ENORMOUSLY ANNOYING BOLDFACE WARNING: These instructions should not be used yet, due to the fact that I'm an SQL newbie myself and I need to independently verify them with someone who knows what they're doing. Plus, it might not be a complete process yet.

UPDATE: Looks good, a couple of outside sources have confirmed that the sky won't fall on your head if you follow these instructions. As far as I know, nobody has actually tried following them yet, so until that happens I'm leaving the warning message. 😀

Upgrading Tiki from one major version to another

This process assumes the following setup; you'll probably need to substitute your own stuff for most of the values here. Things you might/will need to substitute are in **bold** so they're easy to see.

This is the assumed configuration for this example:

- Tiki database username: tiki
- Tiki database password: secret
- Admin database username: root
- Admin database password: topsecret
- SQL database name (current/old version): tiki16
- SQL database name (new/upgrading-to version): tiki17
- Directory where Tiki is currently installed: /home/tikiwiki/public_html/tiki

1) % mysqldump -u **tiki** -p **secret** **tiki16** > tiki16-backup.sql

This exports your old 1.6 database to a file, so you can import it into another database. As a side note, this is also what you would do if you wanted a full backup of your database and for some reason didn't want to use Tiki's own backup function.

2) % mysqladmin -u **root** -p **topsecret** create **tiki17**

This creates the new database to hold your 1.7 Tiki data. I use database names like **tiki16** and **tiki17** instead of just calling them all **tiki** because it makes it easy to run multiple versions on the same machine.

3) % mysql -u **tiki** -p **secret tiki17** < tiki16-backup.sql

This copies all your old 1.6 data into the new 1.7 database. You can not just use this new database, though, because Tiki 1.7 is expecting the database in an updated format (new tables, new columns, etc). That's what the upgrade script does.

4) % mysql -u **tiki** -p **secret tiki17** < tiki_1.6to1.7.sql

This converts the 1.6-format data to 1.7-format data in your new database. You find this upgrade SQL script in the db/ subdirectory.

5) % cd **/home/tikiwiki/public_html**

Change directory to where your Tiki is currently installed (not *into* its directory, but into the directory where the Tiki directory is).

6) % mv **tiki oldtiki**

This moves your Tiki directory out of the way, so that when you unpack the new 1.7 archive it doesn't overwrite your current installation.

7) % tar xvjf tikiwiki_1.7.tar.bz2

Unpack the new Tiki archive. You should now have two directories: one called "oldtiki" with your current installation, and one called "tiki" with the new 1.7 installation.

8) This is the tricky step; you'll need to copy all of your Tiki modifications and stored files over to the same locations in the new "tiki" directory. You can *move* them rather than *copy* them if you like, to save time and disk space, but then you're modifying your existing installation, which may or may not cause problems if you have to do a rollback. Here's a checklist of things you may have to migrate:

- If you store your file galleries in a directory instead of in the database, copy that directory over

- If you store your image galleries in a directory instead of in the database, copy that directory over
- If you store your wiki attachments in a directory instead of in the database, copy that directory over
- If you have games installed, copy them over (don't forget the thumbs directory!)
- If you have any customized or completely new themes, copy them over
- If you have any customized or completely new modules, copy them over
- Re-edit "tiki-admin-include-features.tpl" and "tiki-admin_include_features.php" [under what circumstances would you need to do this? -daaave]
- To preserve images used in wiki pages, move the img/wiki_up directory over
- To preserve Galaxia workflow processes, move the lib/Galaxia/processes directory over

Any comments on this from someone who knows how different versions of Tiki react to using each other's file storage directories would be appreciated. Can you just create directories for these things outside the Tiki hierarchy and just change the pointers rather than move/copy anything? In other words, instead of having something like this:

~/html/tiki/filegallery/
~/html/tiki/imagegallery/
~/html/tiki/wikiattachments/

...you would just have:

~/html/tiki/
~/html/filegallery/
~/html/imagegallery/
~/html/wikiattachments/

You would then point both Tikis at the same set of directories. First question, would this work as an upgrade strategy (in other words, where even though both Tikis would be pointed at those directories, only one Tiki would actually be running at a time, so no problems with multiple Tikis stepping on each other)? Second question, would this actually work in production? In other words, would it be possible to have two (or more) Tikis simultaneously running, all using the same filesystem storage directories? I'm guessing the answer to this second one is no, but I figured it was worth asking. 😀

Another idea I just had; wouldn't it help the upgrade process if Tiki moved toward having a "local/" directory where all site-specific customizations like user modules, new themes, games, extra avatars, filesystem storage, etc are stored? The "local.php" db config file would go in there as well. That way, a properly configured Tiki install could be upgraded using the above steps and then just copying over that one directory. Thoughts?

.