DVN 3.0 Upgrade Procedure

The instructions provided are for upgrading DVN v2.2.5. For an older version, it is the responsibility of the user to first upgrade to v2.2.5, then proceed with the v3.0 upgrade.

0. Introduction.

An important part of the 3.0 release is a transition to Glassfish 3.1. So even though you are upgrading an existing DVN installation, a new Glassfish environment needs to be built from scratch. Because of this, we've determined that the easiest way to proceed is to first create a brand new, unpopulated DVN 3.0 using the provided installer; then upgrade, and point the new DVN to, the preserved v2.2.5 database; and then restore any remaining custom configuration your old installation may have. These steps are described in detail below.

1. Preserve the old database and build a brand new DVN 3.0.

Please install Glassfish v3 and create a new DVN 3.0 environment using the provided installer and following the instructions at *(installation manual)*.

You don't really need to do anything special to preserve your current database: you can simply leave it as is, and create a new, empty database with a different name during the 3.0 install. (Note that the DVN installer would not allow you to specify an already existing database). However, it may be prudent to make a full backup of the database right before you proceed.

We also recommend that you preserve the old version of Glassfish. For example, if you have the version 2.1 installed in `/usr/local/glassfish`, you can install the new version in `/usr/local/glassfish3`. This way the old versions, of both Glassfish and DVN, are preserved for reference.

Once you have created a new, empty DVN 3.0...

2. Upgrade your old (preserved) database

Before you attempt to do this, please verify that the step. 1, above, really succeeded! I.e., check that the application is running, that you see the DVN front page at `http://<YOUR HOST>/dvn`. And maybe, if you are particularly thorough, create a test dataverse and a study with a file or two. It is important to verify that the new DVN 3.0 framework has been successfully created and

all the basic functionality is working. This will allow us to separate any issues that may have happened as the new environment was being built from those with restoring your existing data! But that is not to say that we expect you to run into either. In fact there is a reasonably high chances of the whole process being smooth and painless.

(**Note that the order of the steps below is important!** For example, the SQL update script must be run after the redeployment of the application)

a. Stop glassfish.

b. Edit your domain.xml and change the database settings (under `<jdbc-connection-pool … name="dvnDbPool">`) to point to your old, preserved database. (consult your old domain.xml for details).

c. Delete the EJB timer lock file:

```
rm <DOMAIN DIR>/generated/ejb-timer-service-app
```

d. Start glassfish

e. Re-deploy the application one more time:
(assuming you are still in the `dvninstall` directory)

```
  cd appdeploy
  /usr/local/glassfish3/bin/asadmin deploy --force=true --name=DVN-web dist/
DVN-web.war
```

f. Run the build update script (found in the installer directory), on the server where PostgresQL is running, **as user postgres**:

```
psql -d <DB NAME> -f buildupdate_v2_2_5_v3_0.sql
```

g. Resrtart glassfish one more time, for good measure.

3. Restore your old custom configuration

This part may require some manual work. There's really no way to reliably import all your old custom configuration into the new installation, so you'll have to do it by hand.

Luckily, you have your old domain.xml file preserved, as well as the entire old domain directory, so it shouldn't be difficult to locate and reproduce all the necessary custom additions you might have there.

a. Static files.

If the steps 1. and 2. above have worked correctly, you should now be seeing a new DVN with

all your previously existing dataverses and studies.

Still, some things, particularly some customizations may not look quite right. If you had any images stored locally, you will need to move this directory into the new web root, for example:

```
cp -R <OLD DOMAIN>/docroot/images <NEW DOMAIN>/docroot
```

(Copy everything else from docroot that your installation may need)

Note: DVN 3.0 installer supplies the file `robots.txt` in the `docroot` directory, configured to keep all the automated crawlers (googlebot, etc.) away. If you do want to enable some crawling, do NOT simply copy your old `robots.txt` into the new `docroot`. Many URL patterns have changed in the DVN 3.0. So if you reuse your old configuration, spider bots will likely be encouraged to enter parts of the site that you never want them to crawl. You certainly don't want search engines to crawl your entire site, in particular, the file download URLs and the UI service resources. That will generate a substantial system load, with a potential to cause serious instability. (And it is unlikely that you have any reason to have anything other than your study metadata, if anything, indexed by public search engines). For a sample configuration and a list of the current URL patterns in the DVN application, see the commented-out portion of the robots.txt supplied by the new Installer.

b. JVM options

- The location of the filesystem directory where your studies are stored. This is controlled by the `vdc.study.file.dir` JVM option in your domain.xml. The installer set it to files/studies subdirectory in your domain config directory. Make sure it is set to what you had in your old `domain.xml`.

- Lucene index directory. It's not strictly necessary to point Lucene to the same directory that you were using before. (In fact, if you choose to keep using the same directory, **delete all the old files** that are currently there before restarting the application) But if you had a reason to use a custom location before (larger filesystem, had to be shared with another server, etc.), it may still be valid. If so, restore the `dvn.index.location` JVM option from the old `domain.xml`. Note the "Rebuild Lucene index" section below.

(Please remember that all the modifications made to `domain.xml` require a Glassfish restart!)

- Handles registration setup.
If you maintain your own Handles registration service, you will need to restore your registration credentials configuration. Restore the 3 or 4 `dvn.handle.*` JVM options.

- If using Google Analytics, restore the `dvn.googleanalytics.key` option.

c. Other Glassfish configuraiton

Any custom configuration you may have created to control system performance, memory allocation, garbage collection, etc. Be extra careful, since some of this functionality has changed in Glassfish 3, so some settings you've used in the past may be obsolete now.

The most important settings are:
-Xmx (heap size allocation) <...>
PermSize <...>

d. Security (SSL) certificates

If your setup needed any extra SSL certificates, they will need to be imported again. The certificate store files are kept in the Glassfish domain configuration directory. And since you've switched to a brand-new Glassfish installation, you now have only the certificates that come with it pre-installed.

- If you were already providing https service (until DVN v3.0, https was optional), you will want to keep using the same certificate. This lives in `<DOMAIN>/config/keystore.jks`. The easiest thing to do is to simply copy the keystore.jks from your old domain directory (do back up the existing one, just in case).

- If you were using any extra authority certificates, they will need to be reimported one by one. These live in `<DOMAIN>/config/cacerts.jks`.

For example, if your DVN is serving data from the ICPSR, you need the authority certificate from incommon.org. (This is the signing authority of the certificate used by https on icpsr.umich.edu, and their certificate does not come pre-installed with Glassfish. So it needs to be re-imported)

You can export it from your old cacerts.jks, or download it from incommon.org directly; for example:

```
wget -O /tmp/incommon.crt "http://cert.incommon.org/InCommonServerCA.crt"

cd <GLASSFISH DOMAIN>/config

<JAVA HOME>/bin/keytool -importcert -keystore cacerts.jks -alias
incommonserverca -file /tmp/incommon.crt
```

then restart glassfish.

e. OS-level configuration

You will probably need to modify your system startup script for Glassfish (`/etc/init.d/glassfish` on RedHat), to reflect the new Glassfish directory.

Note that the extra configuration you (should) have in that startup script, such as setting ulimit and “memory overcommit” is still needed.

The same will apply if you were doing backups of the Glassfish directory, and/or to any other system scripts that need to know its location.

f. Rebuild Lucene index

Using the Network Admin Utilities, re-index all studies in your DVN. This will restore the search functionality.