

# ログを使用したCisco CVP Call Studioのトラブルシューティング

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[問題：Call Studioインストールログの入手先](#)

[解決方法](#)

[問題：Call Studioライセンスログの入手場所](#)

[解決方法](#)

[問題：Call StudioアプリケーションのCall Studioエラーログの入手場所](#)

[解決方法](#)

[問題：アプリケーション検証のためのCall Studioログの取得場所](#)

[解決方法](#)

## 概要

このドキュメントでは、トラブルシューティングの一般的なCisco Call Studioのシナリオとそのログの入手場所について説明します。

## 前提条件

### 要件

次の項目に関する知識があることが推奨されます。

- Cisco Call Studio

### 使用するコンポーネント

このドキュメントの情報は、Cisco Call Studio 11.6および/または12.0に基づいています。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

## 問題：Call Studioインストールログの入手先

### 解決方法

Call Studioのインストールログは、C:\Tempディレクトリにあります。

新規インストールのログのスニペットの例を次に示します。

```
[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=
```

## 問題 : Call Studioライセンスログの入手場所

### 解決方法

Call Studioへのライセンスは、バージョン11.6まで有効です。12.0以降、Call Studioにはライセンスは必要ありません。

ライセンスログは次の場所で入手できます。

次にログファイルの例を示します。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.jav  
a: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(EclipseLazyStart  
er.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(RegistryS  
trategyOSGI.java:184) at  
org.eclipse.core.internal.registry.ExtensionRegistry.createExecutableExtension(ExtensionRegist  
ry.java:905) at  
org.eclipse.core.internal.registry.ConfigurationElement.createExecutableExtension(ConfigurationE  
lement.java:243) at  
org.eclipse.core.internal.registry.ConfigurationElementHandle.createExecutableExtension(Configur  
ationElementHandle.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

```

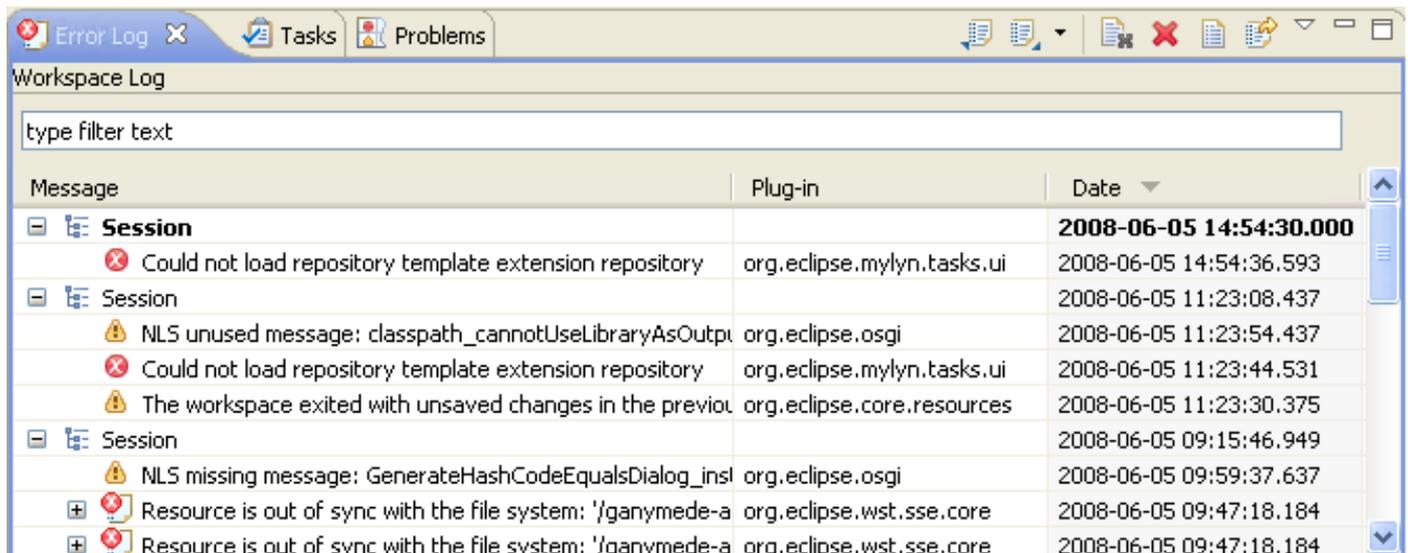
org.eclipse.swt.widgets.Display.readAndDispatch(Display.java:3757) at
org.eclipse.ui.internal.Workbench.runUI(Workbench.java:2609) at
org.eclipse.ui.internal.Workbench.access$4(Workbench.java:2499) at
org.eclipse.ui.internal.Workbench$7.run(Workbench.java:679) at
org.eclipse.core.databinding.observable.Realm.runWithDefault(Realm.java:332) 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.java: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)

```

## 問題：Call StudioアプリケーションのCall Studioエラーログの入手場所

### 解決方法

[Error Log]ビューには、プラグインによって記録されたすべての警告とエラーが表示されます。基になるログファイルは、ワークスペースの.metadataサブディレクトリに保存される.logファイルです。[Error Log]ビューは、[Window] > [Show View] > [Error Log]で使用できます。



### イベントソート

ログビューのイベントは、メッセージ、プラグインID、または日付の順に並べ替えることができます。並べ替えの基準にする列見出しをクリックします。列ヘッダーの下矢印は、降順を示します。ただし、上矢印は昇順を示します。

## イベントのグループ化

ログビューのイベントは、セッションまたはプラグインIDでグループ化できます。ビューのツールバーの山形をクリックし、「グループ化」(Group By)を選択します。

## イベントフィルタリング

ビューをフィルタして、特定のタイプまたはセッションのイベントを表示できます。また、ビューのエントリ数を制限することもできます。フィルタオプションは、ビューのツールバーのドロップダウンメニューの[フィルタ...]で使用できます。

## ログのインポートとエクスポート

任意の.logファイルをビューにインポートするには、*Import Log* toolbarボタンを押すか、コンテキストメニューから*Import Log...*を選択します。次に、ファイルシステムから.logファイルを選択します。

現在のログビューのコンテンツをファイルにエクスポートするには、ログのエクスポートのツールバー・ボタンを押すか、コンテキスト・メニューからログのエクスポートを選択します。次に、ファイル名を入力します。

## ログのクリアと削除

基になる.logファイルを削除せずにビューログの内容をクリアするには、*Clear Log* toolbarボタンを押すか、コンテキストメニューから*Clear Log Viewer*を選択します。

基になる.logファイルを完全に削除するには、ツールバーの削除ボタンを押すか、コンテキストメニューからログの削除を選択してください。

## イベントの詳細

特定のイベントに関する詳細は、[イベントの詳細]ダイアログでは、特定のエントリをダブルクリックするか、そのエントリのコンテキストメニューから[イベントの詳細]を選択して表示できます。各イベントの日付、重大度、メッセージ、例外スタックトレース(可能な場合)、およびセッションデータを表示できます。

Date: 2008-06-04 16:12:58.727  
Severity:  Error  
Message: Could not load library: jWinHttp.dll

Exception Stack Trace:

```
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)
```

Session Data:

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

上向き矢印ボタンと下向き矢印ボタンを使用して、1つのエントリから次のエントリへ移動することができます。

エラーをクリップボードにコピーするには、クリップボードの画像を含むボタンを押します。

## 問題：アプリケーション検証のためのCall Studioログの取得場所

### 解決方法

ワークベンチでリソースを操作すると、さまざまなビルダーが[問題]ビューで問題、エラー、または警告を自動的にログに記録できます。たとえば、構文エラーを含むJavaソースファイルを保存すると、それらのファイルが[問題]ビューに記録されます。問題、エラー、または警告のアイコンをダブルクリックすると、関連するリソースのエディタが自動的に関連するコード行に開きます。

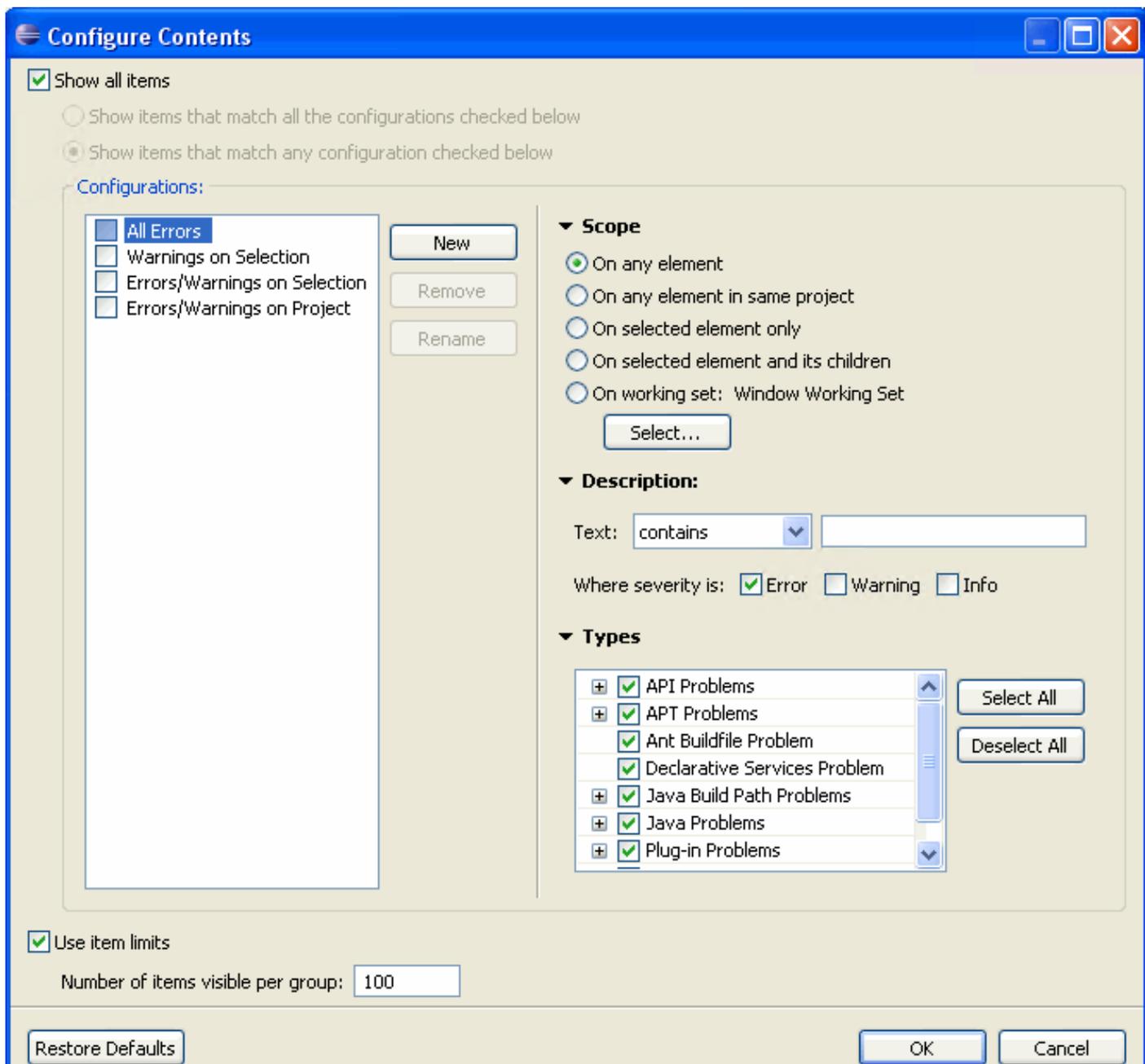
Description	Resource	Path	Location	Type
<b>Fatal Errors (1 item)</b>				
The import java.net.Bug cannot be resolved	Logo.java	/ScreenShots/sr...	line 6	Java Problem
<b>Potential Programming Problems (21 items)</b>				
The argument of type Class[] should explicitly be cast	BaseTestRunner.java	/ScreenShots/sr...	line 111	Java Problem
The argument of type Class[] should explicitly be cast	TestCase.java	/ScreenShots/sr...	line 154	Java Problem
The argument of type Class[] should explicitly be cast	TestCaseClassLoaderTest.j...	/ScreenShots/sr...	line 25	Java Problem

デフォルトでは、[Problems]ビューでは問題が重大度ごとにグループ化されます。タイプ別にグループ化することも、まったくグループ化しないこともできます。特定のコンポーネントが独自のグループ化を追加します。たとえば、Java開発ツール(JDT)のサポートにより、Java問題タイプ(JDT)グループが追加されます。グループ化は、[グループ化]メニューを使用して選択できます。

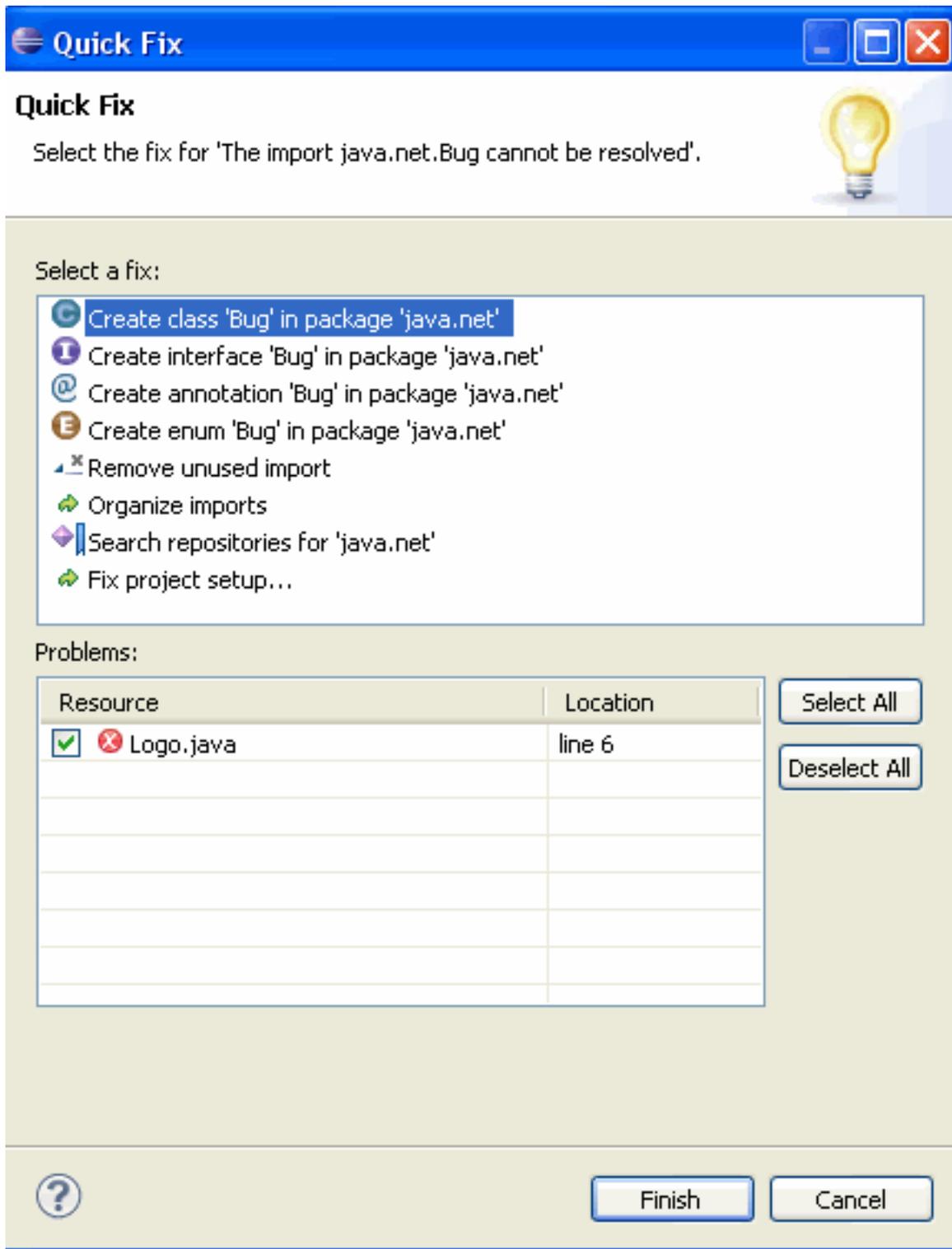
。

[問題]ビューの最初の列に、品目のタイプ、カテゴリ、説明を示すアイコンが表示されます。項目を左クリックしてファイルをエディタで開き、問題を含む行を強調表示します。

[問題]ビューの内容を設定して、特定のリソースまたはリソースグループに関連する警告とエラーだけを表示できます。これを行うには、ドロップダウンメニューから[Configure Contents]ダイアログを使用します。[Problems]ビューに複数のフィルタを追加し、必要に応じてそれらを有効または無効にできます。フィルタは、加算（有効なフィルタの少なくとも1つを満たす問題が表示されます）または排他（すべてのフィルタを満たす問題だけが表示されます）のいずれかです。最も一般的な2つのフィルタ（選択時のエラーと警告）がデフォルトで提供されます。



問題を解決するには、コンテキストメニューから「クイック修正」を選択します。可能な解決策のリストが表示されます。



問題ビューを現在のパースペクティブに追加するには、[ウインドウ] > [ビューの表示] > [その他...] > [全般] > [問題]をクリックします。