Inhaltsverzeichnis

SVN- Revision in Java Programme einblenden

Netbeans

Dies ist eine direkte Kopie von http://dragly.org/2009/10/11/revision-and-build-numbers-for-your-netbeans-apps/und hier nur noch mal abgelegt, damit diese Info bleibt, wenn die Originalseite mal verschwindet..

Revision and build numbers for your NetBeans apps

Posted on October 11, 2009 by Svenn-Arne Dragly

After going through a couple of guides on the net, I figured out how to add build numbers and revision numbers to my NetBeans projects. The build number will be incremented for each build, and the revision number is gathered from your subversion repository. If you don't use Subversion just skip that part and add the revision number manually.

In this tutorial I assume you are creating an application based on the NetBeans Desktop Application wizard. Otherwise, you may just follow along and use whatever fits into your project. All you need is a java project being built by Ant (as all NetBeans projects are). Adding a build number

First of all, figure out where the properties file for your application is located. This should be in [package name].resources (or if you are browsing the files, you will find it under /src/[package name]/resources) and is named the same as your application with the suffix ".properties".

This could for instance be `myapp.resources/myapp.properties` or `/src/myapp/resources/myapp.properties`.

Open it up and find the version property. It should look something like this:

```
Application.version = 1.0
```

Change this to the following

```
Application.version = 1.0.0.${Application.buildnumber}
```

Now, we are ready to add a build number to your `build.xml` file. This is found in the `nbproject` folder if you browse for files (not in the Projects browser).

Add the following lines below the

```
<import file="nbproject/build-impl.xml"/>
```

by replacing [Package Name] and [App Name] with your own values.

```
<target name="-post-jar" description="Sets the buildversion for the current build">
```

```
    <propertyfile file="${src.dir}/[Package Name]/[App Name].properties">
        <entry key="Application.buildnumber" value="1" type="int"
operation="+"/>
    </propertyfile>
</target>
```

This should now add your build number to your version number. If you want to have the version in the main title bar of your application just go back into the properties file and change the Application.title property to something like this:

```
Application.title = My Application Name ${Application.version}
```

Adding the Subversion revision number

Go back to your build.xml file and add the following below everything else:

```
<target name="-post-init" description="Sets the buildversion for the current
build">
    <exec outputproperty="svna.version" executable="svnversion">
        <arg value="-c" />
        <redirector>
            <outputfilterchain>
                <tokenfilter>
                    <replaceregex pattern="^[0-9]*:?" replace="" flags="g"/>
                    <replaceregex pattern="M" replace="" flags="g"/>
                </tokenfilter>
            </outputfilterchain>
        </redirector>
    </exec>
    <propertyfile file="${src.dir}/[Package Name]/[App Name].properties">
        <entry key="Application.revision" value="${svna.version}" type="int"
operation="="/>
    </propertyfile>
    <echo>Revision found from SVN: ${svna.version}</echo>
</target>
```

(replace [Package Name] and [App Name] with your own values)

Then, add the revision number to your version number by editing changing the Application.version property to something like this:

```
Application.version = 1.0.${Application.revision}.${Application.buildnumber}
```

This should be all it takes.

Eclipse

Elipse nutzt kein Ant und damit auch keine Build.xml. Hier trickst man dann folgendermaßen:

unter `res/values` ergänzt man in der strings.xml einen neuen Eintrag, der die SVN- Revision beinhalten soll:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">OpenOnBoardDiagnostics</string>
    <string name="app_name_short">OOBD</string>
    <string name="app_svnversion">foo</string>
</resources>
```

Dann erzeugt man sich ein eigenes Ant - Buildscript, was den obigen Eintrag (trickreich) aktualisiert:

build_svn_revison.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="add SVN" default="foo-update-svnversion" basedir=".">
    <description>adds SVN Revision to Android project</description>
<target name="foo-update-svnversion">
   <exec outputproperty="svna.version" executable="svnversion">
        <arg value="-c" />
        <redirector>
            <outputfilterchain>
                <tokenfilter>
                    <replaceregex pattern="^[0-9]*:?" replace=""
flags="g"/>
                    <replaceregex pattern="M" replace="" flags="g"/>
                </tokenfilter>
            </outputfilterchain>
        </redirector>
    </exec>
         <property name="match.start" value="&lt;string
name=&quot;app_svnversion&quot;&gt;"/>

        <property name="match.end" value="&lt;/string&gt;"/>

   <replaceregexp file="res/values/strings.xml"
                  match="${match.start}.*${match.end}"
                  replace="${match.start}${svna.version}${match.end}">
    </replaceregexp>
    <echo>Revision found from SVN: ${svna.version}</echo>
</target>
</project>
```

Wenn man dann den Build laufen läßt, aktualisiert er die aktuelle Buildnummer in strings.xml

Letztlich auswerten kann man das z.B. so:

```
    TextView versionView=(TextView) findViewById(R.id.versionView);
```

```
    versionView.setText("Build number: "
+getResources().getString(R.string.app_svnversion));
```