

**WebSphere Application Server**

# **Application Migration Tool for WebSphere Version Migration**

Built on Rational Software Analyzer technology

Version 3.0.0  
June 17, 2011

A combined effort:  
**IBM Software Group, Application and Integration Middleware Software**  
**IBM Software Group, Rational Software**

© Copyright IBM Corp. 2009, 2011

# Contents

<b>Overview</b> .....	<b>3</b>
<b>New for this release</b> .....	<b>4</b>
Version 3.0.....	4
Version 2.1.....	4
<b>Support</b> .....	<b>5</b>
<b>Installing and updating the migration tools</b> .....	<b>6</b>
<b>Importing applications</b> .....	<b>10</b>
Shared Java projects.....	10
EAR-level library.....	10
WAR-level library.....	12
Configuring a Java EE Runtime Library.....	13
<b>Configuring the migration tool for analysis</b> .....	<b>16</b>
<b>Analyzing code for migration</b> .....	<b>20</b>
<b>Running additional rules to optimize code quality</b> .....	<b>22</b>
<b>WebSphere Application Server version migration rules and quick fixes</b> .....	<b>23</b>
Java code review.....	23
JSP code review.....	29
XML code review.....	30
Deprecated features.....	31
<b>Troubleshooting</b> .....	<b>39</b>
Software Analyzer options not shown.....	39
Java EE constructs or JSP not read correctly.....	39
Logging and trace.....	39
Reports and history.....	40
Markers.....	40
Known issues.....	40
<b>Copyright and trademarks</b> .....	<b>43</b>

# Overview

---

The Application Migration Tool - WebSphere Version to Version feature is part of the IBM® WebSphere® Application Server Migration Toolkit. This toolkit is part of the overall strategy and plan for assisting customers in migrating from one application server to the IBM WebSphere Application Server. The Application Migration Tool - WebSphere Version to Version feature provides support for migrating applications from older versions of WebSphere Application Server to WebSphere Application Server Version 7.0 or Version 8.0.

The overall migration process between versions of WebSphere Application Server involves a series of steps to assess the migration, plan the work involved, migrate and develop the code, manage the runtime configuration, test the results, and roll out the new release. This tool helps migrate and develop the code.

There are a number of issues that affect the code migration and development step when moving between WebSphere Application Server releases. These include:

- Changes to the Java Runtime Environment (JRE)
- Changes to the level of supported Java Enterprise Edition version
- Removal of previously deprecated features
- Behavior changes in the product
- Deprecated features

The migration process can involve modifying Java source code, JavaServer Pages (JSP), and deployment descriptors. The application migration tool can assist you in performing these types of code changes.

This document explains how to install, configure, and use the tool to assist in the conversion process. The migration tool is based on IBM Rational® Software Analyzer, which provides a single solution to identify, analyze, and optimize the application health. The migration tool uses the scanning capabilities of Rational Software Analyzer to look for specific constructs that have changed between versions of WebSphere Application Server that can affect compilation and application server runtime behaviors. The tool then provides a way to review problematic code, get help on the issue, and in some cases make automatic fixes to the code so that the application can run on IBM WebSphere Application Server Version 7.0 or 8.0.

For WebSphere Application Server version migration, the tool supports:

- Migrating applications from WebSphere Application Server Version 5.1 to Version 7.0
- Migrating applications from WebSphere Application Server Version 6.0 to Version 7.0
- Migrating applications from WebSphere Application Server Version 6.1 to Version 7.0
- Migrating applications from WebSphere Application Server Version 5.1 to Version 8.0
- Migrating applications from WebSphere Application Server Version 6.0 to Version 8.0
- Migrating applications from WebSphere Application Server Version 6.1 to Version 8.0
- Migrating applications from WebSphere Application Server Version 7.0 to Version 8.0

See the Application Migration Tools for Competitive Migration documentation for information about competitive migration.

## New for this release

---

### Version 3.0

---

Application Migration Tool - WebSphere Version to Version V3.0 adds rules to help migrate applications to WebSphere Application Server Version 8.0.

- New Java, XML, and JSP file rules to migrate to WebSphere Application Server Version 8.0.
- New rule sets to help select the rules needed for migrating from Version 5.1, 6.0, 6.1, or 7.0 to Version 8.0.
- New deprecation rules for Version 6.1, 7.0, and 8.0.

### Version 2.1

---

Application Migration Tool - WebSphere Version to Version V2.1 adds support to existing rules and quick fixes to assist with the migration of applications from one version of WebSphere Application Server to another.

The emphasis for this release was to add new rules to flag classes, interfaces, methods, and fields that were deprecated in WebSphere Application Server Version 5.1 and 6.0. Rules were added in the Java, XML and JSP categories.

#### **New WebSphere Application Server version migration rules and quick fixes include:**

- 46 Java file deprecation rules with five quick fixes
- One JSP file deprecation rule
- Two XML file deprecation rules

#### **The tool is translated into the following languages:**

- German (de)
- Spanish (es)
- French (fr)
- Japanese (ja)
- Korean (ko)
- Brazilian Portuguese (pt\_BR)

## Support

---

The application migration tool scans for a number of known issues in applications being migrated from WebSphere Application Server Version 5.1, 6.0, 6.1, or 7.0 to WebSphere Application Server Version 7.0 or 8.0. Where possible, a quick fix is provided to change your code to address the changes between versions. Supported migration rules and quick fixes are documented in the section, *WebSphere Application Server version migration rules and quick fixes* on page 23. As new rules and quick fixes are available, updates are made on developerWorks™.

You can use the quick fix preview support in the migration tool to decide if you want to accept the suggested code change. Also, view the help information provided with the rules to decide if you want to run the quick fix. Always make a backup copy of your source code before starting a migration.

For some rules, the scan detects code that requires design changes and code rewrites. The tool highlights these problem areas but does not provide a quick fix.

The tool does not identify all problems. As you encounter problems that the tool does not flag, provide that feedback through the Application Migration Tool forum, which is available to provide input and get questions answered. (<http://www.ibm.com/developerworks/forums/forum.jspa?forumID=2106>).

If you have access to IBM Passport Advantage, you can open a customer defect. Other users can use the forum to report issues or suggestions and ask questions.

For more information on the Rational Software Analyzer product, go to <http://www.ibm.com/developerworks/rational/products/rsar/>.

## Installing and updating the migration tools

The application migration tools are Eclipse features that you install into an existing Eclipse or Rational Application Developer environment. The preferred environment is Rational Application Developer since it provides all the tools for application development and deployment. However, the tool works with a basic Eclipse IDE. If an Eclipse IDE is not already installed, then download and install Eclipse from <http://www.eclipse.org>.

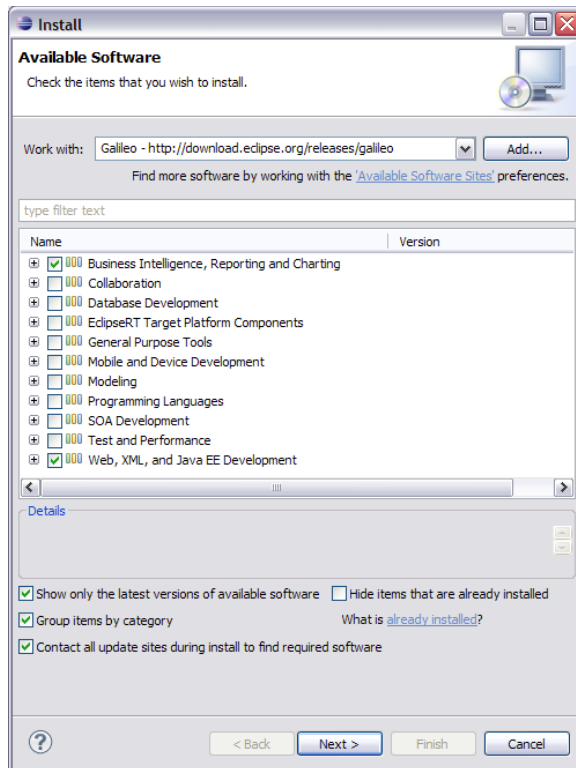
The tool is supported on Eclipse 3.4.2 or higher and Rational Application Developer Version 7.5 and higher.

To install or update this application migration tool, perform the following steps:

1. Download the latest version of the Application Migration Tool - WebSphere Version to Version feature from the developerWorks site [http://www.ibm.com/developerworks/websphere/downloads/migration\\_toolkit.html](http://www.ibm.com/developerworks/websphere/downloads/migration_toolkit.html) and save it locally.
2. Start the IDE.
3. Uninstall versions 1.0, 1.1, 1.2, and 2.0 of the competitive migration tools before installing Version 3.0. Version 2.1 of the application migration tools can be upgraded to Version 3.0 with one exception. Uninstall any previous releases of the Application Migration Tool if you are installing on Rational Application Developer Version 8.0.3.
4. If you are using a Rational Application Developer environment, all prerequisite plug-ins are typically installed by default. During installation, verify that the Business Intelligence and Reporting Tools and the Web Development Tools features are selected. If you are not using a Rational Application Developer environment, install the prerequisite plug-ins from either the Helios, Galileo, or Ganymede site depending on your Eclipse version.

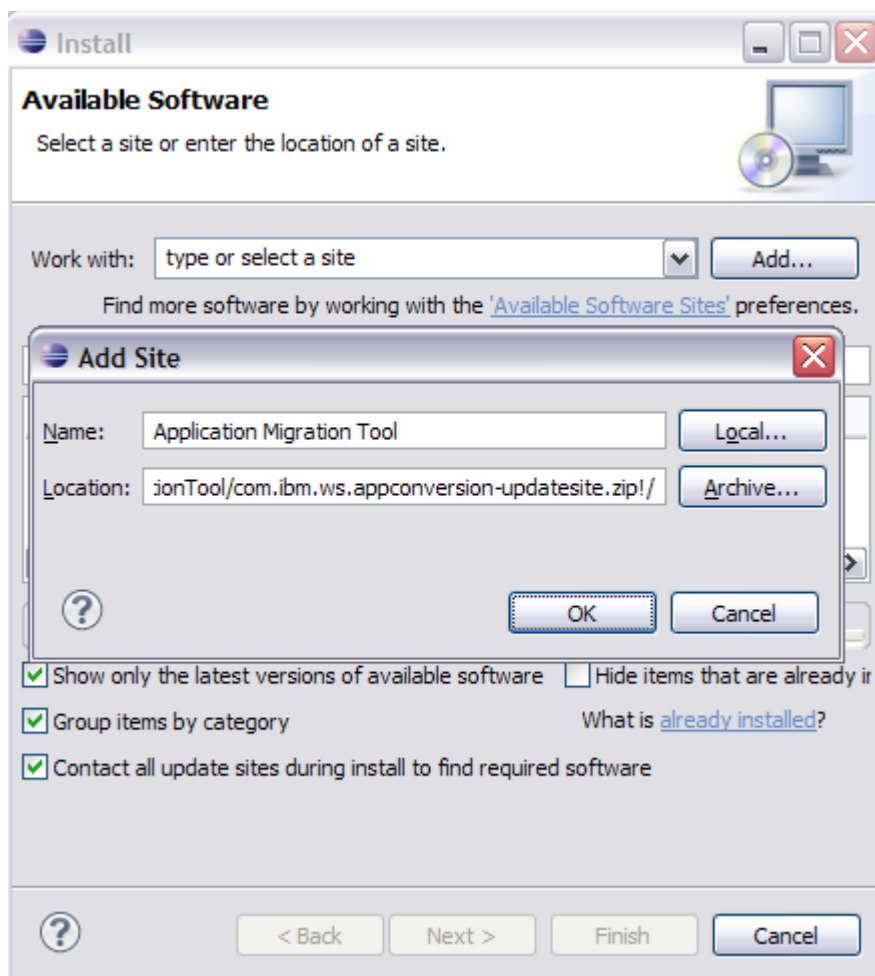
Eclipse version	Install actions
Eclipse 3.6	<ul style="list-style-type: none"> <li>• From the Eclipse menu bar, select <b>Help &gt; Install New Software</b>.</li> <li>• Using the Helios update site (<a href="http://download.eclipse.org/releases/helios">http://download.eclipse.org/releases/helios</a>) install <b>Business Intelligence, Reporting and Charting</b> and <b>Web, XML and Java EE Development</b>.</li> </ul>
Eclipse 3.5	<ul style="list-style-type: none"> <li>• From the Eclipse menu bar, select <b>Help &gt; Install New Software</b>.</li> <li>• Using the Galileo update site (<a href="http://download.eclipse.org/releases/galileo">http://download.eclipse.org/releases/galileo</a>) install <b>Business Intelligence, Reporting and Charting</b> and <b>Web, XML and Java EE Development</b>.</li> </ul>
Eclipse 3.4.2	<ul style="list-style-type: none"> <li>• From the Eclipse menu bar, select <b>Help &gt; Software updates</b>.</li> <li>• Using the Ganymede update site (<a href="http://download.eclipse.org/releases/ganymede">http://download.eclipse.org/releases/ganymede</a>), install <b>Charting and Reporting</b> and <b>Web and Java EE Development</b>.</li> </ul>

**Note:** The catalog of available plug-ins is large and can require several minutes to download.



**Figure 1: Eclipse 3.6 plug-in installation dialog**

5. Use the Eclipse installation dialog to install or update the Application Migration Toolkit. Click **Add** in Eclipse 3.5 or higher or click **Add Site** in Eclipse 3.4 to point to your downloaded update site file. Rational Application Developer Version 7.5 is built on Eclipse 3.4. Rational Application Developer Version 8.0 is built on Eclipse 3.6.
6. In the Add Site window, enter the following information, which is shown in *Figure 2: Tool installation*.
  - **Name:** Application Migration Tool
  - **Location:** Use the **Archive** button to find the compressed file you downloaded.



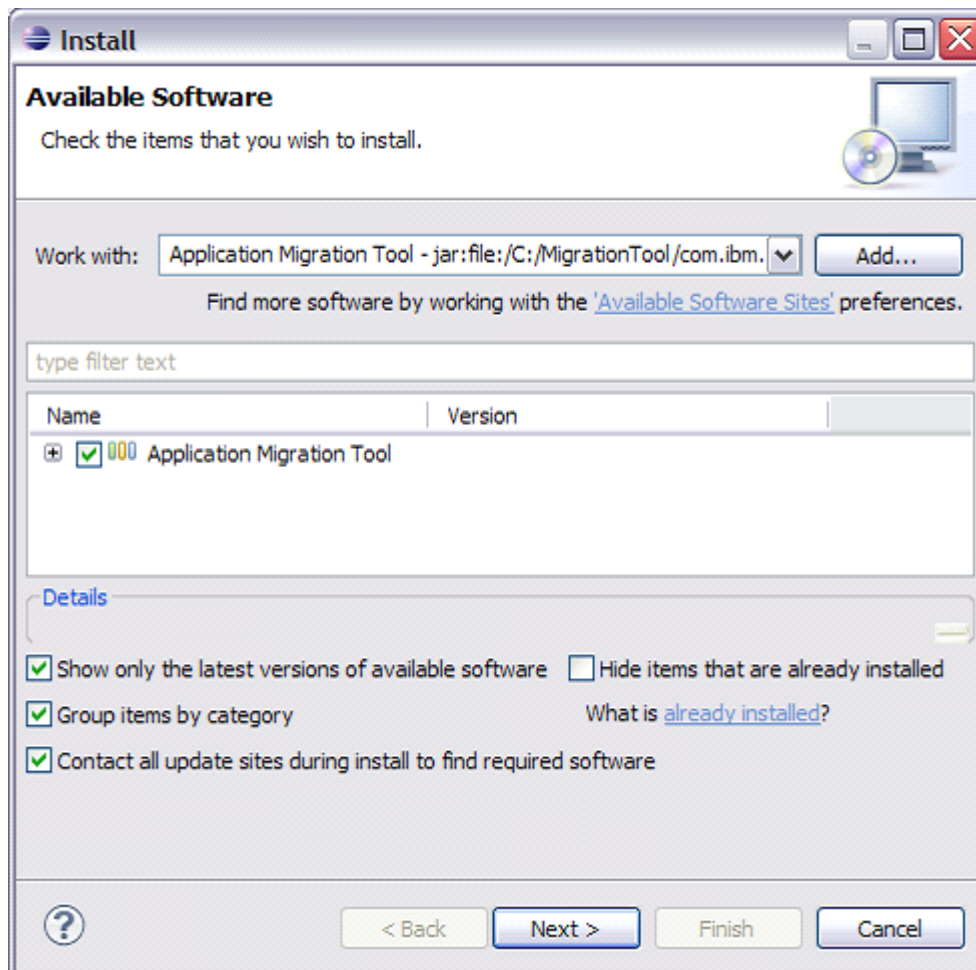
**Figure 2: Tool installation**

7. Click **OK**.
8. Complete the following actions in the **Install** window, which is shown in [Figure 3: Select Plug-In](#).
  - a) Select the check box for **Application Migration Tool** which selects both the main feature and the common feature. Selecting the common feature is particularly important when upgrading.
  - b) Select the check box to **Contact all update sites during install to find required software**, and click **Next**.

**Note:** If you are installing on Eclipse 3.4.2 or Rational Application Developer 7.5, it is preferable to not select **Uncategorized** from the repository site. These plug-ins will be installed automatically regardless of the selection. If **Uncategorized** is selected, the plug-ins will not be automatically uninstalled if you uninstall the application migration tool.

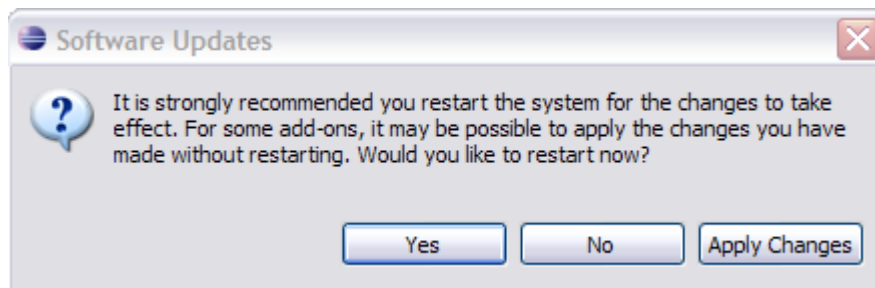
9. Click **Next** on the **Install Details** panel.
10. On the **Review Licenses** panel, read the terms and accept the license agreements, if any are presented. Click **Finish**. The install status window shows the installation progress.





**Figure 3: Select Plug-In**

- Restart the IDE when the Software Updates window is displayed.



**Figure 4: Restart dialog**

## Importing applications

---

If you want to migrate an application that is not already in the IDE, you must import the application enterprise archive (EAR) file or modules with source code into the IDE before performing analysis. The preferred approach for doing this is to create new projects in the workspace for each EAR, web archive (WAR), and Enterprise JavaBeans (EJB) module. Create the EAR file as an enterprise application project. Create each WAR file within the EAR file as a dynamic web project, and create each EJB module as an EJB project. You must install Java EE Developer Tools for Eclipse to create these projects.

Review the following guidelines for structuring projects:

- Java source code for a WAR file (for example, servlet, model, or utility classes), belongs in the `src` folder of the project. The `src` folder is defined and can be changed in the Java Build Path properties for the project. If the Java source code needs to be referenced by more than one WAR file, see the *Shared Java Projects* section of this document.
- Java source code for an EJB module can be placed in the `ejbModule` folder of the EJB project.
- Precompiled Java archive (JAR) libraries for a WAR file belong in the `WebContent/WEB-INF/lib` folder.
- EAR-level JAR libraries can be placed in the `EarContent` folder of the enterprise application project. If your project contains an `APP-INF/lib` folder, it can be placed in the `EarContent` folder; however, you must run the class path rule and its quick fix to update the class paths correctly.

## Shared Java projects

---

There are two options for referencing shared Java classes from a web project:

1. Create a *single copy* of the Java project JAR file in the EAR file. Each WAR file references the JAR file. This approach is preferable, since it reduces the size of the EAR file.
2. Create a copy of the JAR file in *each WAR file*. This approach works well if only one WAR file references the JAR file.

## EAR-level library

Complete the following steps to place Java source files in a separate project:

1. Create a new Java project and add the Java source into the `src` folder of the project.
2. Right-click the project in the Project view and select **Properties** to go to the Properties dialog of the EAR project. Select the **Java EE Module Dependencies** item from the left pane, and on the right pane select the check box next to the Java projects you want to reference.



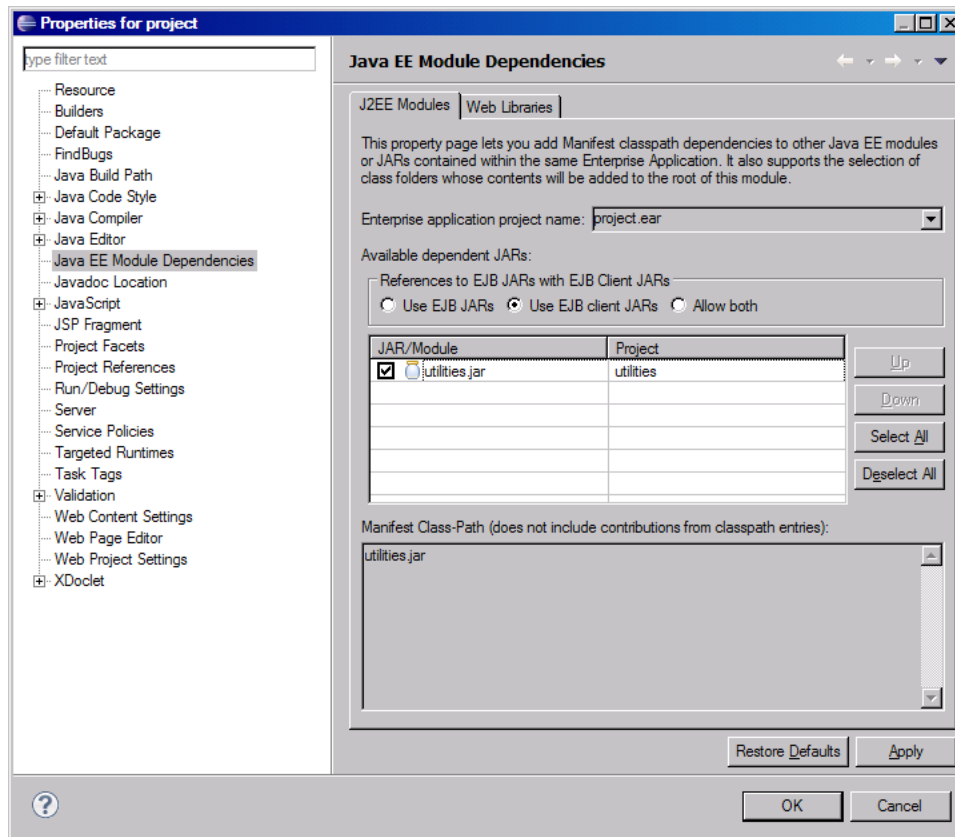


Figure 6: Java EE module dependencies

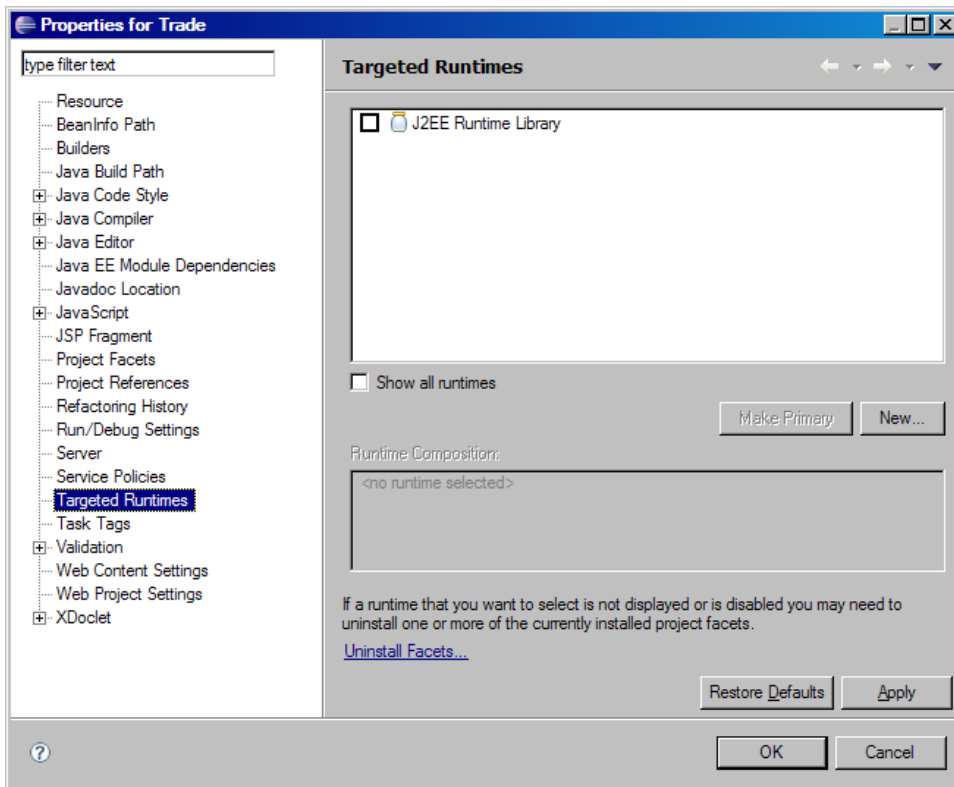
## WAR-level library

For the second approach, which creates a copy of the JAR file in each WAR file, complete the following steps:

1. Open the Properties dialog of the web project.
2. Right-click the project in the Project view and select **Properties**. Select the **Java EE Module Dependencies** item from the left pane, and then click the **Web Libraries** tab in the right pane. The table contains every Java project available.





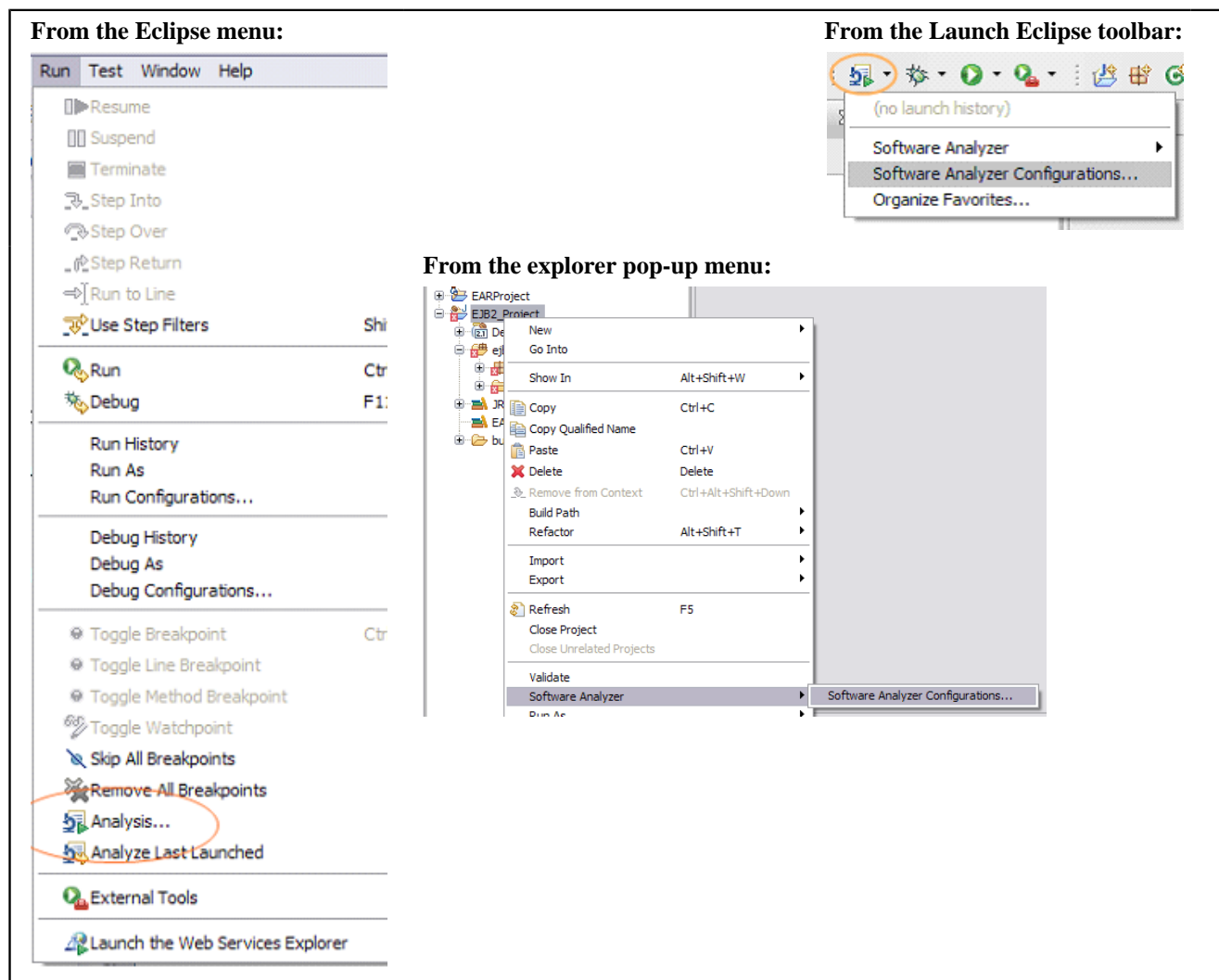


**Figure 9: Targeted Runtimes**

In the Targeted Runtimes window, select the check box next to the recently configured Java EE runtime library, and click **OK**.

## Configuring the migration tool for analysis

You can configure the tool to define a set of rules to run and define the scope of analysis within the workspace. The scope can be a project, a working set, or the entire workspace. After you define the scope, you can save the analysis configuration to use or modify later. With a migration tool installed, you have new analysis options to configure and run an analysis. You can access the options in the Eclipse **Run** menu, in the **Launch** toolbar, and in the explorer pop-up menus. The following figure demonstrates the different ways you can run the analysis using the application migration tools.



**Figure 10: Options for running the tool**

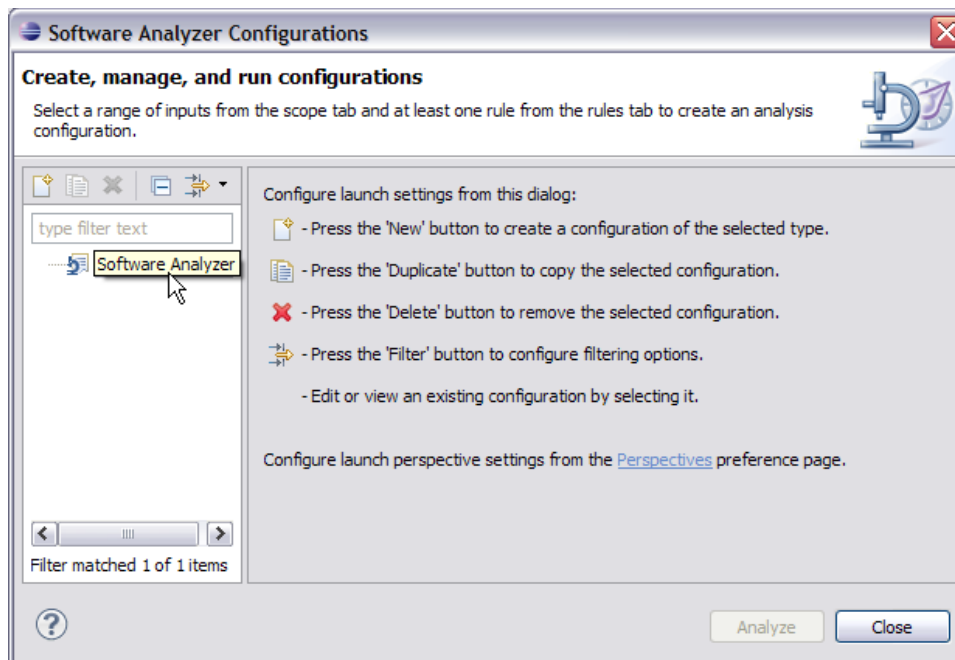
If you do not see Software Analyzer options, see [Software Analyzer options not shown](#) on page 39.


To configure the analysis using the toolbar, complete the following steps:

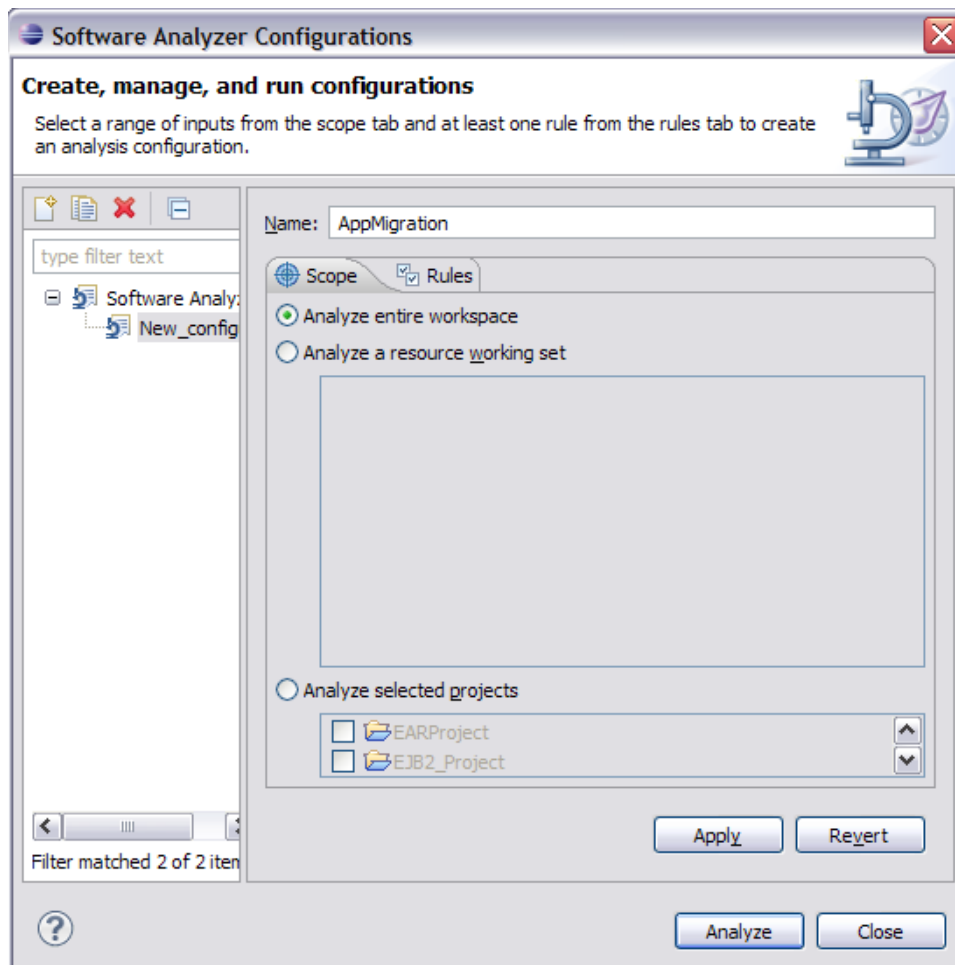
1. On the toolbar, select **Software Analyzer** (🔍) > **Software Analyzer Configurations** to display the main configuration dialog.

You can add or remove analysis configurations by using the icons in the dialog.





2. In the configurations list, select **Software Analyzer**. Then, click **New** . The right side of the dialog changes to show the basic configuration interface.

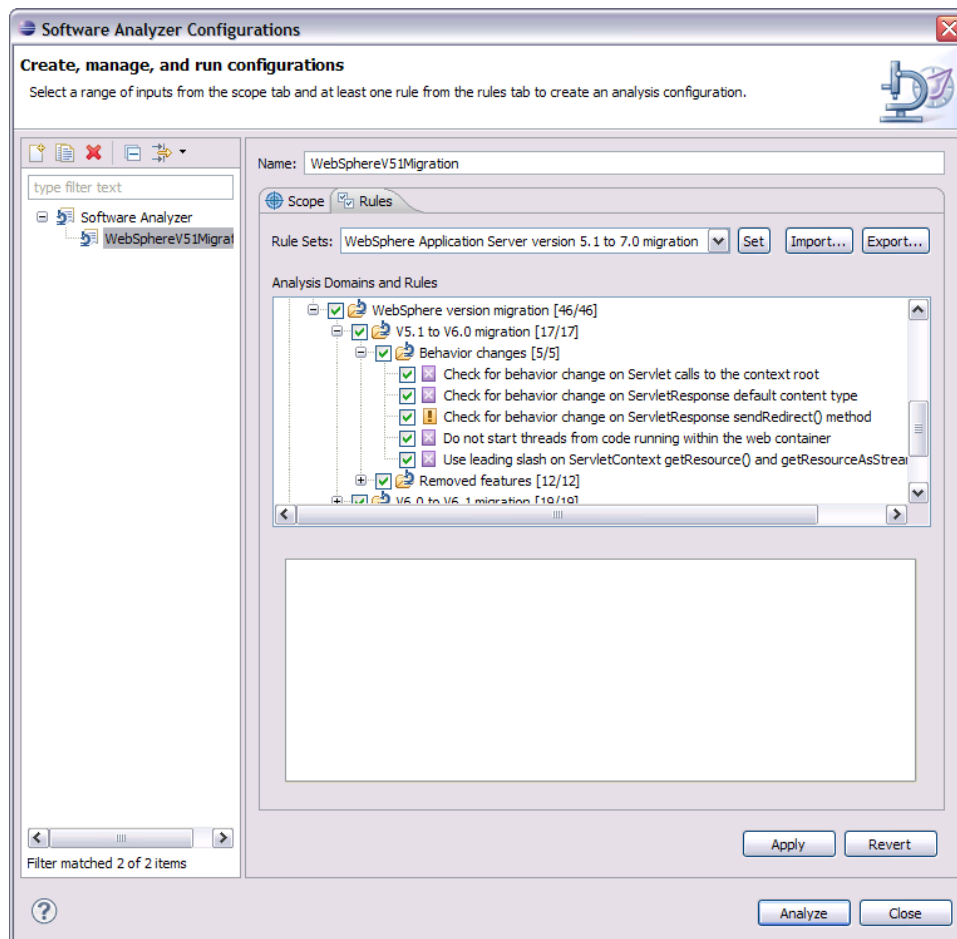


3. In the Software Analyzer Configuration dialog, enter a name for the configuration, such as “AppMigration”.
4. On the **Scope** tab, select **Analyze entire workspace** to scan all projects in the workspace.

You can limit the scope of an analysis by using the other options on this dialog to analyze a working set or a selection of projects.

**Tip:** When you use the explorer pop-up menu to run your analysis, the scope of the analysis is limited to the node in the project where the menu item was selected. This allows you to perform a quick analysis on a limited set of code.

5. On the **Rules** tab, select the type of analysis to perform.



There are seven migration related rule sets for WebSphere Application Server version migration:

- **WebSphere Application Server Version 5.1 to 7.0 migration**
- **WebSphere Application Server Version 6.0 to 7.0 migration**
- **WebSphere Application Server Version 6.1 to 7.0 migration**
- **WebSphere Application Server Version 5.1 to 8.0 migration**
- **WebSphere Application Server Version 6.0 to 8.0 migration**
- **WebSphere Application Server Version 6.1 to 8.0 migration**
- **WebSphere Application Server Version 7.0 to 8.0 migration**

Choose the rule set based on the version of WebSphere Application Server your application is currently running (5.1, 6.0, 6.1, or 7.0) and where you plan to run (7.0 or 8.0).

To automatically select all of the rules from one of these rule sets, select it from the list, and click **Set**.

The migration tools have rules under the following analysis providers:

- Java Code Review
- JSP Code Review
- XML File Review

**Tip:** To obtain additional information about a rule, highlight the rule and press F1. Help for the rule is shown in the configuration dialog. The initial help page includes a short description and a link to more information.

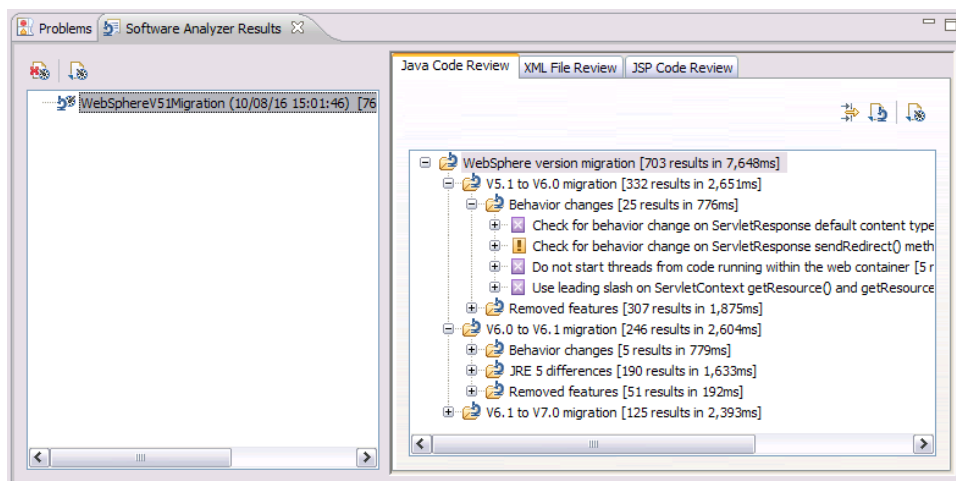
**Note:** The number of rules in the analysis configuration dialog varies depending on the platform on which the tool is installed. Analysis rules are available in several Rational products such as Rational Application Developer; therefore, the included rule sets might be different.

6. To save the rule configuration, click **Apply**.

## Analyzing code for migration

Run the analysis and display the results.

1. To start the analysis, click **Analyze** on the Configuration dialog.
2. The results are displayed in the Software Analysis Results view.

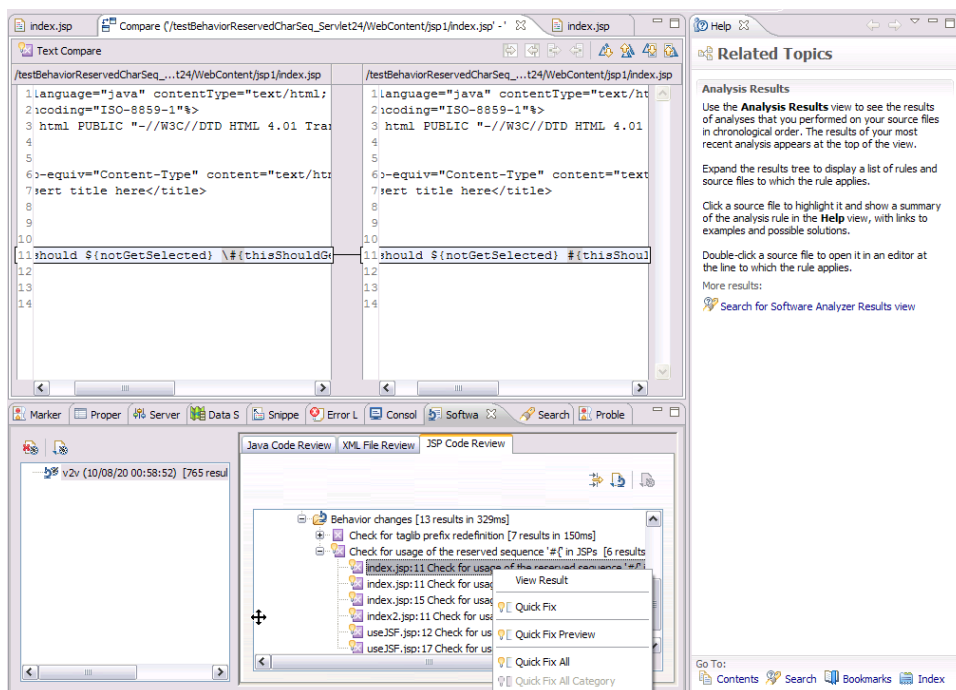


Depending on which rules you are running, the content of the results view might vary. The results generated by a migration tool are displayed in one of the following tabs:

- Java Code Review
- JSP Code Review
- XML File Review

If no results are shown in the panel, no issues were identified while scanning.


Right-clicking on individual results gives access to options such as viewing the source code where the problem occurred or correcting the problem with a provided fix.



Not all rules have all actions available, but the possible actions include:

- **View Result** - Opens an editor showing the source file that triggered the rule. The cause of the problem is highlighted, and a rule violation icon is shown in the left margin of the editor.

```
48 public String getClaim (String str1, String str2)
```

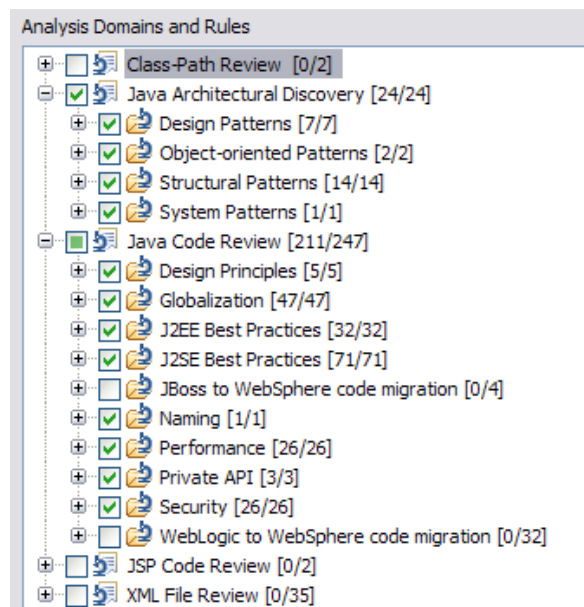
- **Quick Fix** - The light bulb overlay on the result list icon () indicates that this rule has a quick fix. Selecting this option runs the conversion that modifies the affected Java code, XML file, JSP or manifest file, allowing it to run in WebSphere Application Server. The quick fix might change the code directly or it might present the steps needed to complete the fix.
- **Quick Fix Preview** - This option is available for the rules that support showing a side-by-side comparison of the original code and the code after the quick fix is applied. This option allows you to see the changes before they are made.
- **Ignore Result** - This option removes the rule from the list without making a code change. For Java files, a comment annotation is added to the file so that the rule is not triggered on future analysis runs.
- **Quick Fix All** - This option resolves all issues identified for a given rule.
- **Quick Fix All Category** - This option runs all quick fixes identified for the category to which the rule belongs. A rule must have **Quick Fix All** enabled for the **Quick Fix All Category** option to run its quick fix. For example, if you choose this option on a Java rule, the quick fixes for all Java rules that have the Quick Fix All option are run.

Context-sensitive help is displayed in the Help view as you select each result. The first panel is a short description. Click **Detailed Help** to get more information. Press **F1** to open the Help view if it is not already displayed.

## Running additional rules to optimize code quality

---

After you run quick fixes or make manual code changes, and correct the issues highlighted by a migration tool, additional rules are provided by the Rational Software Analyzer product that can help improve the quality of your code. These rules are not required for conversion. *Figure 11: Other architectural and Java rules* shows examples of the additional rules.



**Figure 11: Other architectural and Java rules**

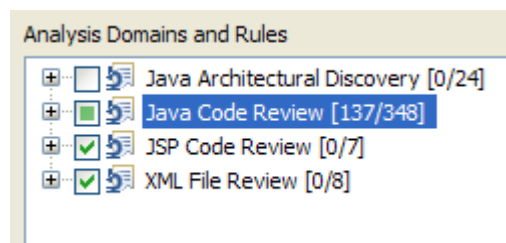
Access these rules by creating a new analysis configuration or modifying an existing one. These rules are not automatically selected as part of rule sets for the application migration tools.

When the selected rules run and necessary changes are made, the updated application must be exported and tested on WebSphere Application Server. If you use Rational Application Developer, tools are available to create deployments and test the application within the Rational Application Developer environment.

# WebSphere Application Server version migration rules and quick fixes

The Application Migration Tool - WebSphere Version to Version feature evaluates Java code, JSP code and deployment descriptors as part of its analysis set. This section provides specific details on the rules and quick fixes that are provided.

WebSphere Application Server conversion rules are included in the following analysis domains:



## Java code review

Under the **Java Code Review** set of rules, the **WebSphere version migration** category contains rules for migrating from WebSphere Application Server Version 5.1, 6.0, 6.1, and 7.0 to Version 7.0 or 8.0. For more information on a rule, press **F1** when viewing the rule in the configuration dialog or in the results viewer.

Quick fixes are available where possible. Rules without quick fixes flag the rule violations so you can evaluate their usage and migrate the code manually.

**Table 1: V5.1 to V6.0 migration**

Rule Name	Quick Fix	Action Taken
Check for a behavior change on the ServletResponse default content type	No	The default content type for HttpServletResponse has changed from "text/html" to "text/plain" for servlets that do not set the content type.
Check for behavior change on URLs containing a plus sign	No	This rule flags calls in Java code that create Uniform Resource Locators (URL) that contain a plus ("+") character in the URI that is not part of the query parameters. The plus ("+") is only reserved in the query string portion of the URL.
Check for expected behavior on ServletResponse sendRedirect() method	No	The WebSphere Application Server implementation of the ServletResponse sendRedirect() method omits path information until the last slash. With a new custom property this behavior can be corrected.
Do not start threads from code running within the web container	No	A restriction was added in WebSphere Application Server V6.0 that code running in the web container is no longer allowed to start threads.  Use the work manager to schedule asynchronous beans instead of creating new threads.

Rule Name	Quick Fix	Action Taken
Use leading slash on ServletContext getResource() and getResourceAsStream() requests	No	This rule flags calls to the <code>ServletContext.getResource()</code> and <code>ServletContext.getResourceAsStream()</code> methods where it cannot easily be determined if the String value passed on the method contains a leading slash (/) as required by the servlet specification.
Do not use activity component features that were removed	No	This rule flags the use of the activity component classes and interfaces that were removed in WebSphere Application Server V6.0.
Do not use Ant task features that were removed	No	This rule flags the use of the Ant task classes and interfaces that were removed in WebSphere Application Server V6.0.
Do not use asynchronous bean features that were removed	No	This rule flags the use of the asynchronous bean classes and interfaces that were removed in WebSphere Application Server V6.0.
Do not use object pool features that were removed	No	This rule flags the use of the object pool classes that were removed in WebSphere Application Server V6.0.
Do not use RAS features that were removed	No	This rule flags the use of the RAS classes and interfaces that were removed in WebSphere Application Server V6.0.
Do not use scheduler features that were removed	No	This rule flags the use of the scheduler classes and interfaces that were removed in WebSphere Application Server V6.0.
Do not use security features that were removed	No	This rule flags the use of the security classes and interfaces that were removed in WebSphere Application Server V6.0.
Do not use the EarUtils class that was removed	No	This rule flags the use of the <code>EarUtils</code> class that was removed in WebSphere Application Server V6.0.
Do not use setJMSPriority() method that was removed	No	This rule flags the use of the method, <code>setJMSPriority</code> , from the interface, <code>com.ibm.websphere.scheduler.MessageTaskInfo</code> . This method was deprecated in the Websphere Application Server V6.0 release and was replaced by the method <code>getJMSPriority</code> .
Do not use user profile features that were removed	No	This rule flags the use of the user profile classes and interfaces that were removed in WebSphere Application Server V6.0.

**Table 2: V6.0 to V6.1 migration**

Rule Name	Quick Fix	Action Taken
Check for behavior change on EJBContext.setRollbackOnly() method	No	A call to <code>setRollbackOnly</code> under a certain scenario can yield a different result on WebSphere Application Server releases prior to V6.0.2.



Rule Name	Quick Fix	Action Taken
Check for JAXP API usage compatibility	No	The JAXP APIs used in JRE 1.4.2 might have compatibility issues when used in JRE 5. This rule detects the import of any JAXP-related packages so that you can check the usage.
Check for JAXP EntityResolver.resolveEntity() exception compatibility	No	JAXP <code>EntityResolver.resolveEntity(String, String)</code> now throws the exception, <code>IOException</code> , in addition to <code>SAXException</code> . This rule detects the missing <code>IOException</code> .
Check for new JAXP DOM APIs	No	New JAXP DOM APIs were added to the following interfaces: <ul style="list-style-type: none"> <li>• <code>org.w3c.dom.Attr</code></li> <li>• <code>org.w3c.dom.Document</code></li> <li>• <code>org.w3c.dom.DOMImplementation</code></li> <li>• <code>org.w3c.dom.Element</code></li> <li>• <code>org.w3c.dom.Entity</code></li> <li>• <code>org.w3c.dom.Node</code></li> <li>• <code>org.w3c.dom.Text</code></li> </ul> This rule detects classes that implement any of these interfaces and flags them so that the new interfaces can be added.
Check Java Object Serialization compatibility	No	Serialization is not consistent between Java 1.4 and Java 5.0. This rule detects the classes that implement <code>java.io.Serializable</code> without a <code>serialVersionUID</code> field.
Do not make direct references to IBMJSSEFIPS provider classes	No	The IBMJSSEFIPS provider has been included in the IBMJSSE2 support. References to the classes of the previous provider, <code>com.ibm.fips.*</code> , must be removed. <p>This rule detects the use of any reference to <code>com.ibm.fips</code> packages.</p>
Do not use APIs from <code>com.ibm.net.ssl</code> packages	Yes	The classes and interfaces in the <code>com.ibm.net.ssl</code> package have been replaced by classes and interfaces in the <code>javax.net.ssl</code> package. This rule detects the use of any reference to <code>com.ibm.net.ssl</code> packages. <p>The quick fix changes package names <code>com.ibm.net.ssl</code> to <code>javax.net.ssl</code>.</p>
Do not use APIs from <code>sun.*</code> packages	No	Some APIs in the <code>sun.*</code> packages changed in JRE 5 and have compatibility issues with JRE 1.4.2. These APIs are not intended for use by developers. This rule detects the use of any reference to <code>sun.*</code> packages.
Do not use JAXP 1.1 internal classes	No	Internal JAXP classes changed. Do not use these internal classes in these packages: <ol style="list-style-type: none"> <li>1. <code>org.apache.crimson.*</code></li> <li>2. <code>org.apache.xml.*</code></li> <li>3. <code>org.apache.xalan.*</code></li> </ol>

Rule Name	Quick Fix	Action Taken
		<p>4. org.apache.xpath.*</p> <p>5. org.apache.xalan.xsltc.*</p> <p>This rule detects the use of these packages and flags them.</p>
Do not use JAXP 1.1 package names in string literals	No	Some of the implementation package names changed between JAXP 1.1 and JAXP 1.3. This rule detects JAXP 1.1 package names used in string literals.
Do not use removed IBMJSSE APIs	No	<p>In the conversion from IBMJSSE to IBMJSSE2 two of the JSSE classes were removed. This rule detects the use of any reference to com.ibm.jsse.KeyManagerFactoryParametersSpec and com.ibm.jsse.SSLContext. Since com.ibm.net.ssl.KeyManagerFactoryParametersSpec was migrated to com.ibm.jsse.KeyManagerFactoryParametersSpec and then removed from use, com.ibm.net.ssl.KeyManagerFactoryParametersSpec is also detected with this rule.</p>
Do not use the Java reserved word enum	No	Beginning in Java 5.0, enum is a reserved Java type and cannot be used as a variable name any longer. If your code uses variables named enum, they must be renamed. This rule detects variables and arguments named enum.
Use the BigDecimal toPlainString() method explicitly when deriving a string value	No	The BigDecimal toString() method behaves differently than in earlier versions. J2SE 5.0 added toPlainString() to BigDecimal, which behaves like the toString() method in earlier versions. This rule detects the implicit use of the toString() method in a class.
Use the BigDecimal toPlainString() method instead of the toString() method	Yes	<p>The BigDecimal toString() method behaves differently than in earlier versions. J2SE 5.0 added toPlainString() to BigDecimal, which behaves like the toString() method in earlier versions.</p> <p>The quick fix changes toString to toPlainString.</p>
Use the IBMJSSE2 Provider	Yes	<p>In the conversion from IBMJSSE to IBMJSSE2, the providers, IBMJSSEProvider and JSSEProvider, are replaced by IBMJSSEProvider2. This rule detects the use of any reference to com.ibm.jsse.IBMJSSEProvider and com.ibm.jsse.JSSEProvider classes.</p> <p>The quick fix changes the reference to com.ibm.jsse2.IBMJSSEProvider2.</p>
Do not use Common Connector Framework features that were removed	No	This rule flags the use of the Common Connector Framework API packages that were removed in WebSphere Application Server V6.1.

Rule Name	Quick Fix	Action Taken
Do not use the WebSphere Ant StopServer.setHost() method that was removed	No	This rule flags the use of the removed method <code>setHost(String s)</code> in the <code>com.ibm.websphere.ant.tasks.StopServer</code> class.
Use the open-source JDOM implementation to replace the JDOM features that were removed	No	This rule flags the use of the JDOM packages that were removed in WebSphere Application Server V6.1.
Use the open-source Mozilla Rhino implementation to replace the Rhino features that were removed	No	This rule flags the use of the Rhino packages that were removed in WebSphere Application Server V6.1.
Use the UserRegistry interface to replace the CustomRegistry interface that was removed	No	This rule flags the use of the <code>CustomRegistry</code> interface that was removed in WebSphere Application Server V6.1.

**Table 3: V6.1 to V7.0 migration**

Rule Name	Quick Fix	Action Taken
Only use JAX-WS annotations in Java EE 5 or later	No	This rule detects the use of JAX-WS annotations in enterprise projects earlier than Java EE 5.
Check exception logic for calls to EventHandler	No	In Java SE 6, the <code>EventHandler</code> constructor and <code>create()</code> methods require non-null parameters to be passed. This rule flags the constructor and <code>create()</code> method calls so that you can verify your logic handles a <code>NullPointerException</code> properly.
Check for Duration and XMLGregorianCalendar equals() compatibility	No	Detect the use of <code>Duration</code> and <code>XMLGregorianCalendar equals()</code> method. Java 6 now returns false if the parameter passed is null. The exception, <code>NullPointerException</code> , was thrown previously.
Check for the OverlappingFileLockException for the FileChannel lock() method	No	In Java SE 6, the <code>FileChannel.lock()</code> method now throws <code>OverlappingFileLockException</code> . This rule flags the <code>lock()</code> method calls without a catch block for <code>OverlappingFileLockException</code> or without a throws declaration for <code>OverlappingFileLockException</code> on the method.
Remove use of double slashes in JMX ObjectName elements	No	Detect the use of the double-slash character string (" <code>///</code> ") in JMX <code>ObjectNames</code> .
Do not use the DistributedLockingMap interface that was removed	No	This rule flags the removed <code>com.ibm.websphere.cache.DistributedLockingMap</code> interface.
Do not use the InvalidationEvent or ChangeEvent constructors that were removed	No	This rule flags the use of the removed constructors in the classes <code>com.ibm.websphere.cache.InvalidationEvent</code> or <code>com.ibm.websphere.cache.ChangeEvent</code> . The new constructor takes an additional field.

Rule Name	Quick Fix	Action Taken
Do not use the SequeLinkDataStoreHelper class that was removed	No	This rule flags the use of the <code>com.ibm.websphere.rsadapter.SequeLinkDataStoreHelper</code> class.
Do not use the WebSphere UserTransactionWrapper class that was removed	No	Do not use the <code>com.ibm.websphere.servlet.session.UserTransactionWrapper</code> class because it has been removed. Store a UserTransaction directly into the HTTP session without wrapping it in the removed class.
Do not use the WSConnectJDBCDataStoreHelper class that was removed	No	This rule flags the use of the <code>com.ibm.websphere.rsadapter.WSConnectJDBCDataStoreHelper</code> class and the <code>com.ibm.websphere.rsadapter.DataStoreHelper.WSCONNECTJDBC_HELPER</code> field that were removed.
Do not use web services gateway customization APIs that were removed	No	Do not use web services Customization APIs. The rule flags the use of the package, <code>com.ibm.wsgw.beans.*</code> .
Use Java EE servlet filters instead of WebSphere Servlet filter class that were removed	No	Do not use <code>com.ibm.websphere.servlet.filter</code> classes because they were removed. Use <code>javax.servlet.filter</code> classes instead.
Use the ConnectJDBCDataStoreHelper class instead of DataDirectDataStoreHelper class	Yes	Do not use the <code>com.ibm.websphere.rsadapter.DataDirectDataStoreHelper</code> object because it was removed.  The quick fix changes the code to use <code>com.ibm.websphere.rsadapter.ConnectJDBCDataStoreHelper</code> instead.
Use MicrosoftSQLServerDataStore helper class instead of the MSSQLDataStoreHelper class	Yes	Do not use the <code>com.ibm.websphere.rsadapter.MSSQLDataStoreHelper</code> class because it was removed.  The quick fix changes the code to use <code>com.ibm.websphere.rsadapter.MicrosoftSQLServerDataStoreHelper</code> instead.

Table 4: V7.0 to V8.0 migration

Rule Name	Quick Fix	Action Taken
Check for a behavior change for EJB presence in a web module	No	This rule flags EJB annotations in Java files if these files are in a web module with version 2.5 or higher.
Check for a behavior change in ApplicationException inheritance	No	This rule flags the EJB <code>ApplicationException</code> annotation which does not have the attribute <code>inherited</code> set. The <code>inherited</code> attribute was added in EJB 3.1 and changed default behavior of EJB 3.0 applications.

Rule Name	Quick Fix	Action Taken
Check for a behavior change on EntityManager refresh(Object entity) method	No	This rule flags the EntityManager.refresh() calls as the behavior of this method has changed.
Check for a behavior change on OpenJPAEntityManager detach(T pc) method	Yes	This rule flags the OpenJPAEntityManager detach(T pc) method. The method return type changed to support the 2.0 JPA specification.  The quick fix changes the detach() method to detachCopy().
Check for a behavior change on SipFactory methods	No	This rule flags certain SipFactory methods using a String to, from, or addr parameter for which there is a behavior change.
Check for a behavior change on some Server MBean operations	No	This rule flags the use of changed Server MBean operations getComponentVersion, getEFixVersion, getPTFVersion, getExtensionVersion, getVersionsForAllComponents, and getVersionsForAllEFixesstartTransports.
Do not use the removed Apache SOAP API	No	This rule flags the use of the removed classes in the org.apache.soap and com.ibm.soap packages.
Do not use the removed method getCause() from ServletException	No	This rule flags the method getCause() in the class com.ibm.websphere.servlet.error.ServletException that was removed.
Use the Oracle 11g helper instead of earlier versions	Yes	This rule flags the use of the Oracle 10g helpers and fields. Version 8.0 only supports the Oracle 11g JDBC driver and helper.  The quick fix changes the code to use the Oracle 11g helper after confirming that the runtime configuration was changed.

## JSP code review

Under the JSP code review set, the **WebSphere version migration** category has rules described in the following table. For more information, press **F1** when viewing the rule in the results viewer.

**Note:** JSP pages written in XML syntax (JSP documents) are not supported in this release.

**Table 5: Migrating from Version 5.1**

Rule Name	Quick Fix	Action Taken
Check for behavior change for included JSP encoding	No	In JSP 2.0, page encoding is done on a per-file basis. This rule detects the statically included JSP files that have different page encoding than the parent JSP.

Rule Name	Quick Fix	Action Taken
Check for behavior change on the request.getAttribute() method	No	This rule flags calls to request.getAttribute() in JSP files that use automatic casting to a String. However in V6, the request.getAttribute() method returns an Object, not a String.
Check for behavior change on URLs containing a plus sign	No	This rule flags a URI in a JSP link tag (<a>) or a form action tag (<form action=...) that contain a plus ("+") character in the URI but not a part of the query parameters. The plus ("+") is only reserved in the query string portion of the URL.
Do not use default packages in JSP import statements	No	As of JSP 2.0, you cannot refer to any classes from the unnamed (also known as the default) package. This rule detects JSP Import Directives that contain classes from the default package.

**Table 6: V6.1 to V7.0 migration**

Rule Name	Quick Fix	Action Taken
Check for taglib prefix redefinition	No	This rule detects the redefinition of a taglib with a different prefix in the same JSP or in the statically included JSP files. If a taglib is defined more than once in a JSP file or a statically included JSP file, it must have the same prefix.
Check for the reserved sequence '#{ ' in JSP files	Yes	This rule flags the use of the '#{ ' reserved sequence in JSP files. In JSP Version 2.1, the syntax #{ } is now a reserved keyword, and older JSP files can generate an error if they contain the #{ character sequence.  The quick fix adds the escape character (\) before the #{ sequence.

## XML code review

---

The XML file review provides rules to detect deployment descriptors and other XML file issues.

**Table 7: V6.1 to V7.0 migration**

Rule Name	Quick Fix	Action Taken
Do not use bean-managed persistence in EJB 3.0 projects	No	This rule detects the use of bean-managed persistence in EJB 3.0 projects, which is valid in the IBM WebSphere Application Server V6.1 Feature Pack for Enterprise JavaBeans 3.0 but not in WebSphere Application Server V7.0.
Use the metadata-complete attribute for Java EE 5 modules without annotations	No	This rule flags Java EE 5 modules that do not have the metadata-complete attribute set.
Use unique EJB 3.0 binding names	No	WebSphere Application Server V6.1 Feature Pack for EJB 3.0 allowed the EJB 3.0 binding file to contain duplicate binding

Rule Name	Quick Fix	Action Taken
		<p>names. The V7.0 server runtime added uniqueness checks for names used in the EJB 3.0 bindings file. Applications with uniqueness errors do not start in V7.0 even though the same application worked on the Feature Pack for EJB 3.0.</p> <p>This rule validates the EJB 3.0 bindings file to verify binding name uniqueness. It also validates that class names for session interfaces and interceptors are fully qualified.</p>

**Table 8: V6.1 to V7.0 migration**

Rule Name	Quick Fix	Action Taken
Check for a behavior change for EJB presence in a web module	No	This rule flags a web.xml file of a Web Module Version 2.5 or higher if that module contains a .class file which has an EJB annotation. The .class file must be in a library (a .jar file in WEB-INF/lib).
Check for a behavior change in ApplicationException inheritance	No	This rule flags EJB ApplicationException definitions in ejb-jar.xml files which do not have the inherited attribute set. The inherited attribute was added in EJB 3.1 and changed the default behavior of EJB 3.0 applications.
Check for a behavior change in web services addressing policy	No	This rule flags addressing policy configuration found in the WSDL definition. The addressing policy was ignored in the WSDL definition in previous releases. This behavior change would cause problems only if the the addressing policy in the packaged WSDL differs in a significant way from the active configured addressing policy.

## Deprecated features

There are deprecation rules under the Java Code Review, JSP Code Review, and XML File Review categories. The rules are organized by target release:

- Migration to V7.0 - currently deprecated in V7 and removed in V8
- Migration to V7.0 and V8.0 - currently deprecated in V7.0 and in V8.0
- Migration to V8.0 - newly deprecated in V8.0

**Table 9: Java deprecated features - Migration to V7.0**

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated Apache SOAP API	No	This rule flags references to the org.apache.soap and com.ibm.soap packages.
Avoid using the deprecated OracleDataStoreHelper class	Yes	<p>This rule flags the use of the deprecated Oracle data store helper and field.</p> <p>The quick fix changes the code to use the Oracle 11g helper after confirming that the runtime configuration was changed.</p>

**Table 10: Java deprecated features - Migration to V7.0 and V8.0**

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated analyzer logging system classes	No	This rule flags the use of the deprecated <code>com.ibm.websphere.als</code> classes.
Avoid using the deprecated Ant <code>setCompileWithAssert</code> method	Yes	This rule flags the use of the deprecated Ant method <code>setCompileWithAssert</code> .  The quick fix replaces the method with <code>setJdkSourceLevel</code> .
Avoid using the deprecated <code>AppDeploymentController</code> methods	No	This rule flags the use of the application deployment controller <code>getTaskInfo</code> methods.
Avoid using the deprecated <code>AppDeploymentTask</code> methods	Yes	This rule flags the use of the deprecated methods from the <code>com.ibm.websphere.management.application.client.AppDeploymentTask</code> class.  The quick fix changes the methods to the recommended replacements.
Avoid using the deprecated application management <code>installStandaloneRAR</code> method	No	This rule flags the use of the application management <code>installStandaloneRAR</code> method.
Avoid using the deprecated application management <code>moveModule</code> method	No	This rule flags the use of the application management <code>moveModule</code> method.
Avoid using the deprecated application profile access intent methods	No	This rule flags the use of deprecated application profile access intent methods.
Avoid using the deprecated Cache interface	No	This rule flags the use of the <code>com.ibm.websphere.Cache</code> interface.
Avoid using the deprecated <code>com.ibm.etools.logging</code> utilities	No	This rule detects and flags references to <code>com.ibm.etools.logging</code> packages that are deprecated.
Avoid using the deprecated command manager methods	Yes	This rule flags the use of the deprecated methods in <code>com.ibm.websphere.management.cmdframework.CommandMgrInitializer</code> class.  The quick fix changes the methods to the recommended replacements.
Avoid using the deprecated <code>ConnectionFactory</code> MBean methods	No	This rule flags the deprecated <code>ConnectionFactory</code> MBean operations <code>getPoolContents</code> , <code>getAllPoolContents</code> , and <code>showAllocationHandleList</code> .
Avoid using the deprecated <code>ConnectionSpecImpl</code> methods	Yes	This rule flags the deprecated methods from the <code>com.ibm.websphere.ola.ConnectionSpecImpl</code> class.  The quick fix changes the parameter to the appropriate boolean value.



Rule Name	Quick Fix	Action Taken
Avoid using the deprecated Connector Architecture interfaces	No	This rule flags the deprecated Connector Architecture interfaces <code>com.ibm.websphere.j2c.ConnectionEventListener</code> and <code>com.ibm.websphere.j2c.ConnectionManager</code> .
Do not used the deprecated distributed locking map field and removed interface	No	This rule flags the deprecated <code>com.ibm.websphere.cache.DistributedObjectCache</code> , <code>TYPE_DISTRIBUTED_LOCKING_MAP</code> field and the removed <code>com.ibm.websphere.cache.DistributedLockingMap</code> interface.
Avoid using the deprecated DumpNameSpace constructor and fields	No	This rule flags the use of the deprecated constructor and fields from the <code>com.ibm.websphere.naming.DumpNameSpace</code> class.
Avoid using the deprecated DynamicCacheAccessor methods	No	This rule flags the use of the deprecated methods from the <code>com.ibm.websphere.cache.DynamicCacheAccessor</code> class.
Avoid using the deprecated dynamic cache methods and fields	No	This rule flags the use of the deprecated methods and fields from the <code>com.ibm.websphere.cache.CacheEntry</code> and the <code>com.ibm.websphere.cache.EntryInfo</code> interfaces.
Avoid using the deprecated EJB persistence createInteraction method	No	This rule flags the use of the <code>com.ibm.websphere.ejbpersistence.EJBToRAAdapter.createInteraction(javax.resource.cci.Connection conn)</code> method.
Avoid using the deprecated ExtendedJTATransaction method registerSynchronizationCallbackFor CurrentTran	No	This rule detects and flags the method <code>registerSynchronizationCallbackForCurrentTran</code> from the class <code>ExtendedJTATransaction</code> .
Avoid using the deprecated HttpServlet request/response proxy classes	No	This rule flags the use of the <code>HttpServletRequestProxy</code> and <code>HttpServletResponseProxy</code> classes.
Avoid using the deprecated interface SIBTransmitMessageRequest	Yes	This rule flags the use of the <code>com.ibm.websphere.sib.admin.SIBTransmitMessageRequest</code> interface.  The quick fix replaces the interface with the <code>com.ibm.websphere.sib.admin.SIBMessageRequest</code> interface.
Avoid using the deprecated J Ras extension APIs	No	This rule flags deprecated Java reliability, availability, and serviceability APIs.
Avoid using the deprecated LocalHomeAccessor class	No	This rule flags the use of the <code>com.ibm.websphere.ejbcontainer.LocalHomeAccessor</code> class.

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated management InvalidDocumentURIException class	Yes	This rule flags the use of the <code>com.ibm.websphere.management.exception.InvalidDocumentURIException</code> class.  The quick fix changes the code to use <code>DocumentNotFoundException</code> .
Avoid using the deprecated management NestedAdminException class	No	This rule flags the use of the <code>com.ibm.websphere.management.exception.NestedAdminException</code> class.
Avoid using the deprecated management NotificationConstants TYPE_AGENT_DISCOVERED field	Yes	This rule flags the use of the <code>NotificationConstants.TYPE_AGENT_DISCOVERED</code> field.  The quick fix replaces references to this field with the <code>NotificationConstants.TYPE_DISCOVERY_AGENT_FOUND</code> field.
Avoid using the deprecated management removeNotificationListenerExtended methods	No	This rule flags the use of the deprecated <code>removeNotificationListenerExtended</code> methods from the <code>com.ibm.websphere.management.AdminService</code> and <code>com.ibm.websphere.management.AdminClient</code> classes.
Avoid using the deprecated management statistics interfaces	Yes	This rule flags the use of the <code>com.ibm.websphere.management.statistics</code> package.  The quick fix changes all references of <code>com.ibm.websphere.management.statistics</code> to <code>javax.management.j2ee.statistics</code> and all reference of <code>MessageBeanStats</code> to <code>MessageDrivenBeanStats</code> .
Avoid using the deprecated methods from WebSphere SIB MQ classes	Yes	This rule flags the use of the <code>getNpmSpeed()</code> and <code>getStatus()</code> methods from SIB MQ classes.  The quick fix changes the methods to the recommended replacements.
Avoid using the deprecated naming properties INITIAL_CONTEXT_FACTORY_LEGACY field	Yes	This rule flags the use of the <code>com.ibm.websphere.naming.PROPS.INITIAL_CONTEXT_FACTORY_LEGACY</code> field.  The quick fix changes all references to this field to <code>com.ibm.websphere.naming.PROPS.INITIAL_CONTEXT_FACTORY</code> .
Avoid using the deprecated PMI Client API	No	This rule flags the use of PMI Client API classes.
Avoid using the deprecated PmiConstants fields	No	This rule flags the use of deprecated <code>PmiConstants</code> fields.
Avoid using the deprecated PmiDataInfo getParticipation method	No	This rule flags the use of the <code>com.ibm.websphere.pmi.PmiDataInfo.getParticipation()</code> method.

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated PMI dynamic cache OBJECT_CACHE_GROUP field	Yes	This rule flags the use of the deprecated <code>com.ibm.websphere.pmi.stat.WSDynamicCacheStats.OBJECT_CACHE_GROUP</code> field.  The quick fix changes OBJECT_CACHE_GROUP to OBJECT_GROUP.
Avoid using the deprecated PMI dynamic cache SERVLET_CACHE_GROUP field	No	This rule flags the use of the deprecated <code>com.ibm.websphere.pmi.stat.WSDynamicCacheStats.SERVLET_CACHE_GROUP</code> field.
Avoid using the deprecated PmiJmxTest methods	No	This rule flags the use of the deprecated methods from the <code>com.ibm.websphere.pmi.PmiJmxTest</code> class.
Avoid using the deprecated PMI MBeanLevelSpec methods	No	This rule flags the use of the deprecated constructors and methods from the <code>com.ibm.websphere.pmi.stat.MBeanLevelSpec</code> class.
Avoid using the deprecated PmiModuleConfig print method	No	This rule flags the use of the deprecated <code>com.ibm.websphere.pmi.PmiModuleConfig.print(PrintWriter)</code> method.
Avoid using the deprecated PMI StatDescriptor methods	No	This rule flags the use of the deprecated constructor and methods from the <code>com.ibm.websphere.pmi.stat.StatDescriptor</code> class.
Avoid using the deprecated PMI statistic classes	No	This rule flags the use of the deprecated classes from the <code>com.ibm.websphere.pmi.stat</code> package.
Avoid using the deprecated PMI WSStats methods	No	This rule flags the use of the deprecated methods from the <code>com.ibm.websphere.pmi.stat.WSStats</code> interface.
Avoid using the deprecated RemoteCommandMgr MBean	No	This rule flags the deprecated <code>RemoteCommandMgr</code> MBean.
Avoid using the deprecated resource adapter fields	No	This rule flags the use of the deprecated resource adapter fields.
Avoid using the deprecated resource adapter methods	No	This rule flags the use of the deprecated resource adapter methods.
Avoid using the deprecated runtime ServerName methods	No	This rule flags the use of methods deprecated from the <code>com.ibm.websphere.runtime.ServerName</code> class.
Avoid using the deprecated scheduler MessageTaskInfo methods	No	This rule flags the use of the deprecated methods from the <code>com.ibm.websphere.scheduler.MessageTaskInfo</code> class.
Avoid using the deprecated Scheduler methods	No	This rule flags the use of the deprecated <code>createBeanTaskInfo()</code> and <code>createMessageTaskInfo()</code> methods from the <code>com.ibm.websphere.scheduler.Scheduler</code> class.

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated security authentication exception classes	No	This rule flags the use of the deprecated exception classes <code>com.ibm.websphere.security.auth.MapCredentialFailedException</code> and <code>com.ibm.websphere.security.auth.MapCredentialNotSupportedException</code> .
Avoid using the deprecated security authentication WSPincipal getCredential method	No	This rule flags the use of the deprecated <code>getCredential()</code> method from the <code>com.ibm.websphere.security.auth.WSPincipal</code> class.
Avoid using the deprecated security printStackTrace() methods	No	This rule flags the use of the deprecated <code>printStackTrace()</code> methods from the <code>com.ibm.websphere.security.WSSecurityException</code> and <code>com.ibm.websphere.security.auth.WSLoginFailedException</code> classes.
Avoid using the deprecated security LoginHelper class	No	This rule flags the use of the deprecated <code>com.ibm.ws.security.util.LoginHelper</code> class.
Avoid using the deprecated servlet cache classes	No	This rule flags the use of the deprecated servlet cache classes.
Avoid using the deprecated servlet cache IdGenerator methods	No	This rule flags the use of the deprecated methods <code>initialize()</code> and <code>getSharingPolicy()</code> from the <code>com.ibm.websphere.servlet.cache.IdGenerator</code> class.
Avoid using the deprecated servlet cache MetaDataGenerator initialize method	No	This rule flags the use of the deprecated <code>initialize()</code> method from the <code>com.ibm.websphere.servlet.cache.MetaDataGenerator</code> class.
Avoid using the deprecated SIMediationBeanMessageContext interface	No	This rule flags the use of the deprecated <code>com.ibm.websphere.sib.mediation.messagecontext.SIMediationBeanMessageContext</code> interface.
Avoid using the deprecated SIMessageContextException class	Yes	This rule flags the use of the deprecated <code>com.ibm.websphere.sib.mediation.handler.SIMessageContextException</code> class.  The quick fix changes the code to use <code>MessageContextException</code> instead.
Avoid using the deprecated TransactionControl interface	No	This rule flags the use of the deprecated <code>com.ibm.ws.extensionhelper.TransactionControl</code> interface.
Avoid using the deprecated UDDI Version 2 interfaces	No	This rule flags the use of UDDI Version 2 related packages.

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated UNTGenerateCallback methods	Yes	This rule flags the use of the deprecated methods from the <code>com.ibm.websphere.wssecurity.callbackhandler.UNTGenerateCallback</code> class.  The quick fix changes the methods to the recommended replacements.
Avoid using the deprecated WASProduct class	No	This rule flags the deprecated <code>com.ibm.websphere.product.WASProduct</code> class was used to get installed product information and history.
Avoid using the deprecated web container custom extension classes	No	This rule flags the deprecated web container custom extension classes from the <code>com.ibm.servlet</code> package.
Avoid using the deprecated WebContainer MBean operations	No	This rule flags references to the deprecated WebContainer MBean operations <code>startTransports</code> , <code>stopTransports</code> , and <code>restartWebApplication</code> .
Avoid using the deprecated WebSphere Ant class ModuleValidator	No	This rule flags the use of the deprecated WebSphere Ant class <code>com.ibm.websphere.ant.tasks.ModuleValidator</code> .
Avoid using the deprecated WebSphere Studio tools runtime classes	No	This rule flags the use of the deprecated WebSphere Studio tools runtime classes from the <code>com.ibm.webtools.runtime</code> package.
Avoid using the deprecated WSAddressing for JAXWS 2.0 classes	No	This rule flags the deprecated WSAddressing for JAXWS 2.0 classes from the <code>com.ibm.websphere.wsaddressing.jaxws</code> package.
Avoid using the deprecated WsnBatchResult fields	Yes	This rule flags use of deprecated fields from the <code>com.ibm.websphere.naming.WsnBatchResult</code> class.  The quick fix changes the fields to the recommended replacements.

Table 11: Java deprecated features - Migration to V8.0

Rule Name	Quick Fix	Action Taken
Avoid extending the AppDeploymentTask class	No	This rule flags any class that extends <code>com.ibm.websphere.management.application.client.AppDeploymentTask</code> .
Avoid using the deprecated AppConstants fields	No	This rule flags the use of the deprecated fields from the <code>com.ibm.websphere.management.application.AppConstants</code> class.
Avoid using the deprecated AppManagementBaseFactory methods	No	This rule flags the use of the deprecated methods from the <code>com.ibm.websphere.management.application.AppManagementBaseFactory</code> class.

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated <code>com.ibm.websphere.product</code> classes	No	This rule flags the deprecated classes from the <code>com.ibm.websphere.product</code> packages.
Avoid using the deprecated elements in the <code>EditionInfo</code> class	No	This rule flags the use of the deprecated elements from the <code>com.ibm.websphere.management.application.EditionInfo</code> class.
Avoid using the deprecated elements in the <code>IFilterConfig</code> interface	No	This rule flags the use of the deprecated elements from the <code>com.ibm.websphere.servlet.filter.IFilterConfig</code> interface.
Avoid using the deprecated field in the <code>AppDeploymentController</code> class	No	This rule flags the use of the deprecated field <code>taskHelperSuffix</code> from the class <code>com.ibm.websphere.management.application.client.AppDeploymentController</code> .
Avoid using the deprecated <code>IRequest.isStartAsync()</code> method	No	This rule flags the use of the deprecated method <code>isStartAsync()</code> from the interface <code>com.ibm.websphere.servlet.request.IRequest</code> .
Avoid using the deprecated <code>WASDirectory</code> methods and fields	No	This rule flags the deprecated <code>com.ibm.websphere.product.WASDirectory</code> methods and related fields.
Avoid using the deprecated WebSphere Studio Application Developer Integration Edition libraries	No	This rule flags the use of the deprecated WebSphere Studio Application Developer Integration Edition libraries.

**Table 12: JSP file deprecated features - Migration to V7.0 and V8.0**

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated JSP <code>&lt;tsx&gt;</code> tags	No	This rule flags the use of the deprecated <code>&lt;tsx&gt;</code> tags in JSP files.

**Table 13: XML file deprecated features - Migration to V7.0 and V8.0**

Rule Name	Quick Fix	Action Taken
Avoid using the deprecated method-level access intent for entity beans	No	This rule flags the use of method-level access intent on entity beans.
Avoid using the deprecated reload attributes of the IBM deployment descriptor extensions	No	This rule flags the deprecated <code>reloadInterval</code> and <code>reloadingEnabled</code> attributes of the IBM deployment descriptor extensions, including both the WAR file extension ( <code>WEB-INF/ibm-web-ext.xmi</code> ) and the application extension ( <code>META-INF/ibm-application-ext.xmi</code> ).

## Troubleshooting

---

### Software Analyzer options not shown

---

After an application migration tool is installed, you have new Software Analyzer menu and toolbar options in the Java, Debug, C++, and Plug-in Development perspectives. When using other perspectives, enable these options manually. To customize other perspectives, complete the following steps:

1. From your perspective, select **Window > Customize Perspective** from the IDE menu.
2. In the Customize Perspective window, click the **Commands** tab and select the **Software Analyzer** check box. If you are running Rational Application Developer, then the Rational Software Analyzer check box might be displayed.
3. Click **OK**.

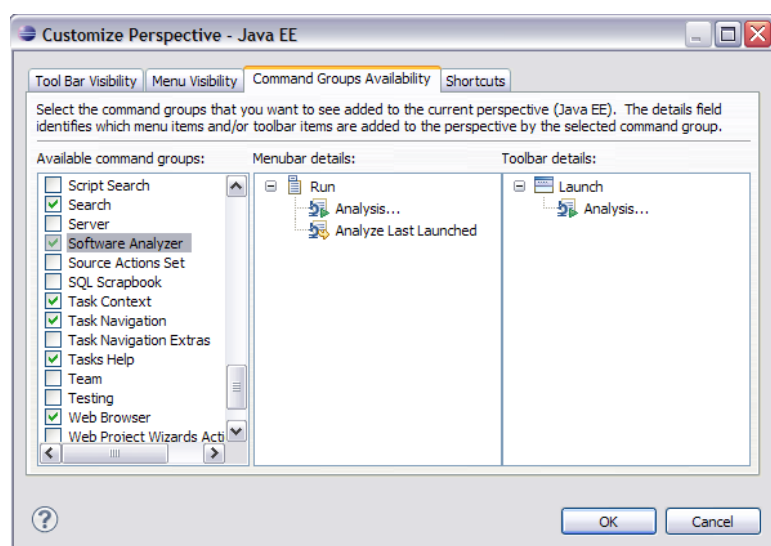


Figure 12: Software Analyzer availability

### Java EE constructs or JSP not read correctly

---

For the tools to read project files correctly, the projects must be set up with their appropriate project facets. For example, projects that contain enterprise beans must have the EJB Module facet. You can see facets for a project by right-clicking on the project in the project explorer and selecting **Properties**. Select the **Project Facets** property. Projects containing web modules must be dynamic web projects. See [Importing applications](#) on page 10 for more information.

### Logging and trace

---

The application migration tools write error information to two log files:

- The workspace log (*workspace/.metadata/.log*) contains messages logged by the application migration tools as well as messages logged by Rational Software Analyzer.
- The service logs for the application migration tools are located in the *workspace/.metadata/.plugins/com.ibm.ws.appconversion.base* directory.

This log contains all error information as well as detailed trace information if trace is enabled.

Enable trace in the migration tool by setting the `appconversion.trace` system variable on the command line to launch the IDE or in the `eclipse.ini` property file; for example:

- Command-line option. Add the system variable to the command line that starts Eclipse:  

```
eclipse.exe -vmargs -Dappconversion.trace=true
```
- eclipse.ini option. Add the following line to the eclipse.ini file found in the same directory as the eclipse.exe file:  

```
-Dappconversion.trace=true
```

## Reports and history

---

In the Software Analyzer Results view, the analysis provider tabs have buttons that are used to export selected analysis history to file and to generate reports. Java Code Review is the only results view that has support for reports.

For exported history, you can specify the location for the generated result.

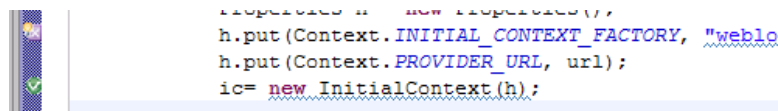
For generated reports, the report is displayed when the report is finished. An HTML report is displayed in a web browser. You can configure the web browser under the Eclipse properties for **General > Web Browser**.

PDF reports are displayed in the default PDF viewer. The generated files are put in the `workspace/.metadata/.plugins/com.ibm.xttools.analysis.reporting/reports` directory.

## Markers

---

Markers are displayed in the left margin of Eclipse editors and provide information about the content of the editor on the line where the marker is displayed. Editors for different file types can have different behaviors for the markers, some of which are described here.



**Figure 13: Analysis rule markers in a Java editor**

### No pop-up window displays when clicking on the marker

This issue affects non-Java based rules. For a Java rule, clicking on the marker allows the quick fix to be performed. However, clicking on the marker for the XML rule has no action, and hovering over the marker only displays the help text. Use the Software Analyzer Results view to select the **Quick Fix** action in non-Java files

### Double-clicking a marker in a manifest file removes the marker

Double-clicking the marker on the first line removes the marker without applying the quick fix. If this happens, run the class path rule again to display the marker again.

### Cannot select multiple markers on the same line

When there are multiple markers on the same line of text, you cannot toggle between the different markers. You must carry out the action of the first marker to select successive markers. To apply a quick fix in such a case, use the Software Analyzer Results view to select the **Quick Fix** action you want, rather than relying on the marker.

## Known issues

---

### Quick Fix All Category

When you use the **Quick Fix All Category** option, let it run to completion before running it on another rule provider. Also, do not run **Quick Fix All Category** again on the same provider where it is already running. Wait until it completes.

If you notice errors being logged when running the **Quick Fix All Category** option, there are a few things you can do in Eclipse to mitigate issues:




- In the **Window > Preferences > General > Editors**, select **Close editors automatically** and limit the **Number of open editors before closing** to a reasonable number. The default value is 8 which should be OK, but you can limit it further.
- You can also choose **Run in Background** to prevent all the editors from opening and possibly running out of window handles.
- Disable the automatic build feature of Eclipse (**Project > Build Automatically**) while running **Quick Fix All Category**. Manually build the projects and enable the option again after running **Quick Fix All Category**.

### Multiple rules on the same node

When multiple rules are flagged on the same node, only the first quick fix applied runs correctly. You might need to run the analysis multiple times to make sure that all code is fixed and processed correctly.

### Viewing external links from help in Linux

Many detailed help pages have external references to information applicable to the specific rule. This information is generally viewed better from an external browser than the Eclipse help view. On Windows platforms, help automatically launches these external references in the default browser. On Linux operating systems, it displays the information in the Eclipse help view. To manually open help in an external browser, use the **Show in external window** icon () on the detailed help page.

### Java Architectural Discovery rules

The **Java Architectural Discovery** rules are not supported in Eclipse 3.4.x or Rational Application Developer 7.5.x. These rules depend on APIs in Eclipse 3.5 and higher. Do not select the **Java Architectural Discovery** rules if your development environment is based on Eclipse 3.4.x.

### Ignore Results

If you select **Ignore Results** on a line of code that is the beginning of a block of Java code with nested results being flagged for the same rule, the nested results line numbers will be set to zero. You can rerun analysis to see the proper line numbers. Examples of statements that cause this issue are method declarations and `if` statements.

### View Results for .xmi files

If you select **View Results** for an `.xmi` file and get an error opening the file, add an `.xmi` file association. Go to **Window > Preferences > General > Editors > File Associations**, and click **Add...** to enter `*.xmi`. Select either **XML Editor** or **Text Editor**, and click the **Default** button. Click OK to save.

### Uncategorized plug-ins

If you are installing the tools on Eclipse 3.4.2 or Rational Application Developer 7.5, it is preferable to not select **Uncategorized** from the repository site. If you do select **Uncategorized**, and you want to uninstall the tool, you must manually uninstall these plug-ins:

- Analysis Reporting n11 version
- Analysis Reporting n12 version
- Command Line Analysis
- Command Line Analysis n11 version
- Command Line Analysis n12 version
- Compatibility feature
- Core\_feature Feature
- JEE Code Review
- JEE Code Review n11 version
- JEE Code Review n12 version
- J2SE Code Review
- J2SE Code Review n11 version

- J2SE Code Review n12 version
- Java Architectural Discovery
- Java Code Review Feature Translations
- Java Globalization Code Review
- Java Globalization Code Review n11 version
- Java Globalization Code Review n12 version
- Java Security Code Review
- Java Security Code Review n12 version
- Java Security Code Review n12 version
- ODA Compatability Feature
- Rational Code Analyst Core Services Feature
- Rational Software Analyzer Architecture Core
- Rational Software Analyzer Code Review Core
- Rational Software Analyzer Java Code Review
- Rational Software Analyzer Reporting
- Rational Software Analyzer Reporting ODA
- RCA Analysis Feature Translations

This problem does not occur in Eclipse 3.5 and higher.

## Copyright and trademarks

---



© Copyright IBM Corporation 2009, 2011.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

IBM, the IBM logo, developerWorks, Passport Advantage, Rational, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other product and service names might be trademarks of IBM or other companies.