Upgrade Management

From time to time, it is necessary to make changes that require some processing beyond just manipulating the database schema. For example, it may be necessary to change the values of existing records, or populate default values for some new attribute. Where possible, automate these changes using the UpgradeVersionShell. (There may still be times when manual intervention is required. Continue to document those steps in the wiki on version specific pages.)

Current vs Target

The *current* version is defined as the contents of `cm_meta`:`upgrade_version`.

The *target* version is defined as the contents of the file `Config/VERSION`.

$versions

Each new release must have an entry in the `UpgradeVersionShell::$versions` array. The key is the version number and the value is an array with the following supported entries:

- `block`: (Required) Boolean indicating whether or not this version blocks future versions in a single upgrade. See below.
- `pre`: (Optional) The name of a function (defined within `UpgradeVersionShell`) to be called *before* the database schema is updated. While any name can be used, in general follow the format `pre094`, replacing `094` with the appropriate version.
- `post`: (Optional) The name of a function (defined within `UpgradeVersionShell`) to be called *after* the database schema is updated. While any name can be used, in general follow the format `post094`, replacing `094` with the appropriate version.

Blocking

When the database schema is updated, it always bring the schema up to the latest version. Currently, schemas are not explicitly attached to releases. UpgradeVersionShell, however, can run pre and post upgrade scripts on a per-version basis. This creates a problem if, for example, a change is introduced in v1.1.0 that requires manual intervention, and an administrator attempts to upgrade directly from v1.0.0 to v1.2.0.

The solution is to flag v1.1.0 as *blocking*, using the `block` keyword described above. When true, UpgradeVersionShell will prevent upgrading past v1.1.0 until the current version is v1.1.0.

If you flag a version as blocking, be sure to document why. This is currently done in `UpgradeVersionShell.php`.