

02 Version Control

Version control, also known as revision control and source control, is the management of changes to information. A version control system keeps track of software versions throughout development. A distributed version control system allows many developers to work on a given project without requiring that they maintain a connection to a common network.

Introduction

We use [git](#) for distributed LabVIEW code version control. Our code repositories are managed externally on a [GitLab](#) server and internally on a Git server.



In order to be able to work on our LabVIEW code you need to follow the steps described in our [knowledge base pages on git](#).

SCC Best Practices

You can find more information on how to work with(in) Source Code Control in our [Best Practices](#) section:

- [Source Code Control](#)
 - [Rules & Guides](#)
 - [Workflow Model](#)
 - [Submodules](#)

Version Numbering

Please read [Version Numbers](#) for more information on our version number formats and policy.

Further Resources

General

- [Official Git Tutorial](#)
- [GitLab Documentation](#)
- [Simple git how-to in german](#)

Password Management

By courtesy of Fabiola De La Cueva

pageant is a tool that runs at startup so you only have to enter your password once:

<http://blog.shvetsov.com/2010/03/making-pageant-automatically-load-keys.html>

Compare / Merge VIs

By courtesy of Fabiola De La Cueva

Instructions on how to configure your client tool to use LabVIEW Compare and LabVIEW Merge towards the end of this article:

<http://delacor.com/configuring-hg-or-git-to-use-labview-compare-and-labview-merge/>

Internal

HSE-internal resources

- [Infrastructure - SCC](#)

From:

<https://dokuwiki.hampel-soft.com/> - **HAMPEL SOFTWARE ENGINEERING**

Permanent link:

<https://dokuwiki.hampel-soft.com/processes/version-control?rev=1581421343>

Last update: **2020/02/11 11:42**

