

IntelliJ IDEA and CVS Suite

WHITEPAPER 2009 Build 7799 April 2021

Legal Notices

There are various product or company names used herein that are the trademarks, service marks, or trade names of their respective owners, and March Hare Software Limited makes no claim of ownership to, nor intends to imply an endorsement of, such products or companies by their usage.

This document and all information contained herein are the property of March Hare Software Limited, and may not be reproduced, disclosed, revealed, or used in any way without prior written consent of March Hare Software Limited.

This document and the information contained herein are subject to confidentiality agreement, violation of which will subject the violator to all remedies and penalties provided by the law.

LIMITED WARRANTY.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, March Hare Software Limited AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, WITH REGARD TO THIS DOCUMENT, AND ANY ADVICE OR RECOMMENDATION CONTAINED IN THIS DOCUMENT.

NO OTHER WARRANTIES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL March Hare Software Limited OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE FOLLOWING DOCUMENTATION INCLUDING ANY RECOMMENDATION OR ADVICE THEREIN, EVEN IF March Hare Software Limited HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, March Hare Software Limited's ENTIRE LIABILITY UNDER ANY PROVISION OF THIS DOCUMENT INCLUDING ANY RECOMMENDATION OR ADVICE THEREIN SHALL BE LIMITED TO THE GREATER OF THE AMOUNT ACTUALLY PAID BY YOU FOR THE DOCUMENT OR £5.00; PROVIDED.

© Copyright 2004 - 2021 March Hare Software Ltd

march-hare.com

consultants@march-hare.com

Table of Contents

LEGAL NOTICES	A
TABLE OF CONTENTS	1
OVERVIEW.....	1
INSTALLATION	1
<i>Installed Components</i>	1
CONTACTING SALES	3
United Kingdom	3
Australia.....	3
Germany (European Sales)	3
USA & Canada	3
FUNDAMENTALS.....	4
WHAT IS CVSNT AND CM SERVER ALL ABOUT	4
CAN I INSTALL THE SOFTWARE AND READ THIS LATER	4
I DON'T LIKE VERSION CONTROL.....	4
I PREFER TO USE SOME OTHER TOOL	4
THEORY	5
PROMOTION MODEL.....	5
Example 1	5
Example 2	5
WHAT ARE BRANCHES, MAGIC BRANCHES AND VENDOR BRANCHES.....	6
<i>When are Branches, Magic Branches and Vendor Branches Used</i>	6
Example 1	6
Example 2	6
Example 3	6
Example 4	6
Example 5	6
<i>What are the benefits to using Branches, Magic Branches and Vendor Branches</i>	7
Making the same changes twice.....	7
Ensure security.....	7
<i>What is different between a Branch and a Magic Branch</i>	7
MIXED MODEL – BRANCHING AND PROMOTING	8
<i>Mixing up the development models</i>	8
PATCH MANAGEMENT – GETTING FIXES TO CUSTOMERS	8
<i>Service Packs</i>	8
<i>Patches</i>	8
WHAT IS THE REPOSITORY AND THE WORKSPACE.....	9
<i>Repository Version History</i>	9
<i>Working Copy or Workspace</i>	10
CVS SUITE AND INTELLIJ WORKFLOW SETUP	11
SETTING UP INTELLIJ IDEA CLIENT ON WINDOWS.....	11
<i>IntelliJ IDEA 2019.2</i>	11
<i>IntelliJ Workflow</i>	11
<i>IntelliJ Rename Workflow</i>	12
<i>Windows CVSNT Server</i>	12
Configure EXTNT	12
IntelliJ Preferences for Ext Connection Method	13
Testing IntelliJ IDEA connection to CVS Suite Server.....	14
Using Tortoise with IntelliJ IDEA Projects	17
Using CVSNT Command Line Client with IntelliJ IDEA Projects	18
<i>Unix, Linux or Mac OS X CVSNT Server</i>	18

Overview

This document gives a practical and technical overview of how to use IntelliJ and CVS Suite 2009 together, and in particular:

- Development integration with IntelliJ
- Set up IntelliJ client for connection to CVS Suite Server
- Use TortoiseCVS on the same workspace/sandbox as IntelliJ
- Add an existing IntelliJ project to CVS or checkout a CVS project from IntelliJ

Additionally CVS Suite 2009 provides support for these common Software Change and Configuration Management requirements not directly addressed by this document:

- Failsafe audit to an enterprise database
- Fine grained Access Control
- Event based Build management
- Secure Enterprise Authentication including password-less token based
- Encrypted communications
- Project Management – facilitation of multiple baselines for development
- Concurrency control (to allow reserved or unreserved workflows)
- Team communication with automated e-mail notification
- Automated Backup and disaster recovery
- Centralised management of SCCM system configuration

Installation

This whitepaper is not a comprehensive guide to installing CVS Suite. For a more detailed explanation refer to the documentation that accompanies the software, the ebook: *All About CM & CVS*.

Installed Components

The following software was installed in the PC environment used for this whitepaper:

<i>Software Installation</i>		
	<i>Version</i>	<i>Location</i>
CVS Suite Server	2009 Build 7799	D:\program files\cvs suite\cvsnt\
IntelliJ IDEA	2019.2 <i>(other versions also tested)</i>	
Operating System <i>(for both CVS Suite Server and IntelliJ client)</i>	Windows 10	D:\windows

Contacting Sales

If you would like more information please contact us:

United Kingdom

March Hare Software Limited
85-87 Bayham Street
Camden Town
London NW1 0AG
United Kingdom

UK: 0800 376 0404
International: +44 (0)207 692 0712
Fax: +44 (0)207 692 0609
Sales: sales@march-hare.com

Support

Install: cvsasuite.support@march-hare.com
Bronze: cvsasuite.support@march-hare.com
Silver: support.cvs@march-hare.com
Gold: support.cvs@march-hare.com
Platinum: support.cvs@march-hare.com

Australia

March Hare Pty Ltd

Our Sydney team are currently working from a temporary office, please call for an appointment.

Postal Address:
PO Box 139
Narrabeen NSW 2101
Australia

Australia: 1800 092 886
International: +61 (0)2 8212 4409
Fax: +61 2 9475 4277

Sales: sales.au@march-hare.com

Support

Install: cvsasuite.support@march-hare.com
Bronze: cvsasuite.support@march-hare.com
Silver: support.au@march-hare.com
Gold: support.au@march-hare.com
Platinum: support.au@march-hare.com

Germany (European Sales)

March Hare Software UG
Europadamm 4
41460 Neuss
Germany

Telephone: +44 (0)207 692 0712
Fax: +44 (0)207 692 0609

Sales: sales@march-hare.com

Support

Install: cvsasuite.support@march-hare.com
Bronze: cvsasuite.support@march-hare.com
Silver: support.cvs@march-hare.com
Gold: support.cvs@march-hare.com
Platinum: support.cvs@march-hare.com

Impressum

Gründung/Gesellschaftsvertrag: 19. Oktober 2018
Sitz: Neuss
Eintragung ins Handelsregister: 05. Dezember 2018.
Steuernummer: 122/5750/6267
Ust-IdNr. DE322774869
Managing Director: Arthur Barrett

USA & Canada

March Hare Software LLC
200 Broadhollow Road
Suite 207
Melville NY 11747
United States

Ph: 800-653-1501

Sales: sales.us@march-hare.com

Support

Install: cvsasuite.support@march-hare.com
Bronze: cvsasuite.support@march-hare.com
Silver: support.us@march-hare.com
Gold: support.us@march-hare.com
Platinum: support.us@march-hare.com

Fundamentals

What is CVSNT and CM Server all about

This software helps computer users keep track of changes to files.

All of the things you create on your computer: Documents, Program Source Code, Web Pages, Pictures, Spreadsheets can be managed using CVS Suite or CM Server.

In addition to tracking the changes, the software also can provide assistance with publishing, reviewing, securing and managing those files and the ability for different computer users to make changes at different stages of the documents life.

CVS Suite was originally designed for tracking changes to files and documents written by computer programmers: computer source code. This is still the primary use of CVS Suite and the focus of this paper, though the same procedures can be used for managing any type of file.

CM Server is a more advanced edition of the software and addresses the requirements of larger teams and organisations.

Can I Install the Software and Read this Later

Effectively managing your computer files and the changes to them largely depends on how you work and what your priorities for management are.

If you attempt to use the software without understanding the theory first then you will almost certainly find it is not optimal for your purposes.

Therefore you are encouraged to read the theory before attempting the practical.

I Don't Like Version Control

We use the term *Effective Configuration Management* often during our on site consulting. As an organisation it took us at March Hare Software Ltd a long time to discover that there is a difference between Configuration Management and Effective Configuration Management.

Most people who do not like version control feel that way because they have been exposed to it in an ineffective environment. Spending time doing things that are ineffective will lead to an enormous level of frustration.

If you don't like version control, please read this whitepaper and also our comprehensive guide to SCCM theory: *All About CVS* and look for a process that you would be happy to use to effectively manage your work.

I prefer to use some other tool

CM Suite is the ideal server software for people who prefer to use other tools as it is client agnostic. Client agnostic CM Server is ideal for a heterogeneous CM environment, allowing each person to choose the tool most effective for them whilst not limiting the choices of other people and retaining the ability to comply with audit and management objectives. CM Suite supports most popular Version Control and SCCM client tools.

Theory

Promotion model

A promotion model for managing changing documents or software is very common in many organizations. Using a promotion model it is easy to ensure that the correct people only “see” the documents and objects that are at the appropriate level of the promotion process.

Example 1

A government department is required to draft a new piece of legislation. The document evolves using a clearly defined model:

- Draft
- Legal Review
- Ministerial Approval
- Parliament

The document may go through several revisions during this entire process, however each time it is *promoted* to the next level it cannot be changed at that level except by authorised people. For example the public servant who authors the document cannot change the *Legal review* copy.

Frequently the document is only ever changed at the lowest level of the hierarchy; however metadata may be added at different levels. For example the legal review may wish to tag certain paragraphs as needing changes, or they may make the changes themselves.

It is important that the department that prints the legislation to table before parliament cannot accidentally print a copy that has not been through the entire promotion process.

Example 2

A software development company releases software four times a year. The software for each release evolves using a clearly defined model:

- Development
- Review
- Test
- Integration Test
- Production

The software goes through several revisions during this entire process, however each time it is *promoted* to the next level it cannot be changed at that level except by authorised people. For example the programmer who authors the bug fix cannot change the *Test* copy.

The software is only ever changed at the lowest level of the hierarchy; however metadata may be added at different levels. For example the code review may wish to tag certain functions as needing changes to comply with company coding standards, or they may make the changes themselves.

It is important that the users only run the Production version of the software and do not accidentally run another version. If a CD is produced and shipped out, the distribution department must have a fail safe way to ensure that they cannot accidentally deploy an untested release.

What are Branches, Magic Branches and Vendor Branches

Branches, Magic Branches and Vendor Branches facilitate software development project management. Often software development managers indicate that they do not require branches in their solution. Take the time to carefully read this section since once they understand what branches, magic branches and vendor branches are and what they can facilitate most customers find that these facilities are very useful and very powerful.

When are Branches, Magic Branches and Vendor Branches Used

Vendor Branches, Magic Branches and Branches are used whenever the evolution of changes to the documents are not sequential.

Example 1

A manager writes the outline of the current product specification for what the business does and gives it to the marketing department so they can develop a “what we do” document for sales people. However the manager has been instructed to also prepare for a new venture in the near future so after sending the document to the marketing department the manager begins to modify it again to bring it up to date with the new plans.

While the manager updates the document, the marketing department begin to change the wording of the document to make it more suitable for a lay audience, adding pictures and changing the formatting. The same document now has *two streams of development*.

Example 2

A software company releases version 1 of their software and immediately begins work on version 2. However the sales department sell the software to a company who discovers that a part of the application has a bug. Version 2 will not be ready for weeks yet, however the customer requires a fix for the broken function much sooner. The software source code now requires *two streams of development*.

Example 3

A freight company uses a software package to track cargo around the country; however the software uses terminology that is different to what the company uses. The names are stored in configuration files that the freight company modified to change the names to more appropriate ones.

However the software company release version 2 with many new features that the freight company want, but it has a new configuration files with a lot of new information.

The configuration files have two independent *streams of evolution*.

Example 4

A web designer manages seven web sites that are identical to each other except for a few files on each. About 90% of the web sites share the same PHP code. The web site has a single stream of development with *multiple variations*.

Example 5

A software vendor manages configuration files for their software as used by the most important 20 customers. The configuration files have a single stream of development with *multiple variations*.

What are the benefits to using Branches, Magic Branches and Vendor Branches

In each of the above examples the organisation finds that the documents do not have a sequential evolution of development but there are *more than one stream of development* (or a primary stream and multiple *variations*). CVS uses branches to track how these changes are made.

However using branches can also offer your organisation some advantages because CVS can automate some of your business activities.

Making the same changes twice

One of the benefits to using good configuration management tools is that if you need to re-apply textual changes made to a document in one stream of development to another stream then it can be automated.

For instance, in example 2 above the software developer can make the “bug fix” in “version 2” and use automated techniques to re-apply the same change to the “maintenance version”.

In the example 3 the freight company can use a vendor branch to apply the vendors changes to the configuration files that they are using and therefore keep their labels.

Changes to binary files (eg: word documents or pictures) cannot be replicated automatically.

Ensure security

If a document is evolving on more than one stream of development then it is possible that one or more have different security requirements. For instance a software vendor may develop enhancements to their application for two customers who are competitors. In this case it may be necessary for the developers making changes for customer A not to be able to see customer B’s changes.

In example 1 above, the developers from the “version 2” team may not be allowed to make changes to the more stable “version 1”, so it is possible to secure the access of the two groups based on the branch.

What is different between a Branch and a Magic Branch

Magic branches are used where the documents as a collection exist as several *variations*, however as individual documents the majority are all identical. This is common for configuration files or data (or configuration) driven web sites. Two Active Server Pages web sites may exist that are identical except for a few graphics files and the `locals.inc`, which sets the title for the pages and the name of the database to get the data from.

Magic Branches organise development so that the site-specific configuration can be developed separately to the web site framework. Provided that the web site framework is never modified on the branch, the magic branch always will contain the same files as the trunk.

Mixed model – Branching and Promoting

Mixing up the development models

Most organizations implementing CVS will want to use a mixture of the branching and promotional models. For example:

- Develop Version 1 on the Trunk
- When Version 1 is feature complete Branch Version 1 Maintenance
- Begin work on Version 2 on the Trunk
- Finalise Version 1 on the Branch
- Promote Version 1 Branch to Test
- Fix Version 1 bugs on the Version 1 Branch and promote to test again

Using a mixture of the techniques can lead to a balance with the strengths of both systems.

Regardless of the choices you make CVS is always capable of reapplying the changes between any two revisions to another revision – whether it is on the same branch or a different one.

Patch management – getting fixes to customers

The CM model that you choose may be heavily influenced by business concerns such as needing to deliver fixes to current software while also allowing development to continue on newer versions. This is known as *patch management*.

Service Packs

Often organizations deliver stable combinations of patches to customers as a service pack – or a point release. For example, version 1 is released and several bugs are found and fixed, and three months after the first release version 1.1 (or version 1 service pack 1) is released containing all bug fixes.

This service pack example can also be described in a time line similar to:

- Version 1
- Fix bug 1
- Fix bug 2
- Fix bug 3
- Release Version 1.1

Patches

What happens if one of those bugs seriously effects the customer's current day-to-day operations? In this case it may be necessary to release one of the bug fixes immediately. Since most customers are not affected the release version 1.1 is not created earlier – but a patch is produced.

This patching example can also be described in a time line similar to:

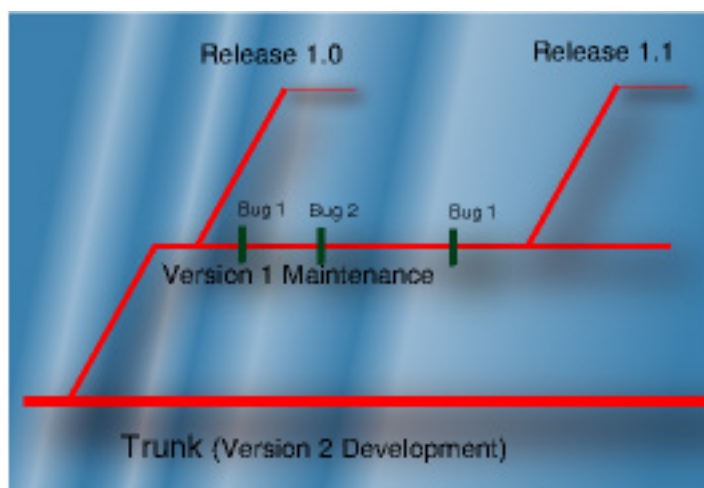
- Version 1
- Fix bug 1
- Fix bug 2
- Release Patch 1
- Fix bug 3
- Release Version 1.1

However the customer does not want any changed functionality *except* their bug fix. In the list above it can be seen that bug fix 1 and bug fix 2 have already been completed and a combined Release Patch 1 contains both fixes. A typical development environment may have made 50 changes, and the customer does not want the responsibility of testing all those fixes to get their environment working again.

The software company needs to balance the customer's requirements with their own. Specifically they need to ensure that the change is only made once but it is then applied to release 1.1 and also later to release 2.0.

Careful consideration of the business requirements is necessary to design an effective CM process. CVS is technically capable of supporting all these decisions.

In this particular example the choices the software organization would make would depend largely on the frequency and the billing methods. If these *individual patch releases* are rare or are charged to the customer then they will be designed as an exception. If they are common then the SCCM solution will be designed with some level of automation for reliability and reproducibility.



What is the Repository and the Workspace

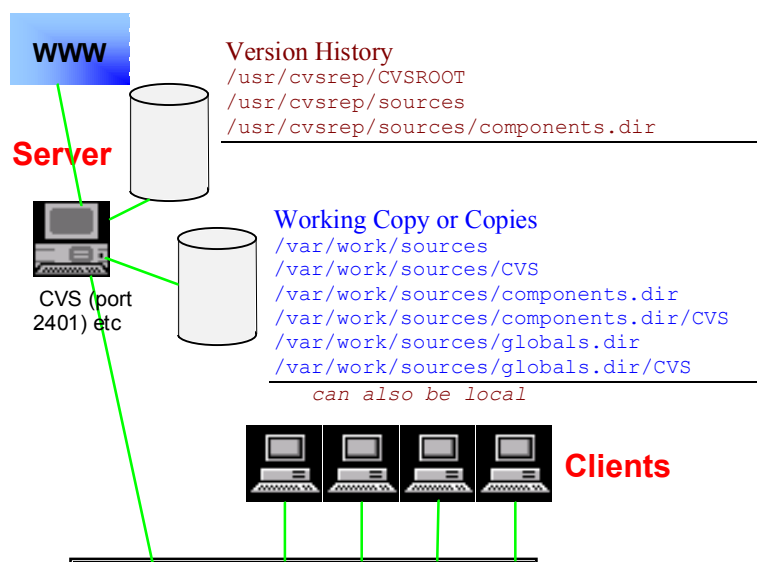
These terms are commonly used in business English but have specific technical meanings for our products and in this whitepaper:

Repository Version History

A repository is a collection of versioned objects with similar business rules managed by our server software in one or more physical locations. E.g.: the method `gencustno` is a versioned object and all instances of it (including dev/test/prod) would be in a single server repository in one or more physical locations.

A repository may contain versioned objects from different projects, eg: `CRM_PROJ`, `ORDER_INV_PROJ`, `STATS_PROJ` etc

Versioned objects from different projects that vary only by role not function (e.g.: `crm_proj_dev` and `crm_proj_test`), will always be stored in a single repository.



Working Copy or Workspace

A working copy (often referred to as a workspace) is a collection of instances of versioned objects outside of the repository. Some people refer to a workspace as *checked out copies* or as a *sandbox*.

The most common storage locations for a working copy are a disk (e.g.: My Documents) or a shared disk (e.g.: e\$ on myfserver OR \\myfserver\\e\$).

Whilst a working copy is always *linked* back to a repository, making a change in a working copy is controlled by the native permissions of target device (e.g.: a disk or a network share). Changes to a working copy may be discarded. Changes to a working copy may be kept (i.e.: committed) back to the repository only if the permissions in the repository allow changes by that person and to that instance (e.g.: John may be allowed to commit changes to custidproj in dev but not in prod).

CVS Suite and IntelliJ Workflow Setup

This section is intended to give the practical workflow when working with one of the popular methodologies supported by CVS Suite.

Setting up IntelliJ IDEA Client on Windows

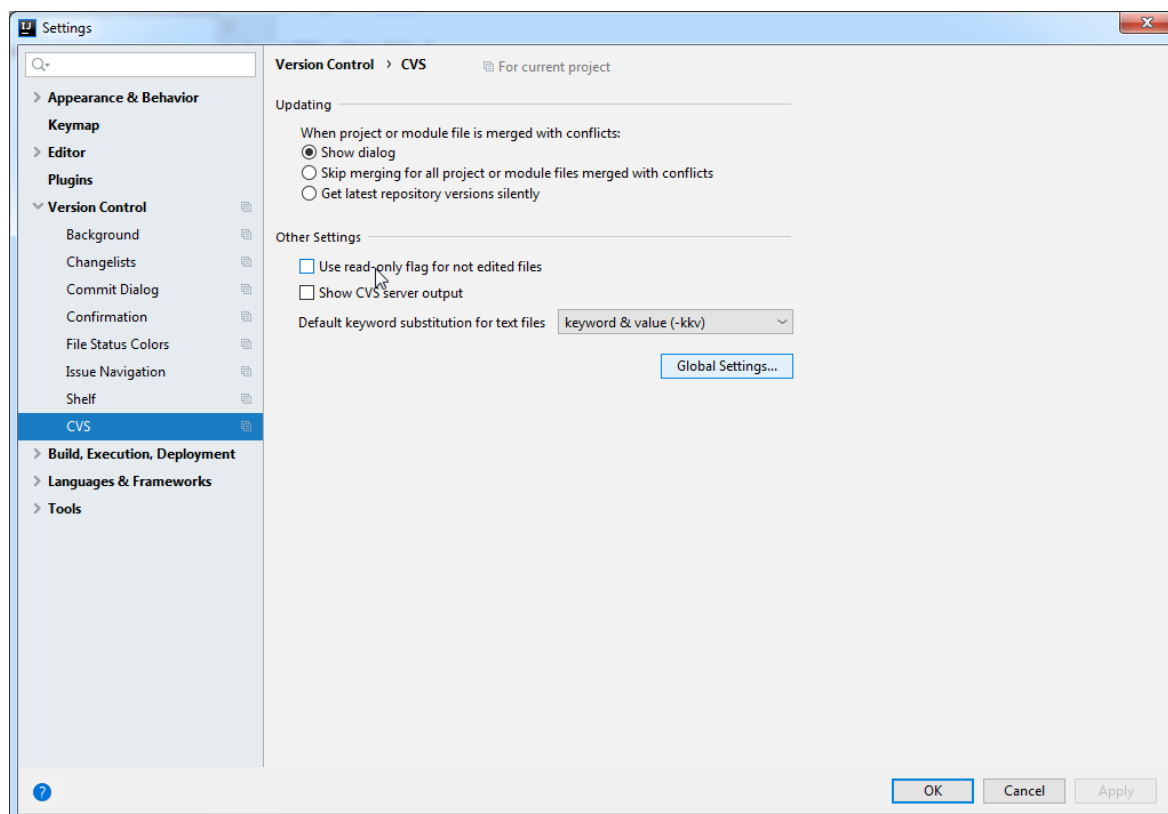
IntelliJ IDEA is a popular IDE / Editor for use in Java programming environments and it is also the basis for Android Studio.

IntelliJ IDEA 2019.2

These examples are created IntelliJ IDEA 2019.2 (Community Edition) [Build #IC-192.5728.98, built on July 23, 2019] with “Runtime version: 11.0.3+12-b304.10 amd64” using “VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o” and the “Non-Bundled Plugins: CVS”.

IntelliJ Workflow

The IntelliJ client is designed primarily to work with an Unreserved Distributed model of versioning. It is possible to configure it to work more co-operatively by enabling “Use read-only flag for not edited files” in the preferences (VCS->CVS->Global CVS Settings...) plus enabling “ignore client side kopt” in the CVSNT Server compatibility settings and setting up a cvs wrappers to use exclusive or co-operative edits by default on your IntelliJ project files (eg: *.java).



IntelliJ Rename Workflow

IntelliJ does not allow files to be renamed, the Refactor->rename function deletes the old file and creates new ones. After refactoring the CVS Plugin will prompt you to add the renamed files to version control, these files will be new and have no historical connection to your previous files.

An alternative is to refactor/rename the code in IntelliJ then using the operating system to rename the physical file(s) to their original names. After that you can use another CVSNT client to rename the file within CVSNT.

Even then the results may not be what you expect. You won't be able to update the new filename to a tag that was placed on the previous filename, because that tag had the other filename. However you would be able to see (with cvs log) the history of the name change and the tags that each name has.

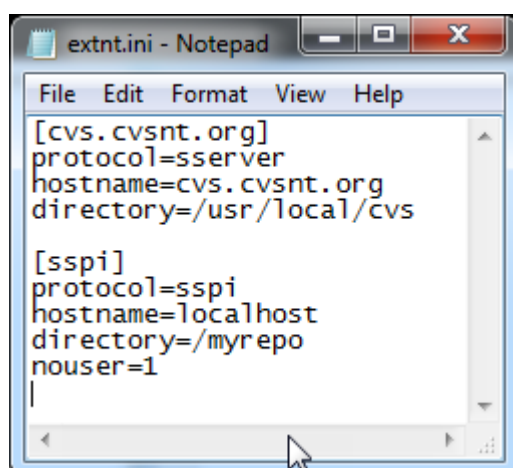
Windows CVSNT Server

To use IntelliJ with a Windows based CVSNT server it is best to use the SSPI protocol. See the section on Authentication with SSPI in the Administration section for more information.

The built in IntelliJ CVS client plugin cannot connect to a Windows based CVSNT server using SSPI, so it is necessary to install the CVSNT command line client on the same workstation as IntelliJ (CVSNT is available for HPUX, Solaris, Mac OS X, Linux and Windows). Eclipse then uses the CVSNT command line client via the EXT protocol.

Configure EXTNT

Use SSPI or SSERVER to connect using the EXT protocol. Open the configuration file extnt.ini which is located in the directory where you installed CVSNT.



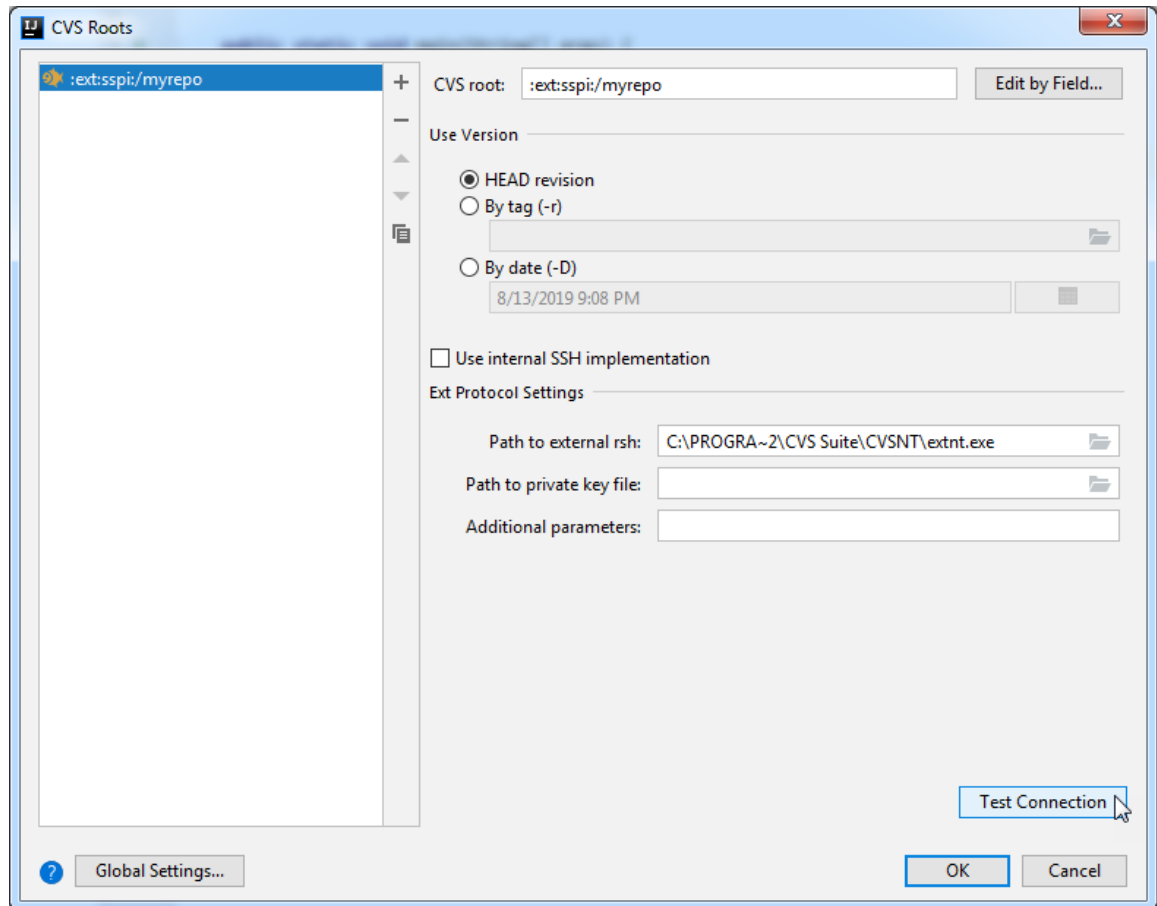
Specify the following in the `extnt.ini` file:

- `[server name]` for use from IntelliJ (this does not have to be the actual server name where the repository is stored)
- `protocol=` This is the authentication technique to use with server (usually SSPI or SSERVER)
- `hostname=` This is the CVSNT server host, and is not the same as the *server name*. This is the actual IPv4 address or the DNS name as resolved on your client PC. If the server is on the same PC as the client, it is usually: `localhost`
- `directory=` This is the repository name on the server
- `nouser=` ignore the 'user' parameter passed from the IDE (for SSPI only)

IntelliJ Preferences for Ext Connection Method

IntelliJ stores the preferences for the EXT connection method with the global list of available CVS Roots.

To set the preferences for the EXT connection, run IntelliJ and from the pulldown menu choose VCS->CVS->Configure CVS Roots... to open the CVS Roots window:



The host name in the CVS Root should be the same as the [server name] in the extnt.ini file. The repository name in the CVS Root should match the directory name in the extnt.ini file.

If you are connecting using the protocol SSERVER (defined in extnt.ini) and your user has a password then the parameters should be (note IntelliJ has a limitation where space characters cannot be used in a password):

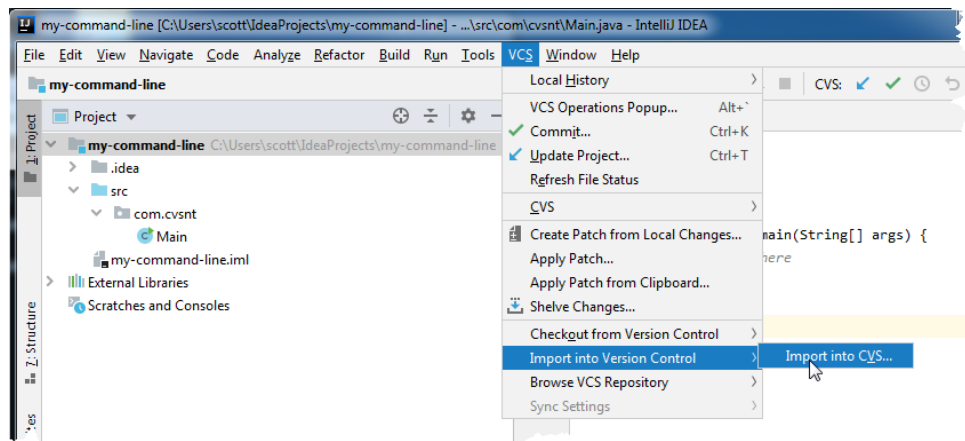
```
-P {password}
```

Testing IntelliJ IDEA connection to CVS Suite Server

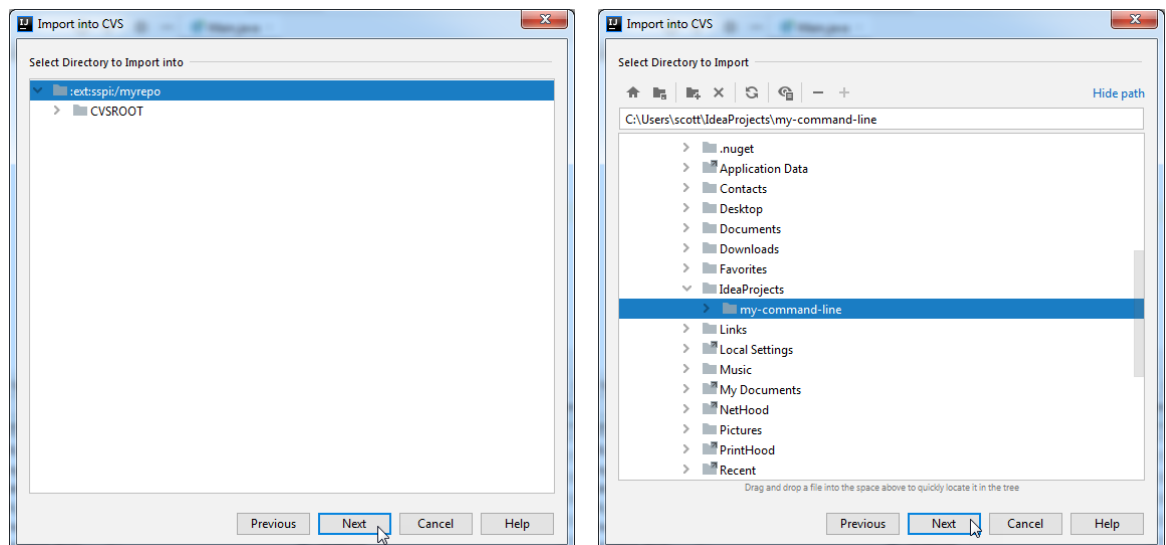
When EXTNT.INI, and the IntelliJ EXT Preferences for the CVS Root are configured you can then use the ‘Test Connection’ button to test that the connection works OK. If you are using SSPI protocol and the CVS Suite Server and the Client are on the same domain then you do not need to enter any password.

Importing a new project into CVS Suite from IntelliJ

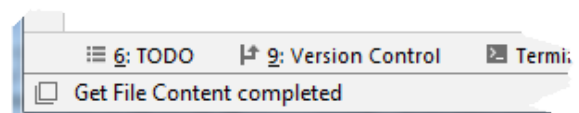
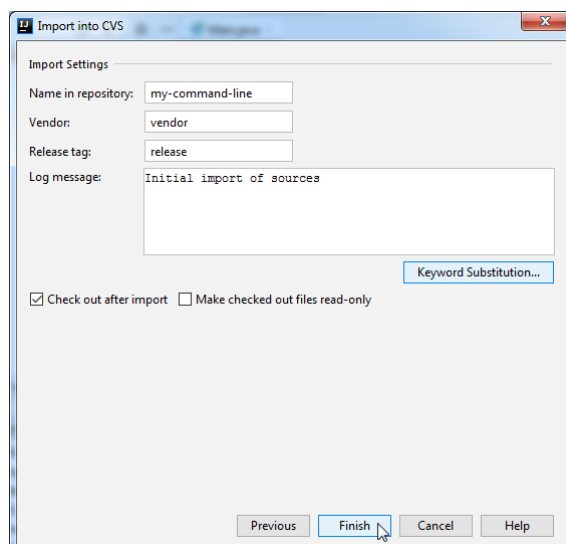
To import a new project into CVSNT, right use the pulldown menu to select VCS->Import into Version Control->Import into CVS...



Choose CVS as the repository type and select the repository location you defined earlier:

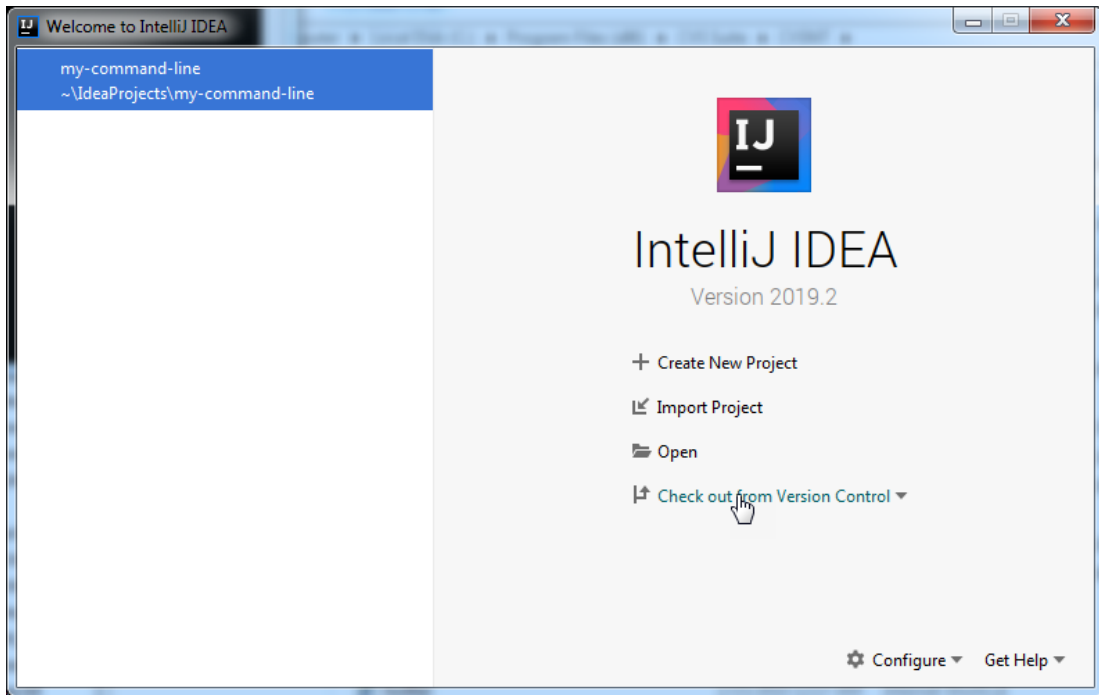


Finally to finish importing existing IntelliJ projects to the CVS Suite Server select a module name. Never import into the CVSROOT which is for CVSNT administration files only.

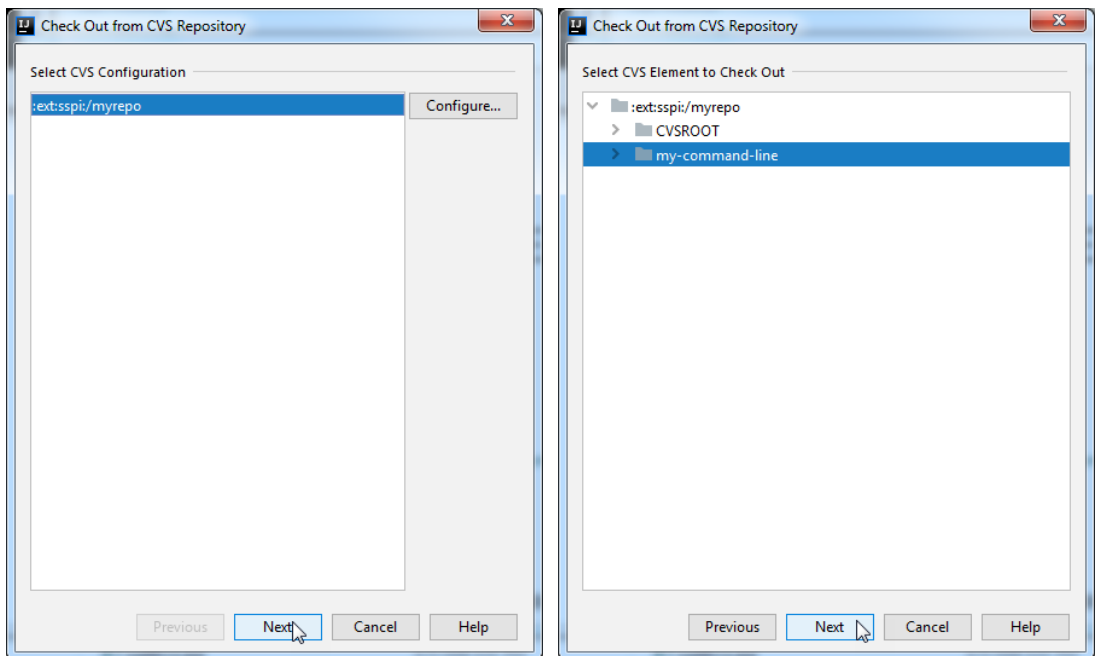


Checkout a project from CVS Suite

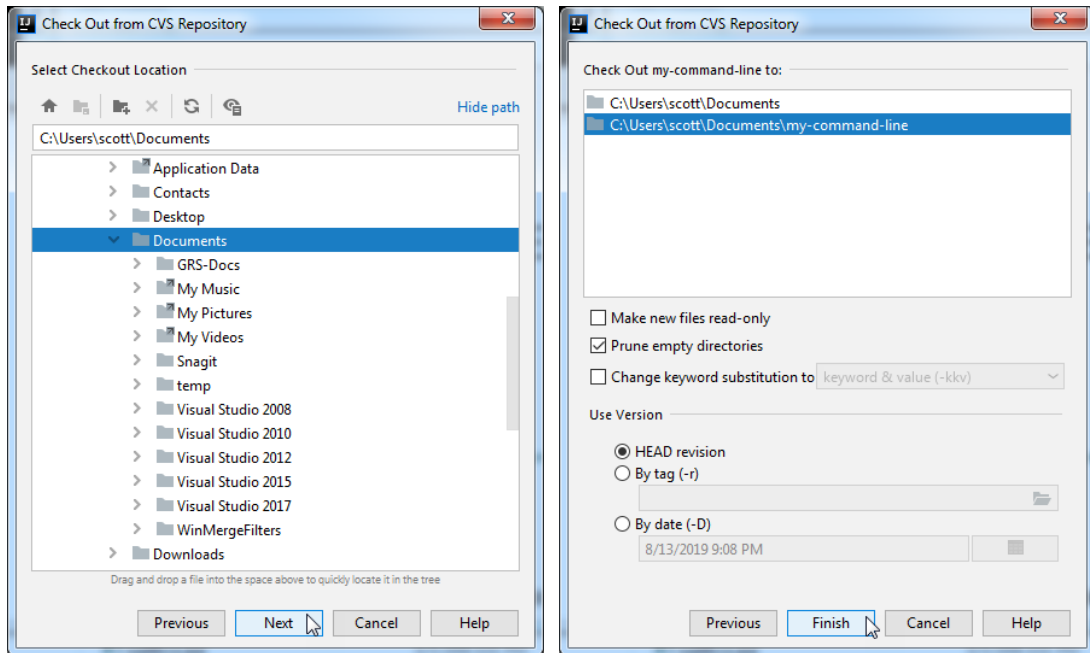
If the project already exists in CVSNT, choose Check out from Version Control on the main screen of IntelliJ IDEA:



On the following screens use 'Select CVS Configuration' and 'Select CVS Element to Checkout' to browse and choose from an existing module in the repository:



Finally use ‘Select Checkout Location’ and ‘Check Out from CVS Repository’ to browse and choose a location on your client PC to create the project files:



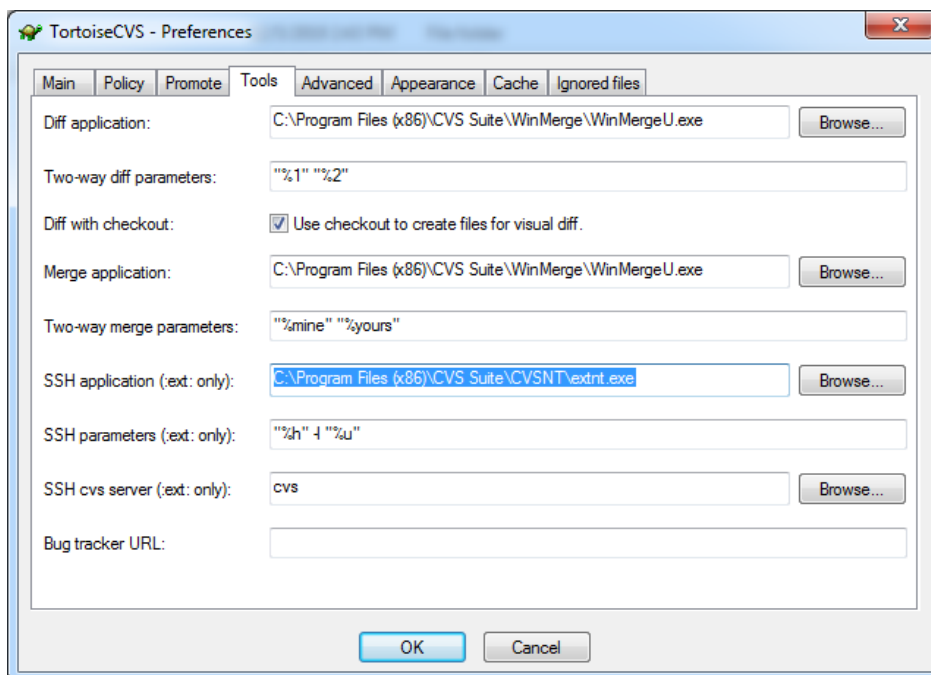
Using Tortoise with IntelliJ IDEA Projects

To use Tortoise with an IntelliJ Project you must do the following:

- Configure Tortoise to use the EXT method
- Use the “real” server name and repository name in both Eclipse and the extnt.ini configuration file.

Configure Tortoise to use the EXT method

Configure Tortoise to also use the EXT method. Open the Tortoise Preferences and set the “EXT” protocol program to extnt.exe:



Using the “real” names in extnt.ini configuration file.

The extnt.ini file maps the server name that you specify in Eclipse to a server, protocol and repository name that will be “really” used. Ideally these should be the same for both Tortoise and IntelliJ, i.e.: your IntelliJ CVSROOT is:

```
Host: mycvsserver
Repository path: /myrepo
Connection type: ext
```

And the extnt.ini would match it:

```
[mycvsserver]
protocol=sspi
hostname=mycvsserver
directory=/myrepo
nouser=1
```

Using CVSNT Command Line Client with IntelliJ IDEA Projects

To use CVSNT command line client with an IntelliJ Project you must do the following:

- Configure CVSNT to use the EXT method
- Use the “real” server name and repository name in both IntelliJ and the extnt.ini configuration file.

Configure CVSNT Command Line to use the EXT method

Configure CVSNT to use the EXT connection method by setting the environment variable CVS_EXT, similar to how you would in Tortoise or IntelliJ:

```
CVS_EXT="C:\PROGRA~2\CVSSUI~1\CVSNT\extnt.exe" %h
```

Unix, Linux or Mac OS X CVSNT Server

Use the Internal SSH client of IntelliJ IDEA to connect using the SSH protocol. This protocol is internal to IntelliJ so therefore the extnt.ini does not affect it.