

SubversionTutorialPart3

From Erh-itproject

This tutorial is designed to show a basic work flow using Subversion and to demonstrate the use of some of the svn commands. Please keep the SVN Book handy for reference, especially Chapter 2, for more details and descriptions of how to interpret the output of the commands.

Part 3

Part 3 is a short example of how to assemble a tar file from a working copy and how to "tag" a specific revision in the repository as specific software version.

This tutorial presents the commands you should run but in general does not show the results (to encourage you to actually perform the commands). A sample log of actually performing this tutorial is at [SubversionTutorialLogPart3](#). Commands you should enter are prefaced with ==> and are in bold.

We will start out where Part 2 left off working as the origin developer.

```
=====  
==> cd $HOME/svnwork/test/myApp  
==> svn update  
=====
```

Since this is a very simple application, it could be distributed by users just obtaining the software for the repository directly via "svn export". However, it is most common to make a tar file for a specific software release and post to a web or ftp site. We will make a simple shell script that creates the tar file with only the files intended to be distributed. The --exclude option for tar will be used to also filter out the .svn directories. Create a new file called make_tarfile.sh with the following content (if you can't cut and paste, enter the lines starting with rm -rf distrib):

```
=====  
#!/bin/sh  
# Script to assemble a tar file for distributing myApp  
.  
usage() {  
echo "  
Usage: ./make_tarfile.sh <version string>  
Must be run from inside the myApp working copy  
"  
}  
.  
# Check if the application file exists  
if [ ! -f myscript.sh ]  
then  
echo "Error. Must run in the myApp directory"  
usage  
exit 1  
fi  
if [ $# -ne 1 ]  
then  
usage  
exit 1  
fi  
rm -rf distrib  
mkdir -p distrib/myApp  
cp -pr myscript.sh doc config distrib/myApp  
cd distrib  
tar zcvf myApp_$1.tgz --exclude .svn myApp  
echo "Done making: `pwd`/myApp_$1.tgz"  
=====
```

After editing, test running the script to check for errors and commit.

```
==> chmod +x make_tarfile.sh
==> ./make_tarfile.sh 1.0
==> svn add make_tarfile.sh
==> svn commit -m "Added script to make a release tar file."
```

We now have a tar file for version 1.0 of the application. However, you normally don't want to save version named tar files in the repository. Instead, we will use the tags directory in the repository to essentially create a bookmark that represents a specific software version. The bookmark will point to a specific repository internal revision number (this is all behind the scenes; users will just see the application's version numbers). To do this, svn copy will be used to copy directly within the repository. The end result will be specific versions listed as subdirectories in tags/myApp:

```
tags/myApp/1.0
tags/myApp/1.1
```

Anyone can then get the software at a specific version by checking out or exporting from tags instead of trunk. For example:

```
svn checkout file://$HOME/svnrepos/test/tags/myApp/1.0 myApp
```

To use svn copy directly in the repository, we have to make sure the target path tags/myApp exists, and create it if it doesn't.

```
==> svn list file://$HOME/svnrepo/test -R
```

Use svn mkdir to create myApp in tags and then svn copy to make the 1.0 release:

```
==> svn mkdir file://$HOME/svnrepo/test/tags/myApp -m "Made myApp dir in tags"
==> svn copy file://$HOME/svnrepo/test/trunk/myApp file://$HOME/svnrepo/test/tags/myApp/1.0 -m "Tag"
```

And now to verify the copy:

```
==> svn list file://$HOME/svnrepo/test/tags -R
```

```
myApp/
myApp/1.0/
myApp/1.0/config/
myApp/1.0/config/myApp.config
myApp/1.0/doc/
myApp/1.0/doc/README.txt
myApp/1.0/make_tarfile.sh
myApp/1.0/myscript.sh
```

A couple of final notes. Note when the svn mkdir and copy commands were performed on the repository, there was a committed message. Editing commands performed directly on the repository imply immediate commits.

```
svn mkdir file://$HOME/svnrepo/test/tags/myApp -m "Made myApp dir in tags"
Committed revision 12.
```

The purpose of the `make_tarfile` script can be seen a little better now. Since we have a script embedded in the repository, we can recreate a tar for any tagged version by checking the specific version from the repository and running the script:

```
==> cd $HOME/svnwork
==> svn checkout file://$HOME/svnrepo/test/tags/myApp/1.0 myApp_1.0
==> cd myApp_1.0
==> ./make_tarfile.sh 1.0
```

This completes the tutorial. You may clean up the files from the tutorial.

Retrieved from "<https://collaborate.werh.noaa.gov/wiki/index.php/SubversionTutorialPart3>"

Category: Tips and Tricks

- This page was last modified on 9 January 2009, at 13:19.