



Migrate2SVN

Administration Guide

Table of Contents

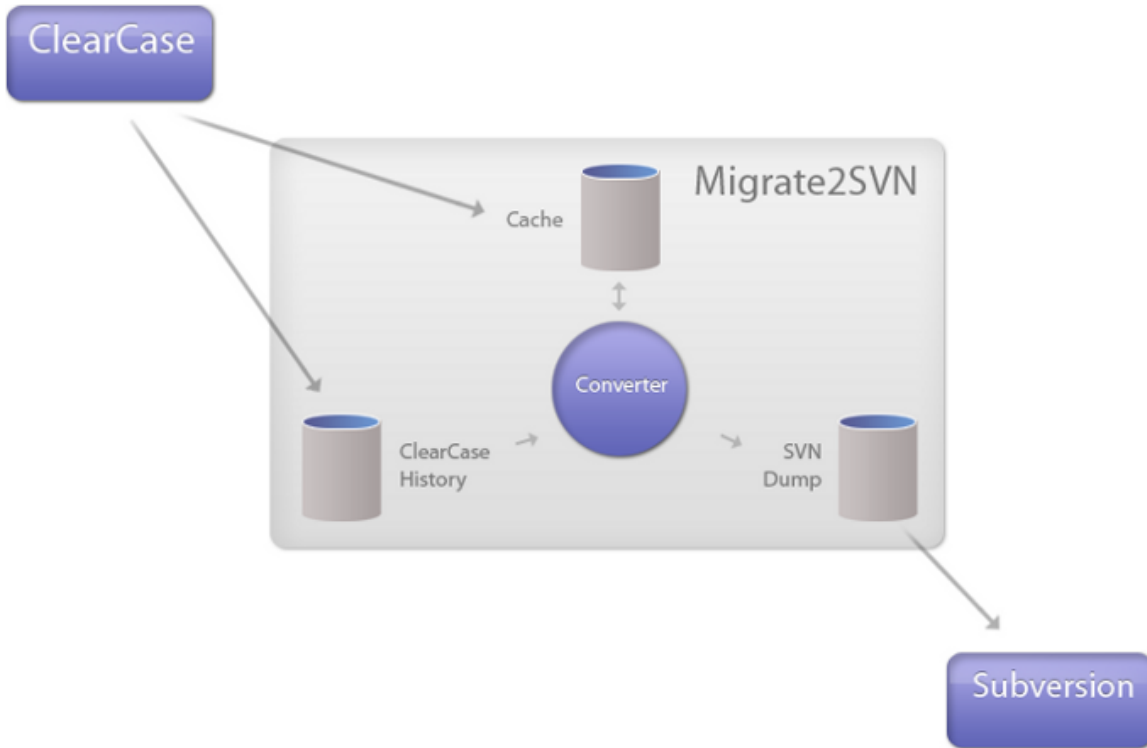
- 1 Introduction
 - 1.1 License
 - 1.1.1 Evaluation License
 - 1.1.2 Full License
- 2 Installation
- 3 Command Line Options
- 4 Configuration Options
 - 4.1 ClearCase Options
 - 4.1.1 cleartool_cmd
 - 4.1.2 cc_view_dir
 - 4.1.3 cache_dir
 - 4.1.4 target_branch_for_cclinks
 - 4.1.5 cspec_list_file
 - 4.1.6 cc_branches
 - 4.1.7 cc_branches_exclude
 - 4.1.8 cc_labels
 - 4.1.9 cc_labels_exclude
 - 4.1.10 cc_config_template
 - 4.1.11 check_zero_size_cache_file
 - 4.1.12 history_file
 - 4.2 Subversion Options
 - 4.2.1 create_branches_tags_dirs
 - 4.2.2 svn_autoprops_file
 - 4.2.3 svn_dump_file
 - 4.3 Logging Options
 - 4.3.1 log_file
 - 4.3.2 log_level
 - 4.4 Migrate2SVN General Options
 - 4.4.1 migrate_encoding
- 5 Appendix A: Generating a Config File
 - 5.1 Introduction
 - 5.2 Running the Tool
 - 5.2.1 Windows
 - 5.2.2 Linux
 - 5.3 Additional Command Line Options

Introduction

Welcome to Migrate2SVN, Clearvision's solution for a full history one-off migration from ClearCase to Subversion. Migrate2SVN allows you to migrate a ClearCase database into a Subversion repository whilst retaining full metadata including authors, timestamps and checkin comments. Migrate2SVN can also migrate branches and labels into Subversion branches and tags.

This Migrate2SVN administration guide documents the installation and configuration options of Migrate2SVN. Please refer to the User Guide for details on how to use the tool.

As shown in the diagram below, Migrate2SVN operates via a ClearCase view and extracts the ClearCase history. Each file version to be migrated is fetched into a cache. The Migrate2SVN converter then converts the file versions into a Subversion dump file, which can be loaded into Subversion.



License

Evaluation License

Evaluation licenses can be requested from the clearvision website: <http://www.clearvision-cm.com>

A link to the license request form for a product can be found on the product's download page. To request a license we need some information about the machine that you will be running the server on. To get this information, you can run the `get_license_info` program on the machine you want to use as the server (the `get_license_info` program can be found within the tools directory of the installation directory). This will retrieve and present system information from the machine.

Once the license request form has been completed and submitted, an email will be sent to the registered email address containing the license information. This information will need to be inserted into the server config file.

Note that this license will only work on a particular server machine, on a particular port, and will allow a maximum number of users. The license is time-limited so it is recommended that Evaluation licenses are not requested until needed.

Full License

Once a license has been purchased, you will receive an email containing a link to the online full license generator. A similar form will need to be filled in as for an evaluation license. Once the form has been completed and confirmed a full license will be emailed. Any evaluation license details in the config file will need to be replaced by the full license.

Licenses are tied to the hardware as well as the IP address of the machine used for the server. It is therefore necessary to ensure these details will not change as this will leave the license invalid.

```
License Request for Clearvision Migrate2SVN:
  Evaluation: yes/no
  IP Address: 192.168.64.1
  MAC Address: 00-50-56-C0-00-05
  Company: You Company name here
  Email Address: yourco@yourco.com
  Expiry Date: 2011-03-31
  Product: Migrate2SVN
```

The `get_license_info` executable is run from a command prompt.

Installation

The installation of Migrate2SVN is normally performed by a Clearvision consultant. In summary, the steps are as follows:-

- Unpack the ZIP file downloaded from the Clearvision web site
- Edit the `migrate2svn.cfg` file and install the licence provided by Clearvision Sales (sales@clearvision-cm.com).
- On Unix, adjust permissions on the installation by running `chmod -R 755 Clearvision`.

Before running Migrate2SVN, you need to configure the tool for your environment and the data to be migrated. Please see the User Guide for information on information how to use the tool including minimal configurations for quick migrations.

Command Line Options

Migrate2SVN accepts the following command line options.

Option Short	Option Full	Description
-h	--help	Print help on options.
-f	--config-file	Specify the config file to use. Default is <code>migrate2svn.cfg</code> .
-c	--check	Check configuration, including any config spec files provided.
-r	--run	Perform migration to Subversion dump file.

Configuration Options

ClearCase Options

All ClearCase-related options live in the `clearCase` section of the config file.

```
{nl:ClearCase}
cleartool_cmd=cleartool
cc_view_dir=/view/myview/vob/myvob/
cache_dir=/tmp/migrate2svn_cache
target_branch_for_cclinks=main
;cspec_list_file=specfiles.txt
;cc_branches=main;mybranch
;cc_branches_exclude=
;cc_labels=
;cc_labels_exclude=
;cc_config_template=main:mybranch
;check_zero_size_cache_file=true
history_file=cchistory.txt
```


cleartool_cmd

The `cleartool_cmd` option allows you to specify where to find the ClearCase cleartool command on your system. If you do not specify an absolute path, Migrate2SVN will use the system search path to find cleartool.

Specifying a full path to cleartool can be useful if you do not normally add cleartool to your path or you need to point to a particular installation of ClearCase.


cc_view_dir

Migrate2SVN operates within a ClearCase view and will only migrate one VOB at a time. Therefore, please specify the full view-extended path to the VOB you would like to migrate via the view you would like to be used for the migration.

 If you are migrating ClearCase labels, Migrate2SVN may change the config spec for this view. It is therefore recommended that a view is created specifically for the migration.

cache_dir

During the migration, Migrate2SVN will fetch elements to be migrated into a temporary area, the location of which can be specified with this setting.

 The cache area is not cleaned in between runs, so if you are experiencing problems, you may want to clean out the cache area to ensure that you start from a clean slate. Provided no instances of Migrate2SVN are running, you can safely remove the cache at any time. However, please note that removing the cache will have a performance impact as all file version will need to be refetched from ClearCase.

target_branch_for_cclinks

ClearCase links cannot be traced back to a particular branch. With this setting you can specify the target branch for any symbolic links in ClearCase. Migrate2SVN will then create all links on the branch specified.

cspec_list_file

With this option, you can point Migrate2SVN to a configuration spec (config spec) list file, which specifies which config spec file to use for a particular label.

Each line within the config spec list file must be in the following format:-

```
<config_spec_file_name> @ [<label>] @ [<target_path_in_subversion>] @ <commit_message>
```

The @ signs in the `cspec_list_file` are field delimiters and must remain in the file.

Example `cspec_list_file`:

```
config_spec_1.txt @ FIRSTLABEL @ @ Commit message 1
config_spec_2.txt @ SECONDLABEL @ @ Commit message 2
config_spec_3.txt @ @ tags/xxx/BR1_VIEW @ Commit message 3
config_spec_4.txt @ @ tags/BR2_VIEW @ Commit message 4
```

When you specify `<label>`, Migrate2SVN will create a tag in the standard Subversion location, e.g. for the first line in the example above, Migrate2SVN will create `tags/FIRSTLABEL` in Subversion.

When you specify `<target_path_in_subversion>`, Migrate2SVN will create (and apply) a temporary Label (in ClearCase) for all elements visible in the config spec. Migrate2SVN will then migrate those elements into the specified tag name in Subversion. e.g. all elements that are visible from `config_spec_3.txt` above will be migrated into a Subversion tag `tags/xxx/BR1_VIEW` tag.



Notes

1. Any Labels listed in the `cspec_list_file` that are also listed in the `#cc_labels_exclude` config option will not be migrated
2. Any labels specified in the `<label>` column must exist in ClearCase **and** must have been applied to file element versions.
 - Only the file elements visible in the config spec provided that have been labelled with `<label>` will be placed in the Subversion Tag
3. The `cspec_list_file` Labels will be included in addition to any labels specified in the `#cc_labels` config option
4. `<label>` and `<target_path_in_subversion>` are mutually exclusive. One (but not both) of them must be provided for each line.
 - If you specify both `<label>` and `<target_path_in_subversion>`, the target path will be ignored and the label will be used instead.
5. You must ensure that the config spec provided never eclipses the top-level view directory (`cc_view_dir`) specified in your Migrate2SVN configuration file. If this happens, the config spec used for `cc_view_dir` may become corrupted.
6. When using the `<target_path_in_subversion>` you will be unable to reuse an existing `history_file` due to the fact that Migrate2SVN will create a unique temporary label for these entries on each run of the tool
 - These Labels are removed from ClearCase when the tool finishes the Migration

cc_branches

The `cc_branches` option allows you specify a subset of branches to be migrated. By default (i.e. This field has been commented out), Migrate2SVN will migrate all branches found in the ClearCase history. Note that `cc_branches_exclude` has precedence over `cc_branches` (IE. If you specify both branches to include AND exclude, only `cc_branches_exclude` will be used)

Branch names are separated by semicolons.

```
cc_branches=main;firstbranch;secondbranch
```

Note: You can also use the `*` wildcard to specify branch names. For example:

```
cc_branches=main;release*
```

cc_branches_exclude

The `cc_branches_exclude` option allows you specify a subset of branches to be skipped during migration. By default (i.e. This field has been commented out), Migrate2SVN will migrate all branches found in the ClearCase history. Note that `cc_branches_exclude` has precedence over `cc_branches` (IE. If you specify both branches to include AND exclude, only `cc_branches_exclude` will be used)

Branch names are separated by semicolons.

```
cc_branches_exclude=branch_release3;branch_release4
```

Note: You can also use the `*` wildcard to specify branch names. For example:

```
cc_branches_exclude=main;release*
```

cc_labels

The `cc_labels` option allows you specify a subset of labels to be migrated. By default (i.e. This field has been commented out), Migrate2SVN will migrate all labels found in the ClearCase history. Note that `cc_labels_exclude` has precedence over `cc_labels` (IE. If you specify both labels to include AND exclude, only `cc_labels_exclude` will be used)

Label names are separated by semicolons.

```
cc_labels=LBL1;LBL2
```


Note: You can also use the `*` wildcard to specify label names. For example:

```
cc_labels=LBL*
```

cc_labels_exclude

The `cc_labels_exclude` option allows you specify a subset of labels to be skipped during migration. By default (i.e. This field has been commented out), Migrate2SVN will migrate all labels found in the ClearCase history. Note that `cc_labels_exclude` has precedence over `cc_labels` (IE. If you specify both labels to include AND exclude, only `cc_labels_exclude` will be used)

Label names are separated by semicolons.

 Any labels listed within `cc_labels_exclude` will be excluded, **even** if they appear within your `cspec_list_file`.

```
cc_labels_exclude=LBL1;LBL2
```

Note: You can also use the `*` wildcard to specify label names. For example:


```
cc_labels_exclude=LBL*
```

cc_config_template

This option points to a template file, which Migrate2SVN can expand when migrating labels. This template is used to generate a config spec for use in migrating each label found.

The template file specified will only be used if the `cspec_list_file` is not provided (i.e. commented out). If the `cspec_list_file` is not provided then the `cc_config_template` must be provided.

Within the config template, Migrate2SVN will replace all occurrences of `X_LBTYPEx` in the `config_template.txt` with each Label name.

 You must ensure that the config spec provided in `specfiles/config_template.txt` never eclipses the top-level view directory (`cc_view_dir`) specified in your Migrate2SVN configuration file. If this happens, the config spec used for `cc_view_dir` may become corrupted and the Migration will fail.

check_zero_size_cache_file

This option indicates if files of zero size in the cache should be refetched when populating the cache. Setting this option to "true" will slightly decrease performance as zero-size files will always be refetched. By default, this option is set to true.

Under certain error conditions, ClearCase may fail and leave an empty file behind in the cache. Setting this option to "true" will ensure that errors from previous runs do not propagate to subsequent runs of the tool.

```
check_zero_size_cache_file=true
```

history_file

During the migration, Migrate2SVN loads the ClearCase history into a temporary file. This options allows you to specify the name of that file.

```
history_file=cchistory.txt
```

By default this file is set to cchistory.txt.

Subversion Options

All Subversion-related options live in the `Subversion` section of the configuration file.

```
[Subversion]
create_branches_tags_dirs=true
svn_autoprops_file=autoprops.txt
svn_dump_file=svndump.txt
```

create_branches_tags_dirs

With this option you can control whether Migrate2SVN should create "tags" and "branches" directories in the Subversion repository.

If you are migrating to an empty repository, then please make sure that this option is set to "true".

If, however, your Subversion repository already contains "tags" and "branches" directories, then please ensure that this option is set to "false" as otherwise an error will occur when the dump file is loaded into Subversion.

```
create_branches_tags_dirs=true
```

By default, this option is set to true.

svn_autoprops_file

Migrate2SVN allows you to automatically apply Subversion properties to files based on file extensions. An autoprops file is used to define which property to set for which file extension.

```
svn_autoprops_file=/home/user/autoprops.txt
```

An example of an autoprops file is as follows:-

```
*.bat = svn:eol-style=CRLF;svn:executable
*.c = svn:eol-style=LF
*.cpp = svn:eol-style=native
*.css = svn:mime-type=text/css
*.doc = svn:mime-type=application/msword;svn:needs-lock=yes
*.dsp = svn:eol-style=CRLF
*.dsw = svn:eol-style=CRLF
*.erl = svn:needs-lock=yes
*.fmb = svn:needs-lock=yes
*.fmt = svn:keywords="Id Rev Date Author URL";svn:needs-lock=yes
*.gif = svn:mime-type=image/gif
*.gz = svn:mime-type=application/x-gzip
```

There is a full example of an autoprops file the config directory of the installation called "autoprops.txt".

svn_dump_file

With this option you can control the file name of the Subversion dump file to be created by Migrate2SVN.

```
svn_dump_file=svndump.txt
```

Logging Options

The `Logging` section allows you to configure the name of the log file and how much log information to produce.

log_file

Please specify the path to the log file to be created. By default, Migrate2SVN will create a log file called `MIGRATE2SVN.log` in the current working directory.

log_level

Migrate2SVN supports a number of log levels to control how much information should be generated during the migration.

The three log levels supported are "ERROR", "WARN" and "INFO".

Log Level	Description
INFO	Generate extra information during the migration. Useful when debugging issues.
WARN	Default log level. Only warnings and errors are reported.
ERROR	Suppress all warnings and other messages and only print errors.

The "INFO" log level is recommended when running Migrate2SVN from the command line. If you are invoking Migrate2SVN from another script, then "WARN" or "ERROR" may be more suitable.

Migrate2SVN General Options

All Migrate2SVN-related options live in the `General` section of the configuration file.

```
[General]
;migrate_encoding=latin-1
```

migrate_encoding

Subversion admin (`svnadmin`) requires that certain parts of the SVN dump file (e.g. file/folder paths) be presented in UTF-8 formatting; therefore Migrate2SVN must convert these values from the current (operating system encoding) to UTF-8 to ensure that the svn dump file will load successfully.

Different operating systems (such as Linux and Windows) use different default encodings and in order to perform the conversion to UTF-8, the Migrate2SVN needs to be aware of what encoding it needs to convert from. Migrate2SVN assumes certain defaults in order to make configuration as simple as possible.

Default Encoding

`migrate_encoding` is (by default) commented out in the config file, and in this situation the tool will automatically use the following defaults:

- Windows: `latin-1`
 - This is also known as ISO-8859-1
- Linux: `utf-8`

These settings should give you the correct migration and should successfully maintain your file names and paths between ClearCase and the final Subversion repository.

If the default settings cause the migration to fail. This failure will result in an error message similar to the following:

- ERROR 'xxx' codec can't decode character u'\xf3 (where xxx is latin-1 or utf-8)

In this situation it may be necessary to change the `migrate_encoding` value.

 It is not recommended to change this setting unless the migration fails when using the default settings.

Appendix A: Generating a Config File

Introduction

This appendix briefly outlines how to use the automatic config file generator tool to generate a default config file. The tool can be found under the 'tools' directory of the Migrate2SVN install directory.

Running the Tool

The tool runs almost identically on Linux and windows (the main difference is the name of the tool).

Windows

Usage: `{{migrate2svn_config_generator_tool.exe options CONFIG_FILE_NAME }}`

- Open a command prompt
- Change directory into the 'tools' directory of the Migrate2SVN install directory
- Run `generate_migrate2svn_config.exe <output_file_name>`
 - This will generate a default config file and save it to the `<output_file_name>` specified

Linux

Usage: `{{migrate2svn_config_generator_tool options CONFIG_FILE_NAME }}`

- Open a command prompt
- Change directory into the 'tools' directory of the Migrate2SVN install directory
- Run `generate_migrate2svn_config <output_file_name>`
 - This will generate a default config file and save it to the `<output_file_name>` specified

Additional Command Line Options

The tool has the following additional command line options

Option	Description
--help	Show this help message and exit
--cleartool_cmd=CLEAR TOOL_CMD	Cleartool command to be used. Default setting picks up cleartool from the user's path.
--cc_view_dir=CC_VIEW_DIR	Full path to ClearCase view from which the migration will be performed.
--cache_dir=CACHE_DIR	Path to a temporary directory in which migrate2svn can cache ClearCase files.
--target_branch_for_cclinks=TARGET_BRANCH_FOR_CCLINKS	As it is not possible to trace the history of a symbolic links, all symbolic links are assumed to relate to the branch specified in this option.
--cspec_list_file=CSPEC_LIST_FILE	Points to file listing configuration spec files to use.
--cc_branches=CC_BRANCHES	Path to a semicolon-separated list of branches to migrate. Default behaviour is to migrate all branches found in the ClearCase history.
--cc_config_template=CC_CONFIG_TEMPLATE	Path to a template file, which Migrate2SVN can expand when migrating labels. The template file specified will only be used if the cspec_list_file is not provided.
--check_zero_size_cache_file=CHECK_ZERO_SIZE_CACHE_FILE	If this option is set to true, the tool will refetch any zero-size files in the cache on incremental runs.
--history_file=HISTORY_FILE	Path to a file for temporary history file created to be created by the tool.

<code>--create_branches_tags_dirs=CREATE_BRANCHES_TAGS_DIRS</code>	If you are migrating to an empty repository, then
	please make sure that this option is set to "true".
	If, however, your Subversion repository already
	contains "tags" and "branches" directories, then
	please ensure that this option is set to "false".
<code>--svn_autoprops_file=SVN_AUTOPROPS_FILE</code>	You can attach Subversion properties to files based on
	file extensions. Autoprops file format is shell-
	pattern = svnproperty=value;svnproperty=value;... The
	space after shell pattern is important.
<code>--svn_dump_file=SVN_DUMP_FILE</code>	Name of Subversion dump file to create.
<code>--company_name=COMPANY_NAME</code>	The name of your Company
<code>--licence_email=LICENCE_EMAIL</code>	A contact email for the licences
<code>--expiry_date=EXPIRY_DATE</code>	The expiry date of your license
<code>--license_key=LICENSE_KEY</code>	Your license_key & certificate as received from us
<code>--license_certificate=LICENSE_CERTIFICATE</code>	Your license_certificate
<code>--server_ip=SERVER_IP</code>	
<code>--log_file=LOG_FILE</code>	The file that the server will use for logging
<code>--log_level=LOG_LEVEL</code>	the level of logging desired: ERROR, WARNING or INFO