Intelligrated®

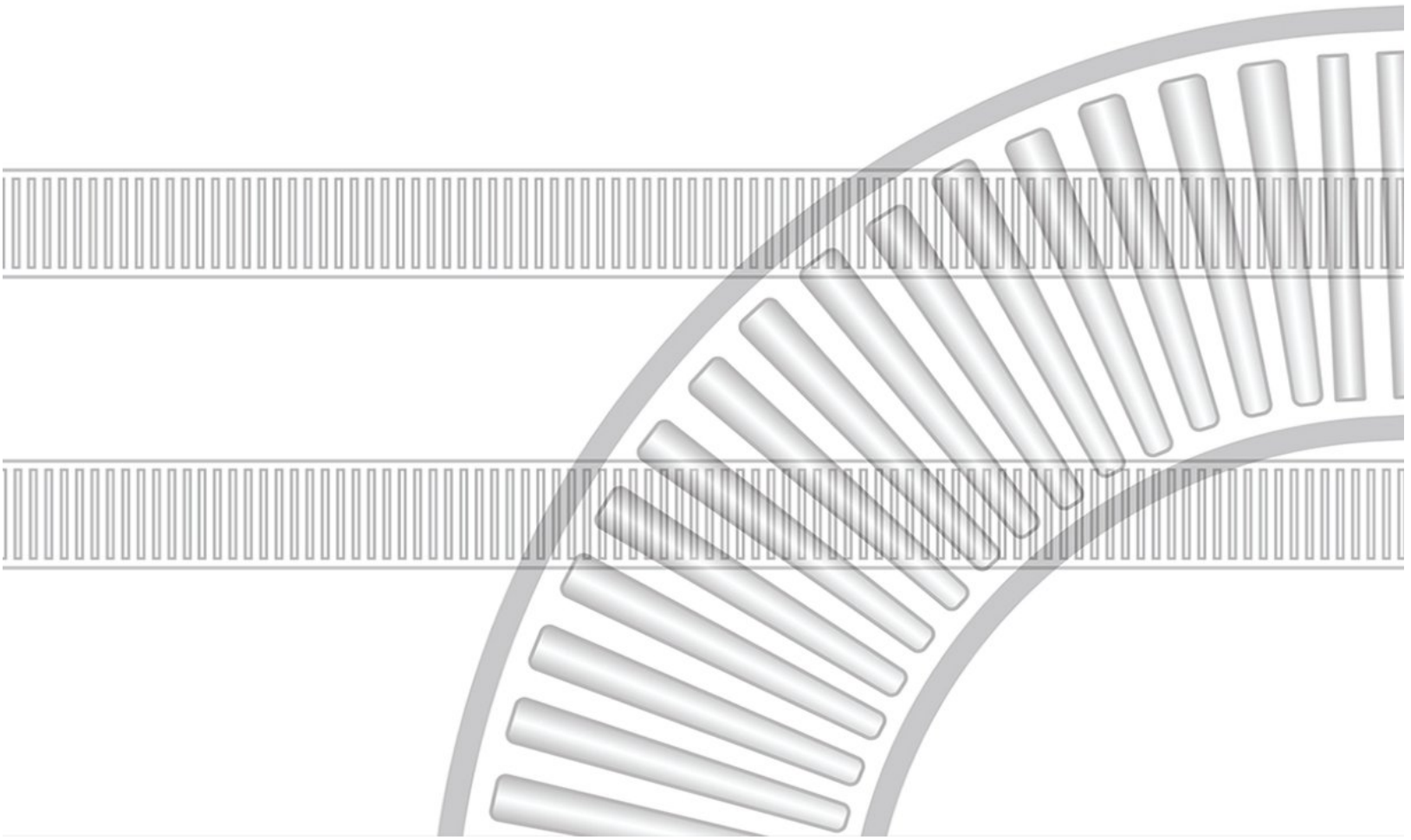
COMPANY CONFIDENTIAL

Tilt-Tray/Cross-Belt Sorters

PLC Machine Control

Version Control Procedure

Publication No. 29383670
June 5, 2015



Legal Notices

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted to parties other than the customer and the customer's employees in any form or by any means, electronic or mechanical, for any purpose, without the express written consent of Intelligrated.

This document contains confidential proprietary information and trade secrets of Intelligrated. This document is distributed with the understanding that it will be not be disclosed to any third party, in whole or part, without the prior written consent of Intelligrated.

Trademarks

InControlWare®, Intelligrated®, and IntelliSort® are registered trademarks of Intelligrated.

Allen-Bradley™ and RSLogix 5000™ are trademarks of Rockwell Automation, Inc.

ControlNet™ is a trademark of ControlNet International.

NetLok™ is a trademark of Intelligrated.

PROFIBUS® is a registered trademark of PROFIBUS & PROFINET International.

Siemens™ is a trademark of SIEMENS AG.

Simatic® is a registered trademark of SIEMENS AG.

Subversion® is a registered trademark of the Apache Software Foundation.

Table of Contents

About This Guide

1

1.1 Introduction

This manual describes the process for maintaining and updating the Subversion repository for Tilt-Tray and Cross-Belt PLC-based machine control code.

1.2 Documentation Conventions

Many typographical conventions are used to distinguish between the different kinds of information presented in software documents, as follows:

Convention	Description
Bold	Used to identify menu selections, toolbar selections, and section references.
Italic	In paragraph text, italic identifies the titles of documents that are being referenced. When used in conjunction with the Code text described below, italics identify a variable that should be replaced by the user with an actual value.
`monospace text`	Text that represents programming code.
`monospace italic text`	Variables in programming code.
CTRL+X	A combination of keystrokes that are pressed simultaneously.
Function | **Function**	A path to a function or dialog box within an interface. For example, “Select File | Open” indicates that you should select the Open function from the File menu.

Convention	Description
( ) and |	Parentheses enclose optional items in command syntax. The vertical bar separates syntax items in a list of choices. For example, any of the following four items can be entered in this syntax: persistPolicy (Never | OnTimer | OnUpdate | NoMoreOftenThan)

1.2.1 Notes, Tips, and Important Information

The following callouts and icons are used to highlight information throughout this guide:

Note or Tip

A Note highlights related information or information that is tangential to the topic being discussed. A Tip highlights useful information that can be used to simplify the tasks that are being discussed.

Important

Important callouts are used to highlight information of great significance or value that the reader should be certain to know before proceeding.

Notes, Tips, and Important callouts are used to call attention to useful information and are not safety notices.

PLC Machine Control Version Control

2

The purpose of this document is to describe procedures to ensure that the most current/running version of PLC machine control code is always captured in the Intelligrated version control system.

Intelligrated uses a system called Subversion to facilitate version control of PLC-based source code for Tilt-Tray and Cross-Belt sorters. The repository can be browsed by clicking on this link (requires TortoiseSVN client, see **Prerequisites** section below):

svn://cinsvn-nl.intellig.local:3834

2.1 New Site Responsibilities

The following details the division of responsibilities for PLC version control management when commissioning a new site implementation.

2.1.1 During Commissioning Activities

If commissioning activities take place in multiple stages, the Commissioning Engineer must ensure that the code is captured in the Subversion repository at the end of each stage.

2.1.2 End of Commissioning Activities

When commissioning activities are completed, the Commissioning Engineer must ensure that the currently running version of PLC code is committed into the repository with a revision number 100 following the procedures below.

The Commissioning Engineer will email the following individuals as a confirmation that the files have been checked in to the repository:

- Commissioning Engineering Manager (Carlos Gutierrez)
- Hotline Engineering Manager (Neil Kaszubski)
- PLC Controls Engineering Manager (RD Wilson)

The files committed must be attached to the email.

2.1.3 Prior to Departing Site

Before leaving the site, the Commissioning Engineer must ensure that the PLC program is downloaded to the non-volatile memory card (i.e., no online code changes must be active).

The Commissioning Engineer must ensure that the Intelligrated NetLok remote support servers are updated with the current/running version of PLC code for all sorters/CPUs on the site. The Commissioning Engineer will coordinate this activity with the Intelligrated Hotline Engineers.

The Commissioning Engineer will email the following individuals to confirm that the files have been checked in to the repository:

- Commissioning Engineering Manager (Carlos Gutierrez)
- Hotline Engineering Manager (Neil Kaszubski)
- PLC Controls Engineering Manager (RD Wilson)

The files committed must be attached to the email.

2.2 Production Site Responsibilities

The following details the division of responsibilities for PLC version control management when implementing changes to a site in production.

2.2.1 Implementing Changes to a Running Site

Before any changes are made to a running site, the currently running version of PLC code should be compared versus the version found in the trunk of the Subversion repository. For **Siemens Simatic Step 7** projects, this means comparing the offline and online program. For **Allen Bradley RSlogix5000** projects, this means uploading the code from the PLC onsite and then using RSLogix 5000 to compare against the version found in the trunk. Working data blocks and tags will produce differences that are expected and can be ignored. If configuration (i.e., DB_AdjChutes, CMCsetup, or DB1-19) or program changes are found, further research must be done to determine which version is correct and should be used as a starting point for the update. This is done to ensure the customer has not performed any updates to the PLC.

After downloading the update to the PLC, a CMC (TT-CB MC) hard reset must be performed and a restart of the upper level control service (i.e., the loop sorter service for ICW sites) to ensure all pointers/indexes have been reset. The flash memory card in the PLC must be updated with the latest code as well.

After changes are completed and installed onsite, the PLC Controls Engineer or Hotline Engineer implementing the changes must ensure that they are captured in the Subversion repository with a new revision number and that the NetLok remote support server code is updated.

2.2.2 Siemens Step 7 Specific

In order to connect to a Siemens PLC, the hardware configuration must be altered to add an additional Ethernet card for the NetLok NAT address. This should NOT be added to the version saved on the Subversion repository, only the file in NetLok. The Ethernet card should be given the label Remote Dummy and will have an IP address beginning with "172.21." When adding this card, the hardware configuration should only be saved and NOT saved and compiled.

2.3 Procedures

2.3.1 Prerequisites

Contact the IT group to be added to the NetLok group in order to connect to the NetLok remote support servers.

In order to update the Subversion repository, you need the following:

1. An account on the Subversion server. Contact Thomas Severinsen (Thomas.Severinsen@intelligrated.com) if you do not already have an account.

2. A Subversion client named "TortoiseSVN" version 1.7.6. It can be downloaded here:

   64-Bit Windows:

   http://sourceforge.net/projects/tortoisesvn/files/1.7.6/Application/TortoiseSVN-1.7.6.22632-x64-svn-1.7.4.msi/download

   32-Bit Windows:

   http://sourceforge.net/projects/tortoisesvn/files/1.7.6/Application/TortoiseSVN-1.7.6.22632-win32-svn-1.7.4.msi/download

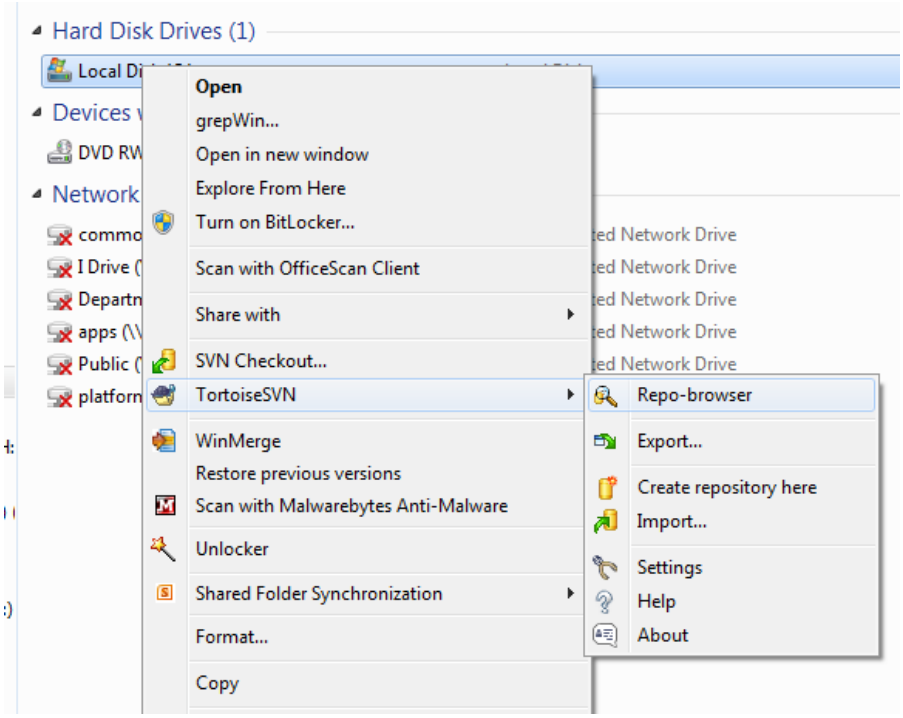
Important

Do NOT download the latest version. Use the link above and download only this specific version.

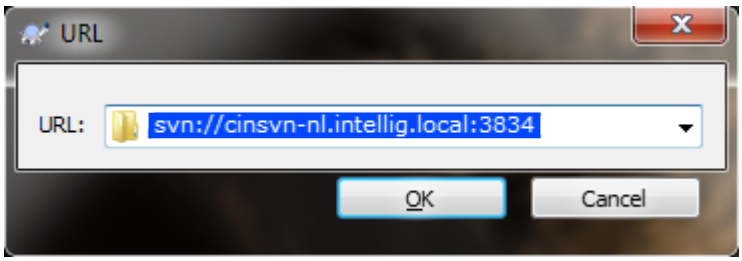
2.3.2 Browsing the Subversion Repository

To browse the Subversion repository:

1. Open up an Explorer window.
2. Right-click on any folder, and click on TortoiseSVN **|** Repo-browser.

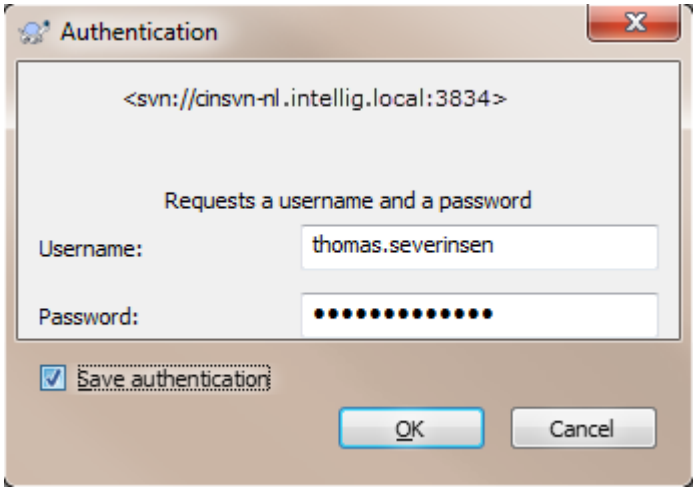


3. Paste this URL into the address bar:
   svn://cinsvn-nl.intellig.local:3834



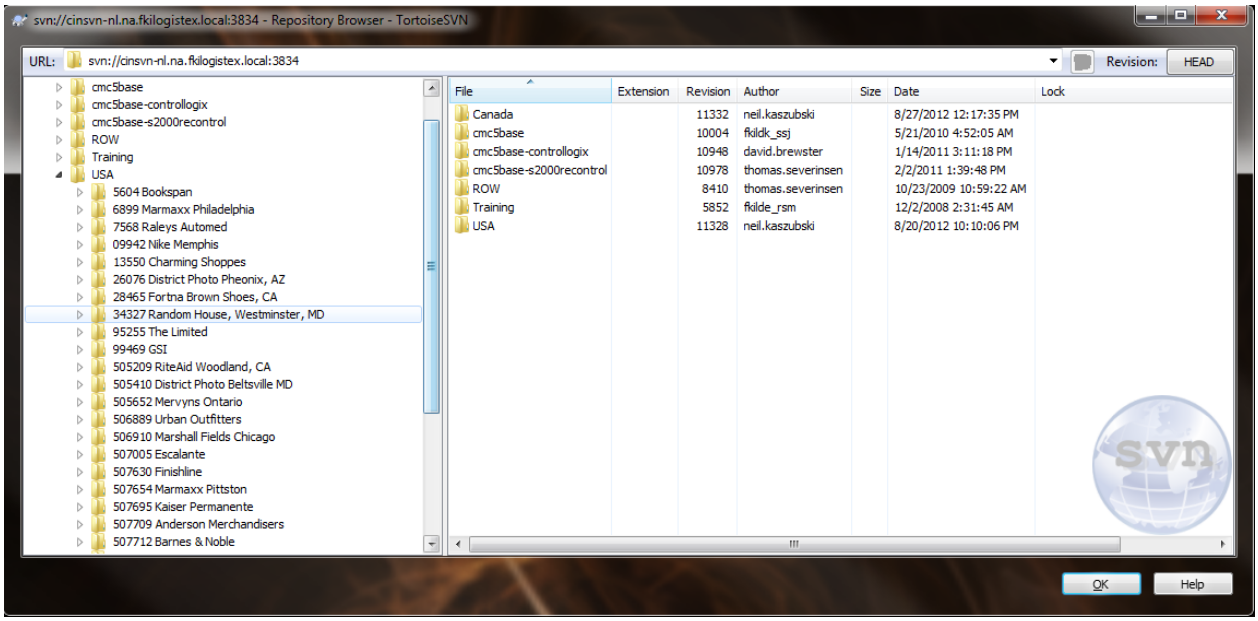
4. Click OK.

5. If promoted for an authentication, enter your Subversion credentials.



6. Ensure the Save Authentication box is checked, and click OK.

   The repository can now be browsed.



   To download a particular version of source code, drag the file onto your desktop or local drive.

2.3.3 Version Numbering and Naming

The Revision number of the project should follow this standard:

- The version number during the **Emulation** phase of the project should start at **E01** and increment E02, E03, etc., for each update
- The version number during the **Factory Testing** phase of the project should start at **F01** and increment F02, F03, etc., for each update
- The version number during **Commissioning** should start at **C01** and increment C02, C03, etc., for each update
- When commissioning efforts are completed or the site is using the system for live production, a new version of the PLC code running at the time must be stored in the Subversion repository with revision 100. This must be done whether or not any changes were made since the last version was checked in.

Due to different limits on file naming conventions, there are different standards for Allen Bradley projects compared to Siemens projects.

Allen Bradley Projects

These source code file names should follow this standard.

Description of file name parameters, with examples:

Project Name/Location (example: RossSE)
PLC Processor (type: Chute, Sorter, or CNETxx. If CNET, xx = ControlNet Number)
Sorter Number (example: TT01)
RSLogix 5000 / Firmware Flashed Version Number (example: V19)
Rev number [xxx] (xxx = starting Revision Number. This number stars at 100, and increments by one for each subsequent revision.)

Examples (place underscores between each element of the file name parameters described above):

RossSE_Chute_TT01_V19_Rev_100.ACD
RossSE_Sorter_TT01_V19_Rev_100.ACD
RossSE_CNET01_TT01_V19_Rev_100.XC

Note

The .ACD files are the files that define the program code, and the .XC files are ControlNet files that define the layout and configuration of the IO network. These are decided for us by the Allen Bradley software.

Siemens Projects

Siemens program file names can only have eight (8) characters, but a project can contain multiple processors and sorters. If multiple sorters have the same function, such as a packing sorter, they should be combined in the same project. These source code file names should follow this standard.

Description of file name parameters, with examples:

Project Name/Location/Sorter (type four characters that are most fitting)
Rev number [xxx] (xxx = starting Revision Number. This number stars at 100, and increments by one for each subsequent revision.)

Examples (place an underscore between each element of the file name parameters described above):

Pywc_100.zip (example for Payless West Coast, version 100)
T555_101.zip (example for Target DC# T555, version 101)
LBIP_c05.zip (example for Limited Brands commissioning version 05, containing upper, lower, and common packing sorter processors)

Files stored in the repository:

- PLC source files
- IO network configuration files
  - ControlNet
  - Profibus

2.3.4 Directory Structure

The following directory structure must be adhered to.

```
root
|-- country
     |-- customer
          |-- site
               |-- sorter (if multiple sorters per site)
                    |-- CPU
                    |   |-- tags
                    |   +-- trunk
                    |-- ControlNet
                        |-- tags
                        +-- trunk

example:

root
 |-- USA
 |    |-- Kohls
 |    |    |-- Edgewood,MD
 |    |    |    |-- Chute CPU
 |    |    |    |    |-- tags
 |    |    |    |    +-- trunk
 |    |    |    |-- Sorter CPU
 |    |    |    |    |-- tags
 |    |    |    |    +-- trunk
 |    |    |    +-- SST
 |    |    |         |-- tags
 |    |    |         +-- trunk
 |    |    |
 |    +-- Ross
 |         +-- RockHill,SC
 |         |    |-- Chute CPU
 |         |    |    |-- tags
 |         |    |    +-- trunk
 |         |    |-- Sorter CPU
 |         |    |    |-- tags
 |         |    |    +-- trunk
 |         |    +-- ControlNet
 |         |         |-- tags
 |         |         +-- trunk
 |         |
```

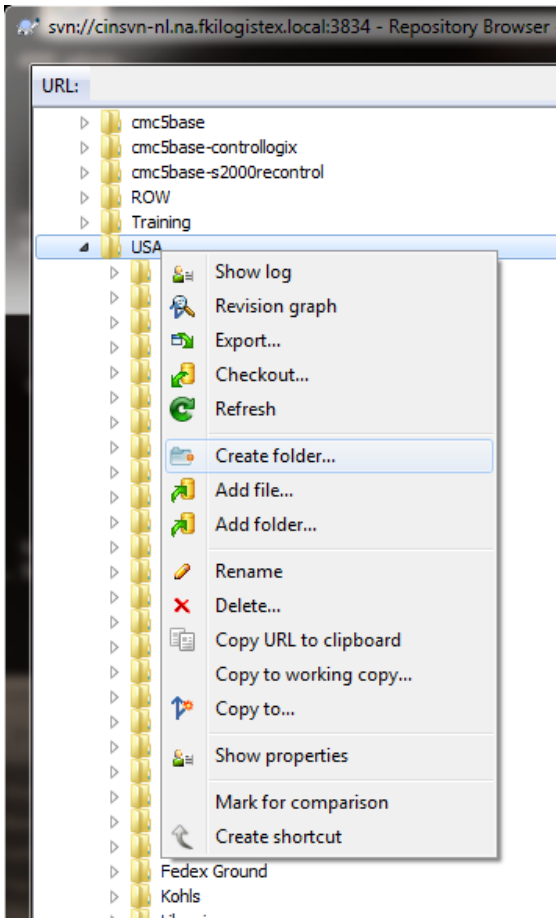
```
|      +-- Perris,CA
|              |-- Chute CPU
|              |    |-- tags
|              |    +-- trunk
|              |-- Sorter CPU
|              |    |-- tags
|              |    +-- trunk
|              +-- ControlNet
|                   |-- tags
|                   +-- trunk
|
+-- Canada
     +-- CanadianPost
          +-- Edmonton,SC
          |    |-- Chute CPU
          |    |    |-- tags
          |    |    +-- trunk
          |    |-- Sorter CPU
          |    |    |-- tags
          |    |    +-- trunk
          |    +-- ControlNet
          |         |-- tags
          |         +-- trunk
          |
       +-- Montreal,XX
              |-- Chute CPU
              |    |-- tags
              |    +-- trunk
              |-- Sorter CPU
              |    |-- tags
              |    +-- trunk
              +-- ControlNet
                   |-- tags
                   +-- trunk
```

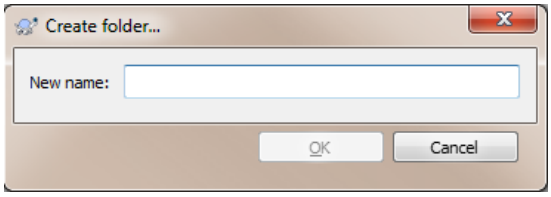
The tags directory should contain the previous revision of the program, so that it can be restored at any time.

The trunk directory should always contain the latest version of the program.

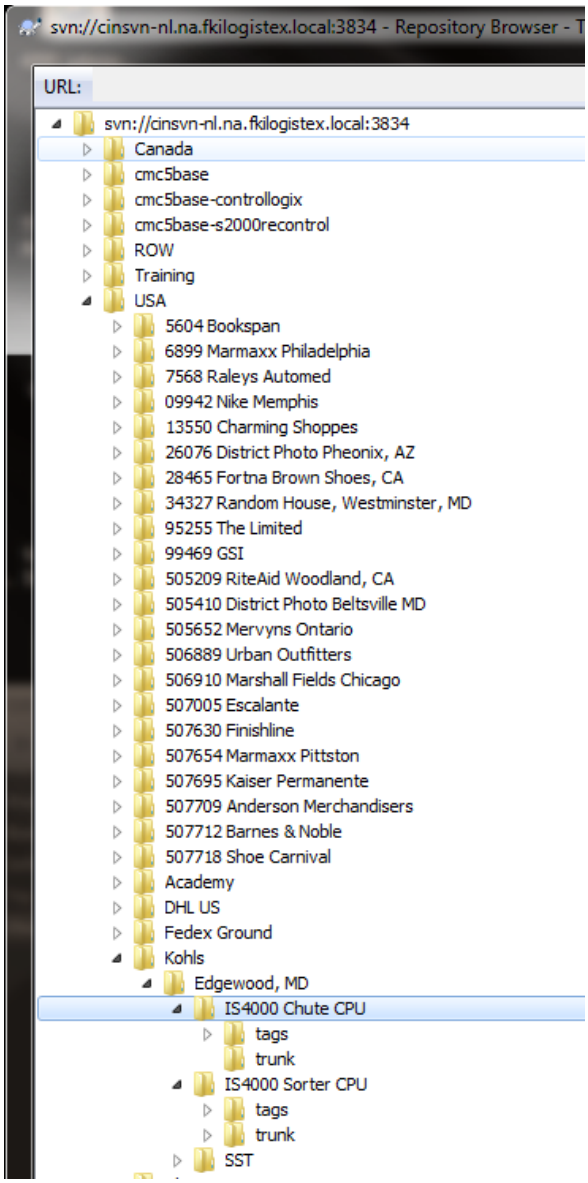
2.3.5 Adding a New Project

1. Create a new project folder by right-clicking on the country where the project is implemented and select Create folder.

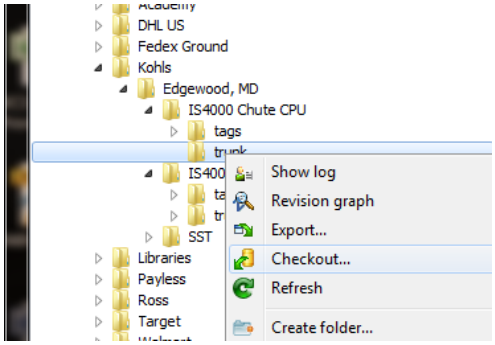




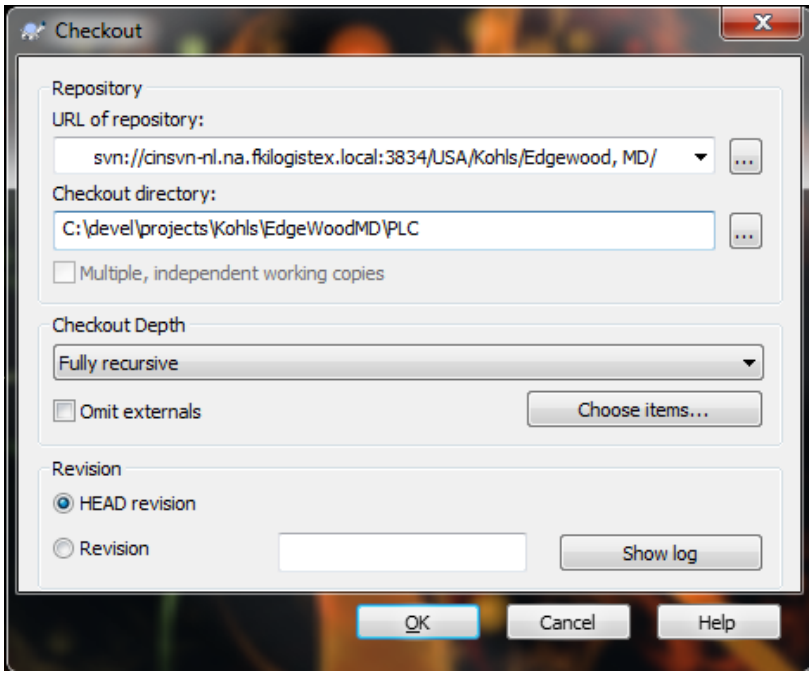
2. Create folders to match the following directory structure:



3. Check out the trunk folder in order to be able to add files to the repository. Check out the trunk folder to your local drive by right-clicking on the trunk folder.



4. Change the Checkout directory to the folder on your local hard drive where you wish to locate the folder, then click OK.

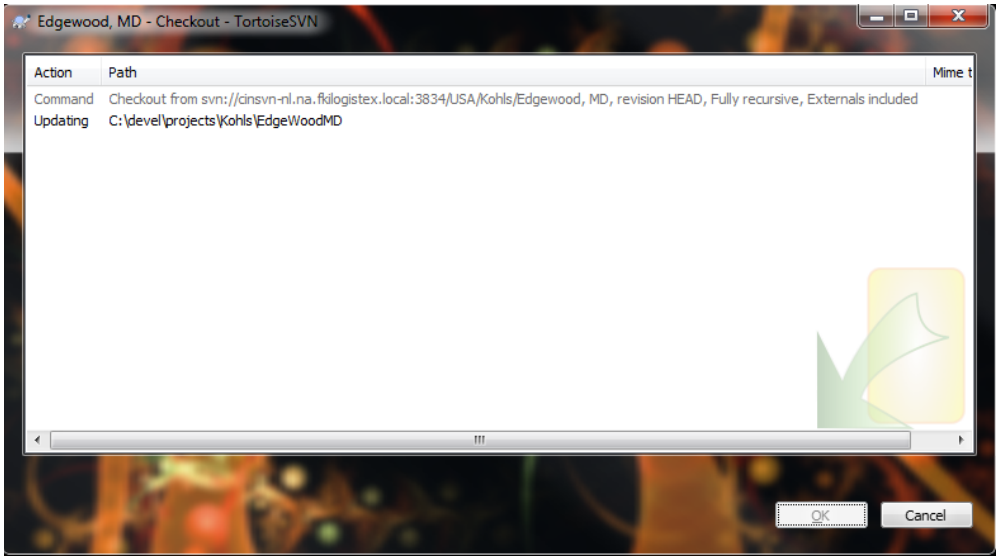


Note

Only the folder is checked out. The program will be added later.

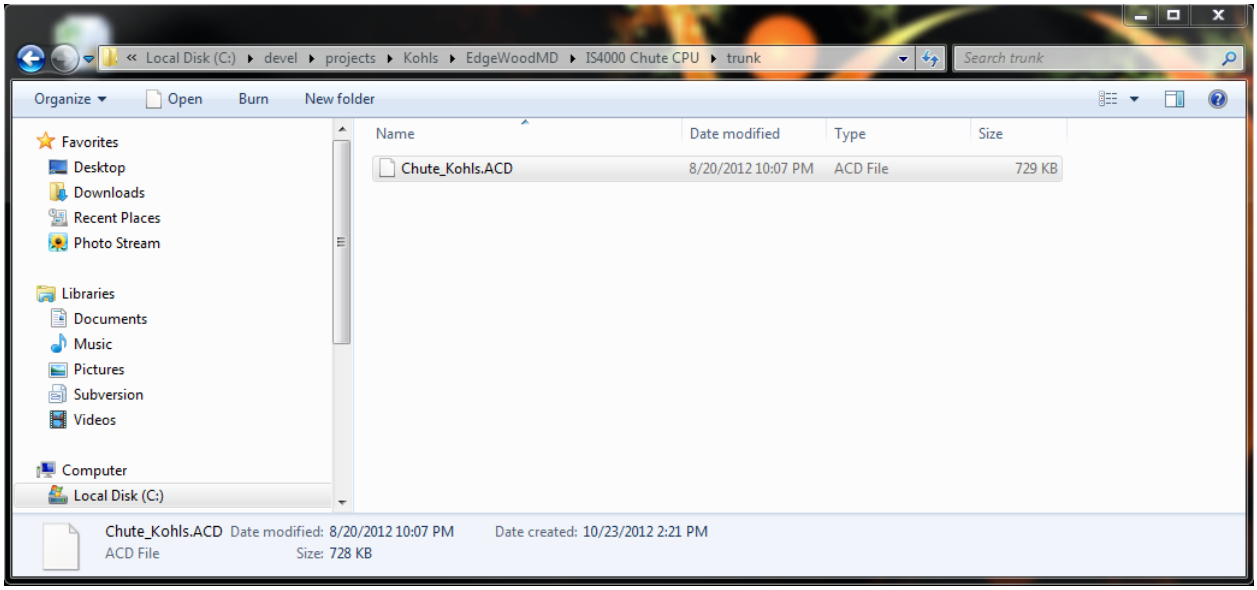
5. Click OK to close the dialog box.

   There will now be a directory structure on your local hard drive that reflects what is in the repository.

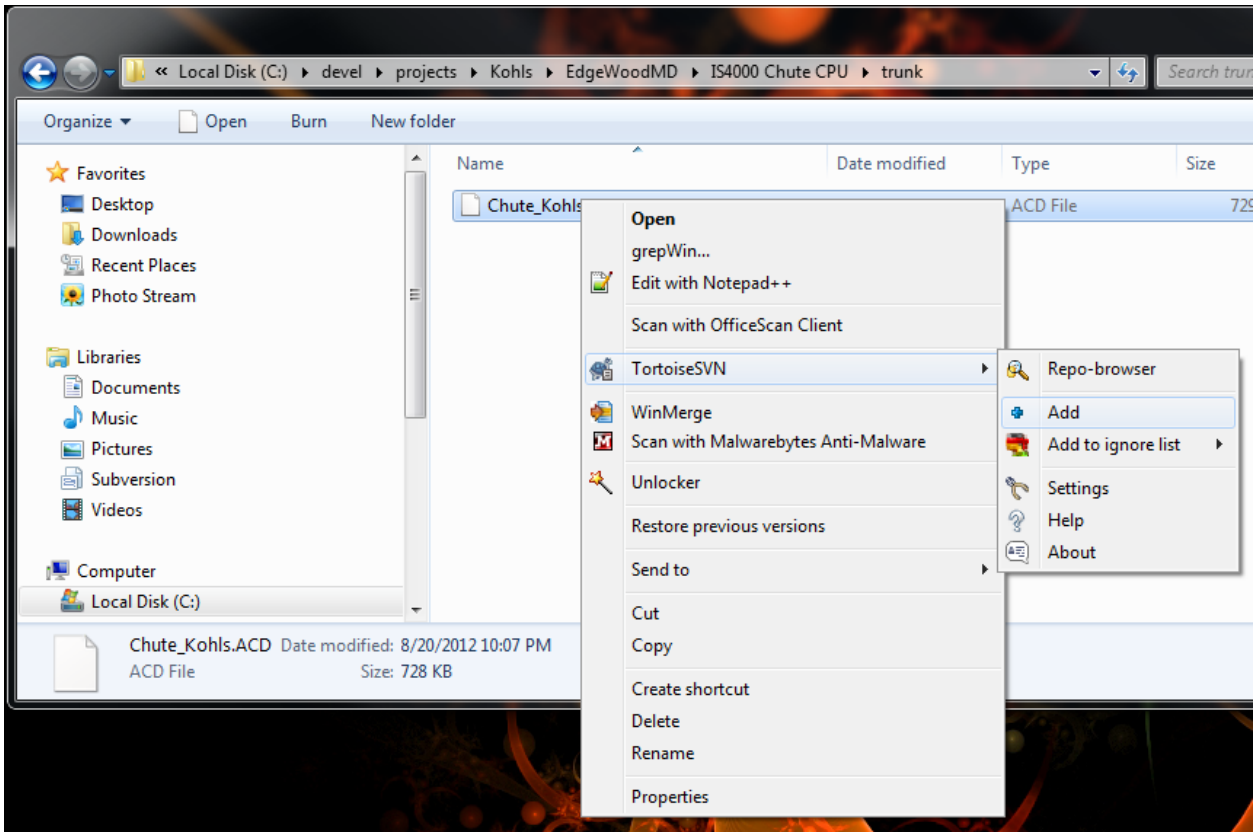


   Ignore any .svn directories; they are used by Subversion to keep track of files.

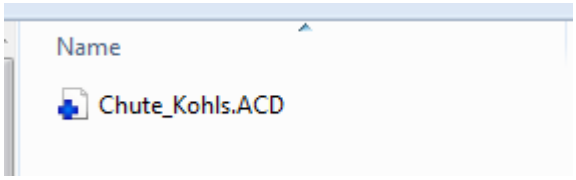
6. Now, to add the PLC software, copy the file into the appropriate directory on your local hard drive.



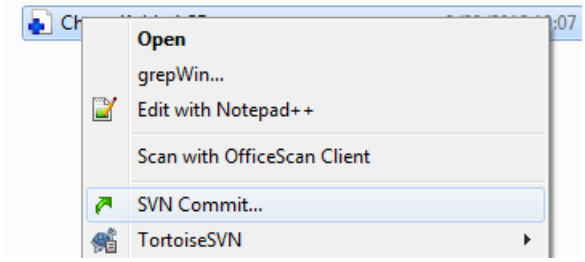
7. Add the file to the repository by right-clicking the file and select TortoiseSVN **|** Add.



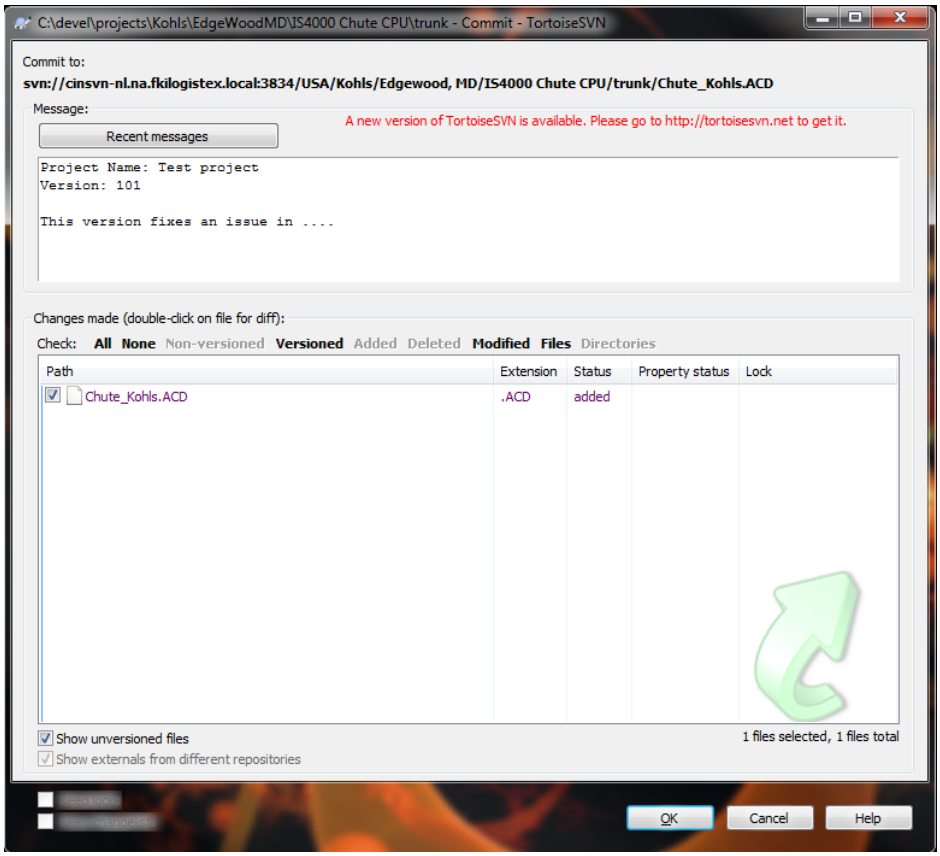
There should now be a blue “+” sign icon for the file you just added. You may need to refresh (F5).



8. To check the file into the repository, right-click on the file and select SVN Commit.



9. Enter revision notes in the message window and click OK to commit.



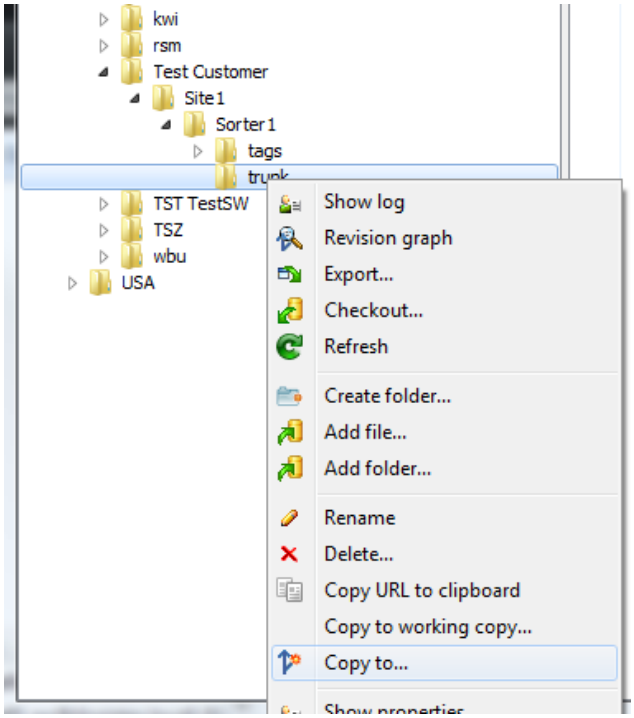
Your local copy of the file has now been committed to the repository. The icon should now be a green checkbox:  (You may need to refresh – F5).

2.3.6 Tagging and Versioning

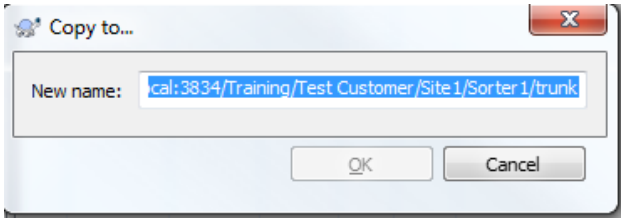
After you have committed a new version to the trunk directory, you must create a new tag.

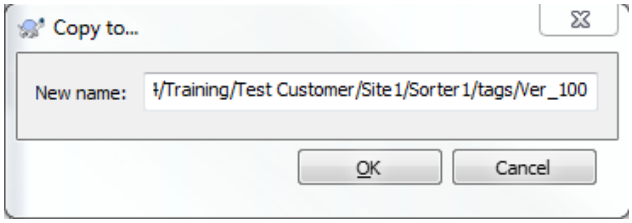
1. Open the repository browser.

2. Right-click on the trunk folder where you have just committed the new version, and select Copy to.



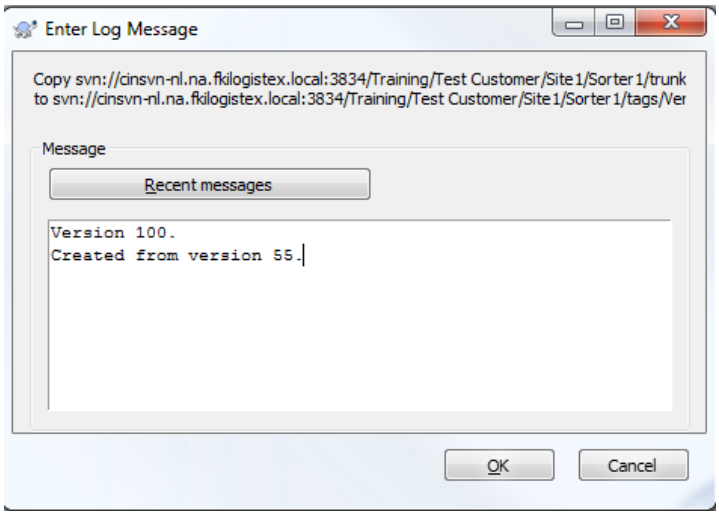
3. Replace “trunk” with “tags/Ver_YYY” – where YYY is the version number.





4. Click OK.

5. Enter any revision notes and click OK.



This enables us to easily restore a previous revision of software by browsing to the tags/Ver_YYY folder and then dragging the file to your desktop/local drive.

Note

When commissioning efforts are completed, a new version of the PLC code running at the time must be stored in the Subversion repository with revision 100. This must be done whether or not any changes were made since the last revision.

Index

R

S

T

U

V

W

X