

# Software Delivery Integration and Source Code Management for Suppliers

## Document Information

Author	
Version	1.0
Version Date	8/6/2012
Status	final
Approved by	
Reference	not applicable

## Table of Contents

1	Introduction.....	3
1.1	Purpose.....	3
1.2	Conventions in this Document.....	3
2	Guidelines .....	4
2.1	Directory Structure.....	4
2.2	General Guidelines.....	4
2.3	Best Practices for Contractors.....	4
3	Roles and Responsibilities .....	5
3.1	R4 Infra/Ex.....	5
3.2	Project Managers .....	5
3.3	Contractors.....	6
3.4	Access Rights.....	6
4	Main Tasks.....	6
4.1	Prepare the Delivery .....	6
4.2	Deliver the Installation Sources .....	7
4.3	Notify the Project Manager.....	9
4.4	Installation Request (DMT/DMP) .....	9
4.5	Installation of the Application .....	9
5	Subversion Clients .....	10
5.1	Windows Client .....	10
5.2	Command Line Client.....	13
5.3	Web Interface.....	14
6	Additional Information .....	15
6.1	Acronyms, Abbreviations .....	15
6.2	URLs.....	15
6.3	Reference Documents .....	15
6.4	Contact Data.....	15

# 1 Introduction

Apache™ Subversion® is a version control system that manages revisions of files and directories and keeps a history of changes.

## 1.1 Purpose

The Publications Office receives software deliveries (application releases, updates and patches) from contractors to be installed in its computer environment. This document describes the use of Subversion (SVN) in software delivery and installation.

Software deliveries usually arrive as binary files, together with the corresponding source code. Subversion will be used to store these files and any other files delivered with the software for safekeeping.

Contractors are obliged to systematically deliver the source code together with the binary installation files. The goal is to have source and binary files in the same place so that software deliveries arrive at the Publications Office in a well-structured way, and are uniquely identifiable.

This document is intended to help contractors and suppliers with the preparation of a software delivery to the Publications Office.

## 1.2 Conventions in this Document



Text that follows this icon indicates a warning; please make sure that you follow these instructions to the letter.



Text that follows this icon indicates important information that should be taken into account.

## 2 Guidelines

### 2.1 Directory Structure

Each Subversion project and sub-project has the following default directory structure:

/trunk	holds the main line of development
/branches	contains branch copies
/tags	contains tag copies
/deliveries	the software delivery in compressed archive format and documentation required to perform the installation
/installations	contains all files and folders created and modified during installation

Additionally, a project may contain optional folders inside the *trunk* or *branches* folder:

/build	build scripts to re-build the binary files from the delivered source code
/doc	all documentation not related to the installation of a specific delivery
/src	source code of the application, unzipped
/test	files related to tests of the application; test scripts, log files and reports

Projects can include sub-projects. The following image shows the typical directory structure for a project without subprojects:



Figure 1: default directory structure of SVN repository



If your project contains sub-projects or is composed of several components, please contact the SVN Administrators of the Publications Office to discuss how to best organise the source code.

### 2.2 General Guidelines

- The Subversion repository should only contain source code that corresponds to software deliveries. Contractors should have their own source code repository to support the development lifecycle. If a contractor would like to use the Subversion repository of the Publications Office during development, an agreement has to be obtained first.
- The contractor should notify the Publications Office before a large delivery (more than 100 MB) is to be uploaded into the repository.

### 2.3 Best Practices for Contractors

Contractors typically manage the source code related to the development of their application in the *trunk/src* folder. Aside from the standard directory structure (please

see section 2.1 *Directory Structure*), the Publications Office does not impose any rule on how the contractor organises the source code. However, contractors are encouraged to adopt a best practice that is followed during the entire project life cycle. An example best practice is available on the Subversion web site (please refer to appendix 6.2 *URLs* for a hyperlink).

The Publications Office recommends adopting the following rules as part of the best practice:

- The *trunk* folder should be used for the main development.
- Branches should be used to indicate diversions from the trunk for a specific purpose (usually bug fixes and patches to a previous release).  
When you choose the name for your branch, please keep it simple (simple descriptive text, no nested folder structure).
- Tags should be created to identify a specific release of an application.  
When you choose the name for your tag, please keep it simple (simple descriptive text, no nested folder structure).
- Each delivery should be tagged using the *tags* folder; the tag should contain a short descriptive name of the release.
- Branches and tags should be created directly on the server by using the `svn copy` command or the *Copy to...* menu item in TortoiseSVN; this will lead to a more space-efficient structure as SVN uses hard links to preserve space.

### 3 Roles and Responsibilities

#### 3.1 R4 Infra/Ex

R4 Infra/Ex (the Exploitation section of the Infrastructures unit of the Publications Office) is in charge of administration and management of Subversion; they create repositories for new projects, create users and manage access and permissions of the project repositories.

Infra/Ex creates the Subversion repository for a particular project. This makes the repository available under the name of the project, for example:

`https://webgate.ec.europa.eu/publications/svn/PROJECT1/`  
URL to access the Subversion server from a Subversion client

Within Infra/Ex, the QA team (quality assurance team) consists of the integrators and testers, who check the provided delivery, perform the installation, verify the installation and perform basic tests of the application after installation.

The Infra/Ex integrators and testers install the delivery that corresponds to the revision number provided in the DMT/DMP request and test the final installation.

#### 3.2 Project Managers

Project managers accept the delivery and create the JIRA DMT/DMP installation request.

### 3.3 Contractors

Contractors who develop software for the Publications Office provide the source code and the installation binaries. Contractors will manage the software deliveries in the *deliveries* root folder of their project.

The contractor has to have a client program installed that enables the use of the Subversion repository. In a Microsoft Windows environment, the Publications Office recommends TortoiseSVN, a Subversion client, implemented as a Microsoft Windows shell extension. In a Unix environment, a command-line client is available from the Subversion home page.

The contractor delivers the application by checking in all contractually specified delivery files (e.g. binary distribution and documentation required for the installation) into the *deliveries* folder. The *trunk* and *branches* folder structure is used to manage the source code and associated documentation related to a software delivery.

### 3.4 Access Rights

In general, personnel internal to the Publications Office have read-only access to all repositories on the Subversion server; external personnel will only be granted access to the repository that is related to their development work.

Write access is assigned based on the directory structure; contractors and project managers have full control over the *trunk*, *branches*, *tags* and *deliveries* folders; R4 Infra Ex has only read access to these folders.

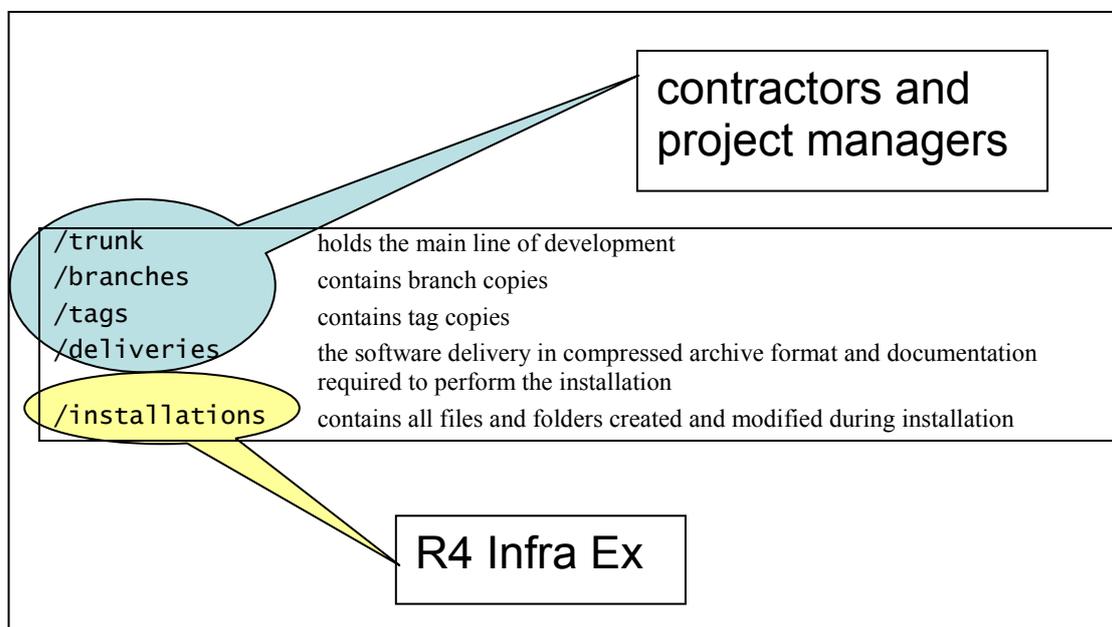


Figure 2: Write Access to Repositories

## 4 Main Tasks

### 4.1 Prepare the Delivery

Prepare the new delivery on a local folder on your workstation.

The local deliveries folder has to be cleaned up (emptied) so that it contains only files related to a specific delivery.



Figure 3: empty deliveries folder before adding new delivery



If you do not clean the deliveries folder before uploading the next delivery, files from the last and the current delivery will be merged, which will cause problems during the installation.

### 4.2 Deliver the Installation Sources

To deliver the packaged installation sources, commit the new delivery using TortoiseSVN; right-click the folder that contains your delivery and select *SVN Commit*.

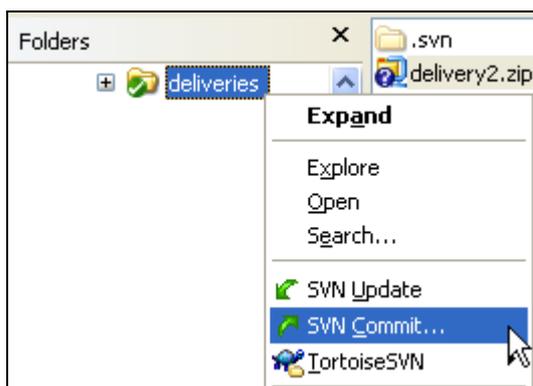
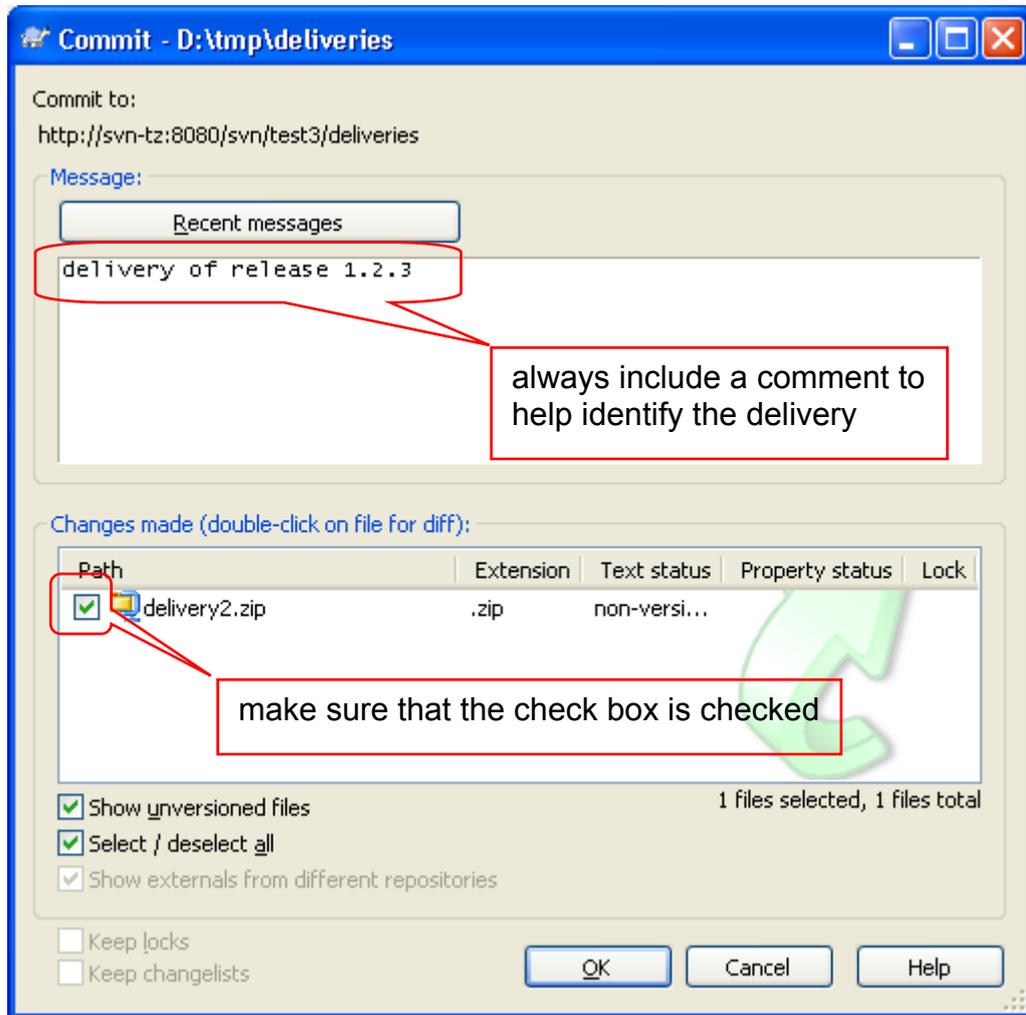


Figure 4: commit new delivery with TortoiseSVN

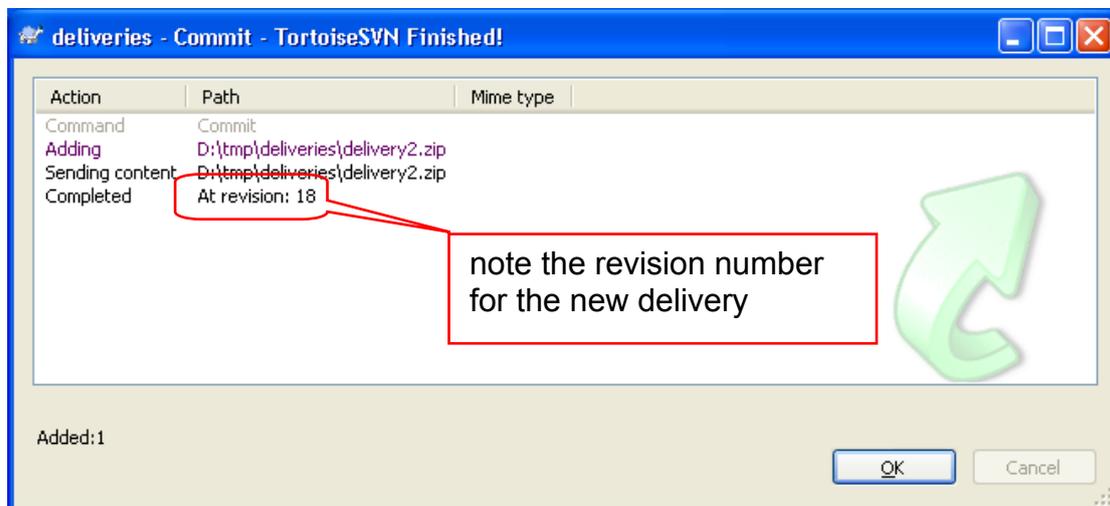


TortoiseSVN requires a user name and password to log on to the SVN server; should you not have your user credentials, please contact [OPDL SVN ADMINS](#) for assistance.



**Figure 5: TortoiseSVN Commit dialog box**

A dialog box will appear and confirm the successful commit. Please take note of the revision number.



**Figure 6: TortoiseSVN Commit finished**



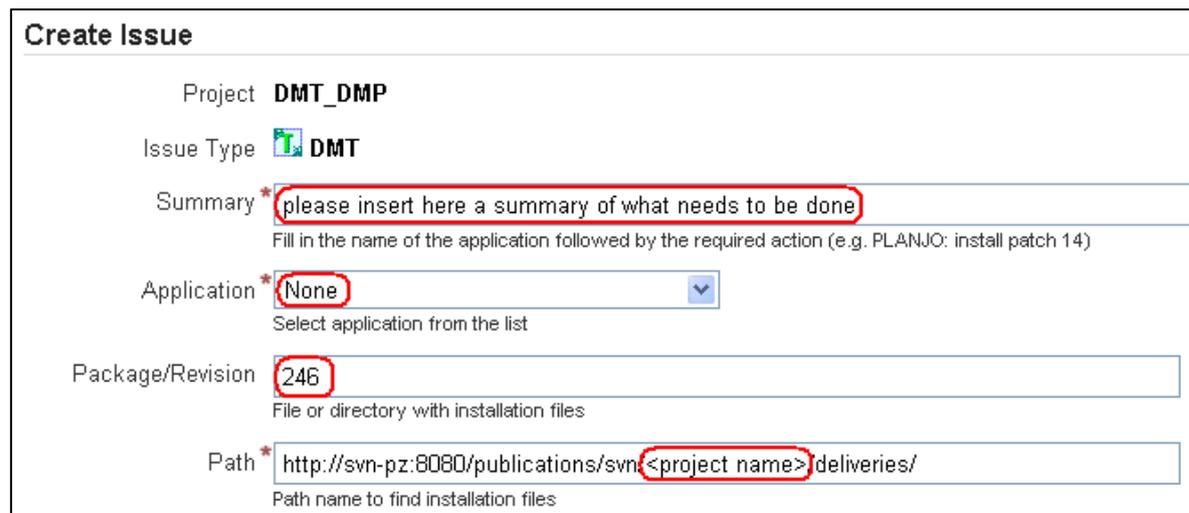
If you are not able to upload the software delivery to the SVN server yourself, please contact the project manager at the Publications Office for assistance.

### 4.3 Notify the Project Manager

Notify the project manager of a new delivery with a request for installation and include the relevant revision number that was obtained in the previous step as reference.

### 4.4 Installation Request (DMT/DMP)

The project manager creates a DMT/DMP (installation request) and includes a reference to the Subversion project repository (URL, for example <https://webgate.ec.europa.eu/publications/svn/<project>/deliveries/>) in the field *Path* and a reference to the revision number in the field *Package* of the DMT. This creates the link between the DMT/DMP request in Jira and the delivery in Subversion.



**Create Issue**

Project **DMT\_DMP**

Issue Type  **DMT**

Summary \*   
Fill in the name of the application followed by the required action (e.g. PLANJO: install patch 14)

Application \*    
Select application from the list

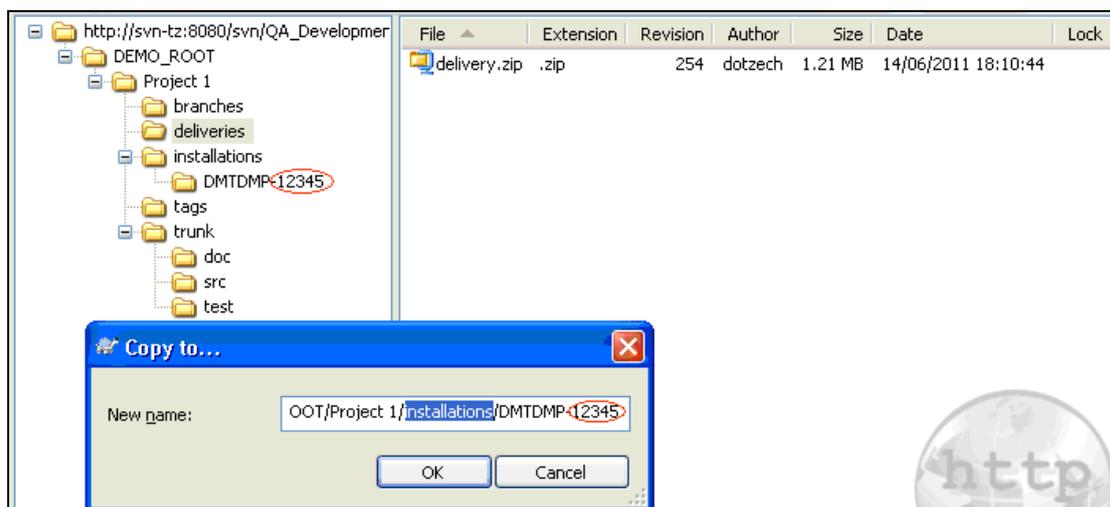
Package/Revision   
File or directory with installation files

Path \*   
Path name to find installation files

Figure 7: the project manager creates an installation request

### 4.5 Installation of the Application

The Infra/Ex integrator copies the *deliveries* folder in SVN to a subfolder of the *installations* folder identified by the DMTDMP number.



**Figure 8: Infra/Ex processes the installation request**



Once the installation is finished, you will find a copy of the original installation instructions, renamed to QA\_<original file name>, which contains the remarks of the integrator who performed the installation



It is mandatory that you take all these comments and remarks into account when you prepare the next software delivery.

## 5 Subversion Clients

Subversion is the name of the server product; in order to access the repositories on the Subversion/SVN server, a Subversion client has to be installed. The most popular client on Windows, TortoiseSVN, installs as a Windows Explorer extension; to open TortoiseSVN you need to right-click on a folder. The Unix client installs a set of command line tools.

### 5.1 Windows Client

The Publications Office recommends to use TortoiseSVN. Alternative SVN clients are available. The screenshots in section 4 Main Tasks demonstrate the use of the TortoiseSVN Windows client for the most common tasks.

#### 5.1.1 TortoiseSVN Windows shell extension

The Windows shell extension makes the Windows Explorer menu context sensitive; depending on whether a folder is known by SVN, you will see a different menu.

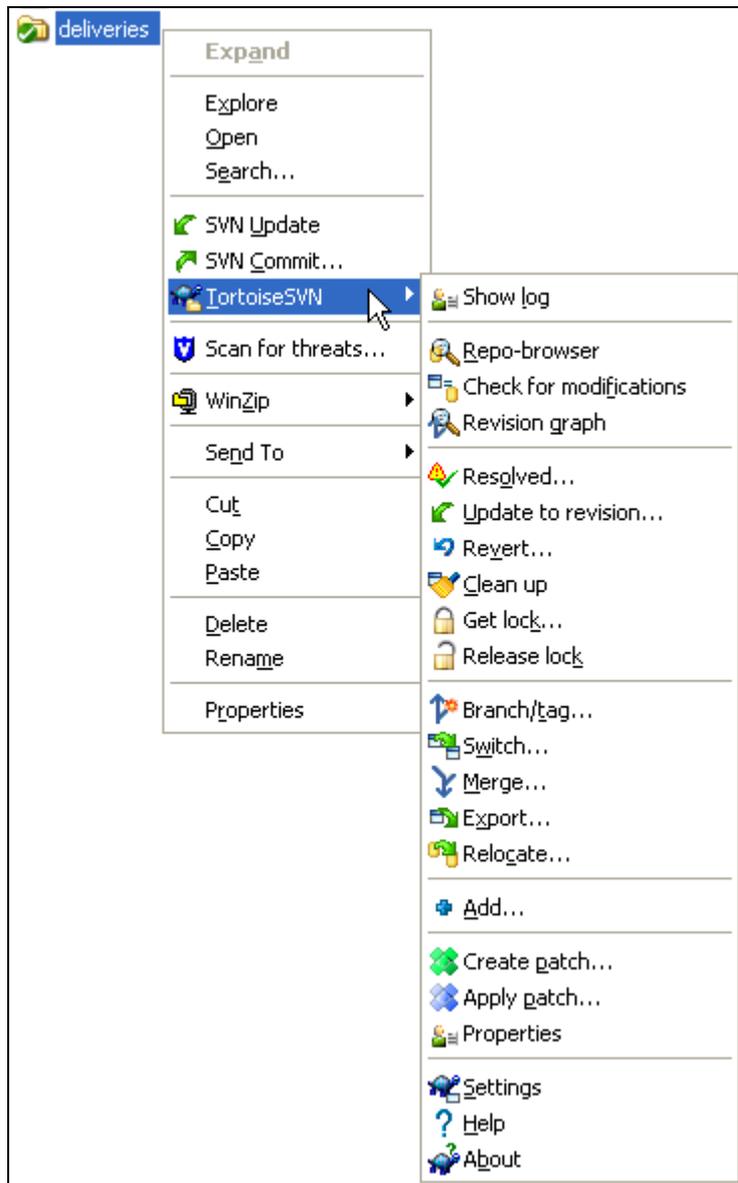


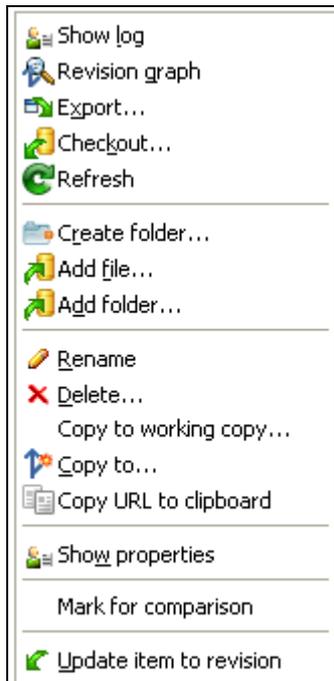
Figure 9: Explorer menu on folder managed by SVN

### Short description of the most useful menu options

- SVN Update: updates the local folder with the latest (HEAD) revision on SVN
- SVN Commit: updates the SVN repository with the contents of the local folder
- Show log: shows a history of changes for this folder
- Repo-browser: opens the repository browser; requires the URL to your repository and your login credentials

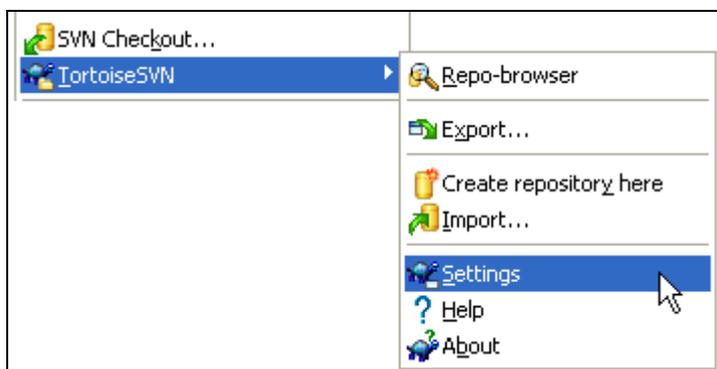
#### 5.1.2 TortoiseSVN Repo-browser

The TortoiseSVN repository browser shows you the contents of your repository on the SVN server. You can use the Repo-browser to upload and download files and directories to and from the SVN server.



**Figure 10: Repo-browser menu items**

Right-click anywhere in a Windows Explorer Window; in the pop-up menu, select Settings from the TortoiseSVN sub-menu to get to the screen where you can configure the general settings of TortoiseSVN:



**Figure 11: TortoiseSVN sub-menu**

Here you will be able to configure for example the proxy settings if you are behind a web proxy:

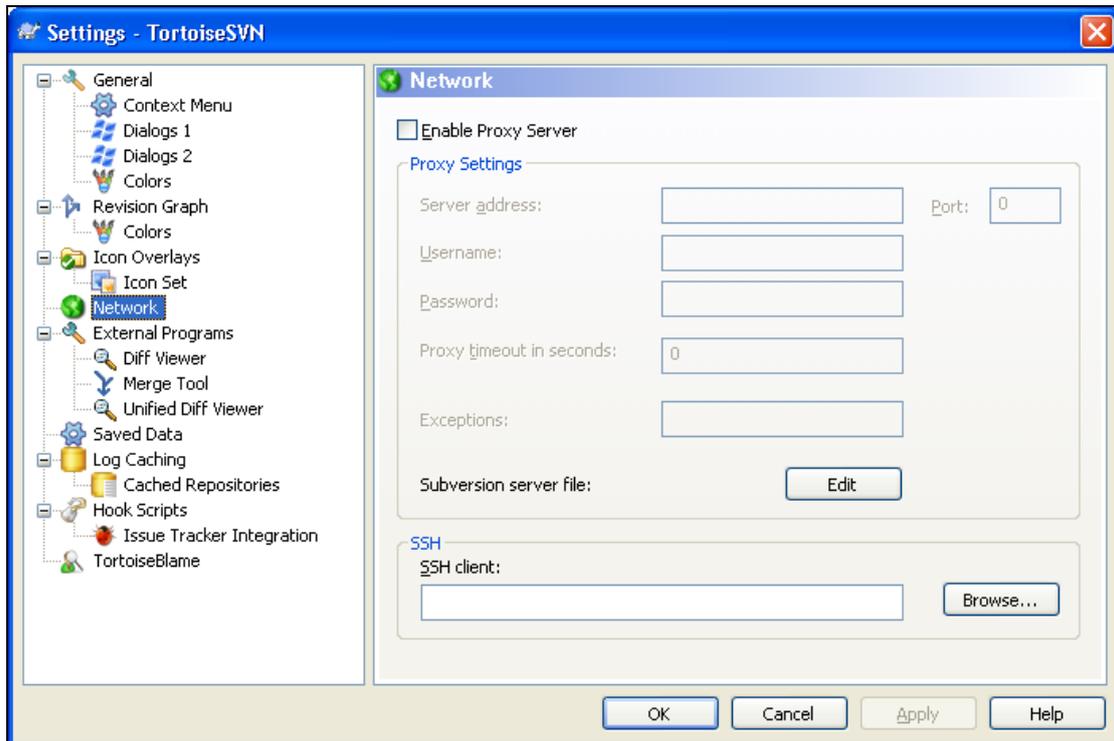


Figure 12: TortoiseSVN Network Settings

### Short description of the most useful menu options

- Show log: right-click on any item in SVN to get a list of the history of changes
- Export...: copies items to a local hard disk without including SVN information
- Checkout...: copies items to a local hard disk together with the SVN information in hidden directories .svn
- Copy to...: copies items within a SVN repository (uses links to preserve disk space)

## 5.2 Command Line Client

The following examples demonstrate how the command line client can be used instead of a Windows client to execute the steps of the procedure.

- Copy contents of deliveries folder to DMTDMP-12345 installations subfolder

```
svn cp -r 246 -m "copying delivery for installation of DMTDMP-12345"
"http://svn-pz:8080/svn/Project 1/deliveries" "http://svn-pz:8080/svn/Project 1/installations/DMTDMP-12345"
```

- List the differences between files previously checked out in the current directory and files in SVN:

```
svn status
A      DMTDMP-12345/
A      DMTDMP-12345/QA_install.doc
```

- Commit modified files to the SVN repository:

```
svn commit -m "QA document added"
Adding      DMTDMP-12345/
Adding (bin) DMTDMP-12345/QA_install.doc
Transmitting file data .
Committed revision 253.
```

- Execute the following SVN command to check out the *installations* folder:

```
svn co "http://svn-pz:8080/svn/Project 1/installations/DMTDMP-12345"
```

### 5.3 *Web Interface*

The contents of the SVN server can also be viewed using a web browser. Note that this access is read-only; to modify the repository on the SVN server you need an SVN client.

## 6 Additional Information

### 6.1 Acronyms, Abbreviations

Value	Definition
SVN	short for Apache Subversion
Jira	Atlassian Jira, an issue tracking tool
DMT	<i>demande de mise en test</i> , request to put in test
DMP	<i>demande de mise en prod</i> , request to put in production
Jira DMT/DMP	project to manage the workflow of changes by application installations
<project>	variable to be replaced by project name

### 6.2 URLs

URL	Description
<a href="http://subversion.apache.org/">http://subversion.apache.org/</a>	Apache™ Subversion® home page
<a href="http://tortoisesvn.tigris.org/">http://tortoisesvn.tigris.org/</a>	TortoiseSVN is a Subversion client, implemented as a Microsoft Windows shell extension.
<a href="http://tortoisesvn.net/support.html">http://tortoisesvn.net/support.html</a>	documentation and support options for TortoiseSVN
<a href="https://webgate.ec.europa.eu/publications/svn/">https://webgate.ec.europa.eu/publications/svn/</a>	external access to Subversion (this URL only works with an SVN client)
<a href="http://svn-pz:8080/websvn/">http://svn-pz:8080/websvn/</a>	WebSVN, access to Subversion through a web browser (access is read only and from OP internal network)
<a href="http://svn.apache.org/repos/asf/subversion/trunk/doc/user/svn-best-practices.html">http://svn.apache.org/repos/asf/subversion/trunk/doc/user/svn-best-practices.html</a>	Subversion Best Practices
<a href="http://svnbook.red-bean.com/">http://svnbook.red-bean.com/</a>	free book about Subversion

### 6.3 Reference Documents

Title	Description
<a href="#">Software Acceptance Procedure</a>	describes the steps to take to introduce an application or application update on the servers of the Publications Office
QA_install	document that contains the original installation instructions and any remarks and additional information recorded by the integrator during installation; it serves as feedback to the project manager and the contractor

### 6.4 Contact Data

Title	Description
<a href="#">DL OPOCE INFRA INT TEST</a>	email distribution list for the Infra/Ex integration and test team
<a href="#">OPDL SVN ADMINS</a>	email distribution list for the SVN administrators