Setting up automatic version control for PeopleCode

Summary: this is how to set up a version control system (either Subversion (SVN) or Git) and have your changes in PeopleCode automatically committed to it via a nightly script. The use of multiple environments is discussed.

Using Subversion

Installing SVN

It is not strictly necessary to run an SVN server – you can also specify a file location in the SVN url in DecodePC.properties (e.g. file:///c:/svn/pshr/trunk/), but this may be slow and/or less secure. If you only use the file:/// protocol, you need not set the SVN passwords (below).

If you don't have it already, get Subversion off the Internet and install it. You can either install it in combination with Apache or with svnserve; if you choose the Apache install, substitute the 'svn://' protocol with 'http://' in the instructions below.

For Windows, you may want to pick the Sliksvn version (http://www.sliksvn.com/en/download ; use the custom install, and select svnserve); for Debian Linux it's just 'apt-get install subversion'.

You probably want Subversion to start automatically. For Linux, this will typically involve some init.d script (details depend on your flavor of Linux); for Windows, it's something like (in a CMD window)

```
sc create SvnServe binpath= "\"C:\Program Files\SlikSVN\bin\svnserve.exe\" --service -
r C:\Svn" start= auto
```

(note the spaces behind the = signs)

```
Net start SvnServe
```

Setting up a repository

Create a repository (called pshr here), and set up local password authentication for your users. The idea is that you set up an SVN user for each PeopleTools user.

```
svnadmin create c:\svn\pshr
```

Edit c:\svn\pshr\conf\svnserve.conf: uncomment lines
'password-db = passwd'
and
'authaccess= write'

Edit c:\svn\pshr\conf\passwd with SVN users you want to map PeopleSoft users to:

```
 [users]
someuser = secret
PPLSOFT = secret
harry = secret
sally = secret
```

Create a directory to with the folder structure you want to have in your repository, and import it

```
cd c:\temp

mkdir svn_template

mkdir svn_template\trunk

mkdir  svn_template\branches

svn import svn_template file:///c:/svn/pshr -m "From template"
```

(this step may not be necessary, since DecodePCODE will attempt to create the paths in SVN, but some SVN servers only accept paths starting with /trunk, /tags or /branches).

View Results

Finally, you'll need a Subversion client to view/retrieve the PeopleCode and SQL text that has been submitted.

Since you won't do doing any direct commits to the repository, the WebSVN viewer is a good choice, as it requires no client installation. However, it sometimes seems to mangle some characters in the PeopleCode – the ubiquitous ampersands in the code appear to mess up the HTML text.

There are also excellent Subversion clients out there that can be installed: TortoiseSVN, or Eclipse with the Subversive or Subclipse plug-in.

Using Git

The other hugely popular open-source versioning system, of course, is Git.

You do not have to install Git or create a repository prior to running DecodePCODE_Git.bat; the repository will be created if necessary.

The required parameters in DecodePCODE.properties are as follows:

```
# parameters for Git processing (DecodePCODE_Git.bat or arg[0]=
ProcessToGit)

# Location of Git work directory (with .git subfolder); repository

will be created if necessary

gitdir=c:\\temp\\git\\myrepo

gitbase=HRDEV

# next parameter will define Git author if PS user not found

gituser=whatever/Some User/unknown@unknown.com

# any number of other PS users, with name/email to use for Git submits

gituser1=Erik/Erik H/erik_h@ourcompany.com

gituser2=JOHN/John Doe/john_doe@ourcompany.com
```

Use DecodePCODE_Git.bat in a way similar to described above for the equivalent SVN script.

There are a number of viewers available for Git, such as GitList.

Configure DecodePCODE to submit changes to SVN

Edit DecodePC.properties. In this file, you'll want to specify the URL and path to use, as well as a mapping from PeopleTools users to SVN users.

```
svnurl=svn://localhost/pshr
svnbase=/trunk/PeopleCode/
svnuser=whatever/someuser/secret
svnuser2=PPLSOFT/PPLSOFT/secret
svnuser3=harry/harry/secret
```

Once you've installed the jar files (JDBC driver and svnkit.jar; see README.txt), you should be able to run DecodePCODE_SVN.bat (or DecodePCODE_SVN.sh).

The first time you run it, you may want to use this:

```
DecodePCODE_SVN.bat custom
```

, which should submit all PeopleCode segments (and SQL definitions) with LASTCHANGEOPRID <> PPLSOFT.

Then, run

```
DecodePCODE_SVN.bat since-days 1
```

which is mostly useful to create a file 'last-time.txt' with a time stamp in the right format.

Finally, you can schedule a nightly task (create a separate .bat file, and use Task Manager) with

```
DecodePCODE_SVN.bat since-last-time
```

There are scripts 'nightly.bat' and 'nightly.sh' available for this.

Working with multiple PeopleSoft environments

Note that you can also have PeopleCode/SQL in other PeopleSoft environments committed to SVN.

For instance, to have to HRACC environment processed, add something like this to DecodePC.properties:

```
processHRACC=ProcessToSVN
userHRACC=sacrm
passwordHRACC=sacrm
urlHRACC=jdbc:sqlserver://130.175.204.77;DatabaseName=HRACC
dbownerHRACC=dbo
```

One way to use this is to have the code in your UAT or PROD environments committed to separate SVN / Git folders, using one of the following properties:

```
svnbaseHRACC=/branches/HRACC/PeopleCode
```

or

```
gitbaseHRACC=/branches/HRACC/PeopleCode
```

Since DecodePcode 0.60, you can also work with a single folder tree, using the 'ancestor' property. In this case, only one environment (typically DEV) needs a svnbase / gitbase property. The other environments (typically PROD and DEMO) are referred through the 'ancestor' property, e.g.

```
ancestor=PROD

ancestorPROD=DEMO
```

this will cause the DEMO and PROD versions of an PeopleCode program to be committed to the repository, prior to the DEV (base) version. These commits will only take place when the program added to the repository.

For example, if the base environment has ancestor 'HCM', and a delivered PeopleCode program is modified twice, with 'since-last-time' runs after each modification, the history will show (here in Eclipse with the Subversive plug-in):