

## Pre-commit hook Example

PHE - Introduction . . . . .	1
PHE - Requirements . . . . .	2
PHE - Workspace Preparation . . . . .	3
PHE - Execution . . . . .	3
PHE - Configuration . . . . .	4

See also main SDK page.

## PHE - Introduction

### PHE - About Pre-commit hook

The pre-commit hook is invoked before a Subversion transaction is committed. Subversion runs this hook by invoking a program (script, executable, binary, etc.) named 'pre-commit', with the following ordered arguments:

1. REPOS-PATH (the path to this repository)
2. TXN-NAME (the name of the transaction to be committed)

If the hook program exits with success, the txn is committed; but if it exits with failure (non-zero), the txn is aborted, no commit takes place, and STDERR is returned to the client.

For more information see: hook scripts.

### PHE - About this example application

This hook runs java application, which will check, if the comment of commit message contains a link of work item, which have to be resolvable and not to be resolved. The idea is to ensure, that each revision will be linked to not resolved work item.

The return code of application equals 0 if the link to work item in commit message is valid. You should set the path to log file, where errors will be written. See ~Configuration and usage~ section to see the configuration file.

### PHE - Some examples of using

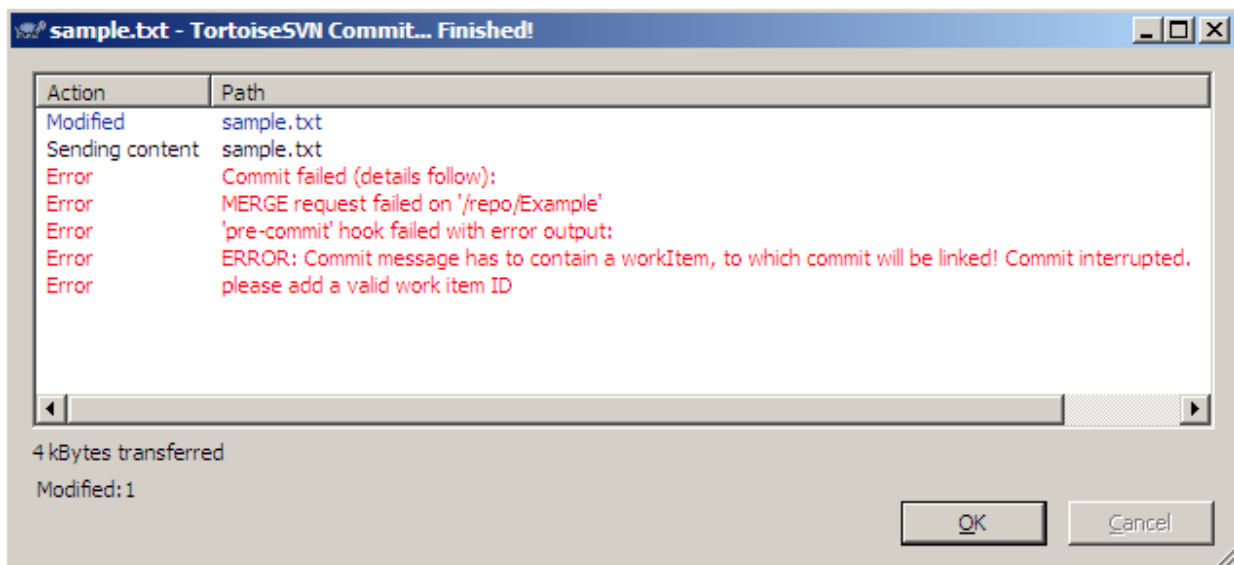


Figure PHE-1: The case, when you do not set any message

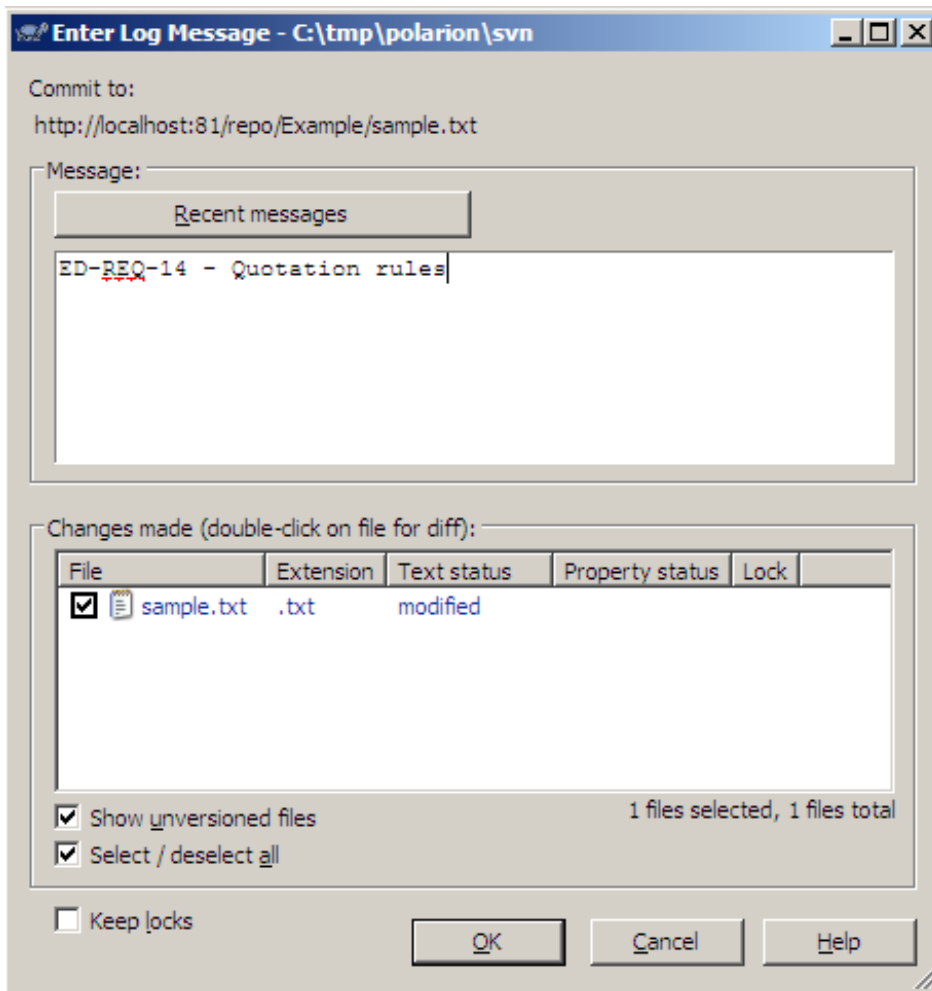


Figure PHE-2: The case, when you set right message and the work exist but is not resolved

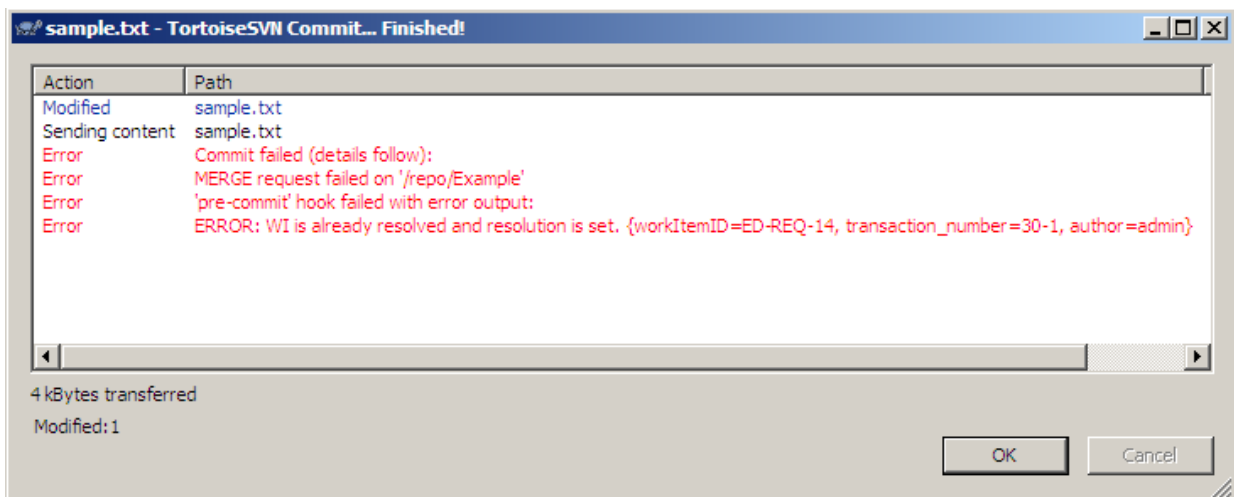


Figure PHE-3: The case, when you set right message, but work item is resolved (state is 'Closed')

## PHE - Requirements

### PHE - Development Environment

- [Eclipse IDE](#)
- [AdoptOpenJDK 11 \(LTS\)](#)

### PHE - Web service client

- You have to use Polarion web service client libraries stored in lib directory of Polarion SDK

(this is usually in MS Windows systems

C:\Polarion\polarion\SDK\lib\com.polarion.alm.ws.client\.)

## PHE - Workspace Preparation

- First, You have to import web service client project `com.polarion.alm.ws.client` from `SDK\lib\` directory. For this action, perform these steps:
  1. Start Eclipse, then select **File > Import...**
  2. In the dialog that appears, select **Existing Projects into Workspace** in **General** section. Submit it by Next button.
  3. By pressing **Browse..** button, select the directory `com.polarion.alm.ws.client`. Press OK.
  4. In subwindow Projects select `com.polarion.alm.ws.client` project and press finish.
- To start working with web services, you can import example project or you can create your own project.
  - To import example project, perform the same steps as the import of web service client project described above, except the name of project - the example project is located in `SDK\examples\com.polarion.example.commithook`.
  - To create Your own new Polarion web service project, perform these steps:
    1. Select **File > New > Java Project**.
    2. Type project name and click Next button.
    3. Select Projects card and click the button Add....
    4. Select `com.polarion.alm.ws.client` project and submit it.
    5. Select Finish.

## PHE - Execution

To execute the example, you have to export it as a jar file and create appreciate shell script. For this action, perform these steps:

1. Select **File > Export...**
2. Select JAR file in Java section, click Next...
3. Mark our project (this example = `com.polarion.example.commithook`) and set location of new JAR archive (together with name of new jar archive, e.g. `pre-commit.jar`).
4. Click Finish button

To run the jar, you can use windows script: (see at `pre-commit.bat` file in commit hook example folder)

So, what You have to do:

- you have to save this script to your svn hook directory (e.g. Polarion svn hook dir = `C:\Polarion\data\svn\repo\hooks\`).
- set **POL\_PROP** variable of properties, where are set the address of Polarion server, login, password, .. For detailed information see Configuration and usage section.
- set **PRE\_COMMIT\_JAR** variable to the same location, where you export the project (in

previous step).

- set **JAVA\_HOME** variable to the same JAVA environment, as you use to compile this project.
- set **POLARION\_SDK\_DIR** variable according your system and Polarion installation.

## **PHE - Configuration**

To start using example, you have to set information about Polarion Server in `settings.properties` file.

### **PHE - settings.properties**

The format is: `key=value`. Following keys have to be set:

- `polarion_server_adr`, e.g. 'http://localhost' - address of Polarion server
- `polarion_server_port`, e.g. '81' - port of Polarion server
- `user`, e.g. 'admin' - name of the user to log in to Polarion
- `passwd`, e.g. 'admin' - password for 'user'
- `project_id`, e.g. 'requ' (Requirements project) - ID of project of WorkItem typed in commit message
- `svnlook_dir`, e.g. 'C:\\Polarion\\bundled\\svn\\bin\\' - path to dir, where svnlook is placed
- `svnlook_cmd`, e.g. 'svnlook.exe' - svnlook program file name
- `apache_log_folder`, e.g. 'C:\\Polarion\\data\\logs\\apache\\' - the path of dir, where log of errors should be placed
- `apache_log_file_name`, e.g. 'commit\_audit.log' - the name of error log file (errors produced during validating commit message)

Info: This example was developed for MS Windows system, for different OSs change path and path separators in source code as well.