**Purpose:**

To update GeoNetwork metadata application and the GeoServer data application on a suitable Linux server using the JETTY9 or later webserver, with:

* existing jetty app running as a service, with separate Jetty HOME and BASE directories,
* jetty startup files in /etc/default/jetty, which point to GN and GS data directory locations
* directories external to geonetwork and geoserver so as to retain previous data, including a metadata\_subversion directory
* metadata database
* actual deployment of the geonetwork and geoserver applications

The approach is to:

* create a copy of the existing metadata database and data file structure
* modify the three configuration files for the database
* install the geonetwork/geoserver application

| **Step** | **Major Activity** | **References, Forms and Details** |
| --- | --- | --- |
| **1** | Notify accessible users of pending upgrade date and time, and consequent downtime | * typically 36 hours or more in advance |
| **2** | Back up the GeoNetwork database. If postgresql/PostGIS:  pg\_dump –U postgres  -f path/to/file/filename\_full.sql  <database\_name> | * Depending upon GeoNetwork version, there may be changes to the structure of the GeoNetwork database. Sometimes these changes are not successful, and the database may need to be migrated manually. * This will be a complete database dump in SQl (text) format, which is the safest and most portable. It is restored with psql * Another alternative is a compressed dump, appending parameter “-Fc” and extension “dmp” (not text). This is later restored using pg\_restore. Size is about ten percent of text version. |
| **3** | Create a geonetwork database copy:  createdb –U postgres -O <owner> <newdb>;  then  pg\_restore –U postgres –Fc –d <newdb> <backup path and file>  -OR-  psql –U postgres –d <newdb> -f <backup path and file>.sql | * create in case something goes wrong. * Use pg\_restore for compressed backup/restore, * Use psql for text backup |
| **4** | Confirm that the new database was successfully cloned:   * In psql and for the <newdb>,   SELECT \* FROM postgis\_full\_version();   * Exit psql ( \q ) | * Should see something like:   POSTGIS="2.1.3 r12547" GEOS="3.4.2-CAPI-1.8.2 r3924" PROJ="Rel. 4.8.0, 6 March 2012" GDAL="GDAL 1.10.0, released 2013/0 4/24"  LIBXML="2.7.8" LIBJSON="UNKNOWN"  TOPOLOGY RASTER |
| **5** | Ensure the GN and GS data directories are:   * external to the gn and gs webapp application subdirectory structure * gn and gs directories and files are backed up * modify the pointers in /etc/default/jetty to point to appropriate files (see step 8) | * gn directory contains the geonetwork data, including Lucerne index and metadata\_subversion directory * gs contains the geoserver data * nominal gn and gs structure * <external-to-gn-install-dir>/gn * < external-to-gn-install-dir>/gs |
| **6** | Remove metadata\_subversion directory from the existing geonetwork data directory, under <gn\_dir>/config/data/ | * Required to avoid implementation error due to version conflicts   metadata\_subversion directory will be automatically rebuilt |
| **7** | ONLY if upgrading from a 3.2.1 or lower version and you have external data files, you need to manually remove the old wro4j cache files before upgrading. In your GN data directory:   * search for the files   (\*wro4j-cache.h2.db\*, \*wro4j-cache.lock.db\*)   * remove them   sudo rm –r wro\* | * added XSRF security function * from [**Login error Invalid CSRF Token**](https://sourceforge.net/p/geonetwork/mailman/message/36073514/) * Otherwise will get the following exception:   \*Invalid CSRF Token '' was found on the request parameter '\_csrf' or header 'X-XSRF-TOKEN'.\*  \*This step should be done only once, before upgrading the version.\* |
| **8** | In the jetty start-up file, ensure keyword locations correctly set. Contents should be similar to:  JETTY\_HOME=/opt/jetty/jetty-distribution-9.4.5.v20170502  JETTY\_BASE=/opt/gn321b  TMPDIR=/opt/jetty/temp  JAVA\_OPTIONS="-Xms512m -Xmx1g -Xss2M"  JAVA\_OPTIONS="$JAVA\_OPTIONS \  -Djeeves.filecharsetdetectandconvert=enabled \  -Dmime-mappings=/opt/gn321b/webapps/geonetwork/WEB-INF/mime-types.properties"  JAVA\_OPTIONS="$JAVA\_OPTIONS  -DGEOSERVER\_DATA\_DIR=/data/v322/gs\_dir \  -Dgeonetwork.dir=/data/v322/gn\_dir \  -Dgeonetwork.lucerne.dir=/data/v322/gn\_dir/index" | * Normally nothing needs changing * If using jetty as a service, the normal start-up location is /etc/default and filename is “jetty” (no extension) * The java options are “chained” for ease in reading and debugging. Do not omit the trailing backslash line-extenders * If the data folders were moved from the default location, the startup file **must** point at the data directories. * Recent start-up files have a prepared keywords in which to change the assignments |
| **9** | Move to the JETTY\_BASE folder and prepare to deploy geonetwork   * Create a folder named geonetwork * Get a copy of the geonetwork.war version desired, and place it in the geonetwork folder * Unzip the war file * Delete the war file, retaining the unzipped contents | * wget <https://sourceforge.net/projects/geonetwork/files/GeoNetwork_opensource/v3.4.3/geonetwork.war> * It is also possible to directly place the war file in the webapps directory, and let jetty do the unpacking. In that case, need a geonetwork.xml file containing the file customizations. * One also wants to ensure the file customization has occurred, as in the following step |
| **10** | Check that the WEB-INF files have correct db name, UserID, PW and port   * jetty-env.xml * config-db/jdbc.properties * config-node/srv.xml | * ensure no backup files are created, especially in config-node   + a retained backup srv.xml file causes “duplicate node” error |
| **11** | Remove any GeoNetwork file/folder from the webapps deployment location   * this is being very safe | * In our case, the location of webapps is at /opt/gn321b/webapps * the webapps folder should not contain any reference to geonetwork |
| **12** | Reboot the server | * This will remove any references to duplicate nodes, which can cause errors * Rebooting can take several minutes |
| **13** | Clear the browser cache while waiting for the server reboot to complete   * generally found under “options/privacy & security” | * If upgrading from GN3.2.1 or lower, the browser may contain a version of GeoNetwork that can inhibit the proper display of the new.   + one indication could be a blank display |
| **14** | Move to the JETTY\_BASE folder and deploy the Geonetwork application   * Move the whole geonetwork folder to the webapps directory for automatic deployment | * JETTY\_BASE = /opt/gn321b in our example   + If errors, then need to start again at step 5. |
| **15** | After a brief delay (minutes), the deployment should complete   * deployment can be monitored at server\_name and port (e.g. http://localhost:8080 )   if errors, examine the JETTY\_BASE/logs and take appropriate action | * Initial monitoring should state “starting” and shortly thereafter “available” * Common errors are   + duplicate node due to not rebooting, or extra (backup) file in config‑node due to editing   + metadata subversion conflict with different database versions   + problems in database migration, necessitating manual migration   + security issues involving tokens |
| **16** | If necessary, restore stuff:   * under settings, Catalogue name, organization, email, logo   + save   + test email * under users,   + add “open data” group,   + change admin PW,   + add other users * under templates, add “all templates” | * providing email address is very important if one forgets or otherwise loses the admin PW * do not add the sample data under templates, unless it is a test installation and you need examples. Otherwise they can get mixed into production metadata. |