Version control using GIT

straton user guide – Rev. 5

sales@straton-plc.com

CP_R31_006_H

Content

1. Overview

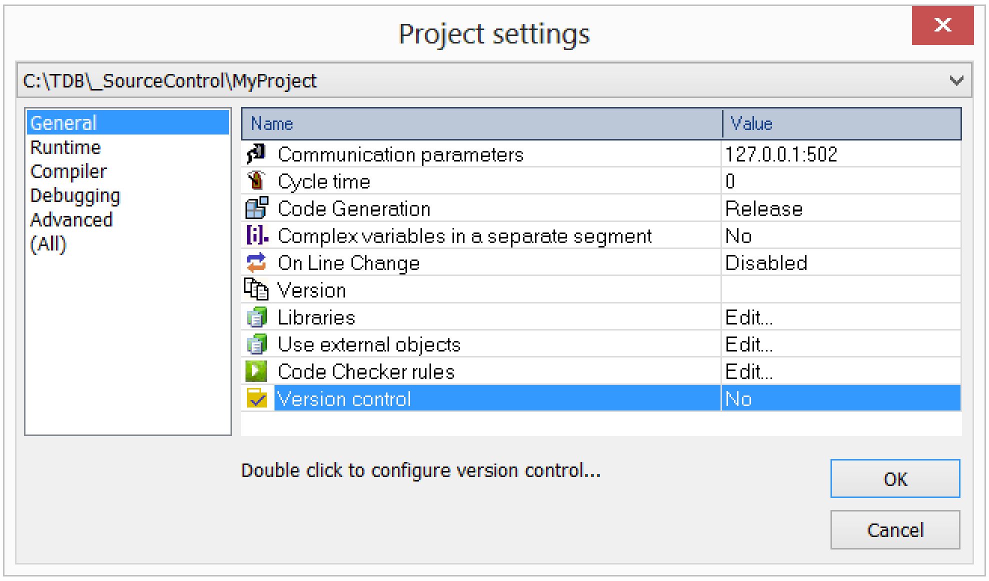
This document introduces the use of GIT as Version Control in the straton Editor.

2. Requirement and setup

Straton Editor version 11 is required.

3. Enable Version Control for a project

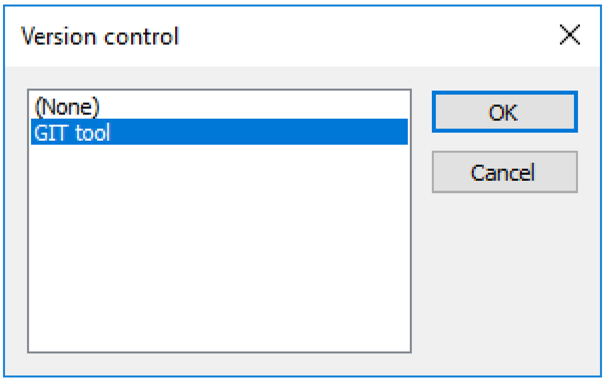
From the "General" page of the Project Settings box, double click on the "Version control" option:



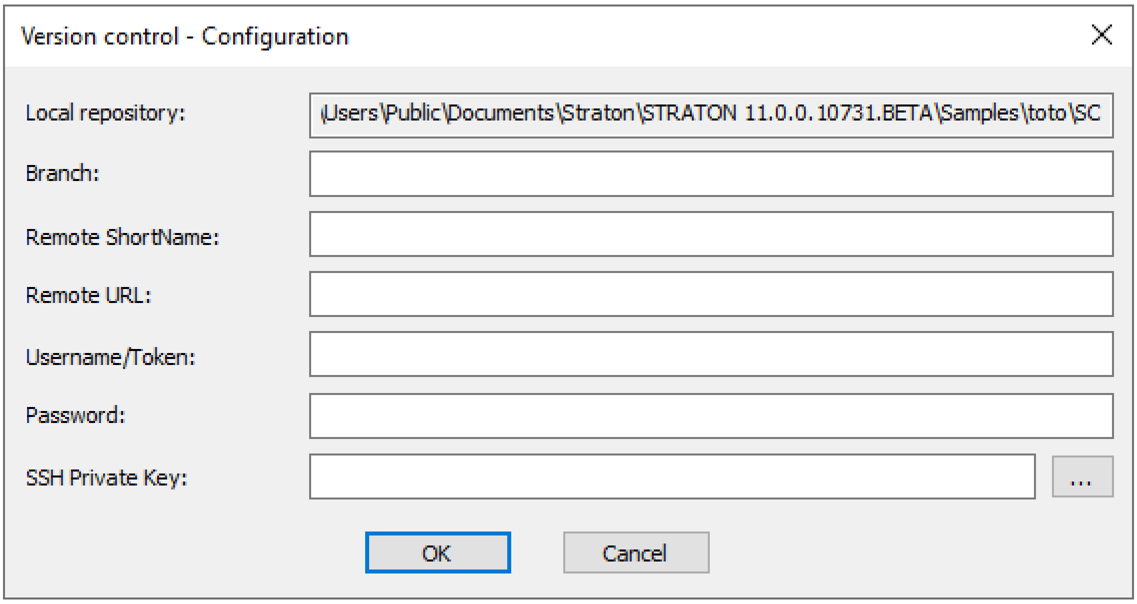
Name	Value
Communication parameters	127.0.0.1:502
Cycle time	0
Code Generation	Release
Complex variables in a separate segment	No
On Line Change	Disabled
Version	
Libraries	Edit...
Use external objects	Edit...
Code Checker rules	Edit...
Version control	No

Double click to configure version control...

Then select "GIT Tool":



Now you need to configure the connection to your GIT repository:



Version control - Configuration

Local repository:	\Users\Public\Documents\Straton\STRATON 11.0.0.10731.BETA\Samples\toto\SC
Branch:	
Remote ShortName:	
Remote URL:	
Username/Token:	
Password:	
SSH Private Key:	

- Local repository: this is the path to the GIT local repository. This path is set automatically by straton and cannot be changed.
- Branch: name of the local branch. If connected to a remote repository, and the remote branch doesn't exist, it will be created automatically when pushing to the remote repository.
- Remote ShortName: ShortName of the remote repository. Leave this field empty if you don't want to use remote repository.
- Remote URL: URL of the remote repository. URL can start with http://, https:// or ssh:// . Leave this field empty if you don't want to use remote repository.
- Username / Token: Username or Personal access Token (part 6.3) used to access remote repository
- Password: Password used to access remote repository

- SSH Private Key: complete path to the SSH private key file.

  Note that the public key file (.pub file) must have the same name as the private key file and must be in the same folder. (more info in part 6.2)

  for example:

my_ssh_key	*(private key file)*
my_ssh_key.pub	*(public key file)*

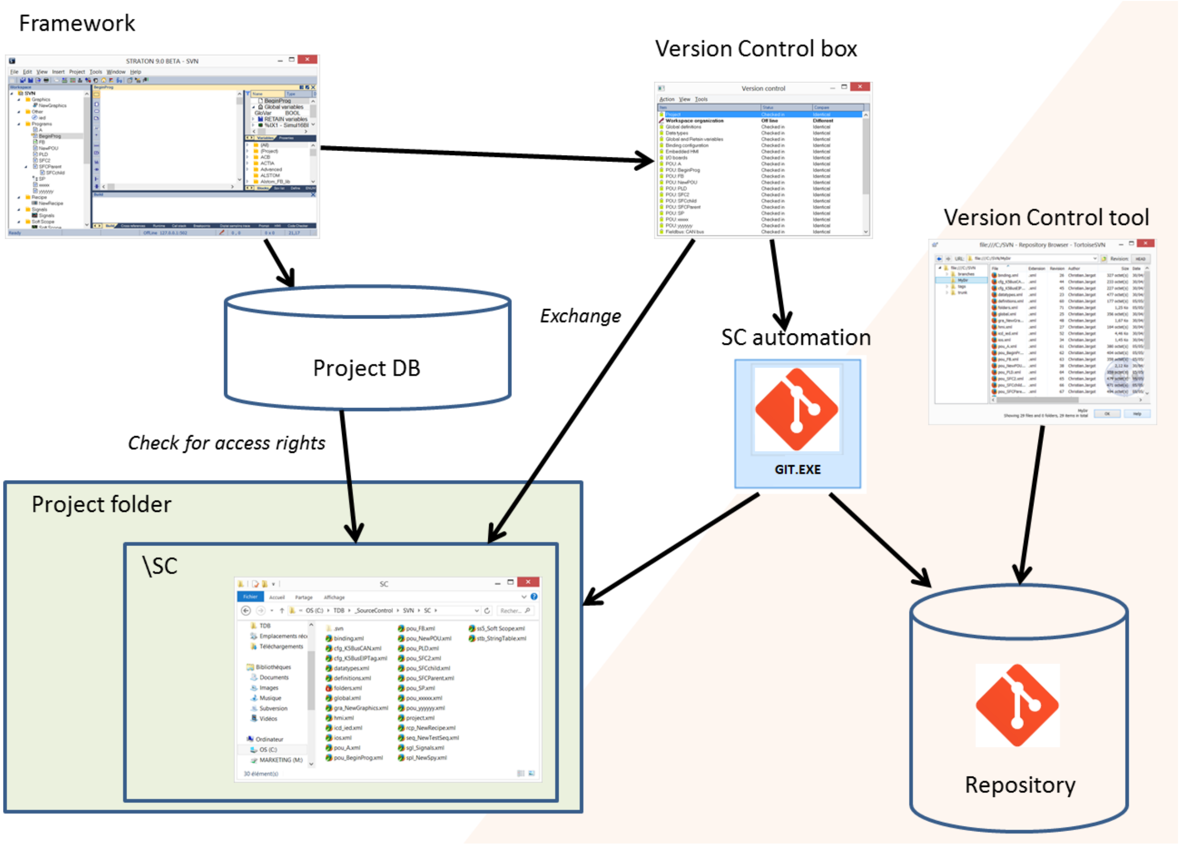
When you press OK, straton will verify the branch and try to connect to the remote repository if you have specified it.

After connection, the Version Control box is opened. You can re-open the Version Control at any time using the “Project / Version Control” menu.

The components of the straton projects are stored in the repository as XML files. straton creates an exchange area where XML files are exported and shared with GIT as a local copy. This area is the directory “\SC” in the project folder.

Due to the GIT’s behavior, all the XML files remain in Read/Write mode on the disc.

Architecture:

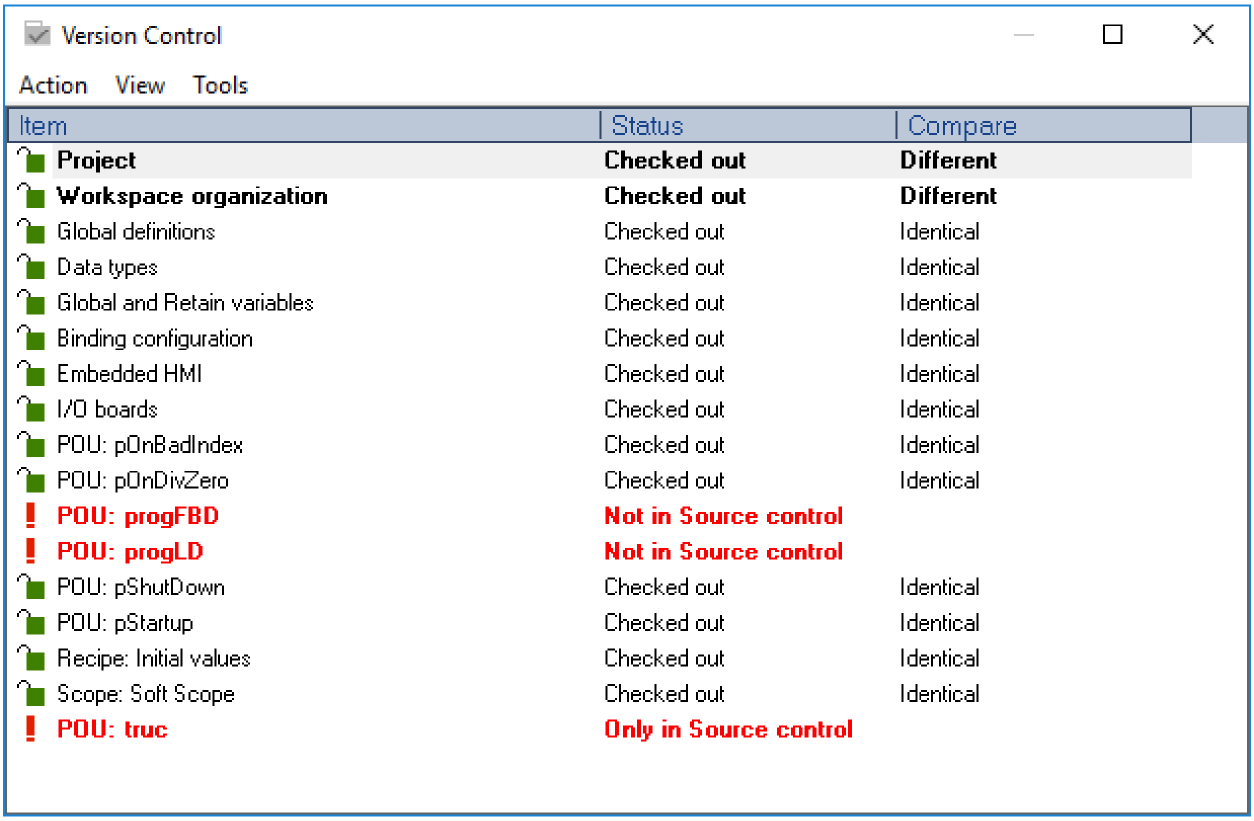


Warning: XML files in the \SC directory are updated only when the Version Control box is open. Some changes made in the local project while this box is closed may not be reflected in these files.

Also, changes of these files using the GIT tool will not be automatically injected in the local project. Always use the Version Control box for any "commit" or "get version" activity.

4. The Version Control box

The box shows the list of the components of the project, with their status in the Version Control tool:



Item	Status	Compare
Project	**Checked out**	**Different**
Workspace organization	**Checked out**	**Different**
Global definitions	Checked out	Identical
Data types	Checked out	Identical
Global and Retain variables	Checked out	Identical
Binding configuration	Checked out	Identical
Embedded HMI	Checked out	Identical
I/O boards	Checked out	Identical
POU: pOnBadIndex	Checked out	Identical
POU: pOnDivZero	Checked out	Identical
! **POU: progFBD**	**Not in Source control**	
! **POU: progLD**	**Not in Source control**	
POU: pShutDown	Checked out	Identical
POU: pStartup	Checked out	Identical
Recipe: Initial values	Checked out	Identical
Scope: Soft Scope	Checked out	Identical
! **POU: truc**	**Only in Source control**	

Below are the various possible components:

	notes
Project	The project skeleton and various options
Workspace organization	The organization of elements with the workspace tree
Global definitions	
Data types	All user defined data types (structures, enumerated data types and bit fields)
Global and Retain variables	
Binding configuration	
Embedded HMI	
I/O boards	"%..." I/O boards
POU …	A POU including its local variables and definitions
Fieldbus …	A fieldbus configuration (e.g. CAN)
Spy list …	A spy list
Recipe …	A recipe
Graphic …	A graphic document
Scope …	A soft scope document
String table …	A string table
Signal …	A "signals" document
Test sequence …	A test sequence
SCL …	A SCL file

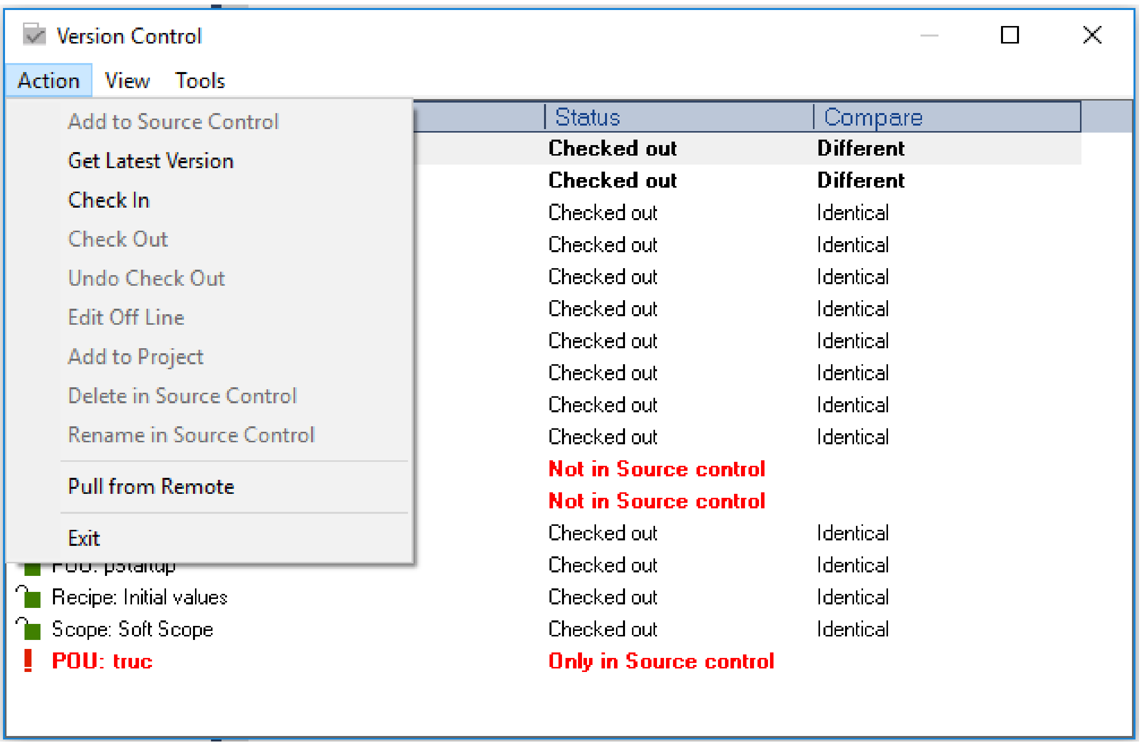
Below are the possible states for components:

	notes
Checked Out	The item is modifiable in the local project
Not in Source control	The item is present in the local project but is not archived in the repository.
Only in Source control	The item is archived in the repository but does not exist in the local project

Below are the possible compare states:

	notes
Identical	The item is identical to the one in the local repository
Different	The item is different from the one in the local repository

Use the commands of the "Action" menu to perform synchronization actions in between the repository and the local project:



	Status	Compare
	Checked out	**Different**
	Checked out	**Different**
	Checked out	Identical
	Checked out	Identical
	Checked out	Identical
	Checked out	Identical
	Checked out	Identical
	Checked out	Identical
	Checked out	Identical
	Checked out	Identical
	Not in Source control	
	Not in Source control	
	Checked out	Identical
POU: pstartup	Checked out	Identical
Recipe: Initial values	Checked out	Identical
Scope: Soft Scope	Checked out	Identical
! **POU: truc**	**Only in Source control**	

Actions:

	notes
Add to Source Control	If the item is present in the local project but not yet archived in the local repository, this command will add the item to the local repository. straton will automatically propose to push the changes to the remote repository at the end of this command.
Get latest version	Update the local project with the latest version of the item in the local repository.
Check in	Commit the changes made to the item to the local repository. straton will automatically propose to push the changes to the remote repository at the end of this command.
Add to project	Add to the project an item existing only in the local repository
Delete in Source Control	Removes an item from the local repository. straton will automatically propose to push the changes to the remote repository at the end of this command.
Rename in Source Control	Rename an item in the local repository. straton will automatically propose to push the changes to the remote repository at the end of this command.
Pull from Remote	Pull all items from the remote repository to the local repository. straton will ask for each file to overwrite it with the remote version if it is different.
Stage All	Stage all the added, modified, deleted files and commit. straton will automatically propose to push the changes to the remote repository at the end of this command.

NOTES:

- All the XML files in the SC directory are always in Read/Write mode.
- For commands “Add to SC”, “Check In”, “Delete in SC” and “Rename in SC”, you can select several items. This will make only one commit and only one push at the end.

5. Supported authentication methods

The following methods are supported by the GIT tool while connecting to the remote repository

- Login/Password
- Token
- SSH Key

NOTE: depending on the remote, some methods could be unavailable. For example, GitHub doesn't allow anymore connections with login/password method.

6. Connexion with GitHub

6.1. Login/Password

In UserName/Token, enter the username used to log in your Github account
In Password, enter the password
Remote URL should start with https://github.com/xxxxxxxx

NOTE: this method is deprecated by Github and may not work anymore

6.2. SSH key

First, create a SSH Key and add it to your Github account.
Follow the instructions here: *https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh/checking-for-existing-ssh-keys*

And: *https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent*

In UserName/Token, enter 'git' (without quote) as the username
No need to enter a password
Remote URL should start with ssh://github.com/xxxxxxxx

6.3. Personal access token

First, Create your token in your Github account.
Follow the instructions here: *https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-personal-access-token*

IMPORTANT NOTE: when creating your token, copy/paste the token somewhere because the generated token won't be displayed anymore after on github

In UserName/Token, enter the token
No need to enter a password
Remote URL should start with https://github.com/xxxxxxxx

7. Common use cases

7.1. Set your project under Source Control

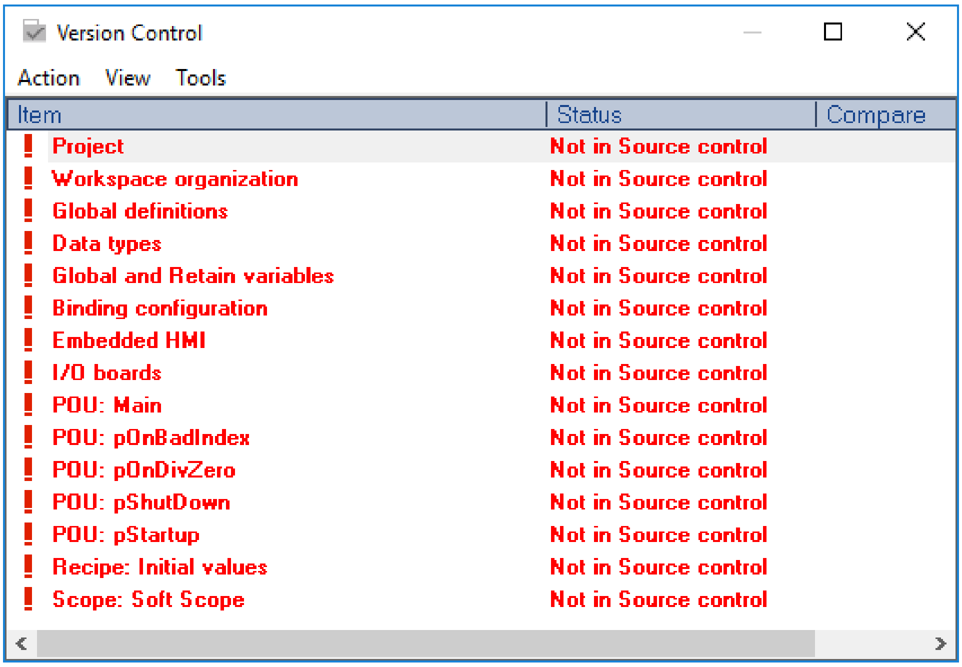
You must follow the procedure below:

1. Open your project in straton

2. From the Project Settings box, activate and configure Version Control.

   Accept to create the local branch if needed.
   Accept to add remote if needed

The list will show all items as "Not in Source Control" because all the XML files in \SC are not added to local repository.



Item	Status	Compare
! Project	Not in Source control	
! Workspace organization	Not in Source control	
! Global definitions	Not in Source control	
! Data types	Not in Source control	
! Global and Retain variables	Not in Source control	
! Binding configuration	Not in Source control	
! Embedded HMI	Not in Source control	
! I/O boards	Not in Source control	
! POU: Main	Not in Source control	
! POU: pOnBadIndex	Not in Source control	
! POU: pOnDivZero	Not in Source control	
! POU: pShutDown	Not in Source control	
! POU: pStartup	Not in Source control	
! Recipe: Initial values	Not in Source control	
! Scope: Soft Scope	Not in Source control	

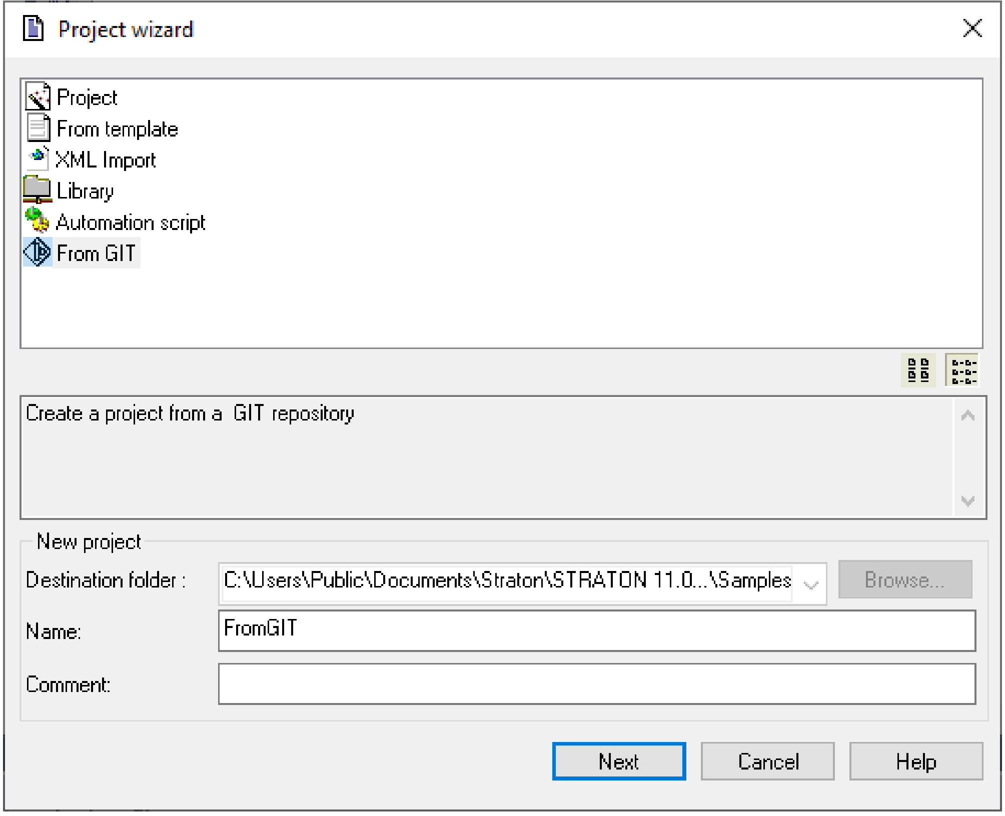
3. Select all the items with status “Not in Source Control” and select the action “Add to Source Control” to add them to the local repository.

If you have specified a remote ShortName and URL, straton will propose you to push to the remote at the end as well.

7.2. Create a local project based on a project archived in Source Control (Clone)

You must follow the procedure below:

1. Create a new project and select “From GIT”



Create a project from a GIT repository

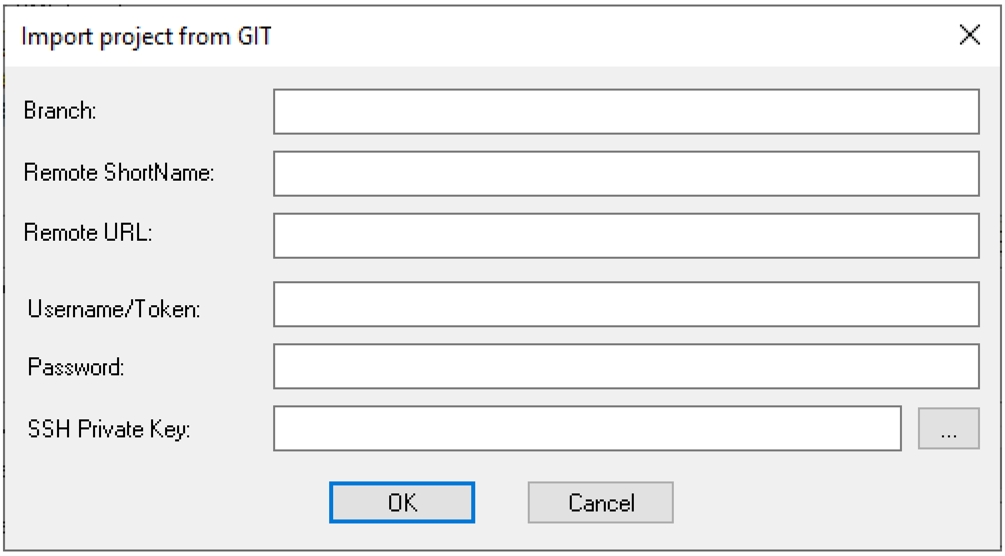
New project

Destination folder : C:\Users\Public\Documents\Straton\STRATON 11.0...\Samples

Name: FromGIT

Comment:

2. In the next dialog, enter all the information needed for GIT



Import project from GIT

Branch:

Remote ShortName:

Remote URL:

Username/Token:

Password:

SSH Private Key:

On OK, the wizard will create the project and local repository and will clone from the remote.