Getting Started with Adobe Version Cue CS3 Development

This document describes how to get started with development using the Adobe® Version Cue® CS3 SDK. The target audience for this document is systems integrators who need to develop plug-ins for Adobe Version Cue Server™.

Terminology

The following terms are used throughout this document:

- **Eclipse** — An integrated development environment (IDE) for Java™ (and other languages) and a run-time environment for applications. Eclipse-based products, like Version Cue Server, use the Eclipse run-time. The Eclipse run-time starts the Eclipse platform base, and it dynamically discovers and loads (on-demand) plug-ins that make up the Eclipse-based product. Version Cue SDK is another Eclipse-based product, based on the Eclipse workbench.

- **Target platform** — The platform for which plug-ins are developed, namely Version Cue Server.

- `<vcsdk>` — The path to the folder where you installed Version Cue SDK.

- **Version Cue SDK** — The SDK used to develop plug-ins for Version Cue Server, which also has resources to help in developing server-driven scripts for Version Cue clients. This is a pre-configured version of Eclipse, branded by Adobe. It has an integrated development environment (IDE) for developing Version Cue Server plug-ins; it includes custom Adobe plug-ins that provide wizards, samples, Help documentation, and additional features to support development of Version Cue Server plug-ins.

- **Version Cue Server** — The server component of Version Cue CS3.

Introduction

To start developing plug-ins for Version Cue Server, perform these steps:

1. Check that your machine meets the basic requirements specified in “Development Platforms” on page 6.

2. Set up your machine for development; follow the instructions in “Setting up your Development Platform” on page 7.
3. Examine the documentation: inspect the Samples guide (accessible from the Welcome page), import the samples from the Welcome page, and compile and run them. See "Exploring the Version Cue SDK Samples and Documentation" on page 22.

4. Write a plug-in by following the instructions in “Creating your First Plug-in” on page 24.

**Development Platforms**

Platforms supported for Version Cue plug-in development, and required components, are listed in Table 1:

**Table 1 Supported Development Platforms and Required Components**

<table>
<thead>
<tr>
<th>Platform</th>
<th>Component</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mac OS 10.4 or</td>
<td>Mac OS 10.4 or later</td>
<td>Mac OS 10.3 and earlier are not supported. You must use 10.4 or later as your development platform on the Mac. Both MacTel and PPC are supported.</td>
</tr>
<tr>
<td>later</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Version Cue Server (3.1.0)</td>
<td>Install Version Cue Server from an installer; the SDK expects the Server to be located in the fixed install location.</td>
</tr>
<tr>
<td></td>
<td>Version Cue SDK (1.1.0)</td>
<td>This is based on Eclipse 3.2 Final (29 June 2006), configured with Adobe plug-ins, preferences, and other features to help developing plug-ins for Version Cue Server. You should install the SDK from the installer.</td>
</tr>
<tr>
<td></td>
<td>JRE 1.5</td>
<td>You must have JRE 1.5 installed on your machine; this is the Java run-time environment required to launch Version Cue SDK. If you have been accepting software updates from Apple, this should be installed.</td>
</tr>
</tbody>
</table>
Setting up your Development Platform

Installation and Configuration

Before you Begin

- You will need a Suite installer, which will enable you to install Version Cue CS3 Server. Please run this Suite installer (to install the Version Cue Server) before installing the Version Cue SDK. If you do not do this, then you must update your target platform manually from within the IDE of Version Cue SDK.

Other components you will need to work with the Version Cue SDK are listed in Table 2:

### Table 2 Other Required Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe Reader 7.0 or later</td>
<td>This is required to view Adobe PDF documentation.</td>
</tr>
<tr>
<td>Adobe Creative Suite® 3 point products</td>
<td>Use point products such as Adobe Bridge™ CS3, Adobe Photoshop® CS3, and Adobe InDesign® CS3 to create and work with content in Version Cue CS3 Server. You must use versions compatible with the Version Cue CS3 Server you are using.</td>
</tr>
</tbody>
</table>
It is recommended to use the latest installer for Version Cue SDK (1.1.0 at the time of writing). If you already have installed Version Cue SDK 1.0.0 (GM), then you should be able to update to Version Cue SDK 1.1.0 by running the latest installer and upgrading your existing SDK.

You should be sure that your Version Cue Server is upgraded to 3.1.0, which is the version required for Version Cue SDK 1.1.0. This also requires you to upgrade your Adobe Bridge CS3 application to 2.1 and be sure that Version Cue client libraries are upgraded to 3.1.0.

You must have at least one point product in Creative Suite CS3 installed, such as Adobe Bridge CS3, and it should be a build that is known to work with the Version Cue Server you have installed. It is recommended that you have Version Cue SDK 1.1.0, Version Cue Server 3.1.0, Version Cue Client 3.1.0 and Adobe Bridge CS3 2.1.

If you already have Adobe Creative Suite 1 installed, you must remove it, since it is not compatible with Version Cue CS3. Before removing Adobe Creative Suite 1, export any valuable files stored in Version Cue CS1 to another computer, using the Administration user interface.

If you already have Adobe Creative Suite 2 installed, export any valuable files stored in Version Cue CS2 to another computer and stop Version Cue CS2 (see “Stopping Version Cue CSx” on page 12).

**Installing the Version Cue SDK**

You already should have installed Creative Suite CS3, and downloaded Version Cue SDK installer for your platform. To start working with the SDK, follow these steps:

1. It is recommended to upgrade Version Cue Server and Client components to 3.1.0 (the latest at the time of writing), and Adobe Bridge CS3 to version 2.1. In this case, you must use Version Cue SDK 1.1.0 (dot release). Note that some of the new samples (sample/remoteproject, for instance) added between Version Cue SDK 1.0.0 and 1.1.0 depend on using Version Cue Server 3.1.0, since there were some additions the API for Version Cue Server 3.1.0.

2. Install Version Cue SDK 1.1.0 (dot release). If you already have installed Version Cue SDK 1.0.0 (GM), then you should be able to update to Version Cue SDK 1.1.0 by simply running the latest installer and choosing the option to update your existing SDK.

3. Install the SDK to a drive with at least 500 MB of free space (and preferably more). On Windows, use an NTFS-formatted drive (not FAT/32 nor a network drive).

4. For the rest of this document, we refer to the folder path containing the installed Version Cue SDK and other components as `<vcsdk>`. Note that the installer will by default install into these folders:

   - `c:\Program Files\Adobe\Adobe Version Cue CS3 SDK\` (Windows)
   - `/Applications/Adobe/Adobe Version Cue CS3 SDK/` (Mac OS)

5. Examine the `<vcsdk>` folder- its contents are shown Table 3. Note the PDF documentation is in the `sdk/docs` folder. Conceptual documentation can also be found under Version Cue SDK Help, along with reference documentation like the server Java API Reference, Extension Point Reference and Client Scripting Reference.
### TABLE 3 Version Cue SDK Content

<table>
<thead>
<tr>
<th>Folder/ file name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“VersionCueSDK”</td>
<td>Folder containing the IDE for Version Cue SDK, which contains Adobe branding and plug-ins that customize the base Eclipse IDE. Sample plug-ins must be imported from the Welcome page (or mini-Welcome page) via a wizard.</td>
</tr>
<tr>
<td>sdk.docs</td>
<td>Folder containing conceptual documentation, such as the PDF documentation in this table. This content also is in the documentation under Version Cue SDK &gt; Help.</td>
</tr>
<tr>
<td>Readme.txt</td>
<td>The Readme file for this SDK.</td>
</tr>
<tr>
<td>legal</td>
<td>Folder containing End-User License Agreements, as shown by the SDK installer.</td>
</tr>
<tr>
<td>sdk.docs/getting-started.pdf</td>
<td>This document.</td>
</tr>
<tr>
<td>sdk.docs/programmers-guide.pdf</td>
<td><em>Adobe Version Cue CS3 Server Programmer's Guide</em>; this is the Version Cue Server guide for this SDK.</td>
</tr>
<tr>
<td>sdk.docs/connecting-external-system.pdf</td>
<td><em>Case Study: Connecting to an External System through Adobe Version Cue CS3</em>. This is based on the reference implementation CustomProject.</td>
</tr>
<tr>
<td>sdk.docs/client-programmers-guide.pdf</td>
<td><em>Version Cue CS3 Client Programmers Guide</em>. Describes the extensibility features at the client enabled by this SDK and working with the scripting interface to Version Cue CS3 clients.</td>
</tr>
</tbody>
</table>

## Running Version Cue SDK for the First Time

To run Version Cue SDK initially, follow these steps:

1. Make sure that you have installed Version Cue SDK from the installer, as described above. If you have updated from Version Cue SDK 1.0.0 (GM) to 1.1.0 (dot release), then you should delete the Eclipse workspace you were using previously and start with a clean, fresh Eclipse workspace.

2. Run the Version Cue SDK application in the <vcsdk>/VersionCueSDK folder. During start-up, you should see an Adobe-branded splash-screen.
3. When prompted to select a workspace, choose a new folder, <vcsdk>/workspace, and check “Use this as the default…”.

4. Note that you should look in the "VersionCue.log" created on first startup to confirm there are no errors; if everything is set up correctly, the only information in the log should be that the target platform has been changed correctly. You can find this log file in the Eclipse workspace you choose, in the folder <vcsdk>/workspace/vc-data/logs.

5. From the Welcome screen, you can import sample plug-ins. They are organized into functional categories; a sample can fall into multiple categories. Choose one or more categories, and one or more sample plug-ins is imported into your workspace. If you already imported a sample plug-in, you are given the chance to replace it. In Figure 1, samples related to notification will be imported.

6. Verify that you can browse the Version Cue SDK programming documentation from both the Welcome screen and the Help system. To check within the Help system, select Help > Help Contents > Adobe Version Cue SDK Help.

**FIGURE 1** Importing Samples from the Welcome Screen via the Wizard

Version Cue SDK sets up the target platform to refer to the Version Cue Server it expects to find in the default installation location for Creative Suite CS3. If you install the Version Cue Server after the SDK, you need to set the target platform manually; see “Plug-in Development Target Platform Preferences” on page 10.

**Plug-in Development Target Platform Preferences**

If you install Version Cue Server and then the SDK, you should not need to change settings related to the target platform, as they are set up automatically by Version Cue SDK. However, if you inadvertently install Version Cue Server after the SDK, then you will need these steps.

To set preferences for the plug-in development target platform, follow these steps:

2. Verify the Location field is set to refer to the installed Version Cue Server. Change it to this path if needed— for instance, if you have installed the Server after the SDK and the target platform is not defined. The installation paths are as follows:

   /Library/Application Support/Adobe/Adobe Version Cue CS3/Server/ (Mac OS)
   C:\Program Files\Common Files\Adobe\Adobe Version Cue CS3\Server\ (Win)

**Java Compiler Preferences**

*This section is for information only.* You should not need to change settings related to compiler preferences, as they already are set up for you within the Version Cue SDK IDE.

To set Java compiler preferences, follow these steps:

1. Select Window > Preferences > Java > Compiler,. The preferences dialog for the Java compiler opens.

2. Verify the option for JDK compliance level is set to 5.0, meaning JDK 1.5.

3. Verify the option “Generated .class files compatibility” is set to 5.0.

4. Verify the option “Source compatibility” is set to 5.0.

From within the IDE, you can switch to the Java perspective by selecting Window > Open Perspective > Java.

**Java Installed JREs Preferences**

*NOTE:* *Windows:* before you can launch Version Cue SDK for the first time, you must have JRE 1.5 installed on your machine. Once you launch Version Cue SDK, you can switch the JRE preference to the one supplied with the installed Version Cue Server.

1. Select Window > Preferences > Java > Installed JREs. The Installed JREs dialog for the Java compiler opens.

2. Choose the JRE for your platform: On Windows, add the JRE that Version Cue Server uses, in a subfolder “jre” of the installation folder. If you have other JREs installed, they also are listed. Check the box for the Version Cue Server JRE; see Figure 2. On Mac OS, verify you are using the Mac OS JRE 1.5; see Figure 3.

3. The IDE may ask whether you want to do a full rebuild. Answer Yes.
Stopping Version Cue CSx

Before running Version Cue Server, stop any other Version Cue Server(s) running on your local machine. For instance, if you installed Version Cue CS2 as part of Creative Suite CS2, make sure it is turned off.

On Windows

To stop Version Cue CSx on Windows, follow these steps:
1. Open the Version Cue CSx preferences by double-clicking its tray bar icon. The icon for CS2 is a green leaf; for CS3, the icon has changed somewhat and is no longer a leaf. If the icon is not visible, open the Windows control panel and double-click the Adobe Version Cue CSx item.

2. Use Settings > Version Cue CSx to stop the installed server.

3. Uncheck Settings > Turn Version Cue On When Computer Starts, to prevent the installed server from restarting automatically.

4. Click OK to save your changes. Wait until shutdown completes.

5. Verify no instance of a Version Cue Server is running: start the Task Manager by pressing Ctrl-Alt-Del, and choose the Processes tab. For instance, if you stop Version Cue CS2, there should be no instance of the “VersionCueCS2.exe” process. You can leave the VersionCueCSxTray.exe process running. Stopping this process does not stop Version Cue CSx.

**On Mac OS**

To stop Version Cue CSx on Mac OS, follow these steps:

1. Open the Version Cue CSx preferences window by clicking its menu bar icon and selecting the Version Cue CSx Preferences menu item. The icon for CS2 is a green leaf; for CS3, the icon has changed somewhat and is no longer a leaf. If the icon is not visible, open the Mac System Preferences window and double-click the Version Cue CSx item.

2. Use Settings > Version Cue CSx to stop the installed server.

3. Uncheck Settings > Turn Version Cue On When Computer Starts, to prevent the installed server from restarting automatically.

4. Click Apply to save your changes. Wait until shutdown completes.

5. Verify no instance of a Version Cue Server is running, by using the Activity Monitor. This application normally is in your Applications/Utilities folder. For instance, if you stop Version Cue CS2, there should be no instance of the “VersionCueCS2” process. You can leave the VersionCueCSxStatus process running. Stopping this process does not stop Version Cue Server.

**Stopping the MySQL Database Service**

If you started Version Cue Server from Version Cue SDK and have not used the “Shutdown Version Cue Server” button to terminate it (see “Stopping Version Cue Server from the IDE of Version Cue SDK” on page 20), you will have a MySQL database service running. You need to terminate this service before re-starting the server.

**On Windows**

To determine whether MySQL is running, use Task Manager > Processes (look for the mysqld-nt.exe process) or execute the following command from a command line:

```
telnet localhost 50702
```
If you see “Connect failed,” terminate the connection with Ctrl-D. If you see anything else, MySQL is running.

To terminate MySQL, open the Task Manager > Processes tab, and terminate the mysqld-nt.exe process.

**On Mac OS**

To determine whether MySQL is running, use the Activity Monitor. This application normally is in your Applications/Utilities folder.

To terminate MySQL, filter for a process called mysqld, and quit that process.

### Preparing to Start Version Cue Server

To configure Version Cue SDK to be able to run Version Cue Server, follow these steps:

1. From within Version Cue SDK, select Window > Open Perspective > Java, to switch to the Java perspective.

2. Select Run > Run, to create a new Version Cue Server configuration.

3. In the Configurations Pane (the left-hand pane), select “Version Cue Server” and click the New Launch Configuration button.

4. Specify “Version Cue Server” as the configuration name.

5. Click Apply, to accept the default settings; see **Figure 4**.

6. Verify the “Main > Program to Run > Run an application” field contains the following value:

   com.adobe.versioncue.tomcat.application

7. Click Close, to complete the “Version Cue Server” configuration.
Running Version Cue Server for the First Time

Regardless of how you start Version Cue Server, before you start it for the first time, you must secure the Version Cue Server by navigating to the Administration page and supply a password for the system admin user. By default, the Administration page is http://localhost:3703/. If you do not set a password for the system admin user, you will not be able to connect to your Version Cue Server from any Version Cue client; your Version Cue Server will be invisible to Version Cue clients.

Starting Version Cue Server from the IDE of Version Cue SDK

To run Version Cue Server from within the IDE of Version Cue SDK, follow these steps:

1. From within the IDE, select Window > Open Perspective > Java, to switch to the Java perspective.
2. Select Run > Run.
3. In the Configurations pane (the left-hand pane), select “Run Version Cue Server.” If you did not yet create the configuration to run Version Cue Server, you will not have a “Run Version Cue Server” configuration. To create it, follow the instructions in “Preparing to Start Version Cue Server” on page 14.

4. Click the Run button. If the Version Cue Server starts successfully, you should see a series of messages, (which may include non-fatal error messages). The following is an example of the start-up messages:

2006-10-12 ... [main] OFF com.adobe.versioncue.boot - ### Starting Version Cue® CS3...
2006-10-12 ... [main] OFF com.adobe.versioncue.tomcat - ### Version Cue 3.0 up and running ###

On Windows, when the server is starting, the Version Cue splash screen should appear briefly. On Mac OS, the splash screen does not display.

A console message within the IDE confirms the server is running. This should look like Figure 5, although you may see additional, non-fatal errors (as described above) after the server starts. The figure shows the typical output on starting Version Cue Server from within the IDE of Version Cue SDK.

Note that in addition to the output shown in Figure 5, when you start Version Cue Server without the system admin password set, you will also see warnings that you need to visit the administration interface at http://localhost:3703/ and set the system admin password.

**FIGURE 5**  
**Console Output on Starting Version Cue Server within Version Cue SDK**

<table>
<thead>
<tr>
<th>Overview</th>
<th>Dependencies</th>
<th>Runtime</th>
<th>Extensions</th>
<th>Extension Points</th>
<th>Build</th>
<th>MANIFEST.MF</th>
<th>build properties</th>
<th>plugin.xml</th>
</tr>
</thead>
</table>

**NOTE:** To further check the server is running, you should follow the instructions in “Checking that Version Cue Server is Running” on page 21.

**Launching Version Cue Server Through the IDE Versus Normal Operation**

**Default Port and the Administration Interface**

When you run an installed Version Cue Server under control of the IDE of Version Cue SDK, you will notice some differences to when it is running normally, if started through the System Preferences > Version Cue CS3 (Mac) or Control Panel > Version Cue CS3 (Win).

For instance, the port number on which Version Cue Server listens by default is 3703 when you launch Version Cue Server under control of the IDE of the SDK; for instance, the adminis-
Getting Started with Adobe Version Cue CS3 Development

Setting up your Development Platform

The administration interface would be http://localhost:3703, for the Version Cue Server running on your local machine. You will find that the port Version Cue Server listens on when started as a system service through the System Preferences (Mac) or Control Panel (Win) is something like 50xxx, and varies dynamically.

Workspaces and Databases

There is another fundamental difference between starting Version Cue Server under control of the IDE of Version Cue SDK, versus starting it through the System Preferences (Mac) or Control Panel (Win). When you create an eclipse-workspace within the IDE of Version Cue SDK, this eclipse-workspace now has its own Server database. In other words, when you start Version Cue Server under control of the IDE of this SDK, then any information added to the Version Cue Server is stored in a database that is within your eclipse-workspace folder. If you had already added assets to the Version Cue Server started through the System Preferences (Mac) or Control Panel (Win), then these will not show up under the projects that appear when you browse the Version Cue Server started under control of the IDE. Vice versa, the database underlying the installed Version Cue Server is not affected when you start Version Cue Server under control of the IDE of this SDK—each eclipse-workspace that you create when using the IDE has its own local database.

Note that a new Version Cue Server database is created within the eclipse-workspace folder when the IDE of this SDK is first started with a clean workspace.

To set up your eclipse-workspace correctly for development within the IDE of this SDK, the Version Cue Server matching the SDK must be installed or must reside as a peer of the VersionCueSDK folder containing the executable for the IDE of this SDK. We strongly recommend to install Version Cue Server first and than install the Version Cue SDK.

Obtaining Debug Information

If you started Version Cue Server under control of the debugger (Debug > Version Cue Server), you can view additional debug information, if you set a breakpoint and inspect assets from within the Variables view of the Debug perspective.

The Variables view in the Debug perspective of IDE shows additional information about assets and asset containers. The IDs and URIs of assets, and the IDs, URIs, and children of asset containers are displayed. This is shown in Figure 6.
FIGURE 6  Debug > Variables View, Showing Information for an Asset Identity

Also see the preferences: Java > Debug > Logical Structures. Here you can find the added structures related to the Version Cue Server API.

Figure 7 shows IDE set-up to trace information about SOAP calls and controller calls to the IDE console, and representative output. This tracing enables you to more easily debug the following operations:

- Registration of SOAP methods and types and invocation of SOAP methods.
- Creation and invocation of controller methods (but only for external namespaces).
To use this feature, follow these steps:

1. Open (or create) the Version Cue Server launch configuration, as described in “Preparing to Start Version Cue Server” on page 14.

2. Open the “Tracing” tab.

3. Check “Enable tracing for selected plug-ins.”

4. Select the checkbox for the plug-ins “com.adobe.versioncue.controller.core” (to trace information about controller calls in external namespaces only) and/or “com.adobe.versioncue.soapdime” (to trace information about SOAP messages for all namespaces).

5. Select the relevant options for those plug-ins.

6. Once you start Version Cue Server, you see log entries like this:
Stopping Version Cue Server from the IDE of Version Cue SDK

To stop Version Cue Server from within the Version Cue SDK IDE, use the “Shutdown Version Cue Server” button, shown in Figure 8. This button appears to the left of the Debug icon.

This stops both Version Cue Server and its MySQL database service in one step.

Alternatively, you can stop Version Cue Server and its MySQL database service in two steps, as follows:

1. Stop Version Cue Server by clicking the Terminate button (red square), in either the Console or Debug view. See Figure 9. If Version Cue Server failed to start properly for any reason, this button is not enabled.

2. Stop the MySQL database service; see “Stopping the MySQL Database Service” on page 13 or “On Mac OS” on page 14.
Checking that Version Cue Server is Running

To verify the server is running, follow these steps:


2. If the server is running, the Version Cue Administration login web page is displayed. Figure 10 shows how the Administration interface appears when running Version Cue Server for the first time.
NOTE: The Administration interface should look similar but not identical to that shown above, due to late-breaking changes. However, the functionality exposed to the user should be equivalent to that shown above.

Exploring the Version Cue SDK Samples and Documentation

Prerequisites
Before exploring SDK documentation, ensure that you:

- Set up your development environment as described in “Setting up your Development Platform” on page 7.
- Have a working knowledge of Java and some idea of how to develop plug-ins for an Eclipse-based product.

Samples

Samples Guide
Samples are documented in the Samples Guide, which is in Version Cue SDK under Help > Help Contents > Adobe Version Cue SDK Help. A Samples Overview page can be accessed directly from the Welcome page, which lists summaries of all the samples and enables you to access the more detailed documentation for each sample.
Setting up your Development Platform

Automatically Importing Samples from the Welcome Page

Sample projects are in the folder structure of an Adobe plug-in. Version Cue SDK samples can be imported from the Welcome page, using a wizard that automates the import of the samples. You already may have imported some or all of the samples, as the Welcome page is one of the first views you see when you open the Version Cue SDK.

On the Welcome page, click on a category to import the samples; one or more sample plug-ins should be imported into your workspace. See Figure 1, where the option to import samples that extend the Version Cue client user interface was chosen.

Manually Importing Samples

We strongly recommend you not import samples manually but instead use the wizard to import samples via the Welcome page. If you want to import the sample projects manually, follow these steps:

1. Locate compressed forms of the samples in the following folder, and uncompress them prior to import:
   
   `<vcsdk>/VersionCueSDK/plugins/com.adobe.versioncue.sdk.samples_3.0.0/samples`

2. Select File > Import > General > Existing Projects into your workspace, and import the samples you require.

3. Select the projects you want to import.

4. Select Finish. You should now see one or more projects imported into your workspace.

Although Version Cue SDK is set up to build the projects automatically (if the Project > Build Automatically menu element is checked), you may have to manually clean up and then trigger a complete rebuild to eliminate some compile errors. Ideally, there should be no compile errors or warnings.

Running Samples

The instructions for running sample plug-ins are like those described in “Running the Hello World plug-in” on page 31. The Samples Guide has instructions specific to each sample. You can browse this guide from within Version Cue SDK; see Help > Help Contents > Adobe Version Cue SDK Help.

Deploying samples

The instructions for deploying sample plug-ins are like those described in “Deploying the Hello World plug-in” on page 32.

API Reference and Other Documentation

After you follow the instructions in this guide, the API Reference is available from the context of Version Cue SDK Help. You can navigate to the API Reference via Help > Help Contents > Adobe Version Cue SDK Help > References. You also will find other useful material, such as the Extension Points Reference and the Scripting Interface Reference.
Creating your First Plug-in

This section describes how to write a plug-in that extends Version Cue Server in a very basic way. The instructions create a Hello World plug-in, which writes a message to the console whenever asset content is stored on (uploaded to) a Version Cue Server.

Prerequisites

Before starting, ensure that you:

- Set up your development environment as described in “Setting up your Development Platform” on page 7.
- Have a working knowledge of how to use an Eclipse-based SDK to develop plug-in projects in Java.

Overview

These are the high-level steps to create and run a HelloWorld plug-in for Version Cue Server:

2. “Adding an Extension to the Manifest” on page 27.

These steps create a plug-in project called com.yourco.versioncue.helloworld, which contains a package of the same name. Throughout these instructions, we use com.yourco as a placeholder for a namespace of your choice. For example, if your company controls the widgetcompany.co.uk domain, you might want to use uk.co.widgetcompany in place of com.yourco.

Creating a new Version Cue plug-in project

To create a new Version Cue plug-in project, follow these steps:

1. Select Window > Open Perspective > Other and choose Plug-in Development. See Figure 11.
2. From the IDE of Version Cue SDK, select the item New > Project > Version Cue Plug-in Development > Version Cue Plug-in Project. See Figure 12.
3. Click Next and proceed to the next screen.
4. Uncheck “Use default location.”
5. Assign a “Project name.” It is good practice to use namespaces in your Java code from the outset, and to reflect the main namespace in the project name; in this example, `com.yourco.versioncue.helloworld`.

6. Use the Browse button to create a new folder for the project, called `<vcsdk>/projects`.

7. Set Directory: `<vcsdk>/projects/com.yourco.versioncue.helloworld`

8. Click Next and proceed to the next screen. See Figure 13.

9. By default, Plug-in Version is set to 1.0.0. The Plug-in Version is independent of the Version Cue Server version, and you can choose the version number you want to use here.

10. You can set Plug-in Name: HelloWorld Plug-in

11. You can set Plug-in Provider to something appropriate. In practice, we expect you to set this to your company name.

12. Check the box “Registering Listener...” to add some plug-in dependencies.

13. Click Finish to dismiss the dialog, and accept the suggestion to open the Java perspective when you have dismissed the dialog.

14. The plug-in you have created is now functional and potentially could be loaded as-is, but we will be adding a listener for an event associated with an asset being added or a version saved to Version Cue Server in the next section.

**Figure 11** Selecting the Plug-in Creation Wizard
Getting Started with Adobe Version Cue CS3 Development

Creating your First Plug-in

**Figure 12** Creating a New Version Cue Plug-in Project

**Figure 13** Finishing the New Version Cue Plug-in Project Wizard
Adding an Extension to the Manifest

To add a listener, follow these steps:

1. In the Plug-in Manifest Editor, double-click MANIFEST.MF and open the Extensions section. Alternatively, the Overview page describing the plug-in—the first thing you see after creating the new plug-in—has a pane entitled Extensions; click on the Extensions link and you will open the Extensions section.

2. Click the Add... button you see on the Extensions section.

3. Select com.adobe.versioncue.controller.core.listener. See Figure 14.

4. Click Finish.

5. Use File > Save to save your edits. Make sure your edits were saved. If there is an asterisk to the left of the filename in the editor tab, the file was not saved.

**Figure 14** Selecting an Extension Point
Defining a Class for the Listener Extension

This section describes how to define the Java class for a listener extension. The class writes a message to the console each time asset content is stored (uploaded) to Version Cue Server; that is, whenever a new file-based asset is added to a project on the Version Cue Server or a new version of an existing asset is created.

To define the Java class for a listener extension, follow these steps:

1. In the Package Explorer, right-click the `com.yourco.versioncue.helloworld` package, and use New > Class to create a new Java class for your listener implementation. See Figure 15.

2. Under Name, enter `StoreContentListener`.

3. Under Superclass, enter `com.adobe.versioncue.controller.asset.api.events.StoreContentListenerAdapter`. Alternatively, click on the Browse button to the right of the Superclass edit box, enter a few characters like Sto and then you will see the choices for the superclass. See Help > Version Cue SDK > API Reference for documentation on the class `StoreContentListenerAdapter`.

4. Click Finish.

5. Define a `preCall` method that writes “HelloWorld” to the console. The code required is shown in Example 1.

6. Use File > Save to save your edits. Make sure your edits were saved. If there is an asterisk to the left of the filename in the editor tab, the file was not saved.
**EXAMPLE 1 StoreContentListener.java Class for HelloWorld**

```java
package com.yourco.versioncue.helloworld;
import org.eclipse.core.runtime.IProgressMonitor;
import com.adobe.versioncue.controller.asset.api.events.StoreContentListenerAdapter;
import com.adobe.versioncue.core.VersionCueException;
/**
 * Writes a message to the console when notified that asset content is being
 * stored.
 */
public class StoreContentListener extends StoreContentListenerAdapter{
    /**
     * 
     */
    public void preCall(IProgressMonitor monitor, IStoreContentEvent event) throws VersionCueException {
        System.out.println("HelloWorld...");
    }
}
```
Associating the Listener Extension to its Implementation Class

To connect the listener extension to its implementation class, follow these steps:

1. In the Plug-in Manifest Editor, double-click MANIFEST.MF to open the Extensions section.
2. Right-click on extension com.adobe.versioncue.controller.core.listener. See Figure 16.
3. Select New > assetStoreContentListener to define the controller event of interest as asset content being stored.
4. Click the Browse button to choose the implementation class.
5. Under Select, enter com.yourco.versioncue.helloworld.StoreContentListener.
6. Click OK.
7. Use File > Save to save your edits and complete the extension definition, as shown in Figure 17. Make sure your edits were saved. If there is an asterisk to the left of the filename in the editor tab, the file was not saved.

Figure 16  New > assetStoreContentListener extension
Your plugin.xml file contains the element that maps the extension to its implementation class, as shown in Example 2.

**Example 2 plugin.xml Extension Definition, on Adding assetStoreContentListener**

```xml
<plugin>
  <extension point="com.adobe.versioncue.controller.core.listener">
    <assetStoreContentListener class="com.yourco.versioncue.helloworld.StoreContentListener"/>
  </extension>
</plugin>
```

**Running the Hello World plug-in**

If you followed the procedures in the preceding sections, your project should resemble that shown in Figure 18.

**Figure 18 The Completed Project**

To build and run your plug-in under Version Cue Server while working in the IDE of Version Cue SDK, follow these steps:
1. In the IDE of Version Cue SDK, select Window > Open Perspective > Java, to switch to the Java perspective.

2. Build the project from within Version Cue SDK. It may be built already, if “Build Automatically” is enabled (its default state) on the Project menu.

3. Run Version Cue Server from within the IDE of Version Cue SDK. See “Starting Version Cue Server from the IDE of Version Cue SDK” on page 15.


5. Use a Version Cue client for CS3, such as Adobe Bridge CS3, to add a file to a project on your Version Cue Server. Alternatively, add a new asset—or version of an existing one—by running code snippets CreateProject and CreateFile from sample.snippetrunner, with default arguments. The sample.snippet plug-in can be imported to your workspace by clicking the “Get started ...” sample category on the Welcome screen.

6. In response to the operation to add a file to the project—a “store-content” operation—the plug-in writes a message to the console like that shown in Figure 19.

---

**Deploying the Hello World plug-in**

To build your plug-in as a JAR file and deploy it so Version Cue Server can run it, follow these steps:

1. Stop the installed Version Cue Server, through the System Tray icon (Windows) or Control Panel (Mac OS).

2. Select File > Export.

3. Select Plug-in development > Deployable plug-ins and fragments.

4. From the list of Available Plug-ins and Fragments, select the newly created plug-in project, com.yourco.versioncue.helloworld.

5. Use Export Destination > Directory > Browse to set the export path for the plug-in. Choose <installdir>/Server; the export itself will deploy to the plugins folder within the chosen folder.
6. Note that <installdir> represents the location where Version Cue Server is installed:

   c:\Program Files\Adobe\Adobe Version Cue CS3 SDK\ (Windows)
   /Applications/Adobe/Adobe Version Cue CS3 SDK/ (Mac OS)

7. Click Finish to complete the export.

8. Verify that the <installdir>/Server/plugins folder to which you exported the plug-in contains the following file:

   com.yourco.versioncue.helloworld_1.0.0.jar

9. Start the installed Version Cue Server, through the System Tray icon (Windows) or Control Panel (Mac OS). Note that you must have already stopped the Version Cue Server from the IDE of Version Cue SDK, if it was already started from the IDE.

10. Use Adobe Bridge CS3 or another Creative Suite CS3 application to add a file to a project on your Version Cue Server. In response to this operation, the plug-in will write a message to the log-file of Version Cue Server.

   **NOTE:** If your plug-in implements a Web application extension and has Java Server Pages (JSP), the plugin should not be exported as a JAR file, because the JSP files cannot be compiled in this stage. If you have JSP files, verify that Export Options > Package plugins as individual JAR archives is unchecked, and export your plug-in to a folder.

---

**Troubleshooting**

Table 4 lists issues that may be encountered in developing Version Cue SDK sample plug-ins, and it suggests possible resolutions.

### Table 4 Troubleshooting Version Cue SDK Sample Plug-ins

<table>
<thead>
<tr>
<th>Issue</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version Cue Server fails to start under control of the IDE, with this error: IllegalStateException:Unable to start database....</td>
<td>An instance of the MySQL database service may be running already. Stop MySQL; see &quot;Stopping the MySQL Database Service&quot; on page 13. Note that in more recent versions of the SDK, this message has been replaced.</td>
</tr>
<tr>
<td>Version Cue Server fails to start under control of the IDE, with this error: FATAL com.adobe.versioncue.core - An instance of VC Server is already running using this workspace. Failed to start Version Cue CS3 Server.... Version Cue CS3 shut down.</td>
<td>This is the error message that is shown when you try to start a Version Cue Server using the same workspace/database as the one that is already running. You should shut down the existing Server- for instance, if you started the running one under control of the IDE, then use the button to Shut Down Version Cue Server. Alternatively, if the Server was started through the Control Panel (Windows) or System Preferences (Mac), then stop the Server through the same mechanism.</td>
</tr>
</tbody>
</table>
### Frequently Asked Questions

#### Why is Version Cue Server the Target Platform for Plug-in Development?

Version Cue Server is based on Eclipse’s universal plug-in architecture, PDE. Although Version Cue Server is not an IDE in the traditional sense, it is an Eclipse environment and can be extended using the Eclipse plug-in architecture. For more information, see *Adobe Version Cue CS3 Server Programmer’s Guide* (programmers-guide.pdf).

#### Why is there a full launch/debug configuration category for Version Cue Server?

In most cases, one configuration suffices; however, many developers prefer to have multiple configurations, supporting different trace settings and so on.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version Cue Server fails to start, with this error:</td>
<td>A Java run-time environment (JRE) matching the Version Cue Server needs to be configured. Set a JRE under Java &gt; Installed JREs; see &quot;Java Installed JREs Preferences&quot; on page 11.</td>
</tr>
<tr>
<td>org.osgi.framework.BundleException</td>
<td></td>
</tr>
<tr>
<td>When you try and start Version Cue Server as described in &quot;Starting Version Cue Server from the IDE of Version Cue SDK&quot; on page 15, an instance of Eclipse workbench starts instead of Version Cue Server</td>
<td>Most likely, the target platform preference was not set correctly when you configured to launch Version Cue Server. Switch to the Java perspective. Use Run &gt; Run to open the configuration dialog. In the left-hand pane, choose the configuration named “Run Version Cue Server.” If the configuration exists, click the Delete button to delete it. Click Close to dismiss the dialog. Follow the procedures in “Plug-in Development Target Platform Preferences” on page 10 followed by “Preparing to Start Version Cue Server” on page 14, and you should be able to start Version Cue Server from the Version Cue SDK IDE.</td>
</tr>
<tr>
<td>Version Cue Server fails to start, with this error: &quot;java.net.BindException: Address already in use: JVM_Bind:3703”</td>
<td>An instance of Version Cue Server already is running; for instance, is Version Cue CS2 Server running? Alternatively, you may have an instance of Version Cue CS3 Server already running- perhaps started from the Control Panel on Windows or System Preferences on Mac OS.</td>
</tr>
<tr>
<td>Version Cue Server fails to start under control of the IDE, with warning about logs/VersionCue.log</td>
<td>This can happen if you installed Version Cue SDK before the Version Cue Server was installed, and you already started Version Cue SDK without the Server being installed. Later, on installing the Server, the workspace that you were using initially is corrupt and you must either delete the workspace folder or delete all its contents.</td>
</tr>
</tbody>
</table>
Where do I find the Error Log for Version Cue SDK itself?

You can find a file named VersionCue.log, which should be created in the Eclipse workspace you have chosen; for instance, <vcsdk>/workspace/vc-data/logs/VersionCue.log. You will find this contains information above and beyond what is shown in the Console window.

How can I have my plug-in contribute entries to the Version Cue Server log file?

Version Cue Server uses the logging framework log4j, and your plug-in can contribute entries to the log file, if you have a file called log4j.properties in the server configuration folder with the appropriate key-value pairs to enable logging for the packages in your plug-in. You will find that there is a sample named sample.customproject in this SDK that has a custom log4j.properties, to enable logging of warnings and errors in that sample to the Version Cue Server log.

If you are running Version Cue Server under control of the IDE of the Version Cue SDK, you can enable logging of information from your own plug-in to the Version Cue Server logfile, if you copy your own custom log4j.properties file to the folder <your-workspace>/vc-data/config/ and restart Version Cue Server.

When Version Cue Server is started through the Control Panel (Windows) or System Preferences (Mac), you must copy your custom log4j.properties file to the folder <Version Cue/Server>/config, where <Version Cue/Server> represents the path to your installed Version Cue Server. This will enable logging of information for the sample.customproject directly into the Version Cue Server log file.

Any key defined in your log4j.properties file will be appended to the Version Cue logger configuration. If a property has already been defined, then the existing value will be overridden by the value supplied in your log4j.properties file. Below is an example of adding a key-value pair to a custom log4j.properties file.

# Print only messages of level WARN or above in the package sample.customproject.
log4j.logger.sample.customproject=WARN, LOGFILE

Note that overriding the default Version Cue logger settings, such as the LOGFILE.layout, may affect the log being displayed in the Administration user interface of Version Cue Server. For more information on log4j, such as what keys and values you can have in your log4j.properties file, see the log4j documentation, available at http://logging.apache.org/log4j/docs/index.html