

# Troubleshoot Cisco CVP Call Studio with Logs

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Problem: Where to Get Call Studio Install Logs](#)

[Solution](#)

[Problem: Where to Get Call Studio License Logs](#)

[Solution](#)

[Problem: Where to Get Call Studio Error Logs For Call Studio Applications](#)

[Solution](#)

[Problem: Where to Get Call Studio Logs for Application Validation](#)

[Solution](#)

## Introduction

This document describes common Cisco Call Studio scenarios to troubleshoot and where to get logs for them.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Call Studio

### Components Used

The information in this document is based on Cisco Call Studio 11.6 and/or 12.0.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

## Problem: Where to Get Call Studio Install Logs

### Solution

Call Studio Install logs can be found in **C:\Temp** directory.

Here's an example snippet of the logs for a new installation:

```
[10-20-2020 08:52:32] Informational: Starting Call Studio installation procedure... [10-20-2020
08:52:32] Informational: Attempting to read version file:
D:\CallStudio\Installer_Windows\VERSION [10-20-2020 08:52:32] Informational: Welcome to the
installation of the Cisco Unified Call Studio, Release 12.0.1. [10-20-2020 08:52:34]
Informational: User has passed the Copyrights notification screen. [10-20-2020 08:52:37]
Informational: User has accepted the license agreement. [10-20-2020 08:52:38] Informational:
User selected directory: C:\Cisco\CallStudio [10-20-2020 08:52:38] Informational: User has
selected target directory = C:\Cisco\CallStudio [10-20-2020 08:52:39] Informational: LaunchApp -
success running: Setting CALLSTUDIO_HOME environmental variable through setx:
C:\Windows\system32\setx.exe [10-20-2020 08:52:39] Informational: Installing the Vc++
redistributable exe from the path: C:\Users\ADMINI~1\AppData\Local\Temp\3\{7F6E4651-C47A-4607-
8E9D-64A8B21F688F}\{763E1DF9-41BC-4C54-9705-A0C6D1594B26}\vcredist_x64.exe [10-20-2020 08:52:41]
WARNING: LaunchApp - Error 5100 while running: Attempting to silently install Vcredist_x64 using
cmd=<C:\Users\ADMINI~1\AppData\Local\Temp\3\{7F6E4651-C47A-4607-8E9D-64A8B21F688F}\{763E1DF9-
41BC-4C54-9705-A0C6D1594B26}\vcredist_x64.exe /q /log C:\temp\vcredistinstall.htm>. [10-20-2020
08:52:41] ERROR: Installation of VC++ redistributable failed. [10-20-2020 08:52:41]
Informational: Installation of VC++ redistributable failed. [10-20-2020 08:53:13] Informational:
Launching program C:\Users\ADMINI~1\AppData\Local\Temp\3\{7F6E4651-C47A-4607-8E9D-
64A8B21F688F}\{763E1DF9-41BC-4C54-9705-A0C6D1594B26}\studio.bat with command line options
"C:\Cisco\CallStudio\eclipse\jre\bin\java.exe" C:\Users\ADMINI~1\AppData\Local\Temp\3\{7F6E4651-
C47A-4607-8E9D-64A8B21F688F}\{763E1DF9-41BC-4C54-9705-A0C6D1594B26}\
"C:\Cisco\CallStudio\eclipse\plugins\com.audiumcorp.studio.license\license" [10-20-2020
08:53:14] Informational: Successfully added install path for registry key SOFTWARE\Cisco
Systems, Inc.\Cisco Unified Call Studio\11.6.1 [10-20-2020 08:53:14] Informational: WriteLine
successfully wrote to C:\Cisco\CallStudio\eclipse\eclipse.ini [10-20-2020 08:53:14]
Informational: WriteLine successfully wrote to C:\Cisco\CallStudio\eclipse\eclipse.ini [10-20-
2020 08:53:15] Informational: New Installation Complete.
```

## Problem: Where to Get Call Studio License Logs

### Solution

License to Call Studio are valid till version 11.6 only. From 12.0 and onwards, Call Studio does not require any license.

You can get the license logs here:

Here's an example log file: **C:\Cisco\CallStudio\eclipse\workspace\metadata**

```
!SUBENTRY 1 com.audiumcorp.studio.core 4 4 2019-07-24 14:33:13.797 !MESSAGE This installation of
Call Studio cannot be used until it has been activated. If you installed Call Studio more than
30 days ago, you are now required to activate it to use it. To activate, restart Call Studio,
enter your licensing information and click on the 'Activate >>' button. !STACK 0
com.audiumcorp.license.client.LicenseException: This installation of Call Studio cannot be used
until it has been activated. If you installed Call Studio more than 30 days ago, you are now
required to activate it to use it. To activate, restart Call Studio, enter your licensing
information and click on the 'Activate >>' button. at
com.audiumcorp.studio.core.license.LicenseManager.initialize(LicenseManager.java:184) at
com.audiumcorp.studio.core.CorePlugin.validateLicense(CorePlugin.java:325) at
com.audiumcorp.studio.core.CorePlugin.start(CorePlugin.java:308) at
org.eclipse.osgi.framework.internal.core.BundleContextImpl$1.run(BundleContextImpl.java:711) at
java.security.AccessController.doPrivileged(Native Method) at
```

org.eclipse.osgi.framework.internal.core.BundleContextImpl.startActivator(BundleContextImpl.java:702) at  
org.eclipse.osgi.framework.internal.core.BundleContextImpl.start(BundleContextImpl.java:683) at  
org.eclipse.osgi.framework.internal.core.BundleHost.startWorker(BundleHost.java:381) at  
org.eclipse.osgi.framework.internal.core.AbstractBundle.start(AbstractBundle.java:299) at  
org.eclipse.osgi.framework.util.SecureAction.start(SecureAction.java:440) at  
org.eclipse.osgi.internal.loader.BundleLoader.setLazyTrigger(BundleLoader.java:268) at  
org.eclipse.core.runtime.internal.adaptor.EclipseLazyStarter.postFindLocalClass(EclipseLazyStarter.java:107) at  
org.eclipse.osgi.baseadaptor.loader.ClasspathManager.findLocalClass(ClasspathManager.java:463) at  
org.eclipse.osgi.internal.baseadaptor.DefaultClassLoader.findLocalClass(DefaultClassLoader.java:216) at  
org.eclipse.osgi.internal.loader.BundleLoader.findLocalClass(BundleLoader.java:400) at  
org.eclipse.osgi.internal.loader.SingleSourcePackage.loadClass(SingleSourcePackage.java:35) at  
org.eclipse.osgi.internal.loader.BundleLoader.findClassInternal(BundleLoader.java:473) at  
org.eclipse.osgi.internal.loader.BundleLoader.findClass(BundleLoader.java:429) at  
org.eclipse.osgi.internal.loader.BundleLoader.findClass(BundleLoader.java:417) at  
org.eclipse.osgi.internal.baseadaptor.DefaultClassLoader.loadClass(DefaultClassLoader.java:107) at  
java.lang.ClassLoader.loadClass(ClassLoader.java:358) at  
com.audiumcorp.studio.builder.core.util.BuilderResources.isBuilderSupported(BuilderResources.java:138) at  
com.audiumcorp.studio.builder.core.CorePlugin.validateLicense(CorePlugin.java:74) at  
com.audiumcorp.studio.builder.core.CorePlugin.start(CorePlugin.java:60) at  
org.eclipse.osgi.framework.internal.core.BundleContextImpl\$1.run(BundleContextImpl.java:711) at  
java.security.AccessController.doPrivileged(Native Method) at  
org.eclipse.osgi.framework.internal.core.BundleContextImpl.startActivator(BundleContextImpl.java:702) at  
org.eclipse.osgi.framework.internal.core.BundleContextImpl.start(BundleContextImpl.java:683) at  
org.eclipse.osgi.framework.internal.core.BundleHost.startWorker(BundleHost.java:381) at  
org.eclipse.osgi.framework.internal.core.AbstractBundle.start(AbstractBundle.java:299) at  
org.eclipse.osgi.framework.util.SecureAction.start(SecureAction.java:440) at  
org.eclipse.osgi.internal.loader.BundleLoader.setLazyTrigger(BundleLoader.java:268) at  
org.eclipse.core.runtime.internal.adaptor.EclipseLazyStarter.postFindLocalClass(EclipseLazyStarter.java:107) at  
org.eclipse.osgi.baseadaptor.loader.ClasspathManager.findLocalClass(ClasspathManager.java:463) at  
org.eclipse.osgi.internal.baseadaptor.DefaultClassLoader.findLocalClass(DefaultClassLoader.java:216) at  
org.eclipse.osgi.internal.loader.BundleLoader.findLocalClass(BundleLoader.java:400) at  
org.eclipse.osgi.internal.loader.SingleSourcePackage.loadClass(SingleSourcePackage.java:35) at  
org.eclipse.osgi.internal.loader.BundleLoader.findClassInternal(BundleLoader.java:473) at  
org.eclipse.osgi.internal.loader.BundleLoader.findClass(BundleLoader.java:429) at  
org.eclipse.osgi.internal.loader.BundleLoader.findClass(BundleLoader.java:417) at  
org.eclipse.osgi.internal.baseadaptor.DefaultClassLoader.loadClass(DefaultClassLoader.java:107) at  
java.lang.ClassLoader.loadClass(ClassLoader.java:358) at  
java.lang.Class.getDeclaredConstructors0(Native Method) at  
java.lang.Class.privateGetDeclaredConstructors(Class.java:2493) at  
java.lang.Class.getConstructor0(Class.java:2803) at  
java.lang.Class.newInstance(Class.java:345) at  
org.eclipse.core.internal.registry.osgi.RegistryStrategyOSGI.createExecutableExtension(RegistryStrategyOSGI.java:184) at  
org.eclipse.core.internal.registry.ExtensionRegistry.createExecutableExtension(ExtensionRegistry.java:905) at  
org.eclipse.core.internal.registry.ConfigurationElement.createExecutableExtension(ConfigurationElement.java:243) at  
org.eclipse.core.internal.registry.ConfigurationElementHandle.createExecutableExtension(ConfigurationElementHandle.java:55) at  
org.eclipse.ui.internal.WorkbenchPlugin\$1.run(WorkbenchPlugin.java:268) at  
org.eclipse.swt.custom.BusyIndicator.showWhile(BusyIndicator.java:70) at  
org.eclipse.ui.internal.WorkbenchPlugin.createExtension(WorkbenchPlugin.java:264) at  
org.eclipse.ui.internal.registry.EditorDescriptor.createEditor(EditorDescriptor.java:235) at  
org.eclipse.ui.internal.EditorManager.createPart(EditorManager.java:875) at  
org.eclipse.ui.internal.EditorReference.createPartHelper(EditorReference.java:609) at  
org.eclipse.ui.internal.EditorReference.createPart(EditorReference.java:465) at  
org.eclipse.ui.internal.WorkbenchPartReference.getPart(WorkbenchPartReference.java:595) at  
org.eclipse.ui.internal.EditorAreaHelper.setVisibleEditor(EditorAreaHelper.java:271) at

```
org.eclipse.ui.internal.EditorManager.setVisibleEditor(EditorManager.java:1459) at
org.eclipse.ui.internal.EditorManager$5.runWithException(EditorManager.java:972) at
org.eclipse.ui.internal.StartupThreading$StartupRunnable.run(StartupThreading.java:31) at
org.eclipse.swt.widgets.RunnableLock.run(RunnableLock.java:35) at
org.eclipse.swt.widgets.Synchronizer.runAsyncMessages(Synchronizer.java:135) at
org.eclipse.swt.widgets.Display.runAsyncMessages(Display.java:4140) at
org.eclipse.swt.widgets.Display.readAndDispatch(Display.java:3757) at
org.eclipse.ui.application.WorkbenchAdvisor.openWindows(WorkbenchAdvisor.java:803) at
org.eclipse.ui.internal.Workbench$33.runWithException(Workbench.java:1600) at
org.eclipse.ui.internal.StartupThreading$StartupRunnable.run(StartupThreading.java:31) at
org.eclipse.swt.widgets.RunnableLock.run(RunnableLock.java:35) at
org.eclipse.swt.widgets.Synchronizer.runAsyncMessages(Synchronizer.java:135) at
org.eclipse.swt.widgets.Display.runAsyncMessages(Display.java:4140) at
org.eclipse.swt.widgets.Display.readAndDispatch(Display.java:3757) at
org.eclipse.ui.internal.Workbench.runUI(Workbench.java:2609) at
org.eclipse.ui.internal.Workbench.access$4(Workbench.java:2499) at
org.eclipse.ui.internal.Workbench$7.run(Workbench.java:679) at
org.eclipse.core.databinding.observable.Realm.runWithDefault(Realm.java:332) at
org.eclipse.ui.internal.Workbench.createAndRunWorkbench(Workbench.java:668) at
org.eclipse.ui.PlatformUI.createAndRunWorkbench(PlatformUI.java:149) at
org.eclipse.ui.internal.ide.application.IDEApplication.start(IDEApplication.java:123) at
org.eclipse.equinox.internal.app.EclipseAppHandle.run(EclipseAppHandle.java:196) at
org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.runApplication(EclipseAppLauncher.j
ava:110) at
org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.start(EclipseAppLauncher.java:79)
at org.eclipse.core.runtime.adaptor.EclipseStarter.run(EclipseStarter.java:344) at
org.eclipse.core.runtime.adaptor.EclipseStarter.run(EclipseStarter.java:179) at
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57) at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at
java.lang.reflect.Method.invoke(Method.java:606) at
org.eclipse.equinox.launcher.Main.invokeFramework(Main.java:622) at
org.eclipse.equinox.launcher.Main.basicRun(Main.java:577) at
org.eclipse.equinox.launcher.Main.run(Main.java:1410) at
org.eclipse.equinox.launcher.Main.main(Main.java:1386)
```

## Problem: Where to Get Call Studio Error Logs For Call Studio Applications

### Solution

The **Error Log view** captures all the warnings and errors logged by plug-ins. The underlying log file is a .log file stored in the .metadata subdirectory of the workspace. The Error Log view is available under **Window > Show View > Error Log**.

Message	Plug-in	Date
<b>Session</b>		<b>2008-06-05 14:54:30.000</b>
✘ Could not load repository template extension repository	org.eclipse.mylyn.tasks.ui	2008-06-05 14:54:36.593
<b>Session</b>		2008-06-05 11:23:08.437
⚠ NLS unused message: classpath_cannotUseLibraryAsOutput	org.eclipse.osgi	2008-06-05 11:23:54.437
✘ Could not load repository template extension repository	org.eclipse.mylyn.tasks.ui	2008-06-05 11:23:44.531
⚠ The workspace exited with unsaved changes in the previous	org.eclipse.core.resources	2008-06-05 11:23:30.375
<b>Session</b>		2008-06-05 09:15:46.949
⚠ NLS missing message: GenerateHashCodeEqualsDialog_inst	org.eclipse.osgi	2008-06-05 09:59:37.637
✘ Resource is out of sync with the file system: 'ganymede-a	org.eclipse.wst.sse.core	2008-06-05 09:47:18.184
✘ Resource is out of sync with the file system: 'ganymede-a	org.eclipse.wst.sse.core	2008-06-05 09:47:18.184

## Event Sorting

Events in the log view can be sorted by **Message**, **Plug-in ID** or **Date** in ascending or descending order. Simply click on the column header that you want the sorting to be based on. The down arrow in the column header indicates descending order; while, the up arrow indicates an ascending order.

## Event Grouping

Events in the log view can be grouped by **Session** or **Plug-in ID**. Simply click on the chevron from the view's toolbar and select **Group By**.

## Event Filtering

You can filter the view to show events of a particular type or session. Also, you can limit the number of entries in the view. Filtering options are available under **Filters...** from the view's toolbar drop down menu.

## Import and Export Logs

To **import** an arbitrary *.log* file into the view, press the **Import Log** toolbar button or select **Import Log...** from the context menu. Then, choose a *.log* file from the file system.

To **export** the current log view content into a file, press the **Export Log** toolbar button or select **Export Log...** from the context menu. Then, enter a file name.

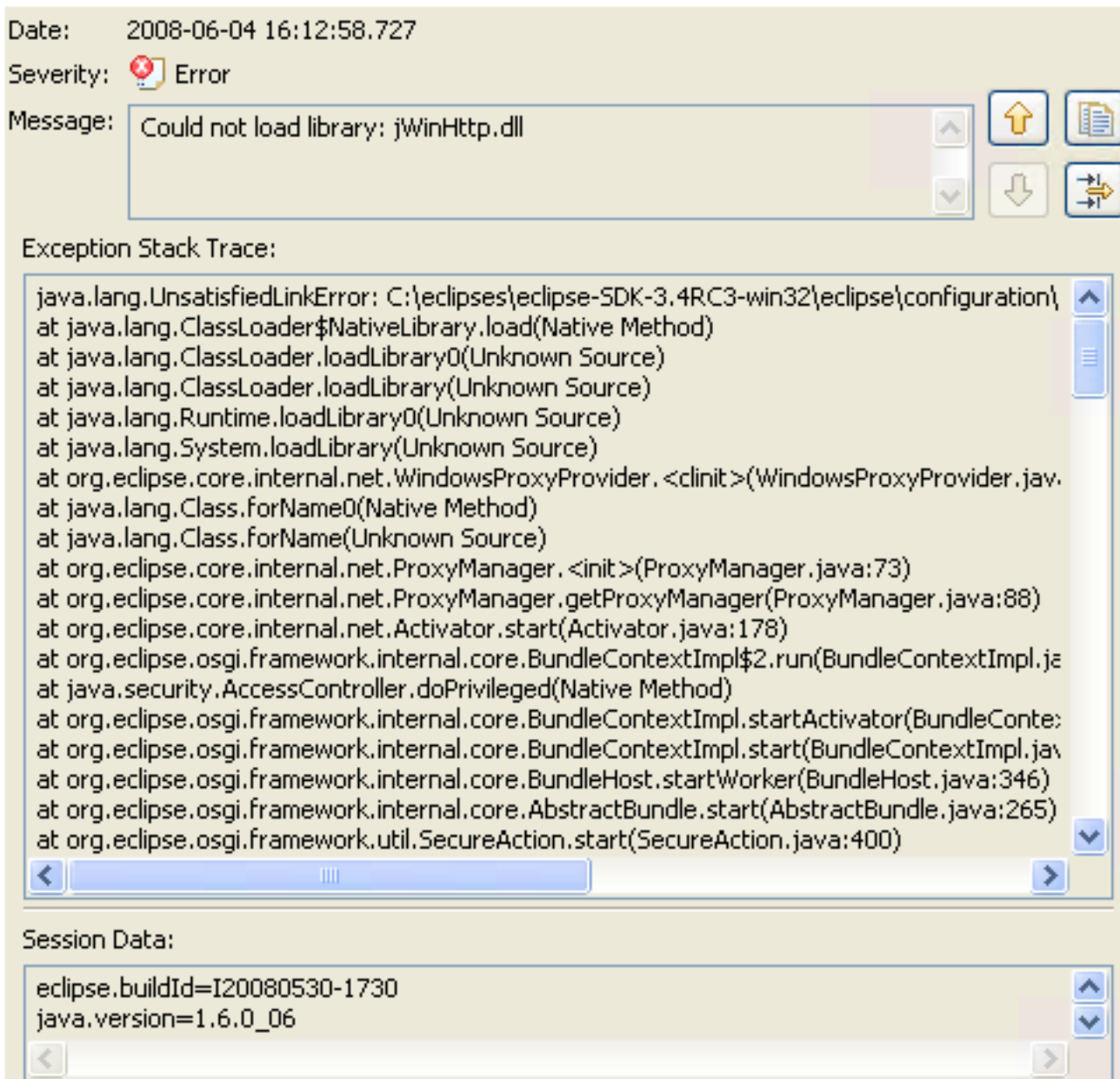
## Clear and Delete Logs

To **clear** the view log content without deleting the underlying *.log* file, press the **Clear Log** toolbar button or select **Clear Log Viewer** from the context menu.

To permanently **delete** the underlying *.log* file, press the **Delete Log** toolbar button or select **Delete Log** from the context menu.

## Event Details

Full details about a particular event can be viewed in the **Event Details** dialog by double-clicking on a particular entry or selecting **Event Details** from the context menu of that entry. You can view the **Date**, **Severity**, **Message**, **Exception Stack Trace** (if available) and **Session Data** of each event.



The screenshot displays the Eclipse Event Details dialog for an error event. The event details are as follows:

- Date:** 2008-06-04 16:12:58.727
- Severity:** Error (indicated by a red error icon)
- Message:** Could not load library: jWinHttp.dll

The **Exception Stack Trace** section shows the following call stack:

```
java.lang.UnsatisfiedLinkError: C:\eclipses\eclipse-SDK-3.4RC3-win32\eclipse\configuration\
at java.lang.ClassLoader$NativeLibrary.load(Native Method)
at java.lang.ClassLoader.loadLibrary0(Unknown Source)
at java.lang.ClassLoader.loadLibrary(Unknown Source)
at java.lang.Runtime.loadLibrary0(Unknown Source)
at java.lang.System.loadLibrary(Unknown Source)
at org.eclipse.core.internal.net.WindowsProxyProvider.<clinit>(WindowsProxyProvider.java:
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Unknown Source)
at org.eclipse.core.internal.net.ProxyManager.<init>(ProxyManager.java:73)
at org.eclipse.core.internal.net.ProxyManager.getProxyManager(ProxyManager.java:88)
at org.eclipse.core.internal.net.Activator.start(Activator.java:178)
at org.eclipse.osgi.framework.internal.core.BundleContextImpl$2.run(BundleContextImpl.java:
at java.security.AccessController.doPrivileged(Native Method)
at org.eclipse.osgi.framework.internal.core.BundleContextImpl.startActivator(BundleConte:
at org.eclipse.osgi.framework.internal.core.BundleContextImpl.start(BundleContextImpl.java:
at org.eclipse.osgi.framework.internal.core.BundleHost.startWorker(BundleHost.java:346)
at org.eclipse.osgi.framework.internal.core.AbstractBundle.start(AbstractBundle.java:265)
at org.eclipse.osgi.framework.util.SecureAction.start(SecureAction.java:400)
```

The **Session Data** section shows the following information:

```
eclipse.buildId=I20080530-1730
java.version=1.6.0_06
```

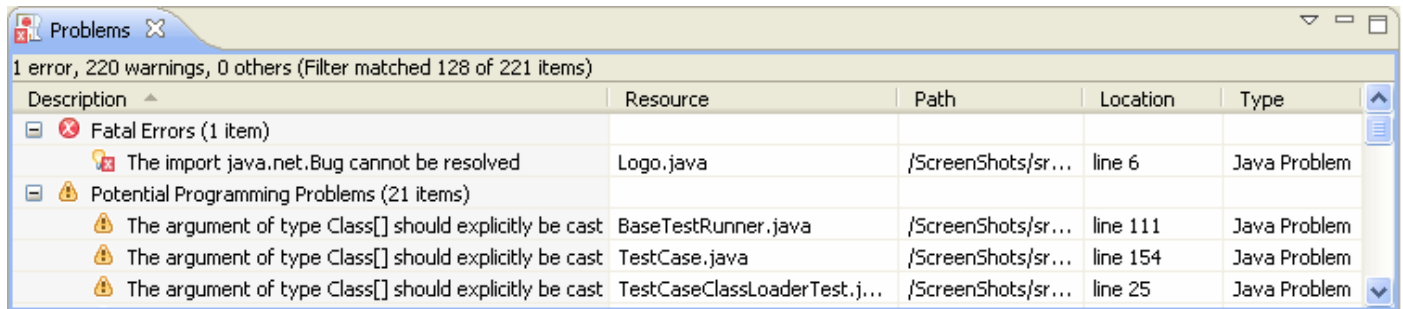
You can navigate from one entry to the next via the **Up** and **Down** arrow buttons.

To copy the error to the clipboard, press the button with the clipboard image.

## Problem: Where to Get Call Studio Logs for Application Validation

# Solution

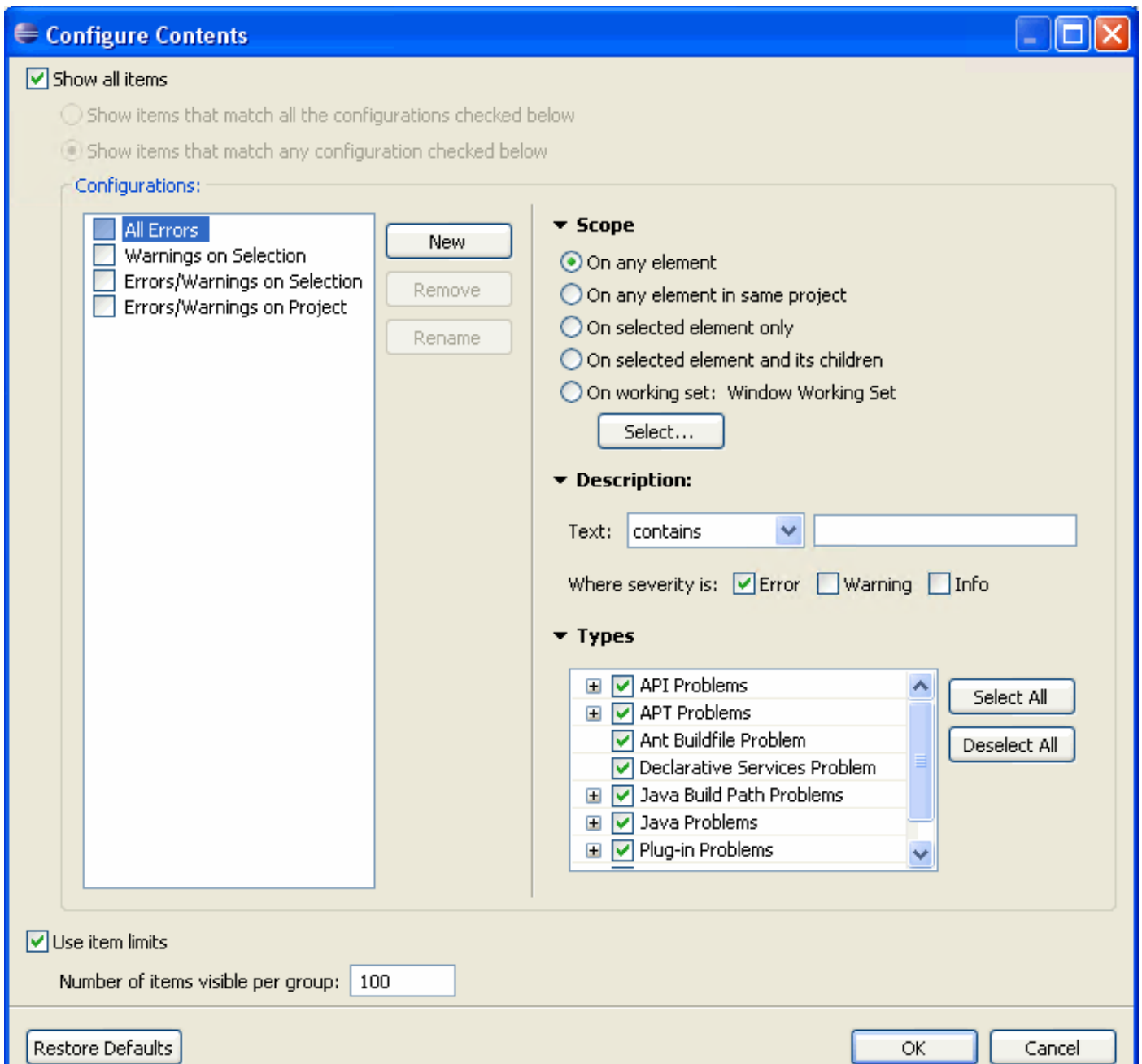
As you work with resources in the workbench, various builders can automatically log problems, errors, or warnings in the Problems view. For example, when you save a Java source file that contains syntax errors, those get logged in the Problems view. When you double-click the icon for a problem, error, or warning, the editor for the associated resource automatically opens to the relevant line of code.



By default the Problems view groups your problems by severity. You can also group them by type or not at all. Certain components add their own grouping. For instance the Java development tools (JDT) support adds a Java Problem Type group. The grouping can be selected using the **Group By** menu.

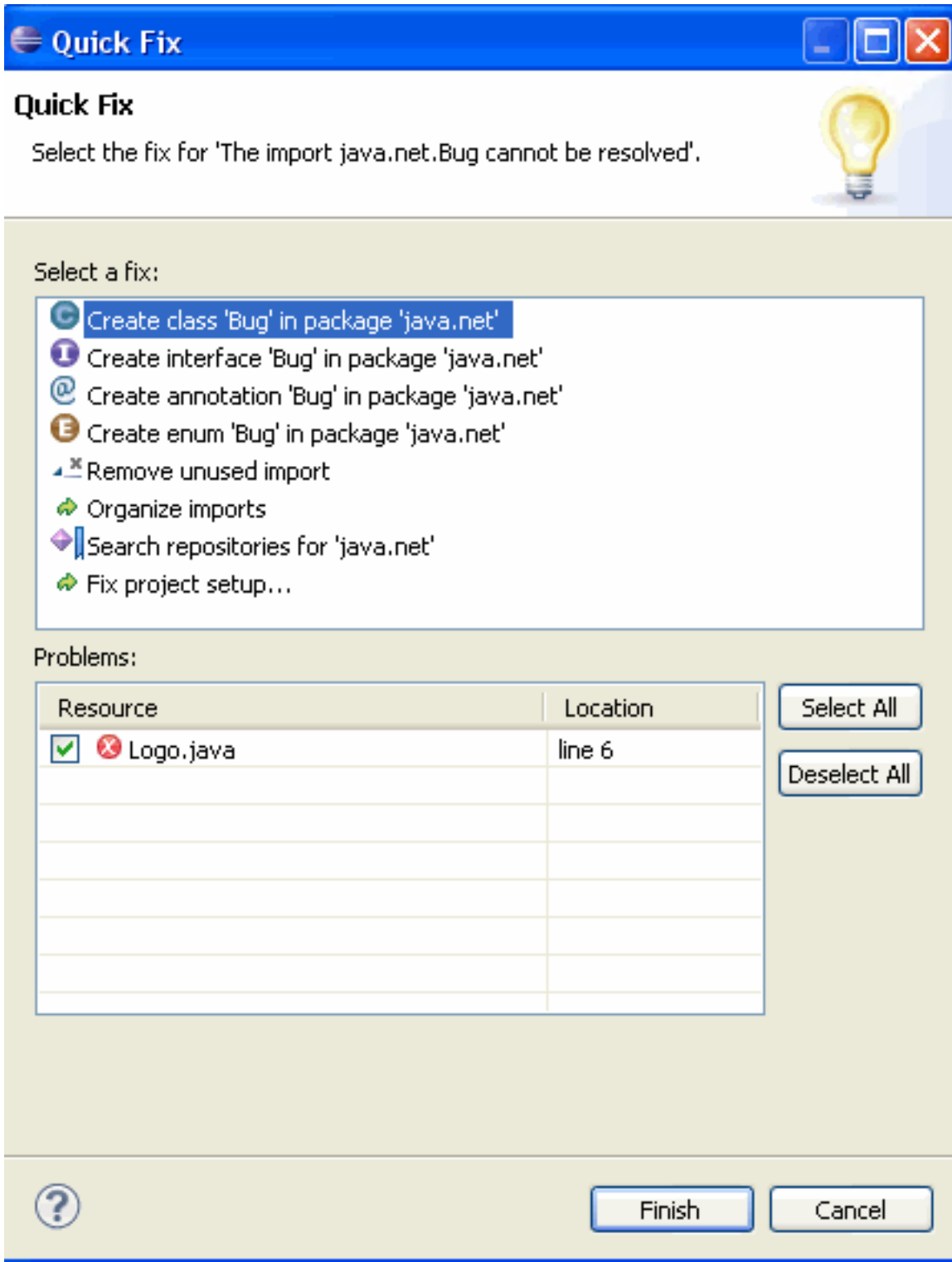
The first column of the Problems view displays an icon that denotes the type of line item, the category and the description. Left-click the item to open the file in an editor and highlight the line containing the problem.

You can configure the contents of the Problems view to view only warnings and errors associated with a particular resource or group of resources. This is done using the **Configure Contents** dialog available from the drop down menu. You can add multiple filters to the Problems view and enable or disable them as required. Filters can either be additive (any problem that satisfies at least one of the enabled filters will be shown) or exclusive (only problems that satisfy all of the filters will be shown) The two most popular filters (All Errors and Warnings on Selection) are provided by default.



Problems can be fixed by selecting **Quick Fix** from the context menu. The list of possible resolutions are presented.





To add the Problems view to the current perspective, click Window > Show View > Other... > General > Problems.