

Simplifying Enterprise Installation of IBM Rational ClearCase

Using the IBM Rational ClearCase Remote Client plug-in and the Eclipse SDK

Fred Bickford IV

Senior Advisory Software Engineer
ClearCase Customer Advocacy Group
IBM Rational Customer Support
fabickfo@us.ibm.com

September 19, 2007

INTRODUCTION 3

CURRENT CLIENTS AND INSTALLATION METHODS 4

 BACKGROUND 4

- CLASSIC CLEARCASE CLIENT 4
- BROWSER BASED CLIENT 4
- REMOTE CLIENT 4

 ClearCase Remote Client 5

 ClearCase Remote Client for Eclipse 5

 ClearCase Remote Client for Visual Studio 6

ECLIPSE SDK AND CCRC PLUG-IN INSTALLATION 7

 OVERVIEW 7

 ASSUMPTIONS 7

 INSTALL AND CONFIGURE ECLIPSE 7

 INSTALL CCRC PLUG-IN INTO ECLIPSE 9

 INSTALL AND CONFIGURE CCRC AS A PLUG-IN TO ECLIPSE 10

SETTING UP TOOL DEFAULTS FOR ALL USERS 15

 SETTING THE DEFAULT PERSPECTIVE TO CLEARCASE 16

 DISABLING CVS CAPABILITIES 17

 CHANGING THE WEB BROWSER 18

 HAVING CLEARCASE IGNORE RESOURCES 19

 ENABLING A DEFAULT INITIAL APPEARANCE 21

 SPECIFYING A CUSTOM SPLASH SCREEN 22

 CONFIGURING AN ALTERNATE "DEBUG" CCRC EXECUTABLE 24

 CREATING DESKTOP ICON[S] 26

OPTIONAL CONFIGURATION STEPS 27

 MULTIPLE CLEARCASE VERSION SUPPORT 27

AUTOMATING THE INSTALLATION AND DISABLING OF CCRC IN ECLIPSE 30

 DISPLAYING WHAT FEATURES ARE CURRENTLY INSTALLED IN A SPECIFIC ECLIPSE INSTANCE 30

 DISPLAYING WHAT FEATURES ARE AVAILABLE ON A SPECIFIC URL TO INSTALL 31

 DISABLING CLEARCASE REMOTE CLIENT FOR ECLIPSE 32

 ENABLING CLEARCASE REMOTE CLIENT FOR ECLIPSE 32

 INSTALLING CLEARCASE REMOTE CLIENT FOR ECLIPSE 32

PACKAGE AND DEPLOYMENT 33

CONCLUSION 34

Introduction

This paper discusses the benefits and ease of deployment an administrator or project lead can realize by deploying the IBM® Rational® ClearCase® Remote Client (CCRC) across an organization in a customized, standardized and user friendly manner.

Discussed will be many customizations and options that are not mandated, but that will serve as an example for users who want to deploy IBM Rational ClearCase in this fashion.

This paper does not replace the standard practices to install IBM Rational ClearCase. Instead it provides an alternative approach that allows for faster deployment of a common, customized client that benefits both the end user and the administrative staff of any organization.

Current Clients and Installation Methods

Background

To better understand all the choices for ClearCase installation, some background on the various clients and typical needs are important. Using CCRC is not an exclusive client decision. CCRC can be used in conjunction or in parallel with any other existing ClearCase client.

For a comparison of the features of all the ClearCase clients, review [technote 1233090](#).

- **Classic ClearCase Client**

Sometimes referred to as the Full ClearCase Client (FCC), this client installation can be performed from media such as a DVD, but in most enterprise applications, FCC requires a network connection to a software release area. An installation of a FCC is really not just a client, it sometimes requires services and complex processes that can rival other robust server applications.

- **Browser Based Client** (*also referred to as CCWeb*)

Using ClearCase through a web browser is simple and has a near zero footprint, but the functionality of this client is very sparse and limited to only the most basic ClearCase operations.

- **Remote Client**

The CCRC was developed to serve as a client that can access ClearCase servers in most network configurations, whether it be on a high speed LAN or a potentially latent WAN. Regardless of the connection type, CCRC still requires a predictable connection.

There are three choices for users who want to use the CCRC in their organization:

1. ClearCase Remote Client
2. ClearCase Remote Client for Eclipse
3. ClearCase Remote Client for Visual Studio.

For the remainder of this document, we will focus on using the CCRC plug-in for the Eclipse SDK, but it is important to be aware of all CCRC versions to make the best choice in any situation.

ClearCase Remote Client (also referred to the “standalone client”)

Meant to serve as an installable self contained client application, this client uses the Rich Client Platform (RCP technology). RCP is used by vendors to leverage Eclipse technologies and serve as a standalone application.

More details regarding RCP can be found at http://wiki.eclipse.org/Rich_Client_Platform.

The benefits of this client is that it is self contained, and once the archive file is downloaded locally and unpacked, the installation will handle installation of RCP, this client includes its own JRE as well.

The standalone version (at the time of this writing ClearCase 7.0.1 is the current release) does not support third party plug-in additions or customizations.

This client is presently available only on Linux® and Microsoft® Windows® platforms.

Review [technote 1224586](#) for a complete list of supported platforms and eclipse versions.

The standalone version of CCRC can be installed only once (as like any typical application), and must be uninstalled at each major version ** (for example ClearCase 2003.06, ClearCase 7.0 and ClearCase 7.0.1) to remove it.

** *Each RCP standalone client is based on different versions of Eclipse.*

If you do not have plans to customize or need integrations with other Eclipse plug-ins in the same interface, this client method is preferred.

The stand-alone client’s footprint is approximately 167 MB after installation.

ClearCase Remote Client for Eclipse (also referred to as “plug-in”)

Throughout the remainder of this document, the focus will be on the usage of CCRC in conjunction with a standard Eclipse SDK, benefits of using the CCRC plugged into a standard Eclipse SDK are Easier ClearCase integration to customers already using Eclipse SDK (or other Eclipse IDE’s)

- Easier ClearCase integration for customers already using the Eclipse SDK (or other Eclipse IDEs)
- Faster installation mechanism (Eclipse Base & JRE is already in place)
- Ability to leverage some existing Eclipse niceties**
- Support for Windows, Linux/GTK, Sun Solaris/GTK, HP-UX/Motif, and IBM-AIX/Motif

** See http://wiki.eclipse.org/The_Official_Eclipse_FAQs for some common questions about Eclipse

ClearCase Remote Client for Visual Studio 2005 (also referred to as "ccrc.net")

Each CCRC server has an archive

http://<ccrc_server_name>/ccrc/update/ccrcvsi_win32.zip available, that clients of Visual Studio 2005 can use to install CCRC to integrate directly into the IDE. This method provides the same web view access for remote ClearCase users who are developing in .NET.

Eclipse SDK and CCRC Plug-in Installation

Overview

In this section we will provide steps for installing the Eclipse SDK and will provide instructions to install the ClearCase Remote Client plug-in. For this paper we will use Eclipse 3.2.1, which is the latest currently supported SDK for CCRC. Eclipse has taken steps in version 3.3 (Europa) to include some of the additional components that we will install, and will make this deployment approach even simpler.

Assumptions

To be able to take advantage of this method of deployment, there are a few things which are assumed:

- Each platform that is planned for support should have its own deployment archive
- All clients will have a common, supported version of Java enabled on the host
- Users need to have a basic understanding of Eclipse
- Deployment requires the ability to decompress an archived file

Install and Configure Eclipse

1. Download Eclipse 3.2.1 for your platform from:
<http://download.eclipse.org/eclipse/downloads/drops/R-3.2.1-200609210945/index.php>
2. Download GEF 3.2.1 from:
<http://www.eclipse.org/downloads/download.php?file=/tools/gef/downloads/drops/R-3.2.1-200609211617/GEF-runtime-3.2.1.zip>
3. Extract the files to some common path like c:\eclipse or /eclipse.
Note: Remember the archive will have an eclipse directory, so that should be taken into account when extracting.

After extraction of Eclipse SDK you should have something like this

```
C:\eclipse
  \plugins
  \features
  \configuration
  \readme
```

CCRC 7.0 and later requires an additional component Eclipse GEF (<http://www.eclipse.org/gef/>) and specifically the draw2d plug-in to support the version tree browser.

Note: There are steps to install GEF formally through the use of the Eclipse update mechanism. In this document we will merely extract the plug-in directly into the SDK file space we have created.

4. Extract the GEF runtime and drw2d plug-ins into the eclipse directory tree:

```
org.eclipse.gef_3.2.1.v20060921.jar
org.eclipse.draw2d_3.2.1.v20060921.jar
```

Validate the install of GEF after restarting Eclipse. Go to *Help > Software Updates > Manage Configuration* (figure 1)



Figure1

Install CCRC Plug-In into Eclipse

1. Start Eclipse (c:\eclipse\eclipse.exe)
2. Store the workspace** file created in the ./ccrc_workspace/default directory, this will create the folder structure in the same directory as the Eclipse SDK, and also will be a way of standardizing the workspaces for all users of this deployment archive.
(figure 2)

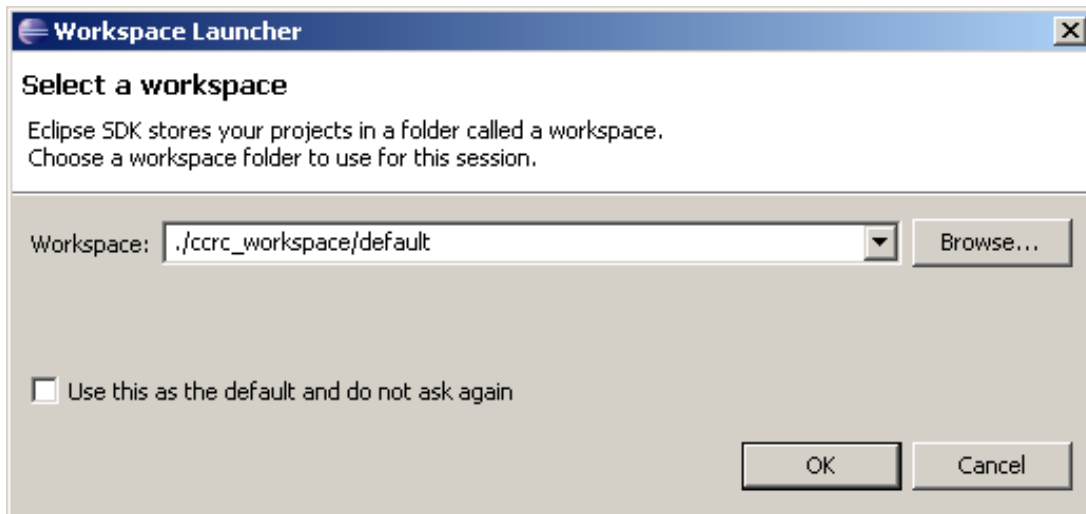


Figure2

This entry and the choices a user has for workspace manipulation is controlled by

{ECLIPSE INSTANCE}\configuration\.settings\org.eclipse.ui.ide.prefs.

This config file is worth noting as there are two (2) directives that can be modified if you choose to do so:

SHOW_WORKSPACE_SELECTION_DIALOG=

Set to "true" by default, this controls whether users see the workspace launcher (see above) to create Eclipse workspaces, if you wish to take the option to create new workspaces from user, set this to "false", else the default is recommended.

RECENT_WORKSPACES=

This is a comma separated path list of the workspaces that have been created by the user.

**** All workspaces for Eclipse must be on local disk**

Install and Configure CCRC as a Plug-in to Eclipse

In the next few steps we will walk through installing ClearCase Remote Client as a plug-in into the Eclipse SDK instance created previously.

To install a plug-in, go to *Help > Software Updates > Find and Install...* (figure 3)



Figure3

Since this is a new feature we will be installing, select "search for new features to install" (figure 4)

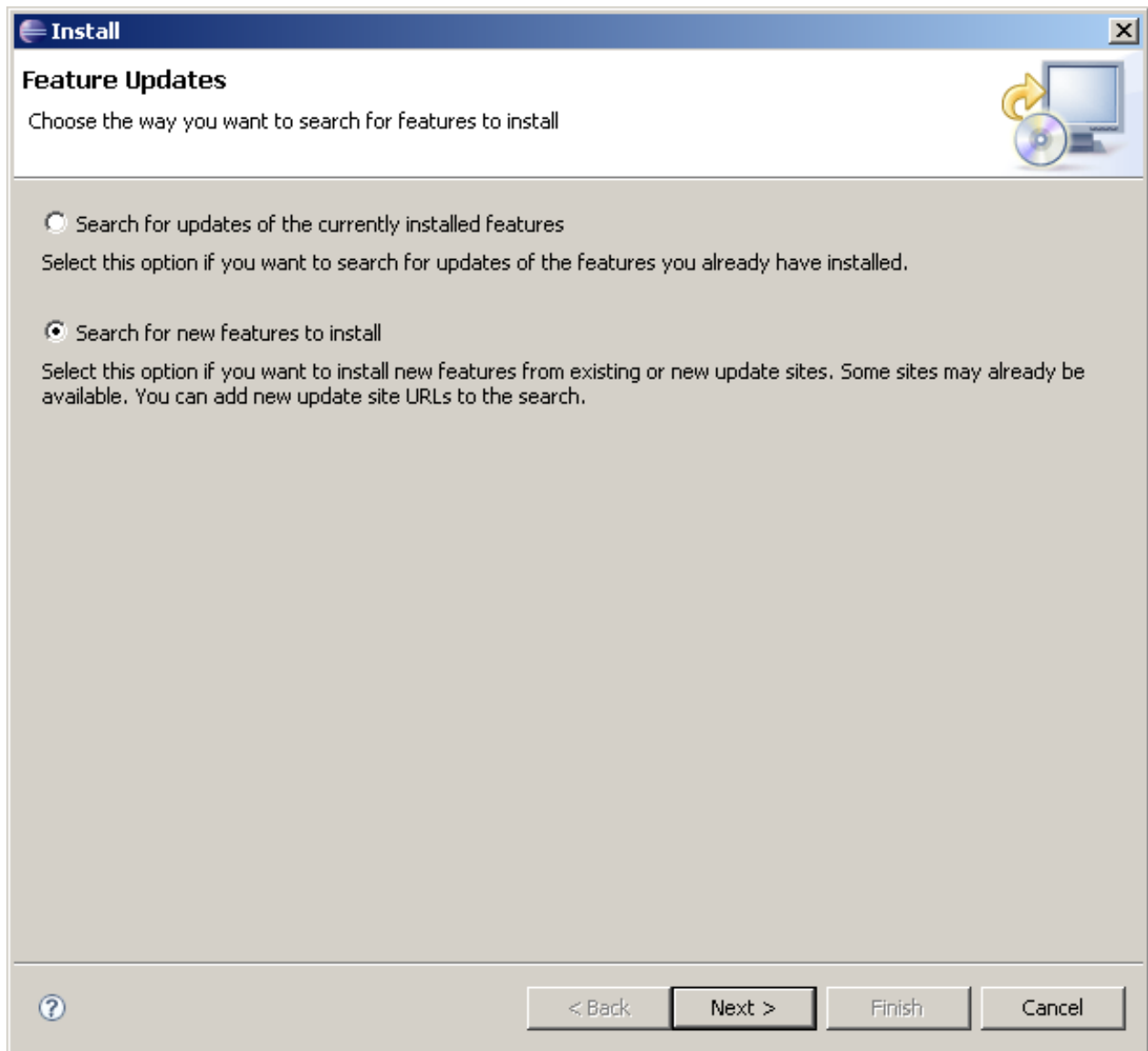


Figure4

For simplicity, we will be installing CCRC by pointing directly to ibm.com

Click on "New Remote Site..." (figure 5)

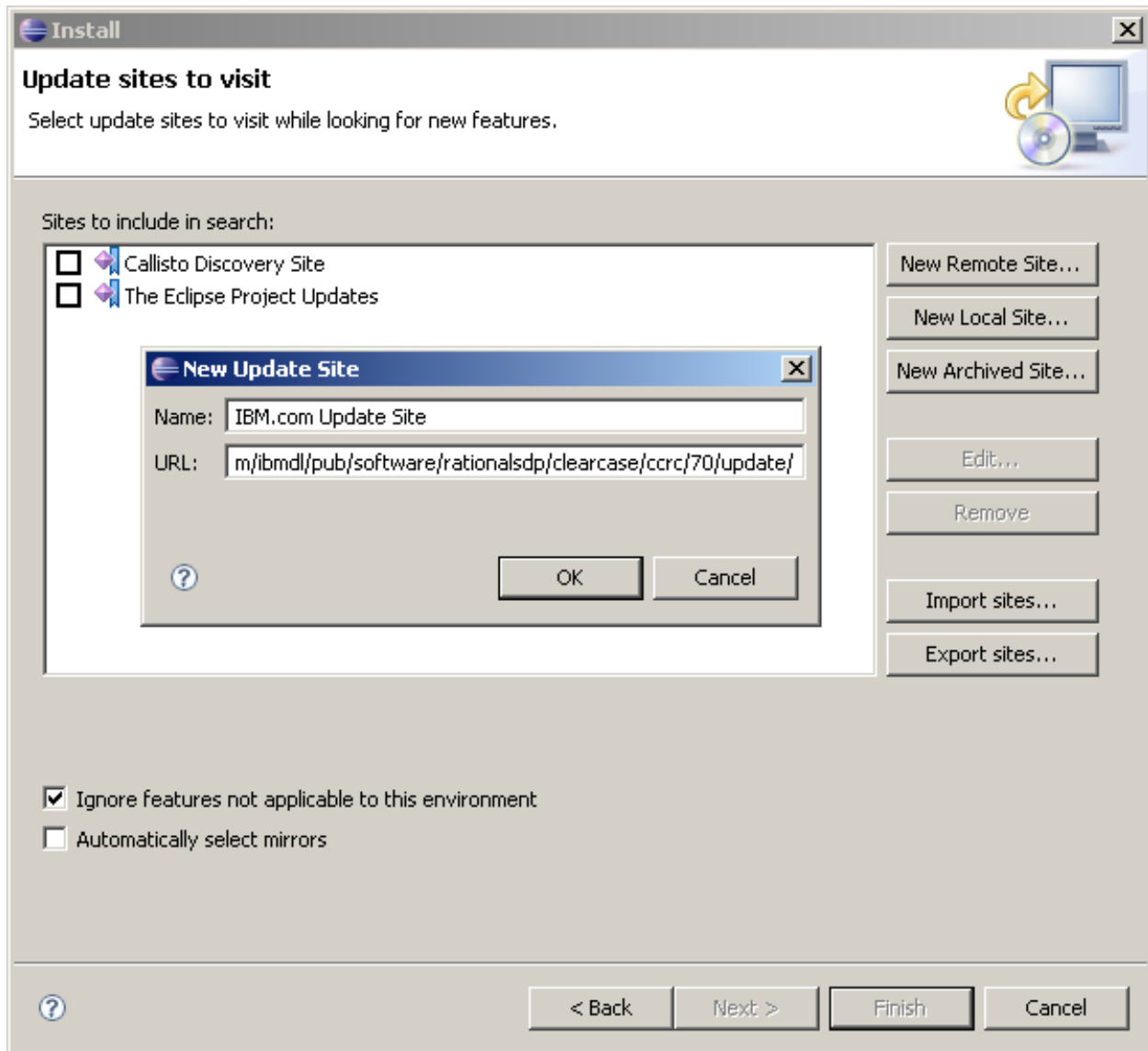


Figure5

For this we will use the following URL as our update path:

<http://www3.software.ibm.com/ibmdl/pub/software/rationalsdp/clearcase/ccrc/70/update/>

This is the version 7.0 product with the latest available Client Update CU003.

You can also install the plug-in by changing the update path to point to a CCRC server set up by your organization has already set up a CCRC server (for example using *http://</servername>/ccrc/update*)

By default CCRC will look to ibm.com in the future to “update” itself. You can change this behavior by creating a site policy. For more information, review [technote 1196280](#).

If your company does not provide direct Internet access or if you wish to enforce that all clients remain at a specific client version, you must use a site policy.

Click “OK” and then Click “Finish”

When the following dialog below is displayed, select, “Rational ClearCase Remote Client for Eclipse 7.0.1.CU003.” (figure 6)

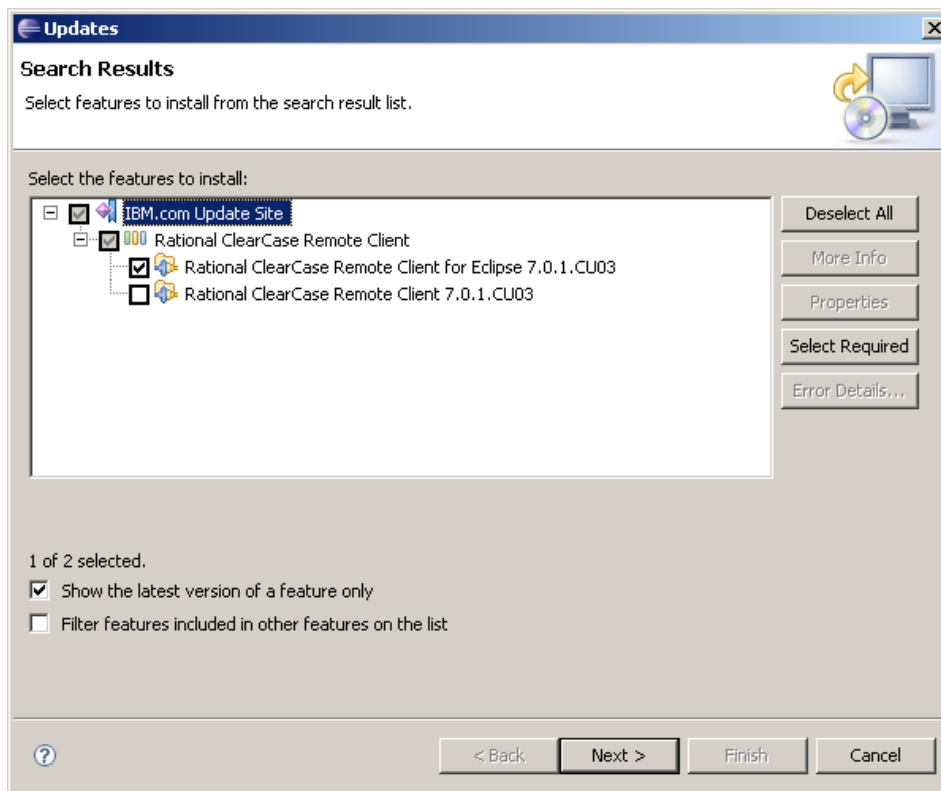


Figure6

1. Click "Next", and then accept the license terms and remaining defaults

After the installation completes, you will be prompted to restart

2. Click "Yes". it is always recommended to restart the workbench after installing a new plug-in. (figure 7)

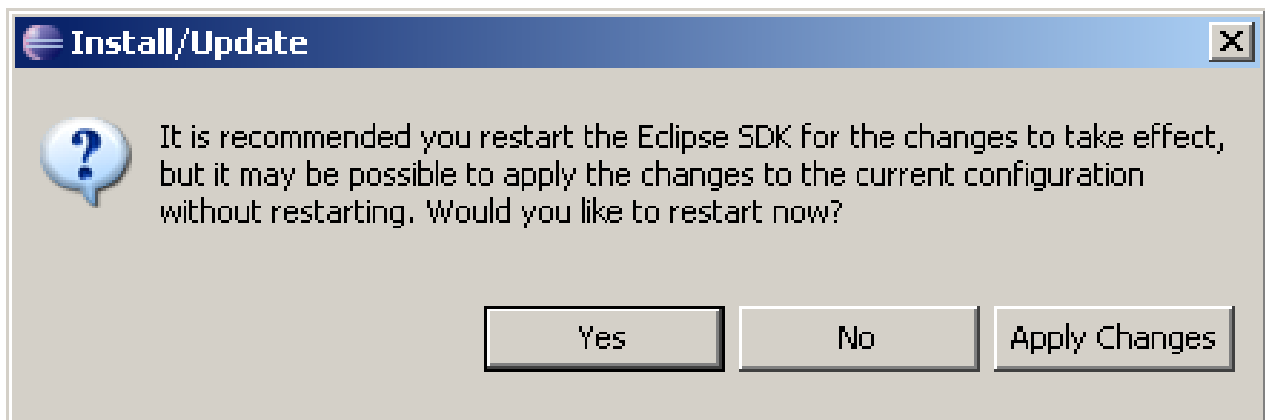


Figure7

Once Eclipse restarts, select the previous workspace location.

Setting up Tool Defaults for All Users

Now that you have a working ClearCase Remote Client instance, we will make some changes to the environment to further assist with standard roll-out of this client.

The suggested changes are not mandatory and can be modified to better fit your organization.

Eclipse preferences are the main way to customize or change the default behavior in Eclipse.

There are various Eclipse preferences that affect the appearance and capabilities for the user. Plug-in providers can have their own preference settings and CCRC provides some key preferences that help you standardize your CCRC deployment.

If you have different needs inside your organization, preferences and settings can be exported using .epf files (*File > Export > General > Preferences*) and then imported to a default or another (override) instance of Eclipse. This allows preferences to be specific down to a team or role.

Setting the default perspective to ClearCase

An Eclipse perspective is a default layout of how Eclipse should present the capabilities of a specific plug-in. Each perspective contains a series of “views” (not to be confused with “ClearCase views”) which are windows or widgets that enable areas of functionality.

Views can be customized, and any perspectives view, can be added or combined with another. For example any view from the ClearCase perspective can be added to the Java perspective.

For this paper we will assume that ClearCase work will be the main focus of the targeted users.

Eclipse will always restart the last enabled or used perspective, but we will change the default perspective to ClearCase.

1. Click *Window > Preferences to access view preferences.*

To make ClearCase the default perspective, expand “General”, and select “Perspectives”. Under “Available Perspectives”, highlight ClearCase and click on the “Make Default” button to the right. (figure 8)

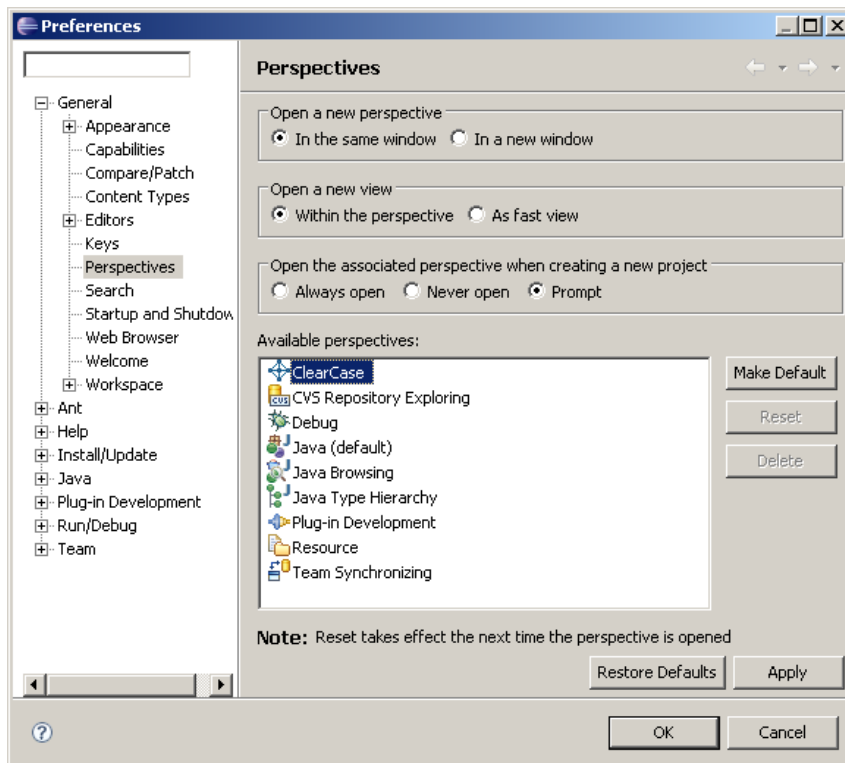


Figure8

Disabling CVS Capabilities

In Eclipse, CVS has a very pervasive existence in the menus when doing team development. To simplify the usage of CCRC, we will remove all CVS capabilities.

Under *General > Capabilities*, click "Advanced" and then select "Team". Clear the "CVS support" option and then click "OK." (figure 9)

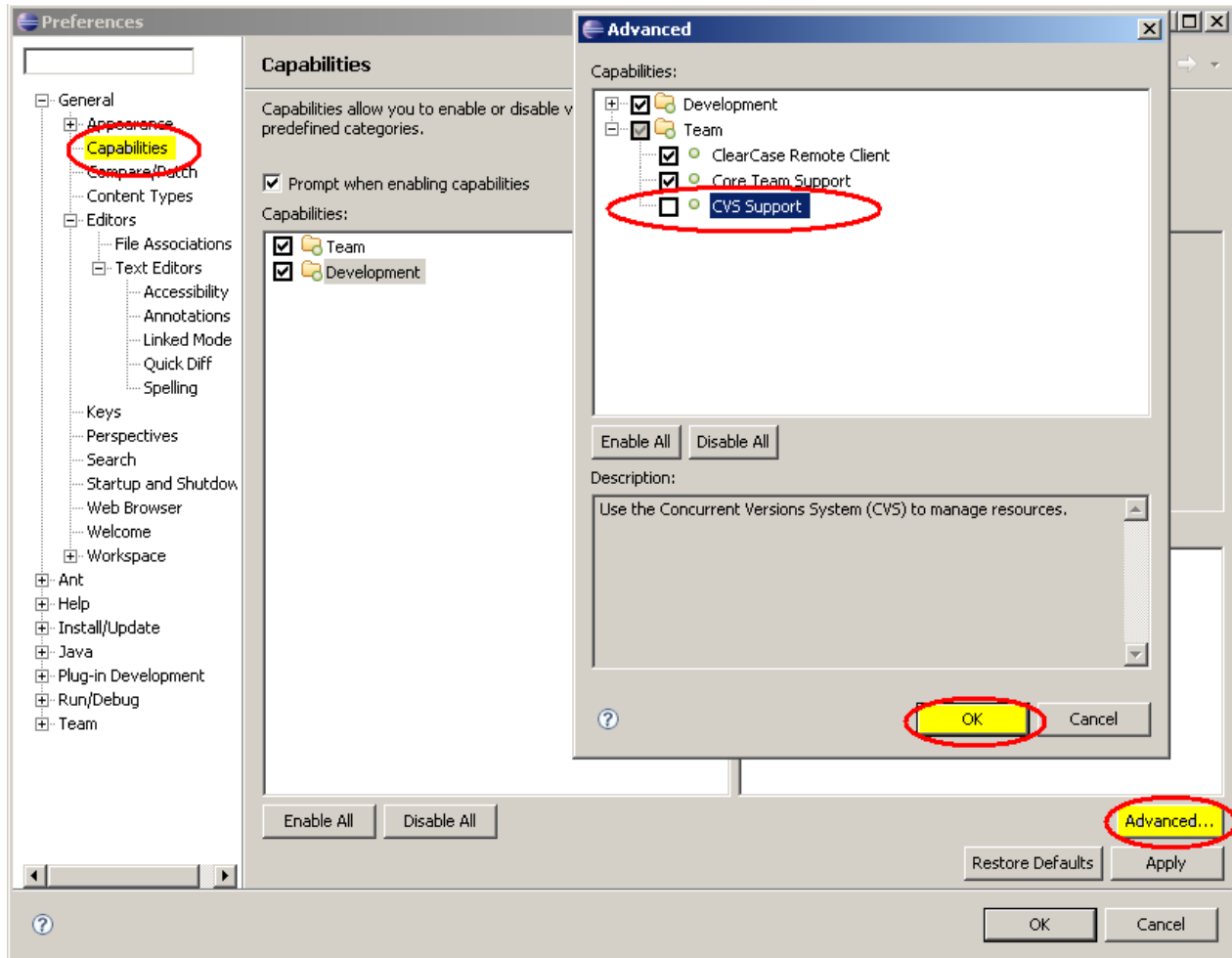


Figure9

Changing the Web Browser

If you require a specific web browser for users to view web pages inside Eclipse, you can change the default (internal browser widget) to a specific browser of your choice.

Under *General > Web Browser*, click "Use external Web browser" and select the browser you want to use (in this example we are switching to Firefox).

Click "Apply" and then "OK". (figure 10)

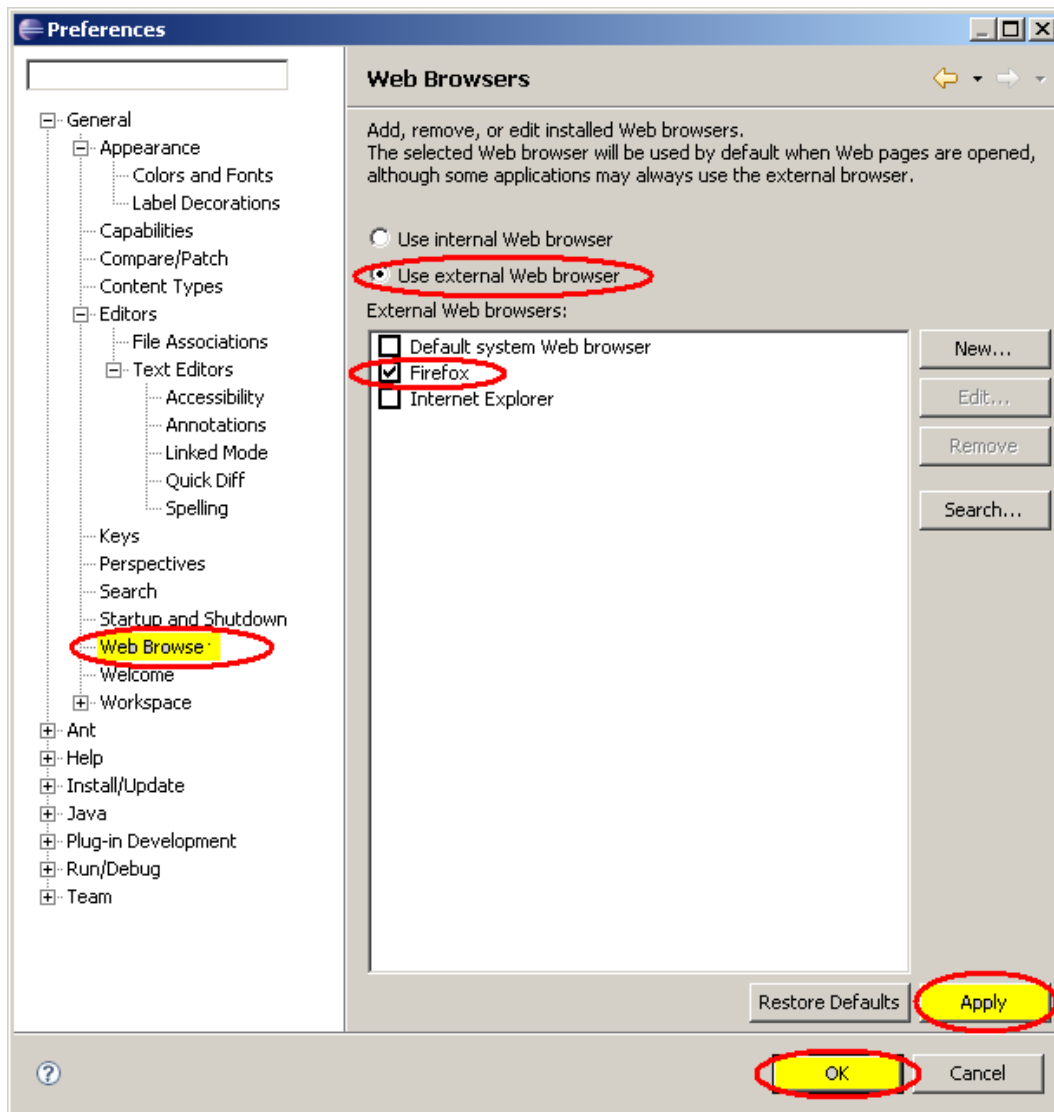


Figure10

Having ClearCase Ignore Resources

If your company develops files that you do not want to be added to source control, you can set up defaults for Eclipse and CCRC to exclude these files when users are doing operations.

Adding a pattern (for example *.foo) will cause CCRC to exclude these file extensions when looking for files to add to source control.

Under *Team > ClearCase Remote Client* there are various ClearCase Specific preferences that you can standardize. Also, if certain users want to change these for certain use cases, they can do so. (figure 11)

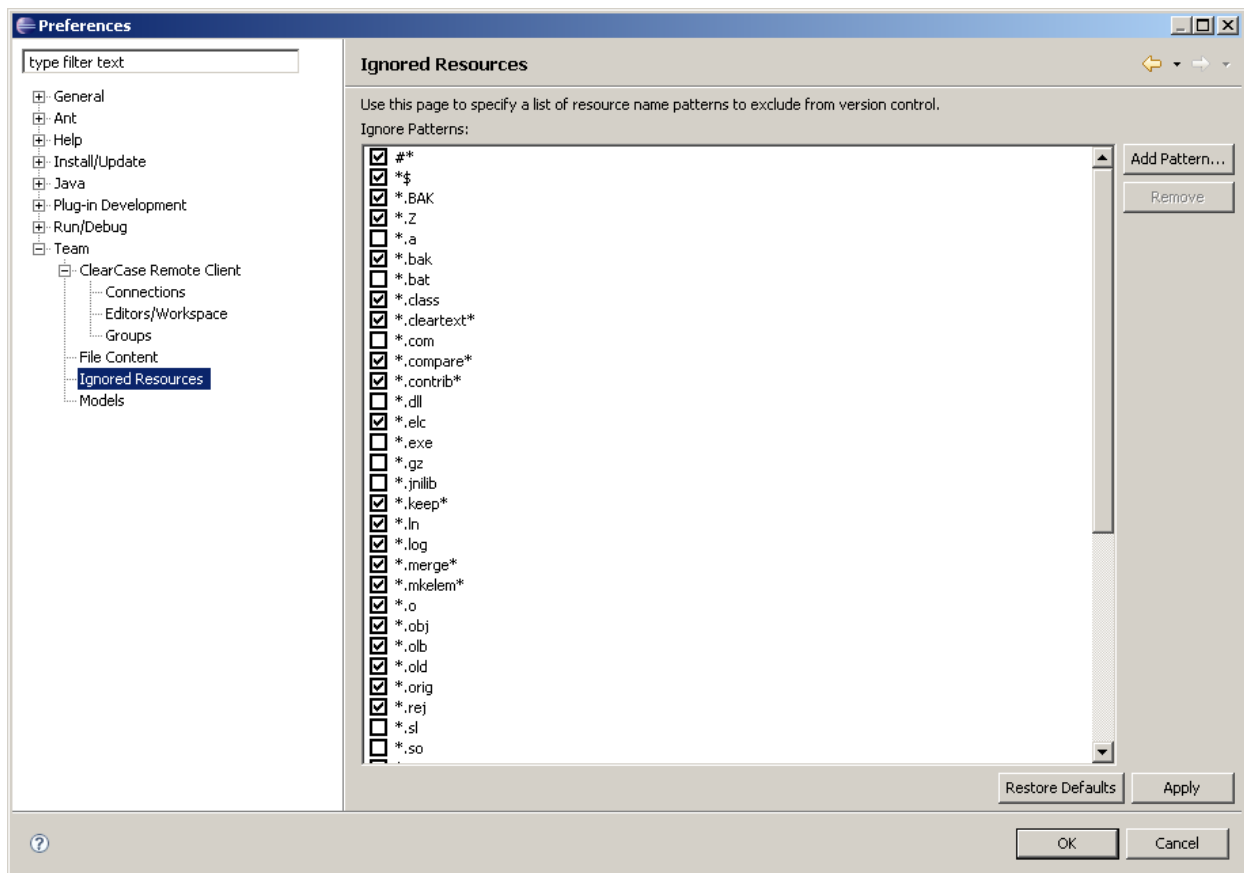


Figure11

Below are a few common preferences that may or may not apply to your usage of ClearCase. (figure 12 & 13)

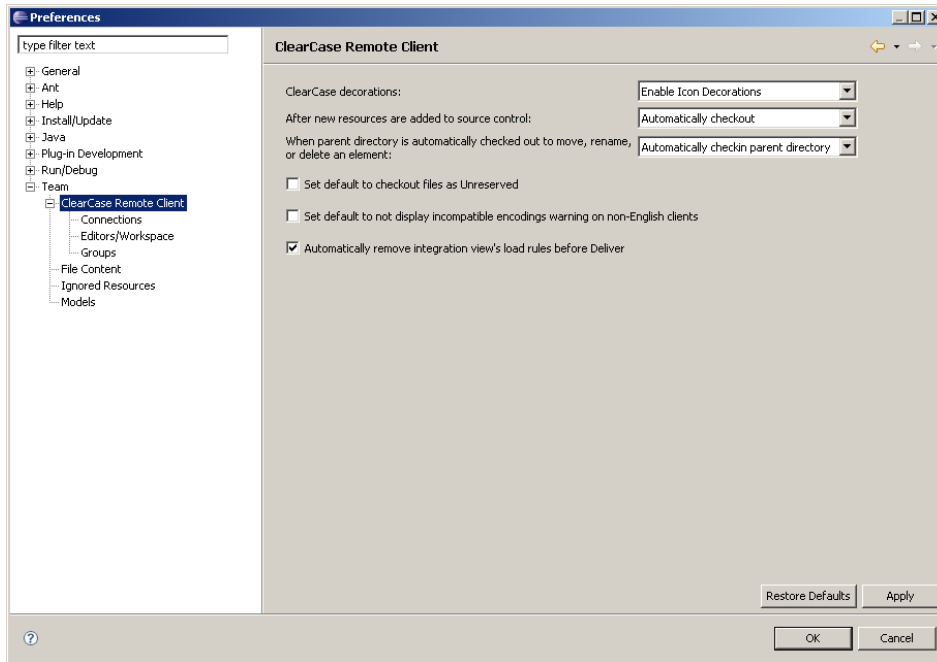


Figure12

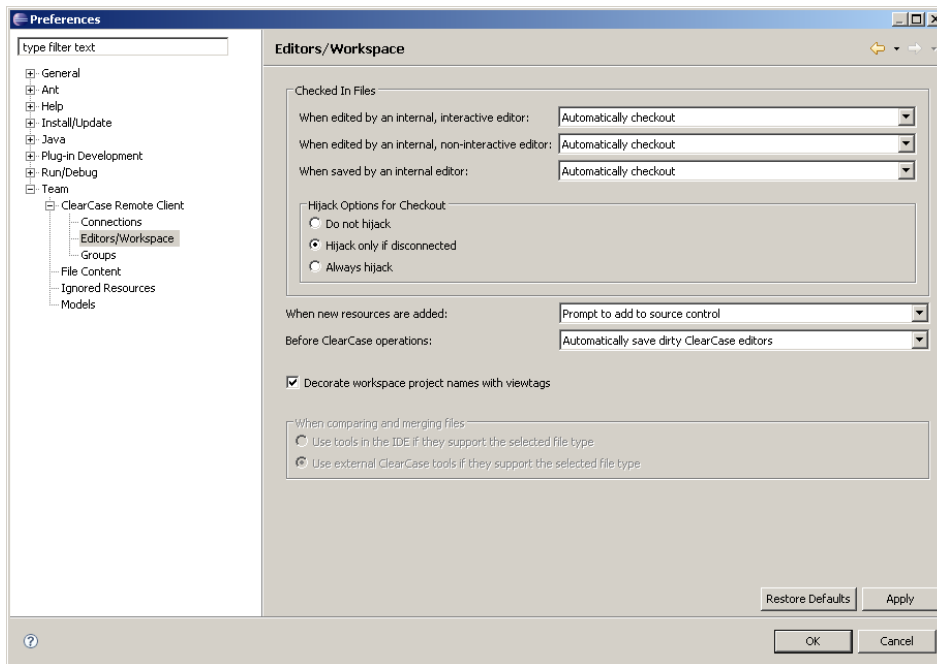


Figure13

Enabling a Default Initial Appearance

The power of Eclipse is the ability to customize the appearance of the UI, yet in our example we want all users to see a unified UI when they first start up CCRC.

Below is one example of how a default appearance could benefit end users. Remember that Eclipse and CCRC will start up with the same perspective and views enabled as last used. (figure 14)

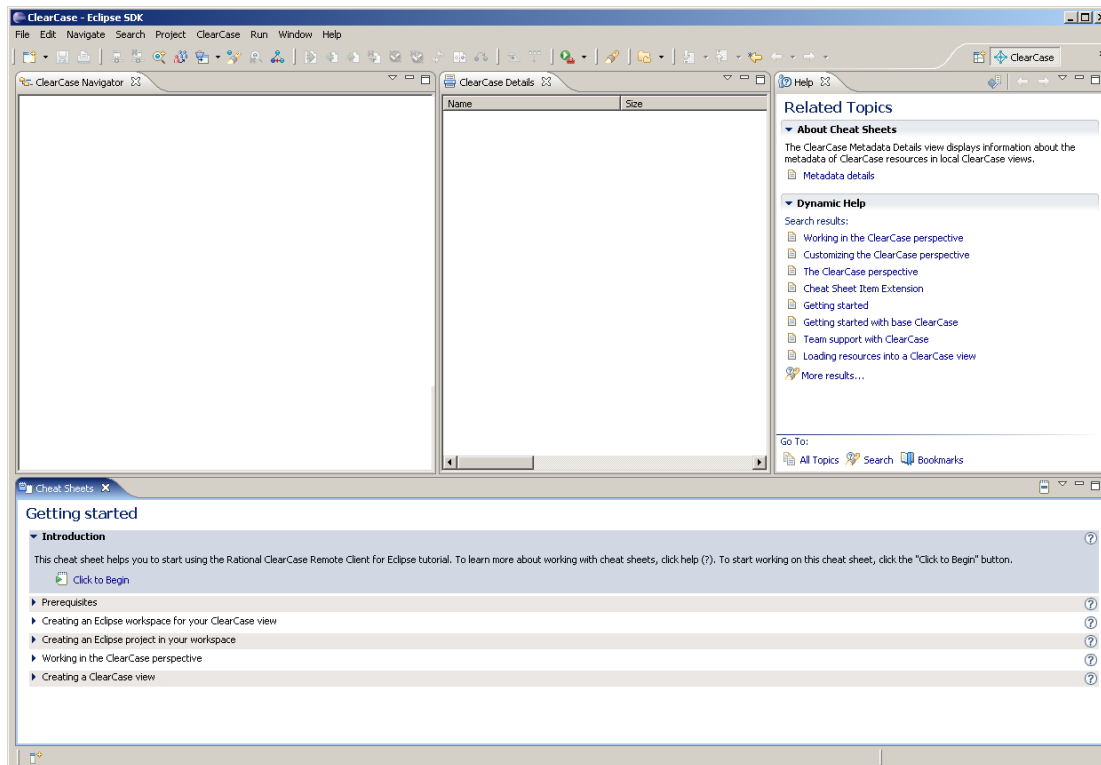


Figure14

In this example we have added the Help view and Cheat Sheet view to the ClearCase Perspective. The ClearCase views in this example are blank (we haven't connected to a server or created any views).

To add any view, go to *Window > Show View* and select the desired view. Think of this like a palette; Eclipse allows you to add what you think your end users might need. These views can be moved around inside the perspective. For example if you want the Help view on the left-hand side of the window, hold down the left mouse button on any tab (where the tabs text is) and drag and drop the view to the desired location.

You can always wipe the palette clean and start over with *Window > Reset Perspective* (this will return it to the factory default layout).

Specifying a Custom Splash Screen

When starting up the Eclipse SDK, there is a default splash screen displayed:

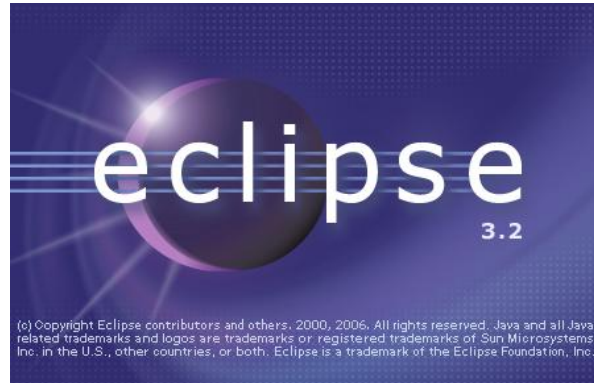


Figure15

If you want to change this from the default Eclipse bitmap to a custom bitmap of your own, you can add an image called splash.bmp to the root directory of the Eclipse instance (for example, c:\eclipse). Then, un-comment and modify the config.ini file in the configuration directory (c:\eclipse\configuration) as indicated below:

```
# The location of the splash screen file. If this value is set at
# system startup it is used in favour of the osgi.splashPath
# searching outlined above. If the value is not set the searching is # done and this key is
bound to the result of the search.
```

```
osgi.splashLocation = C:\\eclipse\\splash.bmp
```

In this example when Eclipse starts up, the following splash will be displayed: (figure 16)

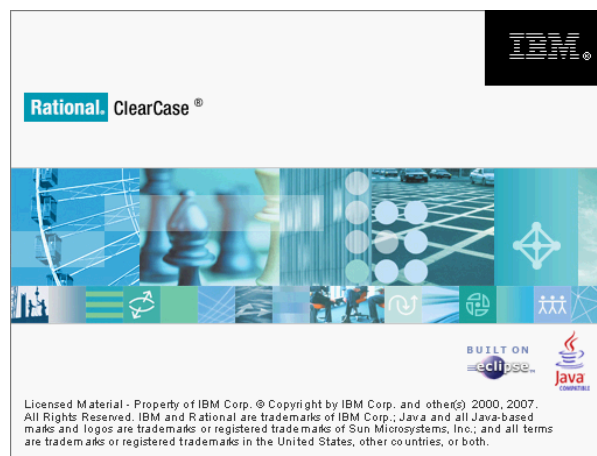


Figure16

Tip: *if you have one instance of the standalone version of CCRC installed, there are different language versions of the above image located in the following path (relative to where you might have installed CCRC)*

`\CCRC\plugins\com.ibm.rational.clearcase.standalone_#.#.#\nl`

Configuring an alternate "debug" CCRC executable

Another advantage to this deployment mechanism, is the fact that along with all the customizations we've made, we can easily pre-configure the Eclipse SDK for debug tracing should the need arise in working with IBM technical support.

Typically all errors for this client would be located at:

```
c:\eclipse\ccrc_workspace\new\.metadata\.log
```

When in debug mode this output is sent to:

```
c:\ccrc_debug.log
```

The steps to trace CCRC operations are detailed in [technote 1234118](#).

We will provide steps that are specific to our example above.

1.) Create a file called .options(notice the file begins with a ".") in c:\eclipse

2.) Copy the following text in the .options file (*lines may wrap*):

```
# ----- CCRC Tracing Options File -----
# All line preceded with a # sign are not evaluated.

# -----Server Tracing-----
# subsystems available:
# CCRC - Core general tracing related to CCRC server operations
# CCWEB- CCWeb(CGI operations)
# SUM - UCM specific

com.ibm.rational.clearcase/server_trace=true
com.ibm.rational.clearcase/server_trace/subsys=CCRC:CCWEB:SUM
com.ibm.rational.clearcase/server_trace/level=4

# -----Client Only Tracing-----
# Parameters available:
# HTTP_CLIENT - Tracing specific to HTTP communications
# CTRC_UI - Tracing specific to the user interface
# CTRC_CORE - For client core tracing

com.ibm.rational.clearcase/client_trace=true
com.ibm.rational.clearcase/client_trace/params=HTTP_CLIENT:4 CTRC_UI:3 CTRC_CORE:3

com.ibm.rational.clearcase/client_trace/output=file
com.ibm.rational.clearcase/client_trace/file_name=./ccrc_debug.log
```


3.) Create a shortcut to eclipse.exe in c:\eclipse, name the shortcut ccrc_debug

4.) Change the target of the shortcut to be C:\eclipse\eclipse.exe -debug

Test the "debug" version of your client and execute the new shortcut. Now you should see a command window and a new file c:\ccrc_debug.log appear.

Close Eclipse at this time.

Creating Desktop Icon[s]

If you want to create desktop icons for your users, you can create a windows shortcut on your desktop. The target of the shortcut will be `c:\eclipse\eclipse.exe`

For an icon, you can use your own or point to one supplied with the plug-in, for example your icon path could point to



This is located at:

`C:\eclipse\plugins\com.ibm.rational.clearcase_7.0.1.CU03\icons\obj48\getting_started.gif`

For normal client use

If you want to add a shortcut on the user's desktop for debug tracing, the steps are the same, but add `c:\eclipse\eclipse -debug` for the shortcut path and point to



this is located at:

`C:\eclipse\plugins\com.ibm.rational.clearcase_7.0.1.CU03\icons\obj48\working_with.gif`

Store these shortcuts in `C:\Documents and Settings\All Users\Desktop`.

Optional Configuration Steps

Multiple ClearCase Version Support

If your organization needs to support users at different major levels of ClearCase server (for example, version 2003.06.16, ClearCase 7.0, or ClearCase 7.0.1) CCRC can support having any combination of versions plugged into the Eclipse SDK.

In the steps above we demonstrated how to install a client that can be used with ClearCase 7.0 (the client may be called 7.0.1-CU003, but we will refer to it as a client of 7.0).

[Technote 1224586](#) explains the current CCRC client platform specifics.

In the next few sections we will walk through how to add the ClearCase 2003.xx version of the CCRC client to the same interface (to provide client capabilities for ClearCase 2003.06.14 or later server connections).

**** Note:** Only one “plug-in” or client can be active at any time. It is possible (with standard Eclipse technology) to switch back and forth between versions (plug-ins), yet that is a use case reserved for specific applications.

First, we need to change our perspective. Choose the “Resource” perspective, or any perspective you want. We will be disabling ClearCase functionality, and would receive a warning that we couldn’t restart the perspective (this warning is normal).

To disable or enable a plug-in, go to *Help > Software Updates > Manage Configuration*. Under Eclipse, expand the c:\eclipse folder, and then expand “Rational ClearCase Client for Eclipse 7.0.1.CU003.” (figure 17)

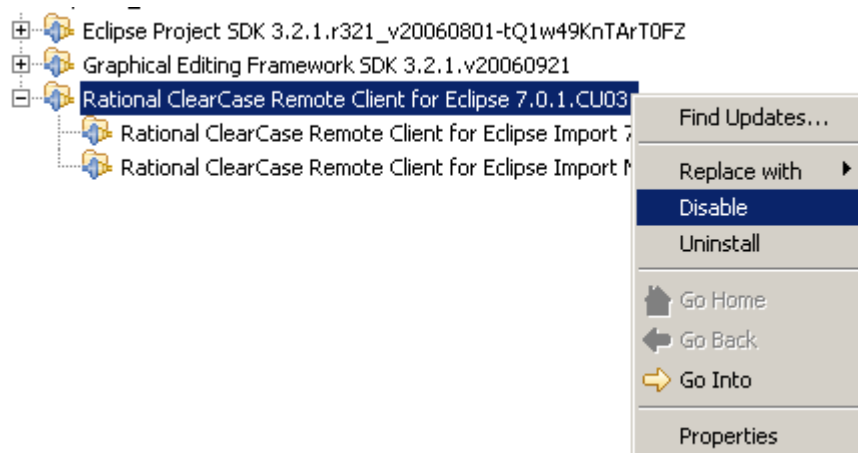


Figure17

Restart the workbench when prompted.

Go back to *Help > Software Updates > Manage Configuration*, CCRC should no longer be present. Don't worry it is merely disabled, and you can display disabled plug-ins by clicking on the "Show Disabled Features" icon. (figure 18)

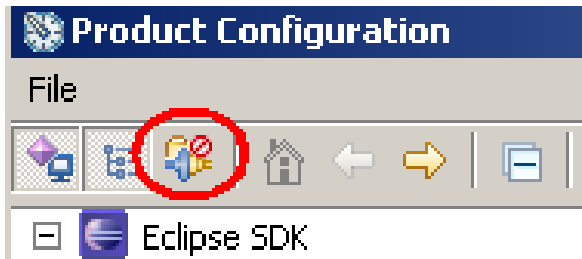


Figure18

The steps to install our second version of CCRC are the same as outlined earlier in the section "Install and Configure CCRC as a Plug-in to Eclipse" except that we will create a new update site called "2003 CLIENT" and that update site will have a path of :

<http://www3.software.ibm.com/ibmdl/pub/software/rationalsdp/clearcase/ccrc/614/update/>

Disable Eclipse to visit for updates/plug-ins from the original update site created. Your Install dialog should look like this (figure 19):

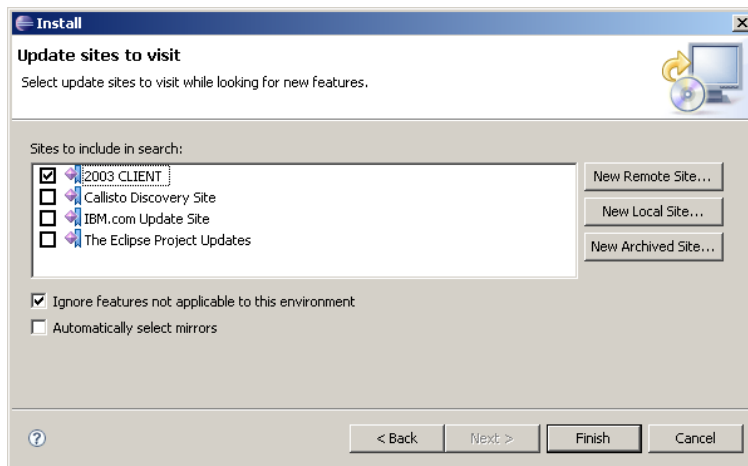


Figure19

Click Finish and take the defaults as you did earlier with the 7.0 client installation. Restart when prompted.

We can confirm the installation by going back to Help > *Software Updates* > *Manage Configuration* (figure 20)

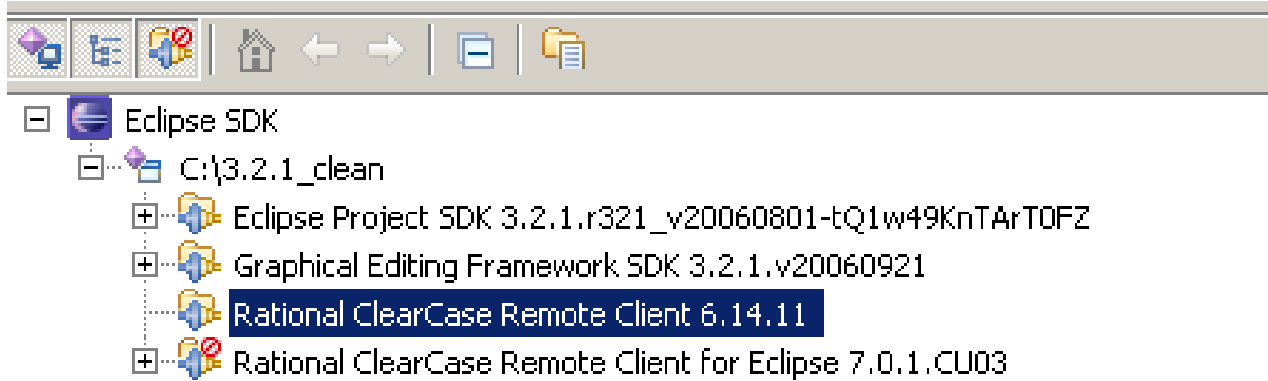


Figure20

As you can see disabling or enabling a plug-in is as simple as a few clicks.

Automating the Installation and Disabling of CCRC in Eclipse

As demonstrated above, a user can enable a specific CCRC version and then follow the steps to package and deploy that "version". They can return and enable a different version and create a separate package to deploy. Be aware that if you wanted to deploy the three currently shipped versions of CCRC, it would require three separate archives and could cause confusion with your users.

Eclipse provides a powerful way to script the installation, enable, and disable the plug-in without any user interaction.

Displaying what Features are currently installed in a specific Eclipse instance

```
C:\eclipse\java -cp startup.jar org.eclipse.core.launcher.Main -application
org.eclipse.update.core.standaloneUpdate
-command listFeatures
```

Run against the above instance, would yield:

```
Site: file:/C:/eclipse/
Feature: org.eclipse.jdt 3.2.1.r321_v20060905-R4CM1Znkvre9wC- enabled
Feature: org.eclipse.sdk 3.2.1.r321_v20060801-tQ1w49KnTArT0FZ enabled
Feature: org.eclipse.gef.sdk 3.2.1.v20060921 enabled
Feature: org.eclipse.jdt.source 3.2.1.r321_v20060905-R4CM1Znkvre9wC- enabled
Feature: org.eclipse.rcp.source 3.2.1.r321_v20060801-2ekW2BxmcpPUOoq enabled
Feature: org.eclipse.gef.source 3.2.1.v20060921 enabled
Feature: org.eclipse.pde.source 3.2.1.r321_v20060823-6vYLLdQ3Nk8DrFG enabled
Feature: org.eclipse.platform.source 3.2.1.r321_v20060921-b_XVA-INSQSyMtx enabled
Feature: org.eclipse.rcp 3.2.1.r321_v20060801-2ekW2BxmcpPUOoq enabled
Feature: org.eclipse.platform 3.2.1.r321_v20060921-b_XVA-INSQSyMtx enabled
Feature: com.ibm.rational.clearcase.standalone.feature 6.14.11 enabled
Feature: org.eclipse.pde 3.2.1.r321_v20060823-6vYLLdQ3Nk8DrFG enabled
Feature: org.eclipse.gef 3.2.1.v20060921 enabled
Feature: com.ibm.rational.clearcase.ccimport.feature.nl1 7.0.0.20061020A disabled
Feature: com.ibm.rational.clearcase.ccimport.feature 7.0.0.20061020A disabled
Feature: com.ibm.rational.clearcase.ide.feature 7.0.1.CU03 disabled
```

Command completed successfully.

Displaying what features are available on a specific URL to install

```
C:\eclipse\java -cp startup.jar org.eclipse.core.launcher.Main -application
org.eclipse.update.core.standaloneUpdate -command search -from
http://www3.software.ibm.com/ibmdl/pub/software/rationalsdp/clearcase/ccrc/614/update/
```

Run against the 2003 site we used before would yield:

```
"Rational ClearCase Remote Client for Eclipse" com.ibm.rational.clearcase.ide.feature
6.14.11
```

```
"Rational ClearCase Remote Client" com.ibm.rational.clearcase.standalone.feature 6.14.11
Command completed successfully.
```

The current state of our Eclipse instance (as documented earlier in this paper) should have ClearCase Remote Client 6.14.11 enabled and 7.0.1.CU03 disabled.

Let's step through changing this. In this example, we assume that we have been using 2003.06 servers in our organization with the ClearCase Remote Client 6.14.11. We now want to upgrade our servers to ClearCase 7.0. The following steps could be used to script this upgrade without the users being involved.

Disabling ClearCase Remote Client for Eclipse

```
C:\eclipse\java -cp startup.jar org.eclipse.core.launcher.Main -application
org.eclipse.update.core.standaloneUpdate -command disable -featureId
com.ibm.rational.clearcase.standalone.feature -version 6.14.11
```

*** Note that the featureId was retrieved from the output we received from inquiring what was installed into this instance*

Enabling ClearCase Remote Client for Eclipse

```
C:\eclipse\java -cp startup.jar org.eclipse.core.launcher.Main -application
org.eclipse.update.core.standaloneUpdate -command enable -featureId
com.ibm.rational.clearcase.ide.feature -version 7.0.1.CU03
```

Installing ClearCase Remote Client for Eclipse

```
C:\eclipse\java -cp startup.jar org.eclipse.core.launcher.Main -application
org.eclipse.update.core.standaloneUpdate -command install -from http://{ - CCRC ServerName -
}/ccrc/update -featureId com.ibm.rational.clearcase.ide.feature -version 7.0.0
```


Package and Deployment

After all modifications, customizations and changes have been made, the packaging of this CCRC client is simple:

- 1.) Compress the entire c:\eclipse directory.
- 2.) Copy the shortcut(s) from C:\Documents and Settings\All Users\Desktop to the archive (saving folder path info)
- 3.) (OPTIONAL) script the decompression or the disabling or enabling of a specific version if multiple versions have been installed previously

To deploy this client, there is no installation required. Merely Decompress the archive to disk and if the default path is taken (or mandated) all clients will have CCRC installed at c:\eclipse and each client will be identical.

Conclusion

This paper is not meant to replace the standard practices used to install IBM Rational ClearCase. Consider it as an alternative approach that allows for faster deployment of a common customized client, that benefits both the end user and the administrative staff of any organization.