



Administration Guide

Unified Change Management for Subversion

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

Copyright (c) Clearvision CM 2009

Contents

<u>1 Introduction</u>	1/13
<u>2 Prerequisites</u>	2/13
<u>2.1 Server Platform</u>	2/13
<u>2.2 Client Platform</u>	2/13
<u>2.3 License</u>	2/13
<u>3 UCM4SVN Setup</u>	4/13
<u>3.1 Create UCM4SVN Userid and Group</u>	4/13
<u>3.2 Install Files</u>	4/13
<u>3.3 Configure the Server Settings</u>	4/13
<u>3.3.1 ucm4svn_dir</u>	5/13
<u>3.3.2 License</u>	6/13
<u>3.3.3 JIRA Settings</u>	6/13
<u>3.3.4 Enable XML-RPC</u>	7/13
<u>3.3.5 JIRA Workflow Matrix</u>	7/13
<u>3.3.6 ClearQuest Settings</u>	8/13
<u>3.3.7 Logging</u>	11/13
<u>3.4 Starting the Server</u>	11/13
<u>4 Creating A Sample Database for Evaluation/Testing</u>	12/13
<u>5 Troubleshooting</u>	13/13
<u>5.1 Emergency Database Access</u>	13/13
<u>5.2 Resetting UCM4SVN Data</u>	13/13

1 Introduction

Welcome to Clearvision's UCM4SVN: Unified Change Management for Subversion

This is the Administration Guide for UCM4SVN, which explains how to install and configure UCM4SVN. There is also a User Guide for a user's perspective, please refer to this for more detailed information about using UCM4SVN. The latest versions of these documents are always available on [our website](#).

2 Prerequisites

UCM4SVN delivers its service over the web, so any web browser can be used to interact with the application. The server is an executable, so you will need a version for your Operating System and Subversion release.

2.1 Server Platform

The UCM4SVN *server* is tested and certified against a particular Subversion release and is currently distributed for specific platforms. (For more information on the product roadmap, please contact us at sales@clearvision-cm.com).

v2.0 is currently built for (please check our website for newer platforms):

- CentOS 5.3
- Windows XP SP2

UCM4SVN has been tested against Subversion 1.6.5 and 1.5.5.

(For more information on the product roadmap, please contact us at sales@clearvision-cm.com).

Note that the `svn` executable must be available on the server's path.

2.2 Client Platform


UCM4SVN should be accessible with any modern web browser. It has been tested with the following browsers:

- Firefox 3
- MS Internet Explorer 7
- Google Chrome 1.0

The Java Checkout Tool, which is executed client-side, requires Java 1.5 or later and has been tested with Java 1.6.0_14.

2.3 License

UCM4SVN is a licensed product. You must obtain a license from sales@clearvision-cm.com before you start.

-  Note: This license will only let you run UCM4SVN 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 will be running the UCM4SVN server. To get this information, you can run the `get_license_info` program on the machine you want to use as the UCM4SVN 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 email them to sales@clearvision-cm.com :

```
License Request for Clearvision UCM4SVN:¶
.... Evaluation: yes/no¶
.... IP Address: 192.168.1.0¶
..... Port: 54325¶
```

```
... MAC Address: 00-50-56-C0-00-05¶
..... Users: 50¶
..... Company: Your Company name here¶
· Email Address: yourco@yourco.com¶
... Expiry Date: 2009-12-31¶
..... Product: UCM4SVN¶
```

Cut and paste the highlighted section; check and fill in all the fields (Company Name, etc) and send to sales@clearvision-cm.com. Note: Any port number can be used (assuming that it's not already in use by your Server - the default of 54325 is a non standard port for web servers, but 80 is used for http traffic, and Apache, and 8080 is the default for JIRA).

3 UCM4SVN Setup

On Windows, all of the configuration can be done via the installer. Make sure you have your licensing information handy as you will need this during the installation. Once installed click on UCM4SVN Server in your Start Menu, located in the Clearvision folder. Then navigate to http://<ip_address>:<port_number> with your favourite browser as per your licensing information, logging in with the user details found below.

The instructions provided in this guide use Unix paths such as `/home/ucm4svn`. If you are installing and configuring UCM4SVN on Windows, you will need to adjust paths accordingly.

3.1 Create UCM4SVN Userid and Group

We recommend that you create a user and group to run the server, e.g. user id 'ucm4svn', group name 'ucm4svn'. Note that for the instructions below, it is assumed that the home directory for the new user id is `/home/ucm4svn`.

3.2 Install Files

If you have created a new user, you can install the UCM4SVN files within that user's home directory, eg:

```
/home/ucm4svn/ucm4svn/
```

and set the permissions so that the ucm4svn user can run the server (you may need to run the command below as ROOT).

```
chown -R ucm4svn:ucm4svn /home/ucm4svn/ucm4svn
```

Alternatively, if you are installing as a system service, you might prefer to install into a system folder, e.g.:

```
/opt/Clearvision/ucm4svn
```

(Note that we will use `/home/ucm4svn` to refer to this location in the remainder of this guide. Please substitute this with your installation directory).

3.3 Configure the Server Settings

The server uses a configuration file to hold license information and other optional settings. A template is provided in `/home/ucm4svn/ucm4svn/ucm4svn.cfg`.

The UCM4SVN section of the config file contains the basic settings required for UCM4SVN to operate, which look

```
[UCM4SVN] ¶
svn_username=svn_username¶
svn_password=svn_password¶
ucm4svn_workspace=/home/ucm4svn/workspace¶
ucm4svn_dir=/home/ucm4svn/ucm4svn¶
```

```

[
smtp_host=smtp.my-company.com[
smtp_username=username[
smtp_password=password[
email_from=issue-tracker@no-reply.com[
[
allow_multiple_activity_sources=1[

```

3.3.1 ucm4svn_dir

This should be set to the directory where you installed UCM4SVN (eg /opt/Clearvision/ucm4svn or /home/ucm4svn/ucm4svn).

3.3.1.1 svn_username/password

The server must be able to write to any Subversion repositories that you are using. Use the `svn_username` and `svn_password` options to set these credentials.

3.3.1.2 ucm4svn_workspace

UCM4SVN needs an area to store temporary data. This option should be set to a suitable location, e.g. /home/ucm4svn/workspace. Please make sure that this directory has full write permissions for the ucm4svn server and has sufficient disk space available.

3.3.1.3 Email Notification Settings (SMTP)

UCM4SVN can be configured to send emails on certain actions, e.g. to notify users that an activity has been assigned to them. To enable this feature, you need to set the `smtp_host`, `smtp_username` and `smtp_password` options in the config file. If you do not require this feature, please leave the smtp settings commented out.

Note that the host specified with `smtp_host` needs to be configured to accept SMTP (Simple Mail Transfer Protocol) connections on port 25 (the standard port for SMTP).

You can also configure the from-address of email notifications using the `email_from` option. Configuring the from-address is required if you want users to be able to reply to email notifications sent by UCM4SVN. In an environment where you are running multiple UCM4SVN servers, the from-address can also be used to distinguish between email notifications from the different servers.

```

[UCM4SVN] [
smtp_host=smtp.your-server.com[
smtp_username=your_username_here[
smtp_password=your_password_here[
email_from=issue-tracker@no-reply.com[

```

3.3.1.4 Activity Sources

An activity source is where an activity is created from. With UCM4SVN's ClearQuest and JIRA integrations, it is not only possible to create activities in UCM4SVN, but you can also create new activities by importing JIRA issues or ClearQuest records.

When you create a project, UCM4SVN allows you to configure which sources are allowable activity sources for new activities for the project. The three options are (provided both JIRA and ClearQuest integrations have been enabled):-

- UCM4SVN
- JIRA
- ClearQuest

By default, when you create a project you can specify more than one activity source. However, if you set the option `allow_multiple_activity_sources` to "0", then you restrict Project Managers to choose one of the options when a new project is created. This means that new activities for the project are then always created from the same activity source.

```
allow_multiple_activity_sources=0
```

3.3.1.5 Baseline Status Settings


UCM4SVN allows you to label baselines in order to indicate their status. You can configure the names for labels using the following settings:-

```
baseline_good=Good¶
baseline_bad=Bad¶
baseline_untested=Untested¶
baseline_failed=Failed¶
baseline_obsolete=Obsolete¶
```

The value for each setting can be any text of your choice. However, it is recommended that you keep these relatively short and meaningful.

Note that the colours used in UCM4SVN are currently only configurable in the HTML stylesheet itself, which lives in the UCM4SVN installation under `media/css/ucm.css`.

3.3.2 License

Set the [License] and [Server] sections of the config file as per your license email ( please note that all information is **case-sensitive**).

3.3.3 JIRA Settings

You can configure UCM4SVN to interface to your JIRA server. To enable this feature, you need to set the `jira_server` option to point to your JIRA server location. Please specify the server location as `http://<host_name>:<port_number>`, or as `https://<host_name>:<port_number>`. If no protocol is defined, then `http://` is assumed.

For example, a valid JIRA server setting would be `http://our-jira-server.domain.com:8080`.

Note: Do not forget the port number!

For the UCM4SVN to be able to access the JIRA server it requires a valid administrator account. This does not have to be the main administrative account, but it does require full admin rights such that it can access user information and import ticket lists.

The JIRA user name and password are specified in the config file using the `jira_username` and `jira_password` options.

Example:-

```
[JIRA]¶
jira_server=http://our-jira-server.domain.com:8080¶
jira_username=myuser¶
jira_password=mypassword¶
```

3.3.4 Enable XML-RPC

Enable the XML-RPC remote JIRA API as per the [instructions](#) , in summary:

- Go to System/Plugins in the Administration page and check that the RPC Jira Plugin is installed and that the 'System XML-RPC Services' are enabled
- Go to Global Settings/General Configuration set 'Accept remote API calls' to 'ON'.

3.3.5 JIRA Workflow Matrix

UCM4SVN can, when integrated with JIRA, progress an issue through the relevant states. For example, when an issue is imported from JIRA, it remains in the Open state. Once an activity has been created from this issue in UCM4SVN and the developer has accepted this activity, it will move on to the 'In Progress' state. Then once the developer has finished his work and delivered it back to trunk (or has gotten an integrator to do so) the activity then progresses to the final state, 'Closed'.

NOTE: If you enable this feature, you must specify a configuration that specifies which JIRA workflow action should be taken for each state.

The example below shows a workflow matrix for the default JIRA workflow. If you use customised workflows in JIRA, you will need to go to the Administration tab and select the Workflows option from the menu on the left. Then click on Steps for your workflow. State ids are shown in parentheses behind each Step Name, and Action ids are shown in parentheses behind each Transition. The `JIRA_workflow` settings need to be defined such that for each state it defines what action to take. The format is as follows:-

```
jira_accept=<state_id>:<workflow_action_id>¶
jira_deliver=<state_id>:<workflow_action_id>¶
jira_close=<state_id>:<workflow_action_id>¶
jira_integrate=<state_id>:<workflow_action_id>¶
```

The default workflow looks as follows. Note that if you are using the default JIRA workflow, you will not need to customise this.

```
jira_accept=1:4¶
jira_deliver=3:2¶
jira_close=3:5¶
jira_integrate=5:701¶
```

The behaviour of the default workflow is described in the following table:

UCM4SVN Action	Previous JIRA State	JIRA Workflow Action	New JIRA State
Accept	Open	Start Progress	In Progress
Deliver	In Progress	Close Issue	Closed
Close	In Progress	Resolve Issue	Resolved
Integrate	Resolved	Close Issue	Closed

If you need to customise this workflow, then please make sure that the action specified for Accept puts the JIRA issue into a state (e.g. In Progress) that makes it possible for the actions specified for Deliver and Close to progress the tickets further through the workflow. The same applies to the Close and Integrate actions. In other words, the New JIRA State for Accept needs to match the Previous JIRA State for Deliver and Close, and the New JIRA State for Close needs to match the Previous JIRA State for Integrate.

3.3.6 ClearQuest Settings

If you are integrating with ClearQuest, you will need to configure ClearQuest-related settings in the configuration file and customise the example CQPerl scripts that are delivered with UCM4SVN for your needs.

3.3.6.1 Basic Configuration

ClearQuest settings are provided in the config file in the [CQ] section:

```
[CQ] ¶
cq_username=user¶
cq_password=password¶
cq_db=db_name¶
cq_db_set=7.1¶
cq_perl=C:\Program Files\IBM\RationalSDLC\ClearQuest\CQperl.exe¶
cq_perl_scripts_dir=c:\Program Files\Clearvision\ucm4svn\scripts¶
¶
cq_accept=open¶
cq_close=resolve¶
cq_deliver=resolve¶
cq_integrate=validate¶
```

The configuration needs to be customised as follows:-

- Set `cq_username` to the name of a suitable account with administrator privileges. The account used needs to be able to retrieve information about user accounts from the server and run queries through CQPerl.
- Set `cq_password` to the password for the administrative account.
- Set `cq_db_name` to the database with which you would like to interface. If you need to be able to interface to multiple databases, you can ignore this setting and you will need to customise the `.pl` scripts in the `scripts` directory.
- Set `cq_db_set` to the name of the ClearQuest schema repository for the database specified in `cq_db_name`.
- Set `cq_perl` to the full path to the CQPerl executable on your system.
- Set `cq_perl_scripts_dir` to the `scripts` directory in the UCM4SVN installation. If you need to be able to maintain a number of different customised versions of the scripts for different UCM4SVN servers, you can take a copy of the scripts for each server and customise the server's config file to point to the correct location.

The remaining four options specify the ClearQuest actions to perform on particular UCM4SVN actions. The actions are used to move ClearQuest records to the correct state on UCM4SVN actions.

- When a developer accepts an activity in the to-do list, UCM4SVN will perform the action specified in `cq_accept`.
- When a developer closes an activity in the to-do list, UCM4SVN will perform the action specified in `cq_accept`.
- When a developer delivers an activity in the to-do list, the action specified in `cq_deliver` will be applied to the associated ClearQuest record.
- And finally, when an integrator integrates an activity, the action specified in `cq_integrate` is applied.

The default actions specified in the config file are suitable for ClearQuest's Defect tracking workflow.

3.3.6.2 ClearQuest Script Customisation

UCM4SVN's ClearQuest integration uses CQPerl scripts to interface to ClearQuest. As ClearQuest is very customisable, it is likely that you will need to customise the scripts provided for your needs. The following table gives an overview for the scripts called by UCM4SVN and their purpose:-

Script	Purpose
<code>validate_password.pl</code>	Called to validate the user credentials of a user against ClearQuest. Users imported from ClearQuest will automatically carry a user type of "ClearQuest" to indicate that they are imported users who should be authenticated against the external database.
<code>get_cq_url.pl</code>	This script is called to retrieve a URL for a ClearQuest record with a particular id.
<code>get_users.pl</code>	The purpose of this script is to retrieve a list of ClearQuest users and return adequate information to make it possible to import the users into UCM4SVN.
<code>get_cq_list.pl</code>	This script is called in New Activity to retrieve a list of ClearQuest records to import into UCM4SVN. May arguments are passed into the script (see the script itself for details). However, please note that one of the arguments is the value of the CQ Selector field for the project for which the script is called. By default the value of the CQ Selector field is used as filter against the Project field. However, you may want to change this to suit your needs.

When customising scripts, you can test your changes by running the script from the command in using the correct command line options. You can see the arguments passed into each script at the top of the script itself, e.g. for `get_cq_list.pl`:-

```
my $cq_superuser_id = @ARGV[0];..... # cq_username from config file
my $cq_superuser_password = @ARGV[1];.. # cq_password from config file
my $project_id = @ARGV[2];..... # Numerical database
project-table row id
my $project_selector = @ARGV[3];..... # Cq Selector specified in
Create/Edit Project
my $cq_db = @ARGV[4];..... # cq_db from config file
my $cq_dbset = @ARGV[5];..... # cq_db_set from config file
```

The following is an example command line for running `get_cq_list.pl` from the command line:-

```
cqperl "get_cq_list" "admin" "pass" "0" "selector" "CQ" "7.1"
```

3.3.6.3 get_cq_list.pl

This script is very likely to require customisation to suit your needs.

Note that the `project_id` argument passed into the script is currently not used. Instead, the script currently uses the value of the CQ Selector field to generate a filter for the project of a record as follows:-

```
... @project_test = $project_selector;¶
... $rootfilternode->BuildFilter("Project", $CQPerlExt::CQ_COMP_OP_EQ,
\@project_test);¶
```

However, if you prefer not to use the CQ Selector field, you could use the `project_id` to distinguish between different projects.

The other test that is of interest is the state test. The default script only returns records that are in "Assigned" state. This is achieved with the following lines of code:-

```
... @state_test = "Assigned";... # AND (state == 'Assigned')¶
... $rootfilternode->BuildFilter("State", $CQPerlExt::CQ_COMP_OP_EQ,
\@state_test);¶
```

Whatever your customisations to the script may be, you must make sure that the result is printed on standard-out (stdout) in the following format:-

```
<id>:<headline>:<owner>:<priority>:<description>; [<id>:<headline>:<owner>:<priority>:<description>];
```

The fields for a each record are separated by colons and the records themselves are separated by semicolons. Due to this, `get_cq_list.pl` must make sure that colons and semicolons in the headline and description fields are filtered not to contain these reserved characters. The default implementation of the script achieves this by filtering the fields and removing the reserved characters using a regular expression.

For example, the following would be a valid result:-

```
CQ00000011:A new ticket from ClearQuest:admin:Critical:This is a ticket
imported from ClearQuest;
```

3.3.6.4 update_cq.pl

The purpose of this script is to attach the closing comment provided by the Developer to the ClearQuest record and move the record to the next state.

The state to move records to can be configured in the UCM4SVN config file. However, you may want to customise the field to which closing comments are attached.

When testing customisations to this script, please note that closing comments are expected to be provided in a file and that the file name is passed into the script.

Also, the username and password combination passed into the script is that of the individual user performing the Accept/Close/Deliver/Integrate action. If users do not have suitable permissions to perform these actions through the ClearQuest API, you may want to change the script to use an administrator's account to perform the modifications.

3.3.6.5 get_cq_url.pl

The script provided is no more than an example, and it needs to be customised to return a URL that allows users to navigate directly to a ClearQuest record. This script is used to generate the URL displayed in Edit Activity for activities imported from ClearCase.

3.3.6.6 get_users.pl

The script is used for importing users into UCM4SVN. Things to consider for this script are the method of authentication in ClearQuest. For example, you may want to customise the script to use an LDAP server.

3.3.6.7 validate_password.pl

This script is called when a user with User Type "ClearQuest" logs in to UCM4SVN. You should not need to customise this script unless individual users cannot authenticate against ClearQuest using the CQPerl API and you need to use an alternative method of authentication.

3.3.7 Logging

The UCM4SVN server will keep a log file. You can configure the level of detail using the `log_level` option, using one of the following:

- **INFO:** output verbose (maximal) informational details about the operations of UCM4SVN. These include creating components, users, projects and activities.
- **WARNING:** show any warning messages. Show details of possible issues when operating UCM4SVN (these are not errors, but may be useful for administration purposes or highlight a potential problem)
- **ERROR:** show only error messages (minimal output)

3.4 Starting the Server

The final step is to start the UCM4SVN server. You can run the server with your config file using the `-f` flag:

```
/home/ucm4svn/ucm4svn/ucm4svn -f /home/ucm4svn/ucm4svn.cfg &
```

This will start the server and make UCM4SVN available to web clients. Now you can connect to the UCM4SVN server in a web browser and login using the username `admin` and password `admin`. From here you can create users, components, projects and activities. See the user guide for more information on how to do this. 🛡️ We recommend that you change the passwords for the default users at this stage.

- **Username:** `admin`
 - **Password:** `admin`
-

4 Creating A Sample Database for Evaluation/Testing

During testing or evaluation, it can be useful to set up some sample data. ⚠ This is a destructive operation, so please **DO NOT USE ON A LIVE SYSTEM!**

Create an empty subversion repository, and make sure you have no data stored in UCM4SVN (**all existing data will be erased**). You can set up the sample database with:

```
/home/ucm4svn/ucm4svn/tools/ucm4svn_reset_db -l /home/ucm4svn/ucm4svn  
--create-sample-database=http://my-repo-server/svn/test
```

This command will initialize UCM4SVN with a set of sample data using the `http://my-repo-server/svn/test` repository.

5 Troubleshooting

5.1 Emergency Database Access

Under exceptional circumstances, if you require access to the underlying database, please use the following login credentials:

- **Username:** dbadmin
- **Password:** clearvisionucm4svn

Note that this is for emergency use only, and will normally only be used by Clearvision Technical Support.

5.2 Resetting UCM4SVN Data

 **Using this will DESTROY all of your UCM4SVN data!**

You can reset UCM4SVN by destroying all of its data. You might want to do this with a test or evaluation system, we do **NOT** recommend ever using this on a live system (please use with utmost caution)!

Run the reset command, and tell it which installation directory to reset.

```
/home/ucm4svn/ucm4svn/tools/ucm4svn_reset_db -l /home/ucm4svn/ucm4svn
```

This command will reset the internal UCM4SVN database to an empty state. Any existing data will be backed up as a new file <DATE>-ucm4svn.db.BAK in the same directory as the original database.