



*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

1 Compatibility	1/9
1.1 What is the minimum version of Clearquest you need to use CO2SVN?	1/9
1.2 We are using Subversion client 'x'. Will we need to use something else to work with CO2SVN?	1/9
1.3 How can I use CO2SVN with an existing project using Clearquest and Subversion?	1/9
1.4 Does CO2SVN integrate with SAP?	1/9
1.5 We have two types of Clearquest records that we need to associate Subversion commits with. They are different records with different fields and, most importantly, a different state transition matrix. Can you support that?	1/9
2 General Functionality	2/9
2.1 Does CO2SVN consume a ClearQuest license	2/9
2.2 Are the changes to Subversion and Clearquest atomic?	2/9
2.3 Can you make a connection after the original commit?	2/9
2.4 How can the list of performed changes and file version history be reported via output/printout?	2/9
2.5 What happens when a user (who does not have CO2SVN installed) tries to commit to a repository with the integration enabled?	2/9
2.6 We use Subversion check-in as part of our automated build process. Is there a way to declare certain users exempt from having to use the CO2SVN process?	2/9
2.7 Is there a way to centralise the configuration file to prevent users from modifying options such as 'commit on error'?	2/9
2.8 Can I use LDAP with Subversion and CO2SVN?	2/9
2.9 Can my ClearQuest user name be different from my Subversion user name?	3/9
2.10 I have a number of projects in my Subversion repository. Can I install CO2SVN such that it only runs for some projects and not others?	4/9
3 Geographic Distribution	5/9
3.1 Our users interface with Clearquest via the web and do not have the Clearquest desktop client installed. Is this a problem?	5/9
3.2 We are running in a geographically distributed environment. Are there performance issues associated with off-shore access?	5/9
3.3 How does CO2SVN work if you have multiple Clearquest repositories geographically, each using a different process?	5/9
4 Security	6/9
4.1 Which protocols are used to communicate between the CO2SVN client and the server – are these secure?	6/9
4.2 What firewall changes do we need to make to allow communication to occur between segregated networks?	6/9
5 Record Type Families	7/9
5.1 What is a Record Type Family?	7/9
5.2 Why would I want to do this?	7/9
5.3 What are the requirements of creating a Record Type Family?	7/9
5.4 Why do some special characters from my commit message not appear correctly in ClearQuest?	7/9
6 Troubleshooting	8/9
6.1 Connection Refused Error	8/9
6.2 How can I automatically restart clients when they shut down?	8/9

1 Compatibility

1.1 What is the minimum version of Clearquest you need to use CQ2SVN?

CQ2SVN uses the Clearquest Open API to communicate between Clearquest and Subversion. To date, we are supporting CQ2003 and Version 7.1 of Clearquest. Because the API has not changed since version CQ2001, CQ2SVN is expected to work on versions up to and including this version.

1.2 We are using Subversion client 'x'. Will we need to use something else to work with CQ2SVN?

No, CQ2SVN is 100% client-independent meaning you can carry on using your favorite Subversion client. CQ2SVN installs one small executable and configuration files on the user's machine. No operating system or client modifications are required.

1.3 How can I use CQ2SVN with an existing project using Clearquest and Subversion?

This is very easy to achieve. Upon setup, CQ2SVN will dynamically query your Clearquest database as if it was installed from the initiation of your project.

1.4 Does CQ2SVN integrate with SAP?

At present, we do not have an SAP integration.

1.5 We have two types of Clearquest records that we need to associate Subversion commits with. They are different records with different fields and, most importantly, a different state transition matrix. Can you support that?

You can customise the Clearquest side as you like. We supply 2 sample perl scripts for the Defect Tracking schema. If you want to use another configuration, you can customize these scripts freely.

2 General Functionality

2.1 Does CQ2SVN consume a ClearQuest license

CQ2SVN uses the approved API calls to extract and update Clearquest records. These tasks are being performed by the CQ Administrator and in our tests, a license is NOT consumed. IBM has stated this is a bug in Clearquest and customers should accommodate for utilizing a license in the future.

2.2 Are the changes to Subversion and Clearquest atomic?

Because CQ2SVN bridges the gap between two completely separate databases, it would theoretically be possible for a crash to cause a write to Clearquest and not to Subversion. However, the probability of this happening is minimal.

2.3 Can you make a connection after the original commit?

There are certain operations which can be modified after a Subversion commit. With regards to the CQ2SVN integration with Clearquest, the fields in Clearquest can be modified by a privileged user (typically your Clearquest administrator).

2.4 How can the list of performed changes and file version history be reported via output/printout?

From the Subversion side, CQ2SVN writes into a Clearquest record ID in a revision property. There are various Subversion commands which allow us to query and report. Clearquest is very similar; with CQ2SVN writing the Subversion change set and Subversion revision ID into a 'Subversion' tab. This can be easily queried and reported using Clearquest's reporting functionality.

2.5 What happens when a user (who does not have CQ2SVN installed) tries to commit to a repository with the integration enabled?

CQ2SVN will allow the administrator to control the behaviour and allow the commit to pass through or fail.

2.6 We use Subversion check-in as part of our automated build process. Is there a way to declare certain users exempt from having to use the CQ2SVN process?

Yes, anyone who is not logged on via CQ2SVN will automatically be excluded from choosing a Clearquest record to associate commits with.

2.7 Is there a way to centralise the configuration file to prevent users from modifying options such as 'commit on error'?

There are two different configuration files. CQ2SVN 'client' manages issues such as the compatibility of the user's name (Subversion and Clearquest require this). The server config file holds the value regarding 'commit on error'. This server config file is central and holds anything that is controversial and should not be changed by users.

2.8 Can I use LDAP with Subversion and CQ2SVN?

In principle yes. However, on occasion there may be problems with LDAP user names being mixed-case (e.g. UserName) and ClearQuest user names being all lower-case (e.g. username).

If you come across this problem, you can customise `get_cq_list.pl` and convert mixed-case user names into lower-case user names for ClearQuest as follows:-

```
[...]
sub get_cq_list {
  ... my ($cq_userid, $cq_superuser_id, $cq_superuser_password, $cq_db,
$cq_dbset) = @_;
}
... # BEGIN CUSTOMISATION: Convert user names to lower case
... $cq_userid = lc $cq_userid;
... # END CUSTOMISATION
}
... my $session;
... my $querydef;
[...]
```

2.9 Can my ClearQuest user name be different from my Subversion user name?

At the moment, with CQ2SVN out-of-the-box, if you want to be able to use ClearQuest reports based on user ids, Subversion and ClearQuest user id's have to match.

However, you can customise `get_cq_list.pl` and map Subversion user names to ClearQuest user names.

Add the following code to the bottom of `get_cq_list.pl`:-

```
sub get_cq_user_name {
  ... my ($svn_user_name, $mapping_file_path) = @_;
}
... open FILE, $mapping_file_path or die $!;
... while (<FILE>) {
  ... $line = $_;
  ... if ($line =~ m/^(#[^#]*)=(.*)$/) {
  ... $old_user_name = $1;
  ... $new_user_name = $2;
  ... if ($old_user_name eq $svn_user_name) {
  ... return $new_user_name;
  ... }
  ... }
  ... }
... }
... print "Error: Cannot map Subversion user name '$svn_user_name' to a
ClearQuest user name\n";
... # Return svn_user_name as a fallback (please customise if you want
different behaviour)
... return $svn_user_name;
}
}
```

Then in the subroutine 'get_cq_list', customise the code as follows:-

```
[...]
sub get_cq_list {
  ... my ($cq_userid, $cq_superuser_id, $cq_superuser_password, $cq_db,
$cq_dbset) = @_;
}
... # BEGIN CUSTOMISATION: Map Subversion user name to ClearQuest user name
... my $mapping_file_path = "c:\cq_user_mapping_file.txt";
... $cq_userid = get_cq_user_name($cq_userid, $mapping_file_path);
... # END CUSTOMISATION
}
```

```

$
... my $session;$
... my $querydef;$
[...]$

```

Then write a user mapping file and store it in the location specified in above customisation, e.g. 'c:\cq_user_mapping_file.txt' in this case. The mapping file should look as follows:-

```

# User mapping file for CQ2SVN$
# Syntax: subversion_user_name=cq_user_name$
# One entry per line$
fred=paul$
gerry=matt$
matt=andy$
ashley=martin$

```

2.10 I have a number of projects in my Subversion repository. Can I install CQ2SVN such that it only runs for some projects and not others?

Yes you can. When you install CQ2SVN Subversion hooks, you need to change the pre and post-commit hooks such that CQ2SVN is only called for commits that relate to a particular locations in the Subversion repository.

A special Subversion command `svnlook` can be used in the hooks to retrieve information about what the user has committed, and you can then make decisions based on that information.

On a Linux system, you could write a hook as follows:-

```

cd $1/hooks$
CHANGES=`svnlook changed· $1 --transaction $2`$
FILES=`sed "s/^\...// " <<EOF$
$CHANGES$
EOF$
`$
$
RUN_CQ2SVN="false"$
$
for FILE in $FILES; do$
... case $FILE in$
..... # Add your repository locations for which to run CQ2SVN hooks here$
..... trunk/proj1/*) RUN_CQ2SVN="true";;$
..... trunk/proj2/*) RUN_CQ2SVN="true";;$
..... trunk/proj4/*) RUN_CQ2SVN="true";;$
... esac$
done$
$
if [ "$RUN_CQ2SVN" == "true" ]$
then$
... $1/hooks/cq2svn-pre-commit $1 $2$
fi$

```

3 Geographic Distribution

3.1 Our users interface with Clearquest via the web and do not have the Clearquest desktop client installed. Is this a problem?

No. The client does not require any parts of Clearquest to be installed.

3.2 We are running in a geographically distributed environment. Are there performance issues associated with off-shore access?

In comparison to registering Subversion commits via a Clearquest record manually, CQ2SVN will obviously speed up the process in comparison with opening the Clearquest web client for example.

The amount of information that travels through communication channels is very small and a lot of this process is performed at the server end. As with a lot of multi-sited software, there will always be slight delays, but one should measure these in milliseconds rather than multiple seconds.

3.3 How does CQ2SVN work if you have multiple Clearquest repositories geographically, each using a different process?

Each server instance of CQ2SVN can only point to a single Clearquest instance. It is possible to have a single Subversion repository call one of many CQ2SVN servers based upon either a username filter or a directory/file filter.

4 Security

4.1 Which protocols are used to communicate between the CQ2SVN client and the server – are these secure?

CQ2SVN uses TCP. We believe CQ2SVN is secure, working with Clearquest record ID's and file names rather than raw data. Further to this, you can configure the port that the CQ2SVN server listens on, and ensure this is the only incoming port that is required to be accessible by the client.

4.2 What firewall changes do we need to make to allow communication to occur between segregated networks?

CQ2SVN allows for routing on any user defined port. The configuration files on the client and the server require a port number in order to communicate and would involve opening any relevant firewalls first.

5 Record Type Families

5.1 What is a Record Type Family?

A Record Type Family is a way of logically grouping together [ClearQuest](#) Record Types.

5.2 Why would I want to do this?

The main reason/advantage for creating a Record Type Family is that they give greater flexibility when you are querying your [ClearQuest](#) user database. When you define a [ClearQuest](#) query you must always specify the record type that the you are querying - therefore you are normally only able to query one record type at a time. However, a record type family can also queried and when you query a family the query will include all record types that are members of the family.

```
Example¶
¶
Two Record Types exist;¶
- Defect¶
- Enhancement¶
¶
- A Record Type Family is created called;¶
ChangeRecords¶
¶
- Both the Defect record type and the Enhancement record type are added as
members of the ChangeRequest family;¶
¶
ChangeRequests¶
... |__Defect¶
... |__Enhancement¶
¶
- Now any query defined to query the ChangeRequest family will automatically
query both Defect records and Enhancement records.¶
```

5.3 What are the requirements of creating a Record Type Family?

When creating a Record Type Family you must consider;

- The record types included in a family must contain one or more common fields, that is to say, fields that are the same in each record type. These common fields are the only fields that can be used when you query the family.
- Because record types and record type families appear in the same selection list when you click on Query > New Query, it is a good idea to use a naming convention that helps users distinguish individual record types from record type families.

5.4 Why do some special characters from my commit message not appear correctly in ClearQuest?

When using different character sets, in particular when committing changes on a Linux platform to a ClearQuest server on Windows, some special characters placed in the commit message may not be translated correctly and appear differently in ClearQuest.

Please note that this does not cause any problems and is only an issue with the display of these special characters.

6 Troubleshooting

6.1 Connection Refused Error

I'm getting the following error, what's the problem?

```
connection failed for reason:␣
[Failure instance: Traceback (failure with no frames):␣
<class 'twisted.internet.error.ConnectionRefusedError'>:␣
Connection was refused by other side:␣
10061: Unknown error.␣
```

This is usually because there is a firewall blocking the port.

You may have a machine-specific firewall service protecting either the Client, Server or both machines. The easy way to test this is to disable any firewall services (such as [Windows Firewall](#)) that are enabled and run CQ2SVN again. If this works, re-enable your firewall and try opening the port that you have configured (remember the client must talk to the server and the server has to be able to talk back to the client).

Other than machine-specific firewalls, you may also have a corporate network firewall which is monitoring traffic between 2 remote networks (eg you have a developer in Chicago who is trying to use a server in London). Consult your network security group for more information.

6.2 How can I automatically restart clients when they shut down?

Clients automatically shut down if they detect a broken connection to the server. Common causes for clients to shut down are:-

- The server was terminated.
- The client machine's IP address changed, e.g. because the user opened a VPN (Virtual Private Network) connection or switched between wired and wireless networks.
- Network cables being temporarily unplugged.

If you are a laptop user who regularly switches between wired and wireless networks or if you are a regular user of VPN, you may want to write a script that restarts the client when it shuts down.

On Windows, such a script would look as follows:-

```
@ECHO OFF␣
:START␣
... start /wait "c:\Program Files\Clearvision\cq2svn\client\cq2svn_client.exe"␣
... REM Wait for five seconds before restarting the client␣
... ping -n 6 127.0.0.1 > nul␣
... ECHO Restarting client␣
GOTO START␣
```

On Linux, the following script would restart the client automatically:-

```
while (true)␣
do␣
... /opt/clearvision/cq2svn/client/cq2svn_client␣
... # Wait for five seconds before restarting the client␣
... sleep 5␣
... echo "Restarting client"␣
done␣
```

Depending on preference, you may want to consider features such as limiting the number of reconnection attempts within a certain time frame (e.g. if it has tried to reconnect more than 5 times within the space of a minute, then wait a few minutes before trying again).