

# **Conversion to GForge AS**

# Overview

The conversion to GForge AS involves dumping out your GForge 4.5 database in a specific format, and transferring your mailing lists, cvs trees, and SVN trees to the new machine.

In GForge 4.x some files, like doc manager, forum, and tracker attachments, are stored base64 encoded in the database. Others, like the FRS files, are stored in the filesystem. The conversion script will move all of these files, along with the data from the database tables, into a /tmp/gf5/ directory structure, which you must create and make writable by the postgres user. When the conversion.php script is completed, simply transport those files to the new machine, along with CVS, SVN, and mailing lists.

If your existing install is not in 4.5 or 4.6 format, it must be upgraded first. There are upgrade scripts for doing this.

## Getting Started

On the new GForge AS machine, do a normal installation using the install scripts.

On the 4.5 GForge, be sure that EVERY user has a valid role in each project. If you are upgrading from 3.x, the roles are not assigned during the conversion to 4.x. This must be corrected manually in each project, so have each project admin go into their project, edit the roles, and assign the roles to each user. The conversion script will abort if even one user has an invalid role.

On the 4.5 box, as root, you must alter the GForge database user so he is a super user:

```
su postgres
psql gforge
alter user gforge createuser;
```

Exit the database and issue these commands. If you already have a /tmp/gf5/, make sure it is empty or delete it and recreate it:

```
mkdir /tmp/gf5
chown -R postgres:postgres /tmp/gf5/
```

Now you need the conversion.php script from GForge AS – this dumps the database and files into /tmp/gf5 in the proper format.

```
php conversion.php
```

In the GForge AS installation directories, there is a db/pgsql/svndump.sh script that you can run to properly dump out your data, which will be required if you are upgrading to a different version of subversion.

On the GForge AS machine, drop the database that was created by the installer. It will have to be re-created in a different way so that it is blank and can have data imported into it. The following set of commands shows how to recreate the database and import the data from the old machine.

```
dropdb -U gforge gforge5
createdb -U gforge gforge5
createlang -U postgres plpgsql gforge5
psql -U gforge gforge5 < /opt/gforge5/db/pgsql/gforge5.sql
```

```
psql -U postgres gforge5 < /usr/share/postgresql/contrib/tsearch2.sql
```

```
psql -U postgres gforge5
gforge5=# GRANT ALL ON TABLE pg_ts_cfg TO gforge;
GRANT
gforge5=# GRANT ALL ON TABLE pg_ts_cfgmap TO gforge;
GRANT
gforge5=# GRANT ALL ON TABLE pg_ts_dict TO gforge;
GRANT
gforge5=# GRANT ALL ON TABLE pg_ts_parser TO gforge;
GRANT
gforge5=# \q
#
```

```
psql -U gforge gforge5 < /opt/gforge5/db/pgsql/FTI-gforge5.sql
```

Make sure the data from the old machine is in /tmp/gf5 and chown it to postgres:postgres so it can be imported.

```
psql -U gforge gforge5 < /opt/gforge5/db/pgsql/import.sql > import.log
```

Apply all updates to the database:

```
php /opt/gforge5/db/pgsql/db-updater.php
```

Check the import log and see if any critical errors are observed. In some cases, there may be some invalid data in the old database that will cause problems (possibly invalid foreign key references, duplicate entries, or fields that are too wide for the new schema).

Now we will move the files from the tmp directory into their location:

```
rm -rf /var/lib/gforge5/filesystem/*
cp -R /tmp/gf5/filesystem/* /var/lib/gforge5/filesystem/
chown -R apache:apache /var/lib/gforge5/filesystem/
mv /var/lib/gforge/filesystem/docmanfolder/
/var/lib/gforge/filesystem/docmanfileversion
```

## Importing CVS

Now you need to import your cvs trees. CVS trees can be decompressed into the /cvsroot/ directory. Run the db/pgsql/cvsclean.sh script to purge out any cvstracker, aclcheck and cvssyncmail lines from the 4.5 installation. These lines will cause any commits to abort and fail.

## Importing SVN

The db/pgsql/svnimport.sh script can help you import all your svn trees from the svndump.sh script. Because of the way SVN works, you must dump/import rather than just transferring the repository like you do with CVS.

## Importing Mailing Lists

The mailing lists generally go into /var/lib/mailman and you will need to chown those and run the

mailing fix permission scripts. You may also need to reconfigure the lists if you used a new hostname. See the mailman documentation for more information.

## **Setup**

Your project administrators should go into the project and do several important thing.

1. Project Admin Setup
  - Go in and edit all of the roles and check the role assignments for all their users
  - Edit the 'Observer' role
  - Edit the SCM Admin preferences
2. Tracker
  - Go into the tracker admin and edit the fields and elements of each and every tracker.
  - Assign new sort\_order preference values, and default\_selected value for each field.
  - Delete any unused fields
  - Add any new 'FRS Release' fields that may be wanted. The new FRS Release field lets you tie defects, tasks, etc, back to the File Release System.

## **Testing**

To test your conversion, it is recommended that you maintain your old installation intact while you compare side-by-side to make sure the files and database were transferred properly. Look at doc mgr, tracker, file release and verify side-by-side that everything is converted and working.

Wiki and URLs that were stored in the doc mgr are not converted. URLs should be put into the wiki instead of into the doc manager in GForge AS, and old wiki content cannot be migrated as it's completely different.