.cgi?id=19775))

#### OSGi and run-time plug-ins cannot revert to the previous version

When a feature containing the OSGi and Eclipse runtime plug-ins is updated to a newer version, performing the Revert operation to undo the action will seemingly work, but the OSGi and runtime plug-ins in use will still be those used prior to the Revert. In effect, updating to a newer version of Eclipse platform is not reversible. All other features can be reverted without problems. (bug [74585](https://bugs.eclipse.org/bugs/show_bug.cgi?id=74585))

**Extension location is lost if the install path changes**

A previously configured extension location may be temporarily removed if the install is moved or mounted under a different path. This only happens when the link file that configures the extension location uses a relative path that points to a directory under the Eclipse install. On a second startup using the same install path, the extension location is added again (bug [95403](https://bugs.eclipse.org/bugs/show_bug.cgi?id=95403)).

### 3.1.9 Platform - Debug

None. (Known problems with the Java debugger appear below in the [JDT](#_35nkun2) section.)

### 3.1.10 Platform - Compare

None.

### 3.2 Java development tools (JDT)

#### Searching for constant field references

Search does not find references to constant fields inside binaries because the Java Language Specification mandates that constant field values be inlined in the class file's byte codes, leaving no trace of a field reference. (bug [12044](https://bugs.eclipse.org/bugs/show_bug.cgi?id=12044))

#### Classpath entry denoting external class folder is now properly rejected

Although external class folders (i.e., folder containing .class files and located outside workspace) were never properly supported by the JDT, a problem was never reported when constructing such a build path. JDT now properly diagnoses a problem in this situation. In order to still benefit from external .class files, a class folder must be mounted as a linked folder in Eclipse workspace (in project properties, select Java Build Path > Libraries > Add Class Folder > Create New Folder... > Advanced. Then a folder name can be associated with an arbitrary file system location by checking "Link to folder on the file system). Once mounted, the linked class folder can normally be referenced on a build path, and from there on programs can be compiled against it. (bug [67631](https://bugs.eclipse.org/bugs/show_bug.cgi?id=67631))

#### Cut, copy, paste not working for linked resources in views showing Java elements

The cut, copy, and paste actions do not work for linked files and folders appearing in views that show Java elements, including the Package Explorer. The workaround is to use these actions from the Navigator view instead. (bug [34568](https://bugs.eclipse.org/bugs/show_bug.cgi?id=34568))

#### Java working sets not working correctly for elements from JRE system library container

Applying a working set consisting entirely of elements from the JRE System library container as a filter to the packages view might result in an empty Package Explorer. (bug [35395](https://bugs.eclipse.org/bugs/show_bug.cgi?id=35395))

#### Cannot generate Javadoc for packages with GB18030 characters in the name

Most class libraries do not properly support the creation of a system process (via java.lang.Runtime.exec(...)) when the specified command line contains GB18030 characters. Since Javadoc is created using the Javadoc executable provided with the JDK, generating Javadoc fails if the package or class name contains GB18030 characters. (bug [32215](https://bugs.eclipse.org/bugs/show_bug.cgi?id=32215))

#### Side effects of Step into Selection and Run to Line

The actions "Step into Selection" and "Run to Line" optimistically set breakpoints on the line the user has chosen to step into or run to. However, the debugger can not determine if or when execution will ever reach the chosen line. The breakpoints set by the underlying implementation are not visible to the user and can cause execution to suspend unexpectedly at a later time, when the associated line is actually executed. (bug [51507](https://bugs.eclipse.org/bugs/show_bug.cgi?id=51507))

#### Default locale initialization incorrect

The default locale is generally initialized from the settings in the operating system when a target VM is launched. However, when using javaw.exe on JDK1.4.2, Windows XP, the default locale is incorrectly initialized to en\_US, no matter what the operating system settings are. (bug [65945](https://bugs.eclipse.org/bugs/show_bug.cgi?id=65945))

#### Suspend on uncaught exception overrides exception breakpoint location filters

Exception breakpoints can be configured with location filters (inclusive and exclusive). When an unchecked exception is configured to **not** suspend execution in a specific class, execution will still suspend when the user preference to suspend on uncaught exceptions is on. (bug [66770](https://bugs.eclipse.org/bugs/show_bug.cgi?id=66770))

#### Running Java programs with non-Latin-1 characters in package or class names

You get a java.lang.NoClassDefFoundError when running Java programs with non-Latin characters in the package or class names. The workaround is to package the class files as a JAR file and run the program out of the JAR and not from the file system directly. (bug [4181](https://bugs.eclipse.org/bugs/show_bug.cgi?id=4181))

#### Cannot run or debug class in a project with GB18030 characters in project name

Most class libraries do not properly support the creation of a system process (via java.lang.Runtime.exec(...)) when the specified command line contains GB18030 characters. This limitation means the debugger cannot launch applications when the command line it generates contains GB18030 characters. (bug [32206](https://bugs.eclipse.org/bugs/show_bug.cgi?id=32206))

#### Cannot detect installed JRE with GB18030 characters in path name

Automatic JRE detection fails when the JRE is stored in a directory containing GB18030 characters in its name. (bug [33844](https://bugs.eclipse.org/bugs/show_bug.cgi?id=33844))

#### Unable to debug stack overflows

If a debug session suspends on a java.lang.StackOverflowError exception (due to an exception breakpoint), the debugger may not be able to retrieve any debug information from the target JVM. As well, the debugger may not be able to reliably interact with the target JVM past this point. (bug [19217](https://bugs.eclipse.org/bugs/show_bug.cgi?id=19217))

#### Evaluation limitation

The debugger uses threads in the target JVM to perform evaluations (both explicit evaluations that the user requests, and implicit evaluations such as toString() invocations in the **Variables** view). The Java Debug Interface (JDI) requires that the thread in which an evaluation is performed be suspended by a user event (that is, a breakpoint or step request). Evaluations cannot be performed on threads suspended by the suspend action. As well, when a breakpoint is configured to suspend the JVM rather than just the individual thread, the threads which did not encounter the breakpoint are not in a valid state to perform an evaluation. When an evaluation is attempted in a thread that is not in a valid state to perform an evaluation, an error message will appear to the effect of "Thread must be suspended by step or breakpoint to perform method invocation". (bug [34440](https://bugs.eclipse.org/bugs/show_bug.cgi?id=34440))

#### Missing debug attributes

The debugger requires that class files be compiled with debug attributes if it is to be able to display line numbers and local variables. Quite often, class libraries (for example, "rt.jar") are compiled without complete debug attributes, and thus local variables and method arguments for those classes are not visible in the debugger.

#### Using Hot Code Replace

Hot code replace is supported on JDK 1.4.x VMs, and IBM J9 VMs. The debugger will attempt to replace all class files that change in the workspace as the user edits and builds source code. However, hot code replace is limited to changes that a particular virtual machine implementation supports. For example, changes within existing methods may be supported, but the addition or removal of members may not be.

Note that hot code replace and stepping on JDK 1.4.0 VMs was unreliable. The underlying VM problems were fixed in JDK 1.4.1, and later.

#### Scrapbook

Setting a breakpoint inside a scrapbook page is not supported.

When a snippet is run in the scrapbook which directly or indirectly calls System.exit(int), the evaluation cannot be completed, and will result in a stack trace for a com.sun.jdi.VMDisconnectedException being displayed in the scrapbook editor.

Terminating a scrapbook page while it is performing an evaluation results in a com.sun.jdi.VMDisconnectedException being displayed in the scrapbook editor.

#### Debugging over slow connections

A global Java debug preference specifies the debugger timeout, which is the maximum amount of time the debugger waits for a response from the target VM after making a request of that VM. Slow connections may require that this value be increased. The timeout value can be edited from the **Java > Debug** preference page. Changing the timeout value only effects subsequently launched VM, not VMs that are already running.

#### Updating of inspected values

When inspecting the result of an evaluated expression in the debugger, it is important to note that the result displayed is the result of that expression at the time it was evaluated. For example, when inspecting a simple integer counter (primitive data type), the value displayed in the Expressions view is the value when the expression was evaluated. As the counter is changed in the running program, the inspected result will not change (since the view is not displaying the value bound to a variable - it is displaying the value of an expression, and the value of a primitive data type cannot change). However, if an expression results in an object, fields of that object will be updated in the inspector as they change in the running program (since the value bound to fields in an object can change).

#### Stepping over native methods that perform I/O

When the debugger steps over native methods that perform I/O to System.out or System.err, the output may not appear immediately unless the native performs a flush on the output buffer.

#### VM and process termination running on IBM 1.3 JVM on Linux (Linux only)

Terminating a launch, debug target, or system process associated with a debug target running on the IBM 1.3 JVM on the Linux platform does not work when the associated debug target has a suspended thread. To remove such debug targets from the debug UI, select **Terminate and Remove** from the debug view's pop-up menu (or use the shortcut "delete" key). Associated system processes in the OS may not be properly cleaned up. If a debug target has no suspended threads, termination works properly. (bug [1631](https://bugs.eclipse.org/bugs/show_bug.cgi?id=1631))

#### Memory View (Linux only)

The feature to automatically load segments of memory while scrolling in the Memory view does not work on Linux. Instead the user must use the "Next Page" and "Previous Page" actions to manually load memory segments (bug [74559](https://bugs.eclipse.org/bugs/show_bug.cgi?id=74559))

#### Java 6 Annotation Processing

Java 6 annotation processors are supported in the batch compiler and in the IDE, with some limitations. Java 6 processors are only executed during a build, not while editing (bug [188558](https://bugs.eclipse.org/bugs/show_bug.cgi?id=188558)). Files generated by Java 6 processors during a build are deleted only when the project is cleaned, rather than during subsequent builds (bug [188559](https://bugs.eclipse.org/bugs/show_bug.cgi?id=188559)). Problem markers contributed by Java 6 processors may incorrectly be removed if the file containing the error also refers to a generated type (bug [186057](https://bugs.eclipse.org/bugs/show_bug.cgi?id=186057)).

### 3.3 Plug-in Development Environment (PDE)

#### Feature manifest editor does not preserve all comments

When a non-source page of the feature manifest editor is used, PDE will convert changes back into XML by regenerating the file. Although the overall content and most of the comments are preserved, some comments may be lost. (bug [59502](https://bugs.eclipse.org/bugs/show_bug.cgi?id=59502))

#### PDE will not unzip source zips of some plug-ins

In the plug-in import wizard, when you choose to import plug-ins as "projects with source folders", PDE will not unzip the source for the org.apache.ant, org.eclipse.core.runtime.compatibility.registry, org.eclipse.osgi.util and org.eclipse.osgi.services. This is because the source ZIPs contains code that will not compile when unzipped as it requires additional JARs that are not part of the SDK. To avoid the creation of plug-in projects that won't compile, PDE will import these plug-ins as binary and attach source, so you would still be able to read the source, you just won't be able to modify it. Also, PDE will not unzip the source for the org.eclipse.swt plug-ins. In this case, it is because, when shipped, the swt code is spread across a plug-in and a fragment, and when unzipped, it will require circular dependencies between the plug-in and fragment projects. These circular dependencies are at minimum marked as warnings by the JDT compiler and may result in unpredictable build behavior. Therefore, PDE always imports org.eclipse.swt as binary with source attached. (bug [66314](https://bugs.eclipse.org/bugs/show_bug.cgi?id=66314))

#### Emacs key bindings do not work in manifest editor fields

Non-default key bindings currently do not work in fields on non-source pages of the PDE manifest editors. (bug [19482](https://bugs.eclipse.org/bugs/show_bug.cgi?id=19482))

#### Plug-in import wizard does not allow plug-ins of different versions to be imported

The Eclipse platform allows two plug-ins with the same ID but different versions to coexist if the only thing they contribute is run-time libraries. However, PDE cannot handle these plug-ins because it creates project names using plug-in IDs during binary project import. (bug [18500](https://bugs.eclipse.org/bugs/show_bug.cgi?id=18500))

#### Export of plug-in may silently drop classes

When exporting a plug-in using the plug-in, feature or product wizards, some classes might be dropped from the resulting archive if their fully qualified name is too long. This typical path limitation can be worked around by creating the jar of the problematic plug-in by using the Jar export wizard. (bug [97150](https://bugs.eclipse.org/bugs/show_bug.cgi?id=97150))

#### Compilation errors when exporting projects not stored outside of the workspace

When exporting multiple plug-ins and one is stored outside of the workspace, compile errors occurs on export. To work around the problem, you can either export the plug-ins one by one, or change their location. (bug [98579](https://bugs.eclipse.org/bugs/show_bug.cgi?id=98579))

#### Headless build needs to be run from a fully qualified path

When running a headless build using the scripts provided by pde build, the properties builder and buildDirectory must refer to a fully qualified path. (bug [139554](https://bugs.eclipse.org/bugs/show_bug.cgi?id=139554))

## 4. Running Eclipse

After installing the Eclipse SDK in a directory, you can start the Workbench by running the Eclipse executable included with the release (you also need a 1.4.2 JRE, not included with the Eclipse SDK). On Windows, the executable file is called eclipse.exe, and is located in the eclipse sub-directory of the install. If installed at c:\eclipse-SDK-3.3-win32, the executable is c:\eclipse-SDK-3.3-win32\eclipse\eclipse.exe. **Note:** Set-up on most other operating environments is analogous. Special instructions for Mac OS X are listed [below](#_2jxsxqh).

### Allocating enough memory and solving OutOfMemoryErrors

By default, Eclipse will allocate up to 256 megabytes of Java heap memory. This should be ample for all typical development tasks. However, depending on the JRE that you are running, the number of additional plug-ins you are using, and the number of files you will be working with, you could conceivably have to increase this amount. Eclipse allows you to pass arguments directly to the Java VM using the -vmargs command line argument, which must follow all other Eclipse specific arguments. Thus, to increase the available heap memory, you would typically use:

eclipse -vmargs -Xmx<memory size>

with the <memory size> value set to greater than "256M" (256 megabytes -- the default).

When using a Sun VM, you may also need to increase the size of the permanent generation memory. The default maximum is 64 megabytes, but more may be needed depending on your plug-in configuration and use. When the VM runs out of permanent generation memory, it may crash or hang during class loading. This failure is less common when using Sun JRE version 1.5.0\_07 or greater. The maximum permanent generation size is increased using the -XX:MaxPermSize=<memory size> argument:

eclipse -vmargs -XX:MaxPermSize=<memory size>

This argument may not be available for all VM versions and platforms; consult your VM documentation for more details.

Note that setting memory sizes to be larger than the amount of available physical memory on your machine will cause Java to "thrash" as it copies objects back and forth to virtual memory, which will severely degrade your performance.

### Selecting a workspace

When the Workbench is launched, the first thing you see is a dialog that allows you to select where the workspace will be located. The workspace is the directory where your work will be stored. If you do not specify otherwise, Eclipse creates the workspace in your user directory. This workspace directory is used as the default content area for your projects as well as for holding any required metadata. For shared or multi-workspace installs you must explicitly specify the location for your workspace using the dialog (or via the "-data" command line argument).

Here is a typical Eclipse command line:

eclipse -vm c:\jdk1.4.2\jre\bin\javaw

*Tip:* It's generally a good idea to explicitly specify which Java VM to use when running Eclipse. This is achieved with the "-vm" command line argument as illustrated above. If you don't use "-vm", Eclipse will look on the O/S path. When you install other Java-based products, they may change your path and could result in a different Java VM being used when you next launch Eclipse.

To create a Windows shortcut to an installed Eclipse:

1. Navigate to eclipse.exe in Windows Explorer and use Create Shortcut on the content menu.
2. Select the shortcut and edit its Properties. In the Target: field append the command line arguments.

Opening this shortcut launches Eclipse. (You can drag the shortcut to the Windows Desktop if you want to keep it in easy reach.)

### Mac OS X

On Mac OS X, you start Eclipse by double clicking the Eclipse application. If you need to pass arguments to Eclipse, you'll have to edit the eclipse.ini file inside the Eclipse application bundle: select the Eclipse application bundle icon while holding down the Control Key. This will present you with a popup menu. Select "Show Package Contents" in the popup menu. Locate eclipse.ini file in the Contents/MacOS sub-folder and open it with your favorite text editor to edit the command line options.

On MacOS X you can only launch a UI program more then once if you have separate copies of the program on disk. The reason for this behavior is that every UI application on Mac can open multiple documents, so typically there is no need to open a program twice. Since Eclipse cannot open more than one workspace, this means you have to make a copy of the Eclipse install if you want to open more then one workspace at the same time (bug [139319](https://bugs.eclipse.org/bugs/show_bug.cgi?id=139319)).

If you need to launch Eclipse from the command line, you can use the symbolic link "eclipse" in the top-level eclipse folder. It refers to the eclipse executable inside the application bundle and takes the same arguments as "eclipse.exe" on other platforms.

On Mac OS X 10.4 and later, you may notice a slow down when working with significant numbers of resources if you allow Spotlight to index your workspace. To prevent this, start System Preferences, select the Spotlight icon, then the Privacy tab, then click the Add button ("+") and find your workspace directory in the dialog that appears.

### Shared Install

The startup speed of a shared install can be improved if proper cache information is stored in the shared install area. To achieve this, after unzipping Eclipse distribution, run Eclipse once with the "-initialize" option from an account that has a write access to the install directory.

## 5. Upgrading Workspace from a Previous Release

### Users who don't use "-data"

If you weren't previously using "-data" to specify your workspace, follow these steps to upgrade:

1. Find the workspace directory used by your old version of Eclipse. Typically this is located inside the directory in which Eclipse was installed in a sub-directory called "workspace". If you are using a shortcut or script to launch Eclipse, then it will be under the current working directory of that shortcut or script in a sub-directory called "workspace". For Windows users, this is specified by the "Start in:" argument in your shortcut properties.
2. Copy this workspace directory to a new, empty location outside of any Eclipse install directory.
3. Install the new version of Eclipse in a new location, separate from any old version of Eclipse.
4. If you had installed additional features and plug-ins into your old Eclipse, you should re-install them in the new Eclipse.
5. Start this new version of Eclipse and select this location using the workspace chooser dialog at startup (or use "-data" command line argument to pre-select the workspace location).

### Users who do use "-data"

If you were previously using the "-data" argument to start Eclipse, your upgrade path is much easier:

1. Optionally copy your workspace directory to a new, empty location outside of any Eclipse install directory as a backup.
2. Install the new version of Eclipse in a new location, separate from any old versions of Eclipse.
3. If you had installed additional features and plug-ins into your old Eclipse, you should re-install them in the new Eclipse.
4. Start this new version of Eclipse and select this location using the workspace chooser dialog at startup (or use "-data" command line argument to pre-select the workspace location).

*Note:* Copying your workspace is recommended because, after you've upgraded your workspace, you won't be able to use it again with an older version of Eclipse. If you ever want to go "back in time" to an earlier release, you will need that backup.

## 6. Interoperability with Previous Releases

### 6.1 Interoperability of Release 3.3 with previous releases

#### Sharing projects between heterogeneous Eclipse 3.3 and 3.2

Special care is required when a project in a team repository is being loaded and operated on by developers using Eclipse-based products based on different feature or plug-in versions. The general problem is that the existence, contents, and interpretation of metadata files in the workspaces may be specific to a particular feature or plug-in version, and differ between versions. The workspace compatibility guarantees only cover cases where all developers upgrade their Eclipse workspaces in lock step. In those cases there should be no problem with shared metadata. However, when some developers are working in Eclipse 3.3 while others are working in Eclipse 3.2, there are no such guarantees. This section provides advice for what to do and not to do. It addresses the specific issues with the Eclipse SDK.

The typical failure mode is noticed by the 3.3 user. 3.3 metadata is lost when a 3.2 user saves changes and then commits the updated metadata files to the repository. Here's how things typically go awry:

* A user working in Eclipse 3.3 creates or modifies a project in a way that results in changes to a shared metadata file that rely on 3.3-specific information. The user then commits the updated project files, including the shared metadata file, to the shared repository.
* Another user working in Eclipse 3.2 shares this project from the same repository. The 3.3-specific information in the shared metadata file is not understood by Eclipse 3.2, and is generally discarded or ignored without warning. The user modifies the project in a way that results in changes to the shared metadata file, causing the shared metadata file to be rewritten without any of the 3.3-specific information. The user commits the updated project files, including the shared metadata file, to the shared repository. The user is generally unaware that shared information has just been lost as a result of their actions.
* A user working in Eclipse 3.3 picks up the changes to a project from the shared repository, including the updated shared metadata file. The user may be unaware that they have just taken a retrograde step until later when things start to malfunction.

Here are some things to watch out for when sharing projects between Eclipse 3.3 and 3.1:

* **Linked resources in the .project file**  
  Eclipse 3.3 supports creating linked resources at arbitrary depth within a project, and supports creating linked resources referring to other file systems. Neither of these scenarios are supported in Eclipse 3.1 or earlier. If such linked resources are created in 3.3, and the project is subsequently loaded into an Eclipse 3.1 or earlier workspace, these links will not be recognized. Recommendation: avoid creating links at arbitrary depth or to other file systems where project compatibility with Eclipse 3.1 or earlier is required.

#### Using Eclipse 3.3 to develop plug-ins that work in Eclipse 3.2

It is also possible (and reasonable) to use Eclipse 3.3 to develop a plug-in intended to work in Eclipse 3.2 or earlier. Use the **Plug-in Development > Target Platform** preference page to locate non-workspace plug-ins in an Eclipse 3.2 install. This ensures that the code for your plug-in is being compiled and tested against Eclipse 3.2 APIs, extension points, and plug-ins. (The above list of concerns do not apply since they affect the layout and interpretation of files in the plug-in *project* but none affect the actual deployed form of the plug-in.)

## 7. Defects Fixed in Maintenance Releases

### 7.1 Defects fixed in release 3.3.1.1 since 3.3.1

Release 3.3.1.1 is a maintenance release to fix serious defects present in release 3.3.1 These changes only affect some plug-ins and features. Modified plug-ins have version id "3.3.1"; plug-ins unchanged since the 3.2 release still have version id "3.3.0"; Note, however, that all features now have version id "3.3.1" (even if none of their plug-ins changed).

Maintenance release 3.3.1.1 contains fixes for the following defects and others:

| ID | Summary |
| --- | --- |
| [201784](https://bugs.eclipse.org/bugs/show_bug.cgi?id=201784) | [CommonNavigator] 2 xml example projects show up in project explorer |
| [203325](https://bugs.eclipse.org/bugs/show_bug.cgi?id=203325) | Wrong eclipse.ini configuration for PermGen size |
| [205622](https://bugs.eclipse.org/bugs/show_bug.cgi?id=205622) | Add -XX:PermGen to eclipse.ini for Mac OS X |
| [206095](https://bugs.eclipse.org/bugs/show_bug.cgi?id=206095) | swt-3.3-gtk-linux-x86 Printer crash under Ubuntu Gutsy |
| [206959](https://bugs.eclipse.org/bugs/show_bug.cgi?id=206959) | NPE in IdReplaceTask: backport to 3.3.1.1 |
| [207046](https://bugs.eclipse.org/bugs/show_bug.cgi?id=207046) | Update features to indicate 3.3.1.1 in about |

Note: Bug fixes since the 3.3 release can be obtained by the following the Bugzilla query: [https://bugs.eclipse.org/bugs/buglist.cgi&query\_format=advanced&short\_desc\_type=allwordssubstr&short\_desc=&classification=Eclipse&product=Equinox&product=JDT&product=PDE&product=Platform&target\_milestone=3.3.1&target\_milestone=3.3.1&long\_desc\_type=allwordssubstr&long\_desc=&bug\_file\_loc\_type=allwordssubstr&bug\_file\_loc=&status\_whiteboard\_type=allwordssubstr&status\_whiteboard=&keywords\_type=allwords&keywords=&bug\_status=RESOLVED&bug\_status=VERIFIED&bug\_status=CLOSED&resolution=FIXED&emailtype1=substring&email1=&emailtype2=substring&email2=&bugidtype=include&bug\_id=&votes=&chfieldfrom=&chfieldto=Now&chfieldvalue=&cmdtype=doit&order=Importance&field0-0-0=noop&type0-0-0=noop&value0-0-0=|](https://bugs.eclipse.org/bugs/buglist.cgi?query_format=advanced&short_desc_type=allwordssubstr&short_desc&classification=Eclipse&product=Equinox&product=JDT&product=PDE&product=Platform&target_milestone=3.3.1&target_milestone=3.3.1&long_desc_type=allwordssubstr&long_desc&bug_file_loc_type=allwordssubstr&bug_file_loc&status_whiteboard_type=allwordssubstr&status_whiteboard&keywords_type=allwords&keywords&bug_status=RESOLVED&bug_status=VERIFIED&bug_status=CLOSED&resolution=FIXED&emailtype1=substring&email1&emailtype2=substring&email2&bugidtype=include&bug_id&votes&chfieldfrom&chfieldto=Now&chfieldvalue&cmdtype=doit&order=Importance&field0-0-0=noop&type0-0-0=noop&value0-0-0=%7C)

Sun, Solaris, Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, Vista, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Apple and Mac OS are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

QNX, Neutrino, and Photon are trademarks or registered trademarks of QNX Software Systems Ltd.

Other company, product, and service names may be trademarks or service marks of others.

(c) Copyright IBM Corp. and others 2007

## Appendix 1: Execution Environment by Plug-in

In the table below, the "3.3 EE" ("3.3 Execution Environment") column indicates the minimum Java class library requirements of each plug-in for the 3.3 release, where the value is one of:

| **Entry** | **Meaning** |
| --- | --- |
| **M1.0** | OSGi Minimum Execution Environment 1.0 - This is a subset of the J2ME Foundation class libraries defined by OSGi to be the base for framework implementations. See the OSGi specification for more details. |
| **M1.1** | OSGi Minimum Execution Environment 1.1 - This is a subset of the J2ME Foundation class libraries defined by OSGi to be the base for framework implementations. See the OSGi specification for more details. |
| **F1.0** | J2ME Foundation 1.0 - indicates that the plug-in can only be run on Foundation 1.0 or greater. Note that with the exception of some MicroEdition IO classes, Foundation 1.0 is a subset of J2SE 1.3. |
| **F1.1** | J2ME Foundation 1.1 - indicates that the plug-in can only be run on Foundation 1.1 or greater. Note that with the exception of some MicroEdition IO classes, Foundation 1.1 is a subset of J2SE 1.4. |
| **1.2** | J2SE 1.2 - indicates that the plug-in can only be run on JSE 1.2 or greater. |
| **1.3** | J2SE 1.3 - indicates that the plug-in can only be run on JSE 1.3 or greater. |
| **1.4** | J2SE 1.4 - indicates that the plug-in can only be run on JSE 1.4 or greater. |
| **1.4/1.5** | Indicates that the plug-in can run on JSE 1.4 or greater, but provides enhanced functionality when run on J2SE 5.0. |
| **1.5** | J2SE 5.0 - indicates that the plug-in can only be run on JSE 5.0 or greater. |
| **1.6** | J2SE 6.0 - indicates that the plug-in can only be run on JSE 6.0 or greater. |
| **n/a** | Not applicable (for example plug-ins that do not contain Java code) |

**Table of minimum execution environments by plug-in.**

| **Plug-in** | **3.3 EE** |
| --- | --- |
| javax.servlet | F1.0 |
| javax.servlet.jsp | F1.0 |
| org.apache.ant | 1.2 |
| org.apache.commons.el | F1.0 |
| org.apache.commons.logging | F1.0 |
| org.apache.jasper | F1.0 |
| org.apache.lucene | n/a |
| org.eclipse.ant.core | 1.4 |
| org.eclipse.ant.ui | 1.4 |
| org.eclipse.compare | 1.4 |
| org.eclipse.core.boot | F1.0 |
| org.eclipse.core.commands | F1.0 |
| org.eclipse.core.contenttype | F1.0 |
| org.eclipse.core.expressions | F1.0 |
| org.eclipse.core.filebuffers | 1.4 |
| org.eclipse.core.filesystem | 1.4 |
| org.eclipse.core.jobs | F1.0 |
| org.eclipse.core.net | 1.4 |
| org.eclipse.core.resources | 1.4 |
| org.eclipse.core.resources.compatibility | 1.4 |
| org.eclipse.core.runtime | F1.0 |
| org.eclipse.core.runtime.compatibility | 1.4 |
| org.eclipse.core.runtime.compatibility.auth | F1.0 |
| org.eclipse.core.runtime.compatibility.registry | F1.0 |
| org.eclipse.core.variables | 1.4 |
| org.eclipse.debug.core | 1.4 |
| org.eclipse.debug.ui | 1.4 |
| org.eclipse.equinox.app | F1.0 |
| org.eclipse.equinox.common | F1.0 |
| org.eclipse.equinox.http.jetty | F1.0 |
| org.eclipse.equinox.http.servlet | F1.0 |
| org.eclipse.equinox.http.registry | F1.0 |
| org.eclipse.equinox.jsp.jasper | F1.0 |
| org.eclipse.equinox.jsp.jasper.registry | F1.0 |
| org.eclipse.equinox.launcher | F1.0 |
| org.eclipse.equinox.preferences | F1.0 |
| org.eclipse.equinox.registry | F1.0 |
| org.eclipse.help | F1.0 |
| org.eclipse.help.appserver | F1.0 |
| org.eclipse.help.base | F1.0 |
| org.eclipse.help.ui | F1.0 |
| org.eclipse.help.webapp | F1.0 |
| org.eclipse.jdt | 1.4 |
| org.eclipse.jdt.apt.core | 1.5 |
| org.eclipse.jdt.apt.ui | 1.5 |
| org.eclipse.jdt.compiler.apt | 1.6 |
| org.eclipse.jdt.compiler.tool | 1.6 |
| org.eclipse.jdt.core | 1.4 |
| org.eclipse.jdt.core.manipulation | 1.4 |
| org.eclipse.jdt.debug | 1.4 |
| org.eclipse.jdt.debug.ui | 1.4 |
| org.eclipse.jdt.doc.isv | n/a |
| org.eclipse.jdt.doc.user | n/a |
| org.eclipse.jdt.junit | 1.4 |
| org.eclipse.jdt.junit.runtime | 1.4 |
| org.eclipse.jdt.junit4.runtime | 1.5 |
| org.eclipse.jdt.launching | 1.4 |
| org.eclipse.jdt.source | n/a |
| org.eclipse.jdt.ui | 1.4 |
| org.eclipse.jface | F1.0 |
| org.eclipse.jface.text | 1.4 |
| org.eclipse.jsch.core | 1.4 |
| org.eclipse.jsch.ui | 1.4 |
| org.eclipse.ltk.core.refactoring | 1.4 |
| org.eclipse.ltk.ui.refactoring | 1.4 |
| org.eclipse.osgi (system.bundle) | M1.0 |
| org.eclipse.osgi.services | M1.0 |
| org.eclipse.osgi.util | M1.0 |
| org.eclipse.pde | 1.4 |
| org.eclipse.pde.build | 1.4 |
| org.eclipse.pde.core | 1.4 |
| org.eclipse.pde.doc.user | n/a |
| org.eclipse.pde.junit.runtime | 1.4 |
| org.eclipse.pde.runtime | 1.4 |
| org.eclipse.pde.source | n/a |
| org.eclipse.pde.ui | 1.4 |
| org.eclipse.pde.ui.templates | 1.4 |
| org.eclipse.platform | F1.0 |
| org.eclipse.platform.doc.isv | n/a |
| org.eclipse.platform.doc.user | n/a |
| org.eclipse.platform.source | n/a |
| org.eclipse.platform.source.\* | n/a |
| org.eclipse.rcp | F1.0 |
| org.eclipse.rcp.source | n/a |
| org.eclipse.rcp.source.\* | n/a |
| org.eclipse.sdk | n/a |
| org.eclipse.search | 1.4 |
| org.eclipse.swt | M1.0 |
| org.eclipse.swt.\* | M1.0 |
| org.eclipse.team.core | 1.4 |
| org.eclipse.team.cvs.core | 1.4 |
| org.eclipse.team.cvs.ssh | 1.4 |
| org.eclipse.team.cvs.ssh2 | 1.4 |
| org.eclipse.team.cvs.ui | 1.4 |
| org.eclipse.team.ui | 1.4 |
| org.eclipse.text | 1.4 |
| org.eclipse.tomcat | 1.4 |
| org.eclipse.ui | F1.0 |
| org.eclipse.ui.browser | 1.4 |
| org.eclipse.ui.cheatsheets | F1.0 |
| org.eclipse.ui.console | 1.4 |
| org.eclipse.ui.editors | 1.4 |
| org.eclipse.ui.externaltools | 1.4 |
| org.eclipse.ui.forms | F1.0 |
| org.eclipse.ui.ide | 1.4 |
| org.eclipse.ui.intro | F1.0 |
| org.eclipse.ui.navigator | 1.4 |
| org.eclipse.ui.navigator.resources | 1.4 |
| org.eclipse.ui.net | 1.4 |
| org.eclipse.ui.presentations.r21 | 1.4 |
| org.eclipse.ui.views | 1.4 |
| org.eclipse.ui.win32 | 1.4 |
| org.eclipse.ui.workbench | F1.0 |
| org.eclipse.ui.workbench.compatibility | 1.4 |
| org.eclipse.ui.workbench.texteditor | 1.4 |
| org.eclipse.update.configurator | F1.0 |
| org.eclipse.update.core | F1.0 |
| org.eclipse.update.core.linux | F1.0 |
| org.eclipse.update.core.win32 | F1.0 |
| org.eclipse.update.scheduler | F1.0 |
| org.eclipse.update.ui | F1.0 |
| org.junit (old) | 1.4 |
| org.junit (JUnit4) | 1.5 |
| org.mortbay.jetty | F1.0 |