# Mods Details

# Morphology

Each Mod is defined by a type and a package name. the types are arbitrary strings to group mods in mods families. The initial set is composed of avatars, icons, menus, themes, languages, sqlupgrade, wikiplugins, features, services ... more types can be added with time, but please do not add wild types. New types should be only added after proper discussion with knowledgeable people (whatever that means for you).

A mod is managed by a control file with following characteristics :

- follows name convention type-name.info.txt
- stored in mods/Packages/ dir typically
- the syntax of this file is intended to be human-readable, composed by parameters and values
- each parameter block is separated by 2 line breaks
- the first line of parameter block is the parameter name, possibly ended by a colomn (:) which is stripped if present
- the following lines, until a blank one appear, is the value of the parameter
- the initial set of parameters is composed of :
    - **contributor** : the login or name of the one that commits, can use the $Author$ cvs macro
    - **revision** : the incremental number of the version of the mod. it can follow the revision number $Revision$ from cvs but not necessarly
    - **requires** : the optional list of required mods by this mods. It must be one line by depend, each line must be of the form: type-name < | > | <= | >= | = revision (and you can have multiple revision tests in one line)
    - **suggests** : the optional list of suggesteds mods. Same syntax than **requires**.
    - **conflicts**: the optional list of mods that conflict with this mod. Same syntax than **requires**.
    - **lastmodif** : the date of last version of the mod, using the macro $Date$
    - **files** : a list of all the files that compose the mod. One line per file, composed of the origin and the destination separated by a space. The origin is relative to mods/ and the destination relative too the tiki documentroot. Files in mods can be prepended with sample: prefix in which case they will be modified by the configuration at install time.

- **Description** : an arbitrarily verbose (but not much) description of the functionalities of the mod
- **Docurl** : the list of relevant urls (one per line) related to documentation of the mod
- **Devurl** : list of urls used for the development of the mod (usually, but not mandatorily, in http://mods.tiki.org)
- **Licence** : the legal form under wish the mod is distributed. If not specified, follows the tikiwiki licence scheme
- **Author** : the name, login or identifier of the author(s) of the mod. If it's a coolactive work, one name per line is accpetable.
- **Version** : the list of restrictive version for which the mod is designed. If that field is absent, any version should be able to use the mod. *(not functional yet)*
- **Changelog** : the list of changes, descending, usually one line per change
- **Configuration** : enables install-time setup of some values. One line per variable, composed of Label, name of the variable and default value from $_SERVER array (for example $HTTP_HOST will propose $_SERVER['HTTP_HOST'] as a default value in configuration screen). The configuration variables suubstituation will be processed only on files begining with "sample:" prefix in mods repository.
- **Configuration help** : the text displayed on configuration screen at install time (remember to avoid empty lines as they are parameters separators).
- **help** : a freeform text instruction about the mod installation and first use
- **sql-install** : the list of sql command issued at install time, one command per line.
- **sql-remove** : same as sql-install, but executed at removal time.
- **sql-upgrade** : that's a special sql statement list. It's splited by version numbers to which the upgrade sql statements apply. The version number has to be on a line prepended with a column. The lines below the version value will be associated to that version upgrade, so multiple upgrade can be run at once if needed.


The maintenance of the mods repository is managed by other text files that have to be present in the mods/Packages/ dir as well. Those files are indexes in CSV format, so they can be used by other tools. The indexes are following the conventions :

- name convention is that all indexes files begin with 00_list. and end with .txt so they are at the top of the list when Packages content is sorted alphabetically
- each line in indexes is composed by strings enclosed with simple quotes and separated with commas
- the fields are 'type','mod name','revision','description','licence' (probably more will be added, but the order shouldn't change).
- of course the content of fields has to be addslashed (simple quotes preceeded by a backslash).
- there is differnt index files for different uses
  - 00_list.txt : is the index of locally available mods. This index can be rebuidl locally by using the 'rebuild local list' link in tiki-mods.php
  - 00_list.public.txt : is the index that is generated when you setup your mods to be able to be mods provider. It lists the mods that are available for other tikiwikis
  - 00_list.xxxxxx.txt : is the index of remotely available from the sevrer configured in the tiki-mods_admin.php panel where xxxxxx is the urlencoded url of the server it comes from

# for developers : mods.sh

In 1.9cvs you can find that script in doc/devtools/mods.sh. This script is recent and has only a few features, but the goal is to make possible to manage the mods only with it. Before use **edit this script** to adapt the configuration to your environment, and when done you can launch it from your tiki root directory.

Usage: ./doc/devtools/mods.sh [copy|diff|install] copy : copy tiki files to mods diff : diff tiki files with mods install : copy mods files to tiki without 2nd arg, returns the list of files from package in tiki tree !!! note that this script only works with installed packages !!! !!! (this is work in progress, use is quite limited for now) !!!

# How to add your own mods

Let's work a simple mods, a wikiplugin

# Get _mods

First get a CVS version of _mods from HEAD. From now on, all paths will be relative to the directory _mods.

# Create a wikiplugin

Create your plugin: