

SubversionRepository

Overview

Apache JDO uses [Subversion](#) to manage its source code. Instructions on Subversion use can be found here [here](#). To receive notice of commits to the repository subscribe to jdo-commits@db.apache.org by sending email to jdo-commits-subscribe@db.apache.org.

Current version

The Apache subversion repository contains the latest code.

Previous versions

Download the 2005-03-08 version [apache-jdo-2005-03-08.zip](#). It implements the remaining JDO TCK 1 query test cases. The subproject structure has been changed by adding directories java, dtd, jdo, etc. under src and test. The tck11 subproject uses some basic test configuration as described in [TestRunner](#).

Download the 2005-02-17 version [apache-jdo-2005-02-17.zip](#). It fixes a problem with running the ri11/tck11 subprojects under windows. Furthermore it updates some TCK query tests.

Download the 2005-02-10 version [apache-jdo-2005-02-10.zip](#). It includes logging changes and support for running the jars with J2SE 1.3.

Download the 2005-01-21 version [apache-jdo-2005-01-21.zip](#). It includes a working copy of the tck11 sub-project.

Download the 2005-01-14 version [apache-jdo-2005-01-14.zip](#).

Download the 2004-12-13 version [apache-jdo-2004-12-13.zip](#).

Web Access

<http://svn.apache.org/viewcvs/db/jdo>

Anonymous access

Apache JDO source can be checked out anonymously with this command:

```
% svn checkout http://svn.apache.org/repos/asf/db/jdo
```

Once you have Apache JDO checked out you can update the source by executing the following command from within the jdo.

```
% svn update
```

Access from behind a firewall

For those users who are stuck behind a corporate firewall which is blocking http access to the Subversion repository, you can try to access it via HTTPS:

```
% svn checkout https://svn.apache.org/repos/asf/db/jdo
```

Access through a proxy

The Subversion client can go through a proxy, if you configure it to do so. First, edit your "servers" configuration file to indicate which proxy to use. The files location depends on your operating system. On Linux or Unix it is located in the directory "~/.subversion". On Windows it is in "%APPDATA%\Subversion". (Try "echo %APPDATA%", note this is a hidden directory.)

There are comments in the file explaining what to do. If you don't have that file, get the latest Subversion client and run any command; this will cause the configuration directory and template files to be created.

Example : Edit the 'servers' file and add something like :

```
[global]
http-proxy-host = your.proxy.name
http-proxy-port = 3128
```

Please use the regular http proxy settings in case you want to access the the repository from the Sun network (SWAN).

Submitting a Patch

If you make changes to Apache JDO, and would like to contribute the to the project, you should create a patch and send it to the jdo-dev alias jdo-dev@db.apache.org. To create a patch, simply execute the following command:

```
% svn diff > your-changes.patch
```

Developer Access

Everyone can access the Apache JDO Subversion repository via HTTPS, but Apache JDO Committers must checkout the Subversion repository via HTTPS.

```
% svn checkout https://svn.apache.org/repos/asf/db/jdo
```

To commit changes to the repository, you must set your password on the Apache Subversion server. To set your password, use ssh to connect to svn.apache.org, and enter the command **svnpasswd**. This will prompt you to enter a svn password of your choice (pick a safe password). Now, now you are ready to commit changes using your username/password. Execute the following command to commit your changes (svn will prompt you for your password)

```
$> svn commit --username your-username
Authentication realm: <https://svn.apache.org:443> ASF Committers
Password for 'your-username': your-password
```

You can also pass your password on the command line directly, but this is a security problem on multiuser unix computers (the command line arguments are available via the ps command). Here is the command if you are Windows or a single user unix computer:

```
$> svn commit --username your-username --password your-password
```

Remember to replace 'your-username' and 'your-password' with your actual username and password on svn.apache.org.

NetBeans CVS Repository Access

The btree subproject checks out the NetBeans mdr btree implementation. This requires cvs being installed on your system. The official NetBeans cvs host might not work if you are behind a firewall that blocks the cvs port. Please consult the [NetBeans sources](#) page for more info.

If you do not have a cvs client installed on your system you can download the NetBeans mdr btree sources [netbeans-mdr-btree.zip](#). Unzip the file in the btree directory and comment out (or remove) the definition of the preGoal java:prepare-filesystem.

Using subversion on Windows with cygwin

If you use subversion on Windows under cygwin, you may find that the subversion client automatically assigns the executable property to non-executable files. In that case, you would see this at the bottom of an *svn diff* of the file:

```
Property changes on: test/sql/derby/datastoreidentity/schemal.sql
_____
Name: svn:executable
+ *
```

This section explains the source of the problem and suggests some actions to avoid it.

Background

Subversion carries executable information in the built-in property called `svn:executable`. This property, unlike others, may be present or absent, but it has no value. You can add it or delete it, but but you cannot change its value.

In theory, subversion ignores Windows file permissions and by default does not set `svn:executable`. However, cygwin `svn` acts like Unix `svn` and determines the `svn:executable` property based on file permissions.

If you create a file from the cygwin command line, by default it is executable only if the filename ends with `.bat`, `.com` or `.exe`, or if its content starts with `#!`. [This is what the doc says, but you may see `-x` for all files.] If you create a file using a Windows tool, by default its Windows permissions are executable by all. Cygwin interprets the Unix-style permissions this way as well. If the file is executable by all, cygwin `svn` sets the `svn:executable` property on the file when you invoke `svn add`.

Removing existing executable properties from the repository

You can use `svn propdel` to remove the `svn:executable` property from your working copy.

```
svn propdel -R svn:executable .
```

will recursively remove the `svn:executable` property from all of the files below the current directory. You can use this and commit the files to clean the repository if necessary.

Preventing subversion from adding unwanted executable properties

Windows/cygwin users who don't want to have to think about using `svn propdel` or `chmod` on each added file can use a non-cygwin version of `svn`. The Subversion 1.2.3 Win32 binaries, downloadable from the link at the bottom of http://subversion.tigris.org/project_packages.html, appear to work well. After installation add the `svn.exe` location to your Windows `PATH` variable. If you are switching from cygwin `svn` to Win32 `svn`

1. Remove the subversion component from your cygwin installation because when `svn` is invoked from a cygwin window, the cygwin version is found even if your `cygwin/bin` directory is later on the path. (In the Select Packages window of the setup wizard, navigate to the subversion package in the Devel. category. Click on the status icon until Uninstall is displayed. Click next and continue through the wizard until installation is complete.)
2. Copy the `servers` file and the `auth` folder from the `sygwin ~/.subversion` directory to `C:\Documents and Settings\<user>\Application Data\Subversion used by Win32 subversion`.

Note that windows `svn` uses backslash as the path separator when displaying file names. You cannot just copy and paste this file name to another `svn` command when running from within a cygwin shell. You need to enclose the file name into double quotes.

Alternatively, Windows users can set file permissions in Windows Explorer. (Right-click on the top-level folder & select Properties. Select the Security tab. Click Advanced. Remove all instances of Read & Execute from the Permission Entries. Click "Reset permissions on all child objects and enable propagations of inheritable permissions". Click Apply. OK. OK.) You will have to do this again when you do a clean checkout to a new directory.