



*The content of this publication is the property of Clearvision-CM  
Reproduction of this content is strictly prohibited © 2007 - 2009*

Copyright (c) Clearvision CM 2007 - 2009

# Contents

<b>1 Introduction</b>	<b>1/30</b>
<b>2 Prerequisites</b>	<b>2/30</b>
2.1 Important Information for Windows Vista Users	2/30
2.2 Supported Platforms	2/30
2.2.1 Operating Systems	2/30
2.2.2 NEW in v4.0: Platform-independent "Portable Hooks"	2/30
2.2.3 Clearquest	3/30
2.2.4 Subversion	3/30
2.2.5 Red Hat Enterprise Linux Pre-installation requirements	3/30
2.3 Configuration planning	3/30
2.3.1 Standard Distributed Configuration	4/30
2.3.2 Shared CO2SVN/Clearquest Configuration	4/30
2.3.3 Shared CO2SVN/Subversion Configuration	5/30
2.3.4 Alternative Configurations	5/30
2.4 Communication Paths	5/30
2.5 License	6/30
<b>3 Clearquest Setup</b>	<b>8/30</b>
3.1 Setup the Clearquest database schema	8/30
<b>4 CO2SVN Installation</b>	<b>10/30</b>
4.1 CO2SVN Server	10/30
4.1.1 Test the Clearquest configuration	12/30
4.1.2 Configuring the Server to run on startup	13/30
4.2 CO2SVN Client	14/30
4.2.1 Configuring a User Id for CO2SVN	14/30
4.2.2 Configuring the Client to run automatically	14/30
4.3 Setting permissions on Linux	15/30
4.4 Subversion and/or Git Hook Setup	16/30
4.5 NEW in 4.0: Portable Hooks	16/30
<b>5 Customization</b>	<b>17/30</b>
5.1 Configuration Options	17/30
5.1.1 Server Configuration Options	17/30
5.1.2 Hook Configuration Options	19/30
5.2 Perl Script Customization	20/30
5.2.1 Change the record selection criteria	20/30
5.2.2 Change the CO details displayed	21/30
5.2.3 Recording all changes against a default CO record	22/30
<b>6 Troubleshooting</b>	<b>23/30</b>
6.1 Installation Support	23/30
<b>7 Appendix A: Using CO2SVN with Git</b>	<b>24/30</b>
7.1 Overview	24/30
7.2 Heterogeneous Environments	25/30
7.3 Installation	25/30
<b>8 Appendix B: RHEL4 Pre-requisites</b>	<b>26/30</b>
8.1 Subversion 1.4.4	26/30

# Contents

<b><u>9 Appendix C: Portable Hooks Examples</u></b> .....	<b>27/30</b>
<u>9.1 Sun Solaris 10</u> .....	27/30
<u>9.1.1 Set path to pick up GNU tools</u> .....	27/30
<u>9.1.2 Python</u> .....	27/30
<u>9.1.3 Twisted 8.2.0</u> .....	27/30
<u>9.1.4 PySVN 1.6.2</u> .....	27/30
<b><u>10 Appendix D: Portable Client Example</u></b> .....	<b>29/30</b>
<u>10.1 Installation of CO2SVN portable client on a Mac (OS-X Leopard)</u> .....	29/30
<u>10.1.1 Requirements</u> .....	29/30
<u>10.1.2 Installing X11</u> .....	29/30
<u>10.1.3 Installing Xcode</u> .....	29/30
<u>10.1.4 Installing MacPorts</u> .....	29/30
<u>10.1.5 Installing Python and other requirements</u> .....	29/30

# 1 Introduction

 **CQ2SVN can now also integrate ClearQuest with Git repositories.**

Welcome to Clearvision's CQ2SVN: Integrating Clearquest with Subversion.

This is the Administration Guide for CQ2SVN which explains how to install and configure CQ2SVN. For a quick guide to installation, refer to the Quick Start and for a user's perspective, please refer to the User Guide. The latest versions of these documents are always available on [our website](#).

## 2 Prerequisites

The CQ2SVN Server is tested and certified against a particular Subversion release (see below for details). CQ2SVN is independent of the database used by Clearquest (CQ2SVN uses the Clearquest API to communicate with the database). CQ2SVN is certified against Clearquest v6(2003.06) and v7 (see below).

CQ2SVN is independent of the Subversion client used.

### 2.1 Important Information for Windows Vista Users

If you're running Windows Vista, there are two steps which you need to carry out before CQ2SVN functions correctly.

1. If using the installer, make sure that it's run as an Administrator (Right click on the installer and select 'Run as Administrator')
2. After installing CQ2SVN, you will need to run the Server and Client as an Administrator. There are a couple of ways to achieve this:
  - ◆ Right Click on the executable and select 'Run as Administrator' (You will have to repeat this every time you load the executable)
  - ◆ Right Click on the executable and select 'Properties'. Click on the 'Compatibility' tab and at the bottom, tick 'Run this program as an administrator' (This will mean every time the executable is run, it will be run as an Administrator)


### 2.2 Supported Platforms

Current support is as follows (for more information on the product roadmap, please contact us at [sales@clearvison-cm.com](mailto:sales@clearvison-cm.com)).

#### 2.2.1 Operating Systems

Platform	CQ2SVN Version
Windows XP Pro SP2 (x86 32bit)	v1.x, v2.x, v4.x
Windows XP Pro SP3 (x86 32bit)	v2.1g+
Red Hat Enterprise Linux 4 AS (x86 32bit)	v1.2, v2.x
Windows Vista	v4.0x

#### 2.2.2 NEW in v4.0: Platform-independent "Portable Hooks"

 NEW in v4.0: As of version 3.1, Cq2SVN provides a special version of the Subversion hooks, which is independent of the operating system. This allows you to run Subversion with Cq2SVN on virtually any platform.

We recommend that you **only use this if your platform is not supported by the executables** as the installation is more involved.

To use the Portable Hooks, you **must download the version that matches your version of Python**. There are also a number of '**pre-requisite packages**' that must be installed.

These packages may already be installed on your system, but if they are not, please note that the software required is Open Source and can be downloaded and compiled from source as required. Note that on Linux, Sun and Mac, binary packages for your OS distribution may also be available.

### Prerequisite Packages:-

- [Subversion 1.5](#) - including the python SWIG bindings
- [python](#) - version 2.5 is recommended
- [Twisted v8](#)
- [pysvn](#)

Note that the version numbers indicated above are the versions of the tools CQ2SVN has been tested with. CQ2SVN may well work with other versions of the tools, but this cannot be guaranteed. Please make sure that the product works in your environment before purchase. An evaluation licence can be made available for that purpose.

Please refer to the appendix for guides on installing pre-requisite software for specific platforms.

### 2.2.3 Clearquest

Clearquest Version	CQ2SVN Version
Clearquest 2003.06	v1.x v2.x
Clearquest v7.0	v2.x
Clearquest v7.1	v3.x, v4.x

The latest releases of CQ2SVN are tested against Clearquest v7.1 (releases prior to v2.1g were tested against v2003.06).

### 2.2.4 Subversion

The current releases are for Subversion 1.4.x, 1.5.x and 1.6.x. Please make sure that you use the correct version for your system.


### 2.2.5 Red Hat Enterprise Linux Pre-installation requirements

CQ2SVN is built against an up to date version of Subversion on RHEL4. We recommend that you download and install the latest Subversion prior to installing CQ2SVN, see the Appendices for more details.

The binary that we distribute contains most dependencies within it, but doesn't include all of the operating system packages that are required. These dependencies are documented in the `README_dependencies.txt` file in the docs folder of the distribution.

## 2.3 Configuration planning

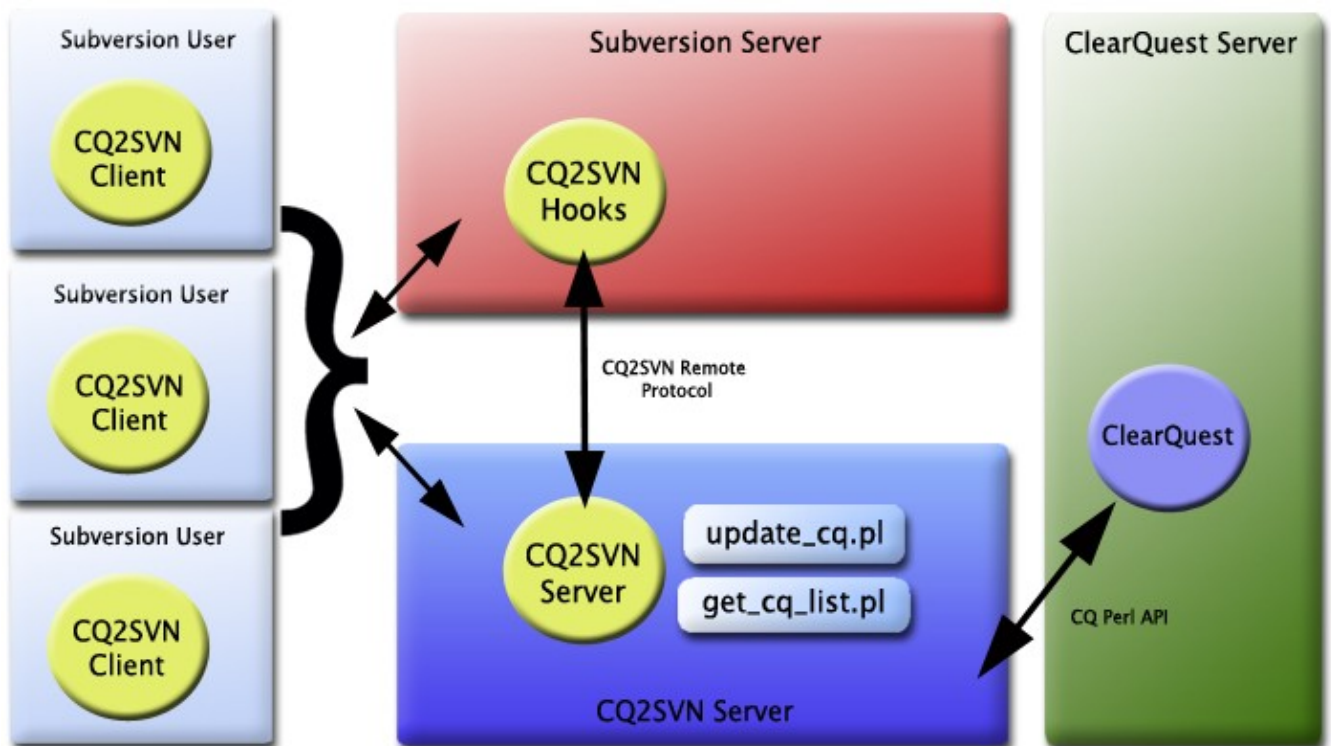
You will need to choose a machine and a port number for CQ2SVN to use (eg 54321). Note that the client machines will require access to this port on the Server (and this may require firewall configuration). As per the IANA recommendations, we recommend that you use one of the Dynamic and/or Private Ports from 49152 through 65535.

 **New in 2.1** Version 2 of CQ2SVN separates the CQ2SVN server from the Subversion hooks so that it can be run on its own machine; or on any other machine. Typically you might want to run the CQ2SVN server on the Clearquest server or it's own machine as depicted below (note that you will have to install the Clearquest Perl API on that machine, configured to access the databases you want to use with CQ2SVN). Note that the perl scripts have moved from the Subversion server (in v1.x) to the CQ2SVN Server machine (in v2.x).

Note that you can run many CQ2SVN Servers (not shown below) for added flexibility (eg to serve separate Subversion servers).

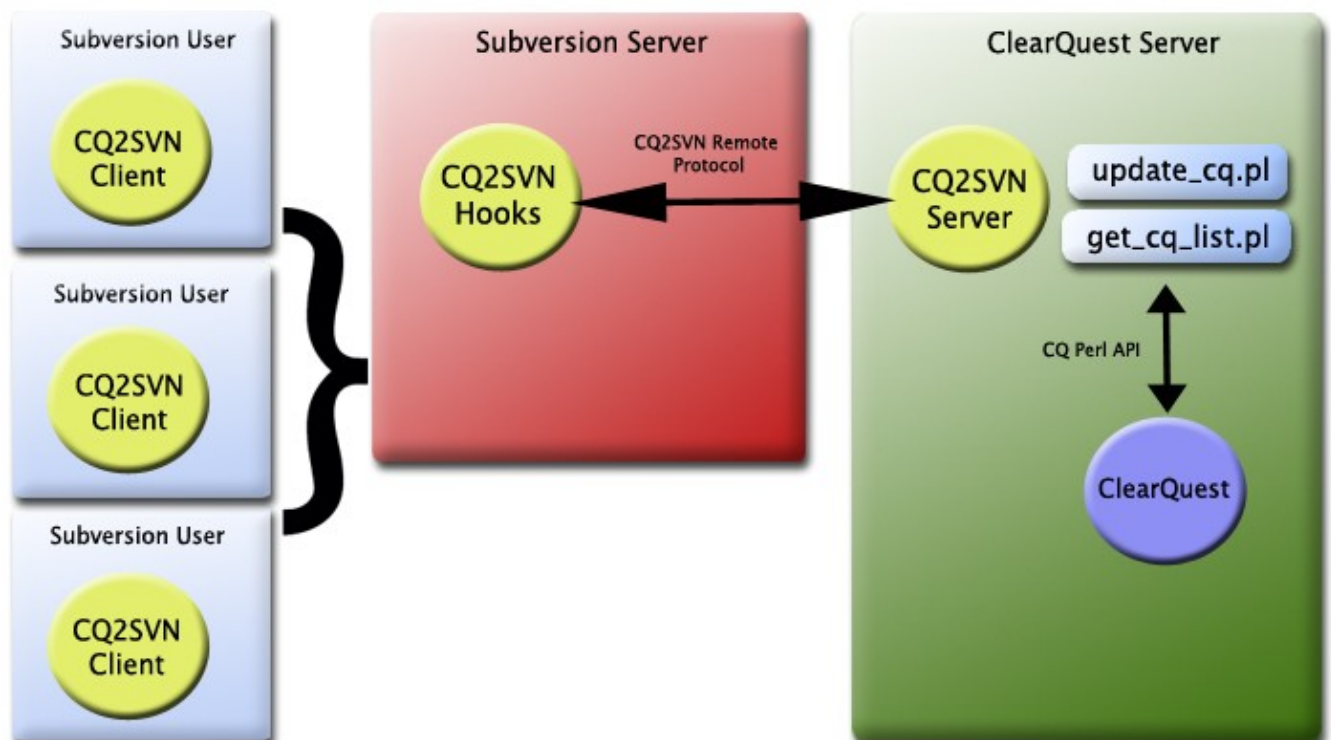
### 2.3.1 Standard Distributed Configuration

This is the standard, fully distributed configuration which allows the maximum flexibility for location of the CQ2SVN Server according to your geographical needs.



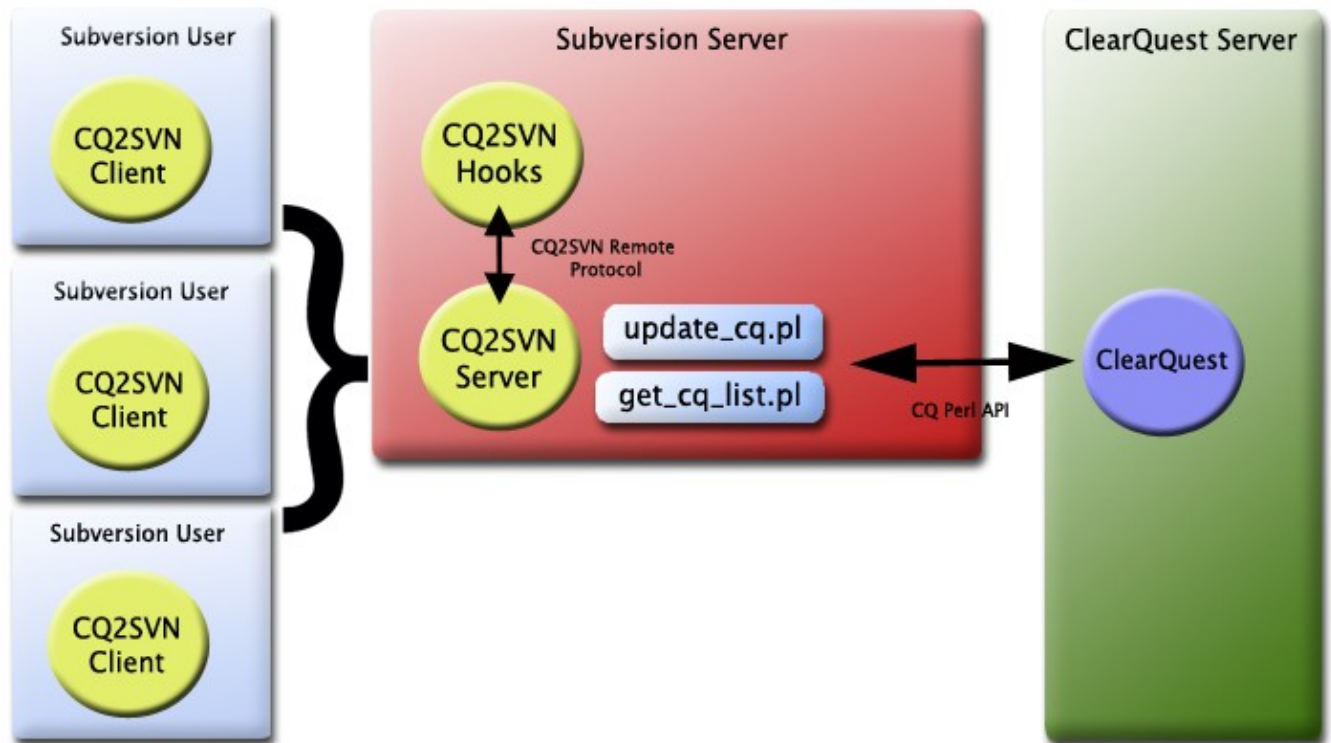
### 2.3.2 Shared CQ2SVN/Clearquest Configuration

If you can share the Clearquest server with CQ2SVN, it makes sense to locate the CQ2SVN Server on the Clearquest machine itself.



### 2.3.3 Shared CQ2SVN/Subversion Configuration

In smaller organisations, it might make sense to locate the CQ2SVN Server on the Subversion server itself.



### 2.3.4 Alternative Configurations

#### 2.3.4.1 Multiple Subversion servers with a single ClearQuest Server

CQ2SVN can support multiple subversion servers either using separate or shared [ClearQuest](#) databases. So, for example, to use one common [ClearQuest](#) database with 2 Subversion repositories, simply setup one CQ2SVN Server for each Subversion repository, each pointing to the shared [ClearQuest](#) Server. On each User machine, run 2 CQ2SVN clients, each with it's own cq2svn\_client.cfg file (pointing to it's respective CQ2SVN Server). This way, whenever a commit is made *to either repository*, the CQ record is requested from the User (one or other of the client processes will respond).

## 2.4 Communication Paths

The following section describes at a high level the various communications High level

- CQ2SVN Client starts and establishes a connection to CQ2SVN Server
- Developers performs a Subversion commit via any Subversion client e.g. Tortoise, Eclipse, command line, etc
- CQ2SVN Hook is initiated and contacts the CQ2SVN Server
- CQ2SVN Server calls [ClearQuest](#) via CQ API calls and asks for a list of records based upon the user performing the commit
- CQ2SVN Server passes the list of record ID's to the CQ2SVN Client which are then displayed in the GUI window
- Developer selects appropriate CQ record(s) and selects OK
- The selected CQ record ID(s) is passed back to the CQ2SVN Server
- CQ2SVN Server writes the Subversion change set of files into the CQ record
- CQ2SVN Server passes the CQ record ID to the CQ2SVN Hook

- CQ2SVN Hook allows the files to be committed and writes into the revision the CQ record ID's
- Subversion informs the Subversion client (Tortoise, Eclipse, command line, etc) the commit has completed

## 2.5 License

You must obtain a license from [sales@clearvision-cm.com](mailto:sales@clearvision-cm.com) before you start. Note that this license will only let you run CQ2SVN on a particular server machine, on a particular port, and will allow a maximum number of users. The license is time-limited.

To request a license we need some information about the machine that you will be running the CQ2SVN server. To get this information, you can run the `get_license_info` program on the machine you want to use as the CQ2SVN server (the `get_license_info` program can be found within the tools directory of the zip package). This will retrieve system information from the machine and present you with a license request to return to Clearvision at the address below. Alternatively, you can fill in the details as follows and send them to [sales@clearvision-cm.com](mailto:sales@clearvision-cm.com) :

```
License Request for Clearvision CQ2SVN:¶
.... Evaluation: yes/no¶
.... IP Address: 192.168.64.1¶
..... Port: 54321¶
... MAC Address: 00-50-56-C0-00-05¶
..... Users: 50¶
..... Company: Your Company name here¶
· Email Address: yourco@yourco.com¶
... Expiry Date: 2008-12-31¶
..... Product: CQ2SVNv2¶
```

The `get_license_info` executable is run from a command prompt, eg on Windows:

```

C:\WINDOWS\system32\cmd.exe
C:\Program Files\Cleavision\CQ2SUN\tools>dir
Volume in drive C has no label.
Volume Serial Number is 74C4-8CF7

Directory of C:\Program Files\Cleavision\CQ2SUN\tools
21/08/2007  18:16    <DIR>          .
21/08/2007  18:16    <DIR>          ..
21/08/2007  18:07                1,403,258 get_license_info.exe
                1 File(s)        1,403,258 bytes
                2 Dir(s)        7,426,170,880 bytes free

C:\Program Files\Cleavision\CQ2SUN\tools>get_license_info.exe

We have examined your machine, you may use the following form
to request a licence to run the CQ2SUN Server on this machine
Please fill out the missing fields below and email the request
to sales@cleavision-cm.com to obtain your licence

-----
CQ2SUN Server License Request:

Company Name=
Contact Email Address=
Required Expiry Date=      (eg 2007-12-31)
CQ2SUN Server port number= (eg 242424) ** IMPORTANT
Number of Users=

MAC='02-00-4C-4F-4F-50'
IP='169.254.25.129'
HOSTNAME=''

-----

C:\Program Files\Cleavision\CQ2SUN\tools>

```

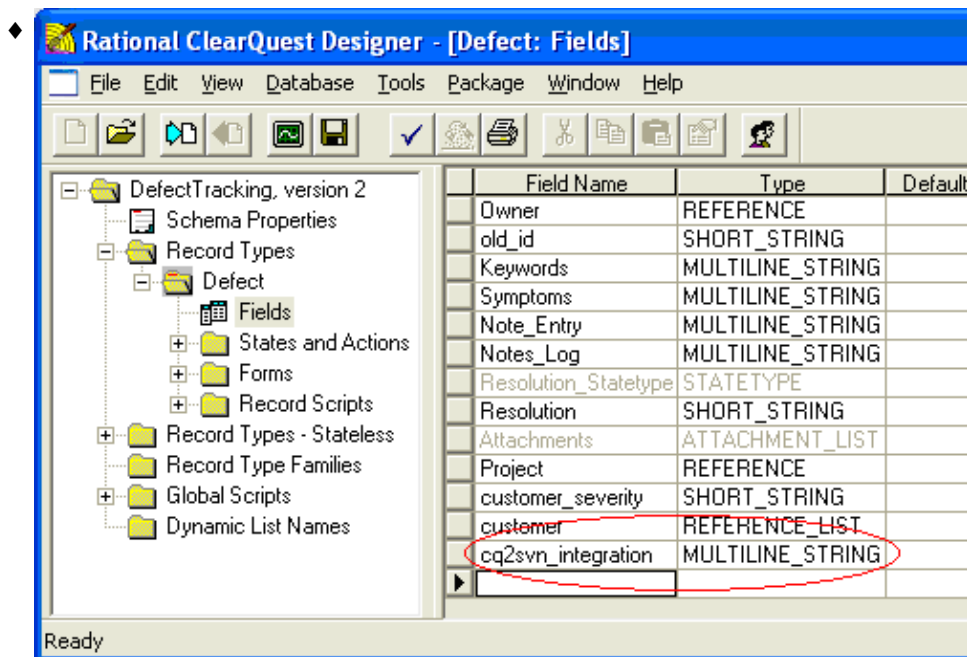
Cut and paste the highlighted section; check all the fields and fill in the empty ones (Company Name, etc) and send to [sales@cleavision-cm.com](mailto:sales@cleavision-cm.com) .

# 3 Clearquest Setup

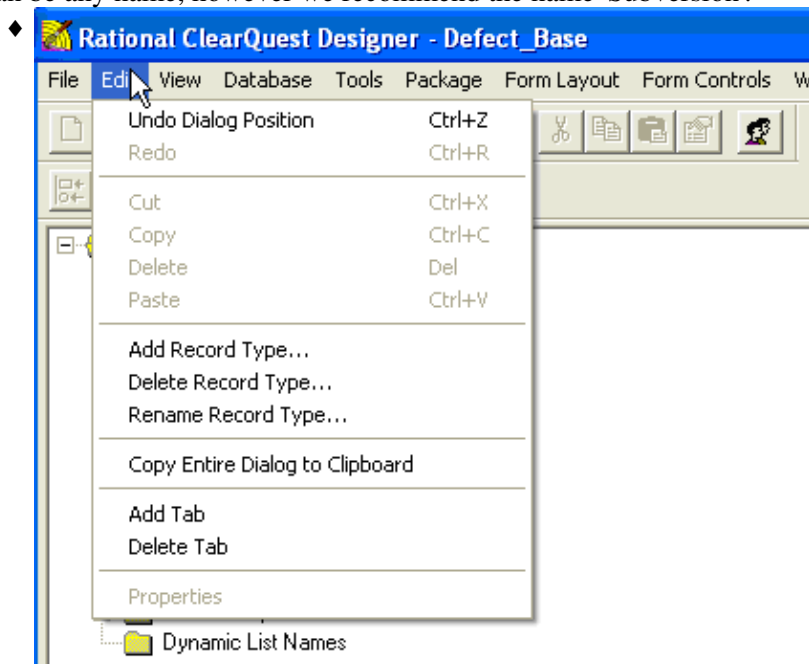
The following guide takes you through steps to add CQ2SVN support to Clearquest. These instructions use the Defect Tracking Schema as an example but CQ2SVN is designed to accomodate any schema (please replace with your local requirements as necessary). The following instructions should be testable using a Clearquest Evaluation database (you can then apply the principles to your local system).

## 3.1 Setup the Clearquest database schema

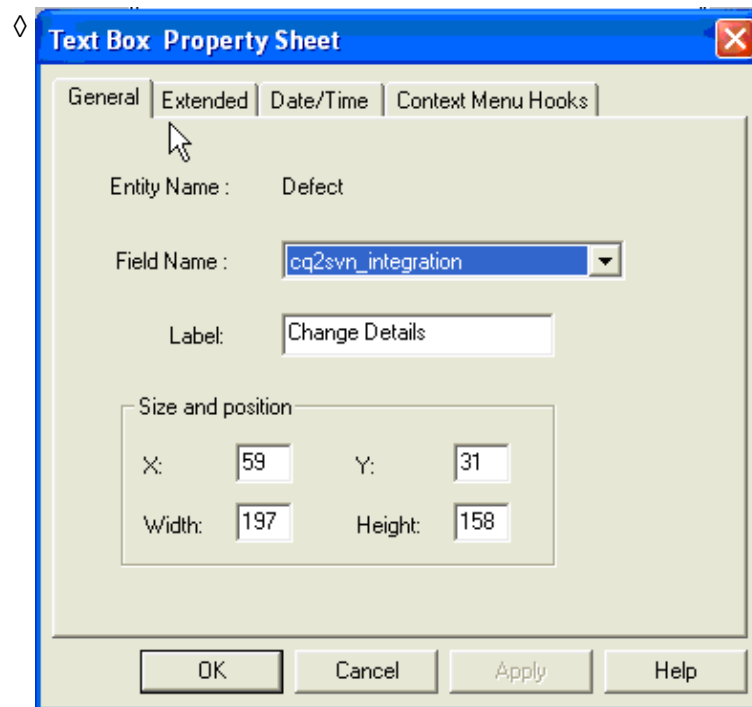
1. Use the Clearquest Designer to edit your schema e.g. Defect Tracking
2. Add a MULTILINE\_STRING field called cq2svn\_integration in the (eg Defect) record.



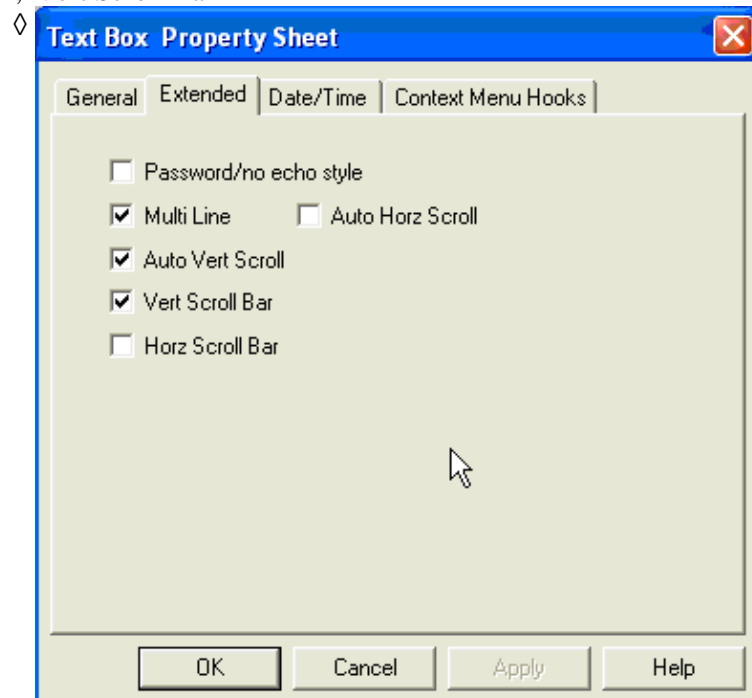
3. Open the Forms folder under the Defect Record Type, edit the Defect\_Base Form and add a tab. The tab can be any name, however we recommend the name 'Subversion'.



- on this tab add a label 'Change Details' and add a Text Box for the cq2svn\_integration field.



- ◆ edit the 'Extended' tab of the Text Box Property Sheet and tick: 'Multi Line'; 'Auto Vert Scroll'; 'Vert Scroll Bar'



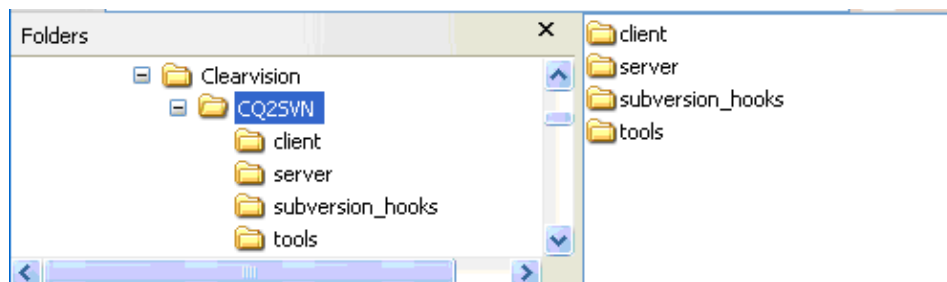
4. save and upgrade your databases as normal.

---

## 4 CQ2SVN Installation

CQ2SVN has 3 parts: a client, a server; and a set of files which must be installed in your Subversion repository's `hooks` directory. There is also a set of sample scripts for Unix platforms and a tool to get license info in the `tools` directory.

Start by configuring the Subversion server by unpacking the zip file into the `C:\Program Files` location (any location can be used, however all the configuration files and instructions assume the use of `C:\Program Files`. On Linux, you may want to replace this with `/opt`). The following directory structure will be created;



- `client` directory contains the executable which must be started on every client.
- `server` directory contains the CQ2SVN server executable & perl scripts
- `subversion_hooks` contains the set of files which reside in the `hooks` directory of every subversion repository. These programs interact with Subversion and the CQ2SVN Server.
- `tools` contains the license request tool and sample Unix scripts
- `docs` contains documentation for CQ2SVN (including this guide)

**⚠ NOTE:** Please take care when editing any configuration files. We recommend the use of professional text editors such as Cream/Vim. Wordpad should also work, **do not use Notepad or MS Word.**

**💡 New in 2.1** There are 3 separate configuration files for the server, client and hooks called `cq2svn_server.cfg`, `cq2svn_client.cfg` and `cq2svn_hook.cfg`.

### 4.1 CQ2SVN Server

You must have Clearquest installed on the CQ2SVN Server machine as the Clearquest Perl API is used. Clearquest databases do NOT need to be located on this machine.

- **💡 New in 2.1** unpack the zip file into `C:\cq2svn_server` (**⚠ Please do not use a path name with a space such as `c:\Program Files`**) as CQ2SVN does not presently support paths with white spaces to the perl scripts)
- edit the config file  
`C:\cq2svn_server\Clearvision\CQ2SVN\server\cq2svn_server.cfg` and set the information **exactly** as per your license request.
- edit the config file to point to the Clearquest perl executable for your installation.

```
◆ cqperl=C:\Program Files\Rational\Common\ratlperl.exe
```

*(note: the `svn_hook_dir` option is no longer required)*

- *optionally* set the `cqperl_scripts_dir` configuration option if you want to store the perl scripts anywhere other than with the server executable (the server looks for these scripts in its 'current working directory' by default) - please do **not** use a path with a space in it.

```
◆ cqperl_scripts_dir=C:\my_perl_scripts
```

- edit the `get_cq_list.pl` and `update_cq.pl` scripts to point to your CQ Database and use appropriate Clearquest Ids. Clearquest administrators with Perl knowledge may additionally customize these scripts for localized functionality (in particular configuring the schema that you use).

Example of configuration settings in `get_cq_list.pl`

```
# *****¶
# PLEASE CONFIGURE THESE VARIABLES¶
# ACCORDING TO YOUR INSTALLATION¶
# *****¶
¶
my $cq_superuser_id = "admin";¶
my $cq_superuser_password = "";¶
my $cq_db = "cvsvn";¶
my $cq_dbset = "2003.06.00";· # If you have multiple connections, set
this appropriately¶
```

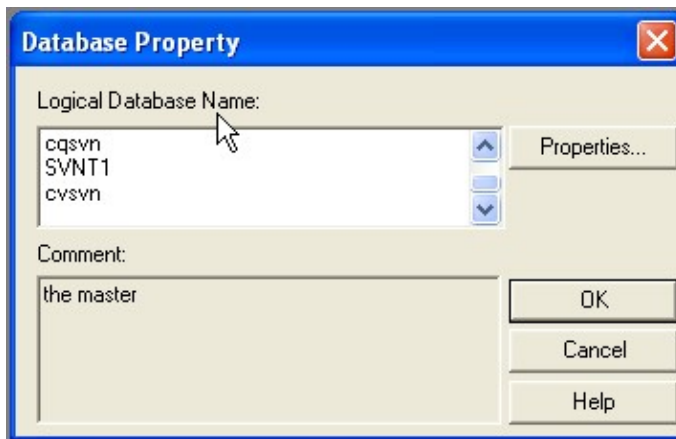
Example of configuration settings in `update_cq.pl`

```
# *****¶
# PLEASE CONFIGURE THESE VARIABLES¶
# ACCORDING TO YOUR INSTALLATION¶
# *****¶
¶
my $cq_superuser_id..... = "admin";¶
my $cq_superuser_password = "";¶
my $cq_db..... = "cvsvn";¶
my $cq_dbset..... = "2003.06.00";¶
my $cq_record_type..... = "defect";¶
my $cq2svn_field..... = "cq2svn_integration";¶
```

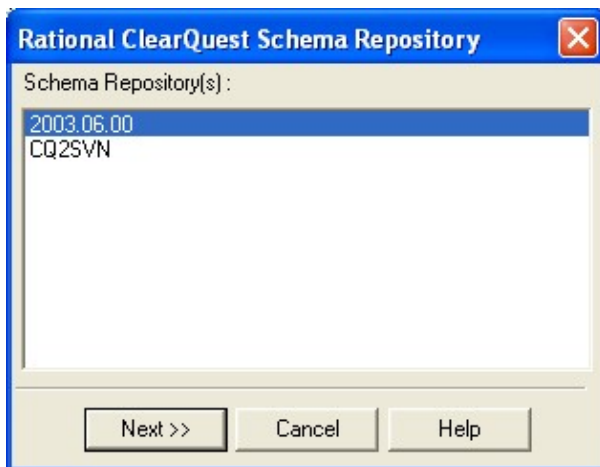
In the Clearquest GUI, the settings relating to `my $cq_db = "cvsvn"`; would look like this



Or



Example of my `$cq_dbset = "2003.06.00";`



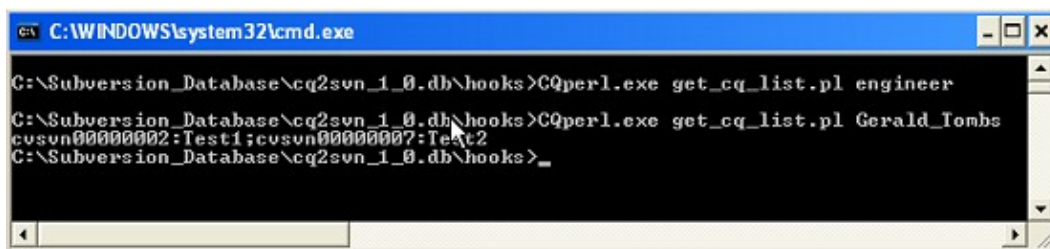
#### 4.1.1 Test the Clearquest configuration

To make sure that you've configured the perl scripts properly, you can run them directly from the command line, eg (in a Command window) (*Change engineer to a valid Clearquest login ID with appropriately assigned records; change SAMPL00000012 to a valid Clearquest Record Id; and on Linux use `cqperl` rather than `CQPerl.exe`.*)

```
CQperl.exe get_cq_list.pl engineer
```

You should see the list of eligible Clearquest records for the user 'engineer'.

Example - Windows



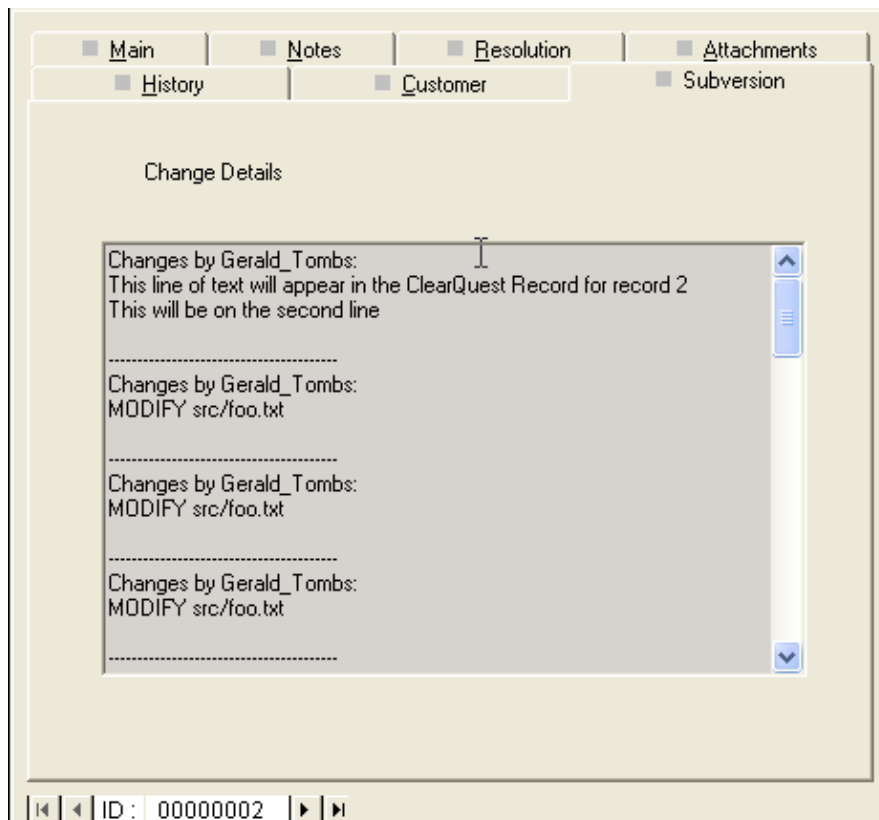
**⚠ Migration Note:** *this has changed in v3.0: there is an extra parameter to `update_cq.pl` to indicate if the record should be moved to it's next default state.*

```
CQPerl.exe update_cq.pl SAMPL00000012 0
```

This will wait for you to type some text on standard input (terminate with Ctrl-Z followed by the Enter key, or Ctrl-D on Linux). Refresh and view the Subversion tab on the Clearquest record to check the text has been created correctly.

Example - Windows

```
C:\Subversion_Database\cq2svn_1_0.db\hooks>CQperl.exe get_cq_list.pl engineer
C:\Subversion_Database\cq2svn_1_0.db\hooks>CQperl.exe get_cq_list.pl Gerald_Tombs
cvsun00000002:Test1;cvsun00000007:Test2
C:\Subversion_Database\cq2svn_1_0.db\hooks>
C:\Subversion_Database\cq2svn_1_0.db\hooks>
C:\Subversion_Database\cq2svn_1_0.db\hooks>CQperl.exe update_cq.pl cvsun00000002 Gerald_Tombs
This line of text will appear in the ClearQuest Record for record 2
This will be on the second line
^Z
C:\Subversion_Database\cq2svn_1_0.db\hooks>_
```



#### 4.1.2 Configuring the Server to run on startup

On Windows, we recommend that you add the CQ2SVN Server as a 'Scheduled Task', 'When my computer starts'. An alternative would be to add an item in the `autoexec.nt` file so that the server is started on boot and is available for all users. Alternatively, you may want to start the Server manually, eg whenever subversion is started.

There is a `-f` command argument that you can use to choose a particular config file like this:

```
cq2svn_server.exe -f c:\my_server_cq2svn.cfg
```

You may also want to refer to the `tools/run_server.sh` sample script for Unix platforms.

## 4.2 CQ2SVN Client

- Either unpack the whole zip file as above or copy just the `c:\Program Files\Clearvision\CQ2SVN\client` folder. (*An alternative location can be used, eg on Linux you may use `/opt/Clearvision/CQ2SVN`, however all the configuration files and instructions use `C:\Program Files` as an example*).
- edit the file `C:\Program Files\Clearvision\CQ2SVN\client\cq2svn_client.cfg` and set the information **exactly** as per your license request, pointing to the CQ2SVN server related to the Subversion instance you want to use.

### 4.2.1 Configuring a User Id for CQ2SVN

CQ2SVN uses a user name for a number of things:

1. to identify users of CQ2SVN
2. to identify commits by that user in Subversion
3. to get the list of Clearquest records for a user.

Because CQ2SVN needs to correlate these systems, you need to be careful to make sure that they can be matched. If you already have a Clearquest system and are installing a Subversion system, you will probably want to use your Clearquest User Ids for Subversion (and hence for CQ2SVN).

By default, CQ2SVN uses your operating system user id as it's id for Subversion and Clearquest. If you want to use an alternative id for CQ2SVN, configure this on the client's machine with the `svn_uid` (*in version 1 of CQ2SVN this was called `cq_uid` and existed in the `[CQ2SVN Client]` section*) option in the `cq2svn_client.cfg` file, eg:

```
[Options]¶
svn_uid=engineer¶
```

The default server configuration disables the ability to use `svn_uid` on the clients. If you want to enable this functionality, please see the CQ2SVN Server Customization section later.

### 4.2.2 Configuring the Client to run automatically

You will probably want to have the CQ2SVN Client running whenever a User is logged on to his/her machine so that CQ2SVN is available whenever they commit changes to Subversion.

#### 4.2.2.1 On Windows

##### 4.2.2.1.1 logon

We recommend that you add a shortcut to the CQ2SVN Client as an item in the `StartMenu/Programs/Startup` folder for each CQ2SVN User. This setup ensures that whenever a User logs on to Windows, they automatically register with CQ2SVN. You may want different configurations for each user, or you may be able to share one configuration (eg if there are no clients that require custom user ids and all Users use the same repository). There is a `-f` command argument that you can use to choose a

particular config file like this (remember to add this to your shortcut):

```
cq2svn_client.exe -f c:\Program
Files\Clearvision\CQ2SVN\client\cq2svn_client.cfg
```

Depending on your setup, an alternative may be to add the startup as a 'logon script' in the Profile tab of the Computer Management/Users settings.

Another alternative is to use a Windows 'Scheduled Task' and run the client 'When I Log On'.

#### 4.2.2.1.2 logoff

You may be able to assign a logoff script as [documented here](#) and write a simple batch script (using the DOS [taskkill](#) command) to kill the process on shutdown, for example:

```
taskkill /f /t /im cq2svn_client.exe
```

#### 4.2.2.2 On Linux/Unix

You can run the CQ2SVN Client by adding a line to your login script (eg ~/.bash\_login), and you may want to copy the configuration file to your home directory ~/.cq2svn\_client.cfg, and use your personalised settings like this:

```
/opt/Clearvision/CQ2SVN/client/cq2svn_client -f ~/.cq2svn_client.cfg
```

You may also want to refer to the tools/run\_client.sh sample script for Unix platforms.

## 4.3 Setting permissions on Linux

On Linux, you will have to set permissions appropriately. Log files are written to the current working directory (ie wherever you run the process from), so this directory will require write-access. All of the executables, ie the client, server and all of the hook files will require execute permission. As an example, we've provided the tools/post\_install.sh script which will need to be customized, eg:

```
# This is an example post-installation script which you
# will need to customize according to your installation.
#
# Run this before copying the hook files to your Subversion repository.
# Also please make sure that the files and directories are readable.
CQ2SVN_HOME="/opt" # change this appropriately
chmod +x $CQ2SVN_HOME/Clearvision/CQ2SVN/client/cq2svn_client
chmod +x $CQ2SVN_HOME/Clearvision/CQ2SVN/server/cq2svn_server
chmod +x $CQ2SVN_HOME/Clearvision/CQ2SVN/subversion_hooks/*
# if you run the client and server in-place, you will need
# to make the directories writable for the log files:
chmod +w $CQ2SVN_HOME/Clearvision/CQ2SVN/client
chmod +w $CQ2SVN_HOME/Clearvision/CQ2SVN/server
```

## 4.4 Subversion and/or Git Hook Setup

Two CQ2SVN Hooks are required for each Subversion or Git repository which requires CQ2SVN to be enabled. The hooks and related programs are delivered in the C:\Program Files\Cleavision\CQ2SVN\subversion\_hooks and C:\Program Files\Cleavision\CQ2SVN\git\_hooks directory respectively. These need to be copied to your hooks directory, and may optionally require customization for your local setup. If you have local (non CQ2SVN) hooks to run, you may integrate these with CQ2SVN (see the .bat/shell files for more info). When a hook fires it calls the CQ2SVN\_hook executable, this in turn refers to the accompanying CQ2SVN\_hook.cfg file in order to know where the CQ2SVN server is running plus any other configurable options

edit the file {repository}\hooks\cq2svn\_hook.cfg and configure the following information in the [Server] section;

example

---

```
Server_ip=192.168.1.172
Server_port=54321
```

---


Change the values according.

Configurable options are;


- allow\_commit\_without\_change\_id=1
- allow\_anonymous\_commits=0
- allow\_commit\_on\_error=1
- log\_issue\_ids\_in\_svn=1
- log\_file={path}


Each of the above option must be configured according to requirements.

On unix platforms, please remember to make the hook files executable.

 If you customise these scripts, please be sure to correctly handle/pass-on errors.

## 4.5 NEW in 4.0: Portable Hooks

 **NEW in 4.0:** To use the portable hooks rather than executables, please replace `pre-commit` and `post-commit` with `portable-pre-commit` and `portable-post-commit`. Note that you **must rename** the portable-xxx files as otherwise they will be ignored by Subversion.

 Also note that, unlike the executables, the portable version requires a set of software tools to be installed on your system. Please refer to the Prerequisites section at the top of this document for details.

---

## 5 Customization

CQ2SVN can be customized according to your local requirements in a number of ways. This section details some of the enhancements that can be made locally by editing the perl scripts, and describes all of the configuration options.

### 5.1 Configuration Options

There are a number of options to control the behavior of CQ2SVN. The default options are configured to enable an **enforcement** policy. Using the options below, you can allow a more *permissive* policy for your system.

The options are set in the configuration files in a section called `[Options]`. There are options for the server, for the hooks and for the clients. The server configuration options control site-wide behavior of clients, such as allowing multiple records to be selected, or allowing configuration of the subversion user id. The hook configuration options control commit behavior for that repository such as permitting anonymous commits or allowing clients to not choose a change record for their changeset. The clients can set their user id for Subversion (if this is allowed by the server).

All of these options are detailed below.

#### 5.1.1 Server Configuration Options

##### 5.1.1.1 Allow Multiple CQ Records per Changeset

The default configuration pops up a GUI which allows the User to select **a single** CQ record to associate with the Subversion change-set. This policy can be changed site-wide by setting the `allow_multiple_choices=1` option in `cq2svn_server.cfg`, for example:

```
[Options]¶
allow_multiple_choices=1¶
```

##### 5.1.1.2 Allow Records to be Moved to the Next State

The default configuration pops up a GUI which allows the User to select a CQ record to associate with the Subversion change-set. Sometimes it's useful to let a user also move that record to it's next state (for example saying that development is complete, and a fix is ready to be tested). The `allow_next_state` option enables this functionality for all clients. Once enabled, the clients will be presented with an enhanced popup window that allows them to optionally progress the state of a record (please see the User Guide for an image of the GUI).

*PLEASE NOTE:* If you enable this feature, your Clearquest Schema must support a next default state for all of the issues that can be selected. This is particularly important if you have customized the `get_cq_list.pl` script to get records in a mixture of states. Also note that it will be impossible to move to the next state if the schema also requires a mandatory field to be completed at the same time.

To enable this feature, set the `allow_next_state` option to 1 in the `[Options]` section of `cq2svn_server.cfg`

```
[Options]¶
allow_next_state=1¶
```

See

[http://publib.boulder.ibm.com/infocenter/cqhelp/v7r0m0/index.jsp?topic=/com.ibm.rational.clearquest.schema.doc/t\\_d](http://publib.boulder.ibm.com/infocenter/cqhelp/v7r0m0/index.jsp?topic=/com.ibm.rational.clearquest.schema.doc/t_d)  
for more details as to how to set up default states.

### 5.1.1.3 Allow Subversion User Id Configuration on Clients

Normally a client will use the operating system's user id as its Subversion user id. Depending on your local policy and setup, you may need to allow clients to tell CQ2SVN that they have a different user id for Subversion.

To enable this configuration option on all clients, you can set the `allow_client_svn_uid_configuration=1` option in the *server* configuration file.

```
[Options]¶
allow_client_svn_uid_configuration=1¶
```

Then, a client can configure their `svn_uid` option (see below)

### 5.1.1.4 Enabling CQ2SVN Command Line Mode

CQ2SVN can scan for record id's in the Subversion log entered by the Subversion user. This is useful if you cannot or prefer not to use graphical applications such as the CQ2SVN client GUI.

You can turn this option on by adding the `allow_change_ids_in_svn_log=1` option to the `[Options]` section in your server config file (the default is off).

```
[Options]¶
allow_change_ids_in_svn_log=1¶
```

Record id's need to be provided at the beginning of the log message with no preceding white spaces and followed by a space character. Multiple record id's can be provided, separated by semicolons. The following examples are Subversion log messages that would result in [ClearQuest](#) records being updated:-

```
Test17 Fixed problem¶
¶
Test12;Test20 Fixed two problems¶
```

Note that if this option is enabled and you specify a log message that starts with a word that looks like a [ClearQuest](#) record id, then you can prevent CQ2SVN from scanning the log message by starting your log message with a space (or any other character):-

```
Java2 is a great platform and fixes this problem.¶
¶
This problem is fixed by using the Java2 platform.¶
```

Both examples above work fine as CQ2SVN will not consider Java2 a [ClearQuest](#) record id as it does not appear at the beginning of the log message. Note that if you specify a [ClearQuest](#) record id at the beginning of the log message that is not part of the set of issues available in [ClearQuest](#), CQ2SVN will give an error.

Note that with the `allow_change_ids_in_svn_log` option it is possible to run CQ2SVN without a client. When running without a client, CQ2SVN will report errors if the information provided in the log message is not sufficient. For example, if CQ2SVN is configured to require a record id with every commit, and no id is provided in the log message, CQ2SVN will raise an error.

### 5.1.1.5 Configure Connection Timeout

When network problems occur, clients may not be able disconnect cleanly from the server, causing the server to believe that clients are still connected and wait for a response.

With the `connection_timeout` setting in the `Options` section, you can configure the time in seconds before the server considers a connection broken.

Setting the value too low can cause clients to be disconnected on slow systems. A value that is too large can cause commits too hang until the server recognises the broken client connection.

Note that connections are checked regularly and in most cases a reasonable large timeout such as 30 or 60 seconds should be suitable.

```
[Options]¶
connection_timeout=30¶
```

### 5.1.1.6 Time Zone Configuration

You can configure the time zone name and offset for commits when they are logged in ClearQuest. By default, all commits are logged with a UTC time code.

To choose a different time zone, specify the name of the time zone in `svn_timezone_label` and configure the offset from UTC (Coordinated Universal time) in hours. For example, for CET specify `svn_timezone_offset=+1` or for PST specify `svn_timezone_offset=-7`.

```
[Options]¶
svn_timezone_offset=+1¶
svn_timezone_label=CET¶
```

## 5.1.2 Hook Configuration Options

### 5.1.2.1 Allow Commit Without Change Id

A user who is logged in to the CQ2SVN system normally must choose a record to go with their changeset. If they cancel their choice, CQ2SVN will abort the Subversion commit.


If you want to allow them to optionally **not** select any of their current change records and 'cancel' their choice in the GUI (ie allow commits **without any change record**, you can set this option.

 **WARNING:** Enabling this will mean that **some changes are NOT recorded in your change control system!**

```
[Options]¶
allow_commit_without_change_id=0¶
```

### 5.1.2.2 Allow Anonymous Commits

Default behaviour is to check that a user has logged on to the CQ2SVN Server, and to abort any commits from users who are not yet logged on (**a policy-'enforcing' mode**). You can disable this behaviour here (for example if some of your users do not use change records)

 **WARNING:** this will mean that those users not running a Client will NOT be using this system to correlate Subversion with your change control system, and effectively bypassing CQ2SVN altogether!

```
[Options]¶
allow_anonymous_commits=1¶
```


### 5.1.2.3 Allow Commit On Error

Normally, if an error occurs in the system, CQ2SVN will try to abort the commit. If you want to let the commit proceed (for example as a temporary measure until you fix a system problem), you can set this option:

```
[Options]¶
allow_commit_on_error=0¶
```

Note that this is *not* guaranteed to work as it depends on the exact problem that is occurring, so, please use with caution.

### 5.1.2.4 NEW in 4.0: Write ClearQuest Record Ids to Subversion Log

 **New in v4.0** You can configure CQ2SVN to add [ClearQuest](#) record ids to the subversion log message. If you enable this option and commit a change, CQ2SVN will add the ids of all records chosen by the user to the front of the log message.

This makes it possible to see from the subversion log which records a change-set relates to.

```
[Options]¶
log_issue_ids_in_svn=1¶
```

Please note this feature is not supported with Git repositories.

## 5.2 Perl Script Customization

### 5.2.1 Change the record selection criteria

The standard setup selects all CQ records owned by the User which are in an 'Assigned' state. You can change this by modifying the Clearquest Query in the `get_cq_list` function in the `get_cq_list.pl` script.

The default snippet of code looks like this:

```

..... # filter the results on:
... @ownertest = "$cq_userid"; #..... (owner == $cq_userid)
... $rootfilternode->BuildFilter("Owner", $CQPerlExt::CQ_COMP_OP_EQ,
\@ownertest);
... @statetest = "Assigned"; # AND (state == 'Assigned')
... $rootfilternode->BuildFilter("State", $CQPerlExt::CQ_COMP_OP_EQ,
\@statetest);

```

Suppose we wanted to show all of the CQ records owned by that User, in a 'Pending' State, we could modify the snippet to look like this:

```

..... # filter the results on:
... @ownertest = "$cq_userid"; #..... (owner == $cq_userid)
... $rootfilternode->BuildFilter("Owner", $CQPerlExt::CQ_COMP_OP_EQ,
\@ownertest);
... @statetest = "Pending"; # AND (state == 'Pending')
... $rootfilternode->BuildFilter("State", $CQPerlExt::CQ_COMP_OP_EQ,
\@statetest);

```

Thus you can see how you have the flexibility to use any query available through the Clearquest API to define the list of CQs to display.

## 5.2.2 Change the CQ details displayed

The contents of the CQ2SVN Selection box (which pops up to ask the User to choose a record) can be customized. The default is to show the CQ id and the headline fields. You can change this as you wish to include any of the fields that are in the CQ record. For example, suppose that we want to add the Priority field after the headline. Edit the `get_cq_list.pl` script to add an extra field to the Clearquest query, so the code snippet :

```

... $querydef = $session->BuildQuery($record_type);
... $querydef->BuildField("id");
... $querydef->BuildField("headline");

```

can be extended to:

```

... $querydef = $session->BuildQuery($record_type);
... $querydef->BuildField("id");
... $querydef->BuildField("headline");
... $querydef->BuildField("priority");

```

and the later snippet:

```

... while ($resultset->MoveNext() == $CQPerlExt::CQ_SUCCESS) {
..... $id..... = $resultset->GetColumnValue(1);
..... $headline = $resultset->GetColumnValue(2);
..... $headline =~ s// /g;..... # detail text must not

```

```

contain semi-colons
..... $cq2svn_choice = "$id:$headline;";
..... $cq_list .= $cq2svn_choice;
... }

```

modified to add the priority field like this:

```

... while ($resultset->MoveNext() == $CQPerlExt::CQ_SUCCESS) {
..... $id..... = $resultset->GetColumnValue(1);
..... $headline = $resultset->GetColumnValue(2);
..... $priority = $resultset->GetColumnValue(3);
..... $headline =~ s/;/ /g;..... # detail text must not
contain semi-colons
..... $cq2svn_choice = "$id:$headline:$priority;";
..... $cq_list .= $cq2svn_choice;
... }

```

and finally, declare the new \$priority variable before it is used by adding it as follows:

```

sub get_cq_list {
... my ($cq_userid, $cq_superuser_id, $cq_superuser_password, $cq_db,
$cq_dbset) = @_;
... my $priority;

```

So, as long as you maintain the format of the final list (which is a semi-colon-separated list of items; each item can have colon-separated fields within it), CQ2SVN will handle it.

### 5.2.3 Recording all changes against a default CQ record

Occasionally, projects enter a period when there are no legitimate Clearquest records, for example, at the beginning of a project. In such cases, the Subversion change sets can be recorded against a single, dummy record. This can be achieved by uncommenting the following code in `get_cq_list.pl` and customizing appropriately:


```


... # If you have a business policy to record all changes against a
default record
... # in Clearquest, you may uncomment the following (with appropriate
customization).
... my $default_cq = "SAMPL00000001";
... my $default_cq_text = "you have no CQ records assigned, so please
use this default";
... if ($cq_list eq "") {
..... return "$default_cq:$default_cq_text";
... }

```

## 6 Troubleshooting

To check that a CQ2SVN client or server is running on Windows, open the Windows Task Manager. You should see 2 entries for the server (on a server machine) and 2 for the client (on a client machine). To kill the process, right click on the process in the Processes tab and select 'End Process Tree' (to kill both processes at once). If the process is not running, check your log file for any errors. On Linux, use `ps -ef | grep cq2svn` - these processes can be safely killed.

 To investigate setup problems, it can be helpful to increase the logging level from the default of 'Warning' to 'INFO' (see `cq2svn_*.cfg`)

 Note that when CQ2SVN detects an error; the default behavior is to *abort* the commit. Some possible reasons are:

1. When your User Id is not recognized as a Clearquest User Id (you may want to set `svn_uid`) in your client's `cq2svn_client.cfg` file to override the default user id: note that this id must be usable in both Subversion and Clearquest)
2. When the user is not logged on to CQ2SVN (check that `cq2svn_client.exe` is running in Task Manager or `ps` on Linux)
3. When there are no valid Clearquest records assigned to the user (or there are no eligible records according to the new criteria you've implemented in `get_cq_list.pl`) (run `get_cq_list.pl` as above with the userid and check the results.).
4. When Subversion `svnserve.conf` file uses `anon-access = write`

If you have a local repository and are using the file protocol via the command line client, your commits will **not** use your windows logon id, they will be *anonymous*. So, to work with CQ2SVN you must use something like:

```
svn commit myfile.txt --username engineer
```

Note that you can use this command, but Subversion *still works anonymously* if you have asked it to (eg using `svnserve` with `anon_access=write`).

The perl and config files use *unix* line-endings, so please be careful to use a well-behaved text editor such as [Cream/Vim](#) or Wordpad rather than Notepad or MS Word.

### 6.1 Installation Support

If you experience any issues during the installation please report your issues via the Clearvision Support Centre, which can be found at <http://www.clearvision-cm.com/support-home/support-center.html>.

---

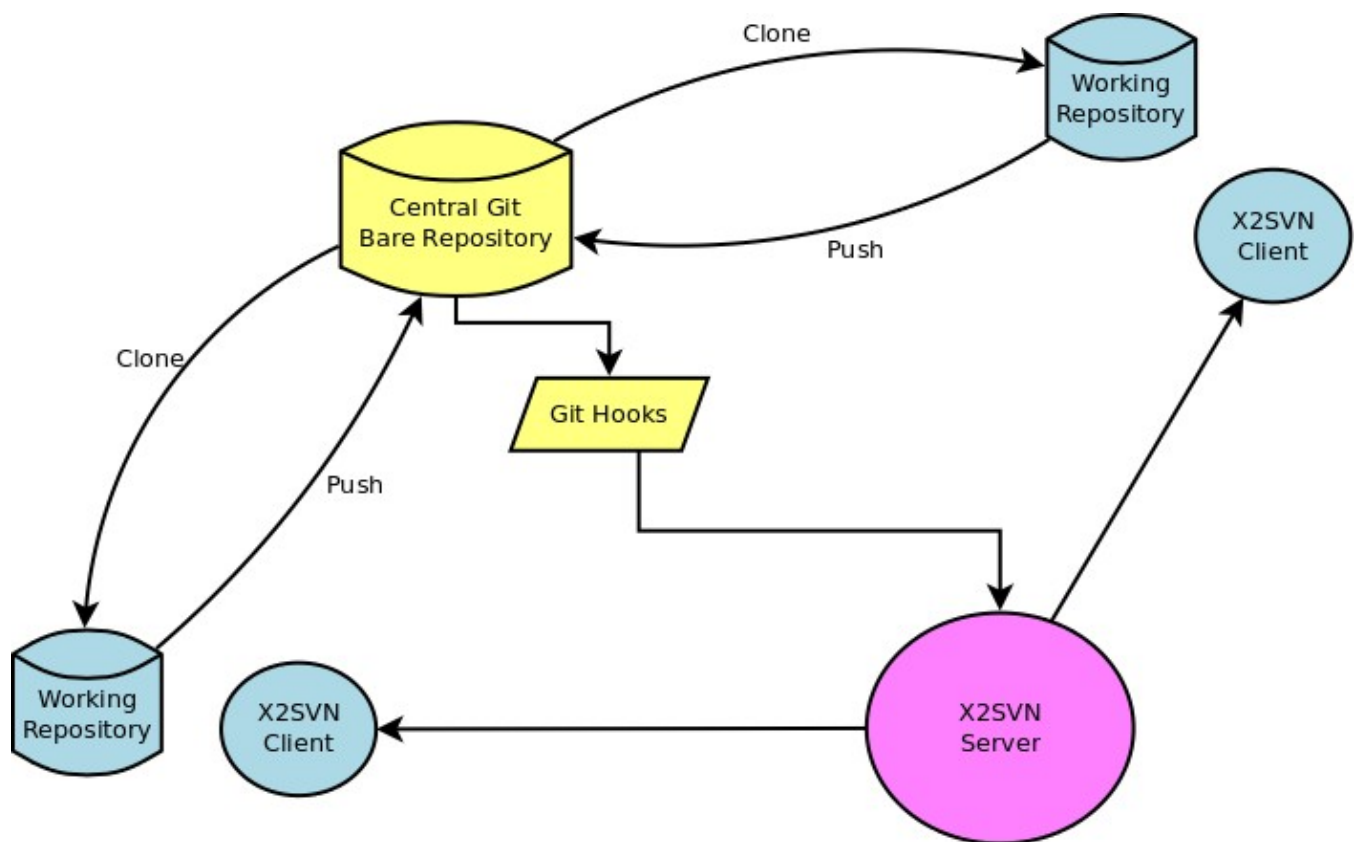
# 7 Appendix A: Using CQ2SVN with Git

## 7.1 Overview

The X2SVN product suite, which currently provides solutions for CQ2SVN, Jira2SVN, CodeBeamer2SVN and Trac2SVN, has been extended to support Git managed repositories.

The principle for Git repositories follows along the same lines as the Subversion-based solution. Hooks are installed in the Git repository which fire on changes to the repository and prompt the user to select the records to attach to the new revision.

Each user runs a client process, which prompts the user for a record id when changes are applied to the repository. Note that even if you work with multiple clones of the same Git repository, you only need to run one client process as the connection of the client is with the server for the central repository and not your clone.



There are some notable differences between Subversion and Git when integrating with change management systems using the Clearvision X2GIT product suite. The most notable change is that with Git you will not be prompted for a change record to attach on every commit. It is not until the point that changes are pushed back into a central repository that they become visible to users of the same repository.

It is therefore on the `push` operation that the X2GIT solution triggers and prompts the user to select a record to attach to the change-set being pushed to the server.

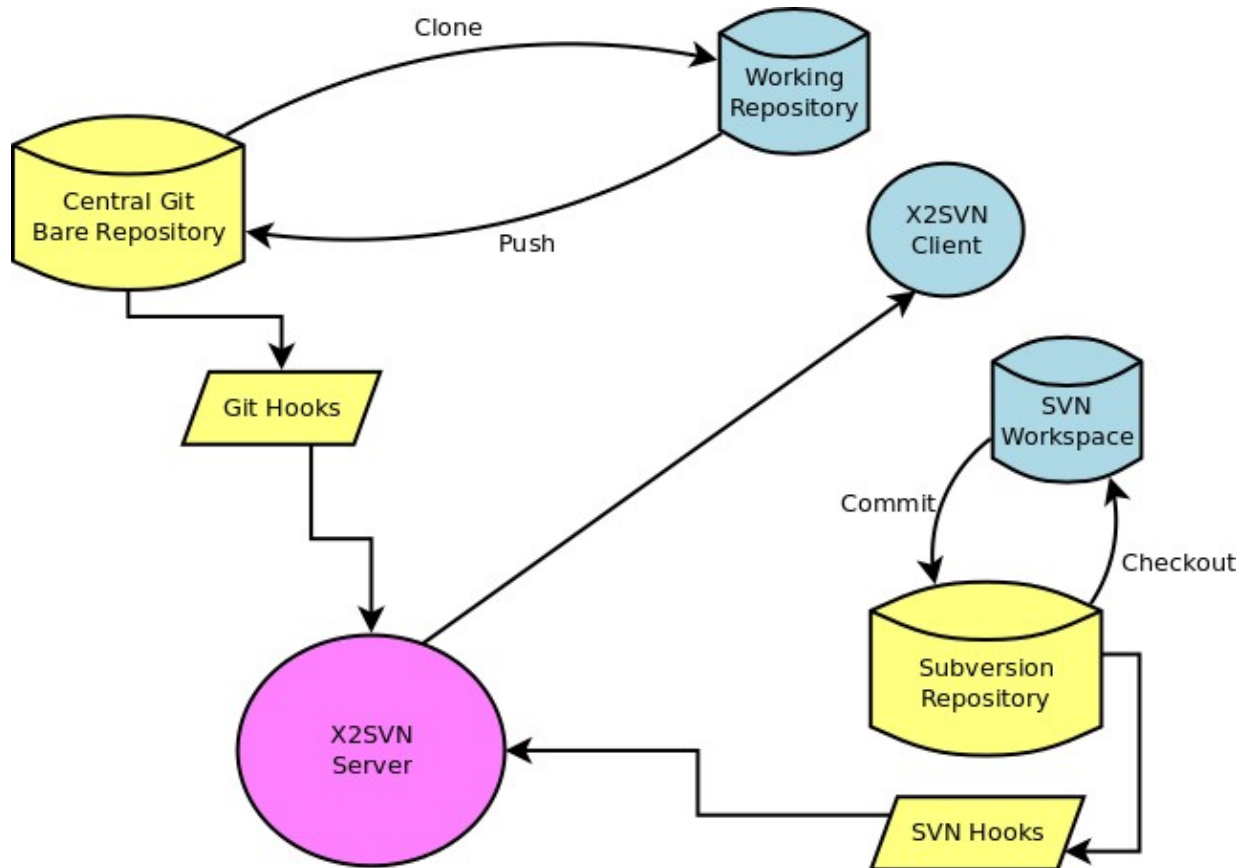
If you are pushing a number of commits in one operation, you will only receive one pop-up, and the selected change record(s) will be associated with all the commits being pushed.

Note that due to the nature of the products, the Git integration is only useful in environments where users push into central repository. For more distributed development environments with informal sharing of repositories

it would not be appropriate to connect these informal and potentially transient repositories to a central change management repository.

## 7.2 Heterogeneous Environments

The same server process can be used as the target for Git as well as Subversion hooks. This makes it possible for developers to use the same client process to receive pop-ups for pushes to the central repository on the Git side and commits to a Subversion repository on the Subversion side.



Note that in this scenario the same server process is being used, meaning that Subversion commits and Git pushes will present the user with the same list of issues to attach to a change-set. If you need to use a different list of issues or connect to a different change management repository, you will need to use multiple server processes.

## 7.3 Installation

Please install the server and client elements of the product as per the normal installation instructions. Once the product is installed, you can install the Git hooks in a bare repository as follows.

- Copy the contents of the hook files from the installation directory `<install_dir>/subversion_hooks` to the Git repository `<git_repo_dir>/hooks`.
- Configure the hooks by editing the config file and changing the `server_ip` setting to point to the correct server.
- There are a number of other hook config options which are described above. Note that the options are the same as for the Subversion hooks.

## 8 Appendix B: RHEL4 Pre-requisites

### 8.1 Subversion 1.4.4

CQ2SVN is built against Subversion 1.4.4. Pre-built RPM packages for Subversion and its dependencies are available from the following site. Note that this is **not** part of the Clearvision product, but is required for CQ2SVN to run. The following site is referenced from [the standard Subversion download page](http://the.earth.li/pub/subversion/summersoft.fay.ar.us/pub/subversion/latest/rhel-4/i386/):

- <http://the.earth.li/pub/subversion/summersoft.fay.ar.us/pub/subversion/latest/rhel-4/i386/>
  - ◆ apr-0.9.12-2.i386.rpm
  - ◆ apr-util-0.9.12-1.i386.rpm
  - ◆ mod\_dav\_svn-1.4.4-1.i386.rpm
  - ◆ subversion-1.4.4-1.i386.rpm
  - ◆ subversion-perl-1.4.4-1.i386.rpm
  - ◆ subversion-python-1.4.4-1.i386.rpm
  - ◆ subversion-tools-1.4.4-1.i386.rpm

To install these rpms after downloading, run `rpm -Uvh <rpmname>` on each rpm file.

## 9 Appendix C: Portable Hooks Examples

### 9.1 Sun Solaris 10

This appendix gives guidance on installing prerequisite tools for portable hooks on a Solaris 10 system. Please note that the instructions below are for *guidance only* and your system may require different steps.

#### 9.1.1 Set path to pick up GNU tools

```
export PATH=/usr/ccs/bin:/usr/sfw/bin:$PATH
```

#### 9.1.2 Python

Python 2.4 is available on the Solaris CD, and other versions (including python 2.5) are available from Sunfreeware.com. These use the standard Sun pkgadd. We would recommend using python 2.5 if possible.

#### 9.1.3 Twisted 8.2.0

⚠ This package has long file names, therefore it **requires 'gtar'** to be unpacked.

```
python setup.py install
```

##### 9.1.3.1 Zope 3.3.1

Twisted requires the interfaces package in Zope. ⚠ This package has long file names, therefore it **requires 'gtar'** to be unpacked.

```
./configure --with-python=/opt/python/bin/python # use your python  
location here.¶  
make¶  
make install¶
```

#### 9.1.4 PySVN 1.6.2

Note that PySVN **requires** GNU's version of make, "gmake".

The version of gcc that we used had a problem with finding the right linker. GCC has [hard-coded search paths](#) (for as, ld, etc.). Our version was missing the path to the GNU linker, but had the Sun linker, so failed with an invalid (GNU-specific parameter, ie --rpath). To work around this problem, we moved /use/ccs temporarily (see below)

```
cd Source¶  
python setup.py configure¶  
mv /usr/ccs /usr/ccs.BAK... # see above¶  
gmake¶  
cp -r pysvn /opt/python/lib/python2.5/site-packages/... # use your python  
location here.¶
```

```
mv /usr/ccs.BAK /usr/ccs¶
```

## 10 Appendix D: Portable Client Example

### 10.1 Installation of CQ2SVN portable client on a Mac (OS-X Leopard)

#### 10.1.1 Requirements

- [Subversion 1.5](#) - including the python SWIG bindings
- [CQ2SVN Python “portable” version](#)
- [MacPorts](#) (latest snapshot)
- [Xcode](#) (Apple’s Xcode developer tools as appropriate for OS-X release)
- X11 windowing environment (A.K.A “X11 user) in Mac
- [Python 2.5.x](#)
- [wxPython](#)
- [wxWidgets](#)
- [Zope](#)
- [Twisted](#)

#### 10.1.2 Installing X11

The X11 windowing environment is installed by default in Snow Leopard and Leopard releases of OS-X. For Tiger release, it is available on the installation media, but must be either installed as an optional package during OS installation OR added on after installation.

#### 10.1.3 Installing Xcode

Xcode is the free developer toolkit from Apple. It requires registering with the Apple developer network (which is free) before it will permit downloading of the installer. NOTE: You will need version 3.2 of XCode to install on Snow Leopard, version 3.1 for Leopard, or version 2.4.1 for Tiger releases). Installation instructions are provided on the XCode site at <http://connect.apple.com>.

#### 10.1.4 Installing MacPorts

[MacPorts](#) is an initiative that aims to bring 99% of the most popular \*nix packages to Darwin (OS-X’s BSD based core). It can be installed from source via downloading of source in tar or from Subversion, it is also distributed as a binary Mac OS-X .DMG file specific to what release of OS-X (Snow Leopard, Leopard, and Tiger) the client is running. Installation instructions are located here: <http://www.macports.org/install.php>

NOTE: The remainder of the instructions require [MacPorts](#) to be installed first.

#### 10.1.5 Installing Python and other requirements

1. Open a terminal window
2. Obtain root access (via su or you will need to sudo each of the following steps)
3. Install Python – Python **MAY** already be installed, if not, use ‘port’ to install python

◆ `(sudo) port install python25`

4. Install `wxWidgets` (this installs `wxPython` as a dependency)

◆ `(sudo) port install wxWidgets`

5. Install Zope

◆ `(sudo) port install Zope`

#### 6. Install Twisted

◆ `(sudo) port install py25-twisted`

At this point, all of the required dependencies should be installed. You can proceed to the actual CQ2SVN portable version install at this point.

Installing CQ2SVN portable:

1. Unzip the Platform Independent zip file to a temporary location
2. Move the expanded “Clearvision” folder to a location that is accessible by all users of the system. We standardized on /opt
3. Create a client configuration file by hand or using the new generator provided in 5.0d with appropriate values. This can reside wherever is appropriate for your environment (i.e., in the user’s home folder or system-wide for all users)
4. Create a startup shell script with the content below in a location within the path

```
# Run the CQ2SVN portable client¶
CQ2SVN_HOME="/opt"¶
export PYTHONPATH="/opt/Clearvision/cq2svn/lib"¶
¶
#ensure that .cq2svn_client config file exists in your home directory¶
python
$CQ2SVN_HOME/Clearvision/cq2svn/lib/cq2svn/client/cq2svn_client.pyc -f
~/cq2svn_client¶
```

The above code assumes the Clearvision folder is placed in /opt and that each user of the system would have their own private config file in his/her home folder.

1. Change permissions on the script to make it executable as appropriate to your environment
  - ◆ `chmod a+rx <path to script>/cq2svn_client.sh`
  - ◆ Or `chmod 755 <path to script>/cq2svn_client.sh`
2. Ensure permissions are suitable on Clearvision folder
  - ◆ `chmod -R 755 /opt/Clearvision`

The script mentioned above can also be used directly within ‘automator’ to create a dock icon or to launch the client manually as needed/wanted. Or, it can be scheduled to start automatically upon login as well.

```
Copyright (c) 2007, 2008, 2009 Clearvision CM All Rights Reserved¶
Registered in England No. 5643578 VAT Registration No. UK881364801¶
Registered Address Laurel Farm, Winters Hill, Durley Southampton, S032
2AH¶
```

<http://www.clearvision-cm.com>