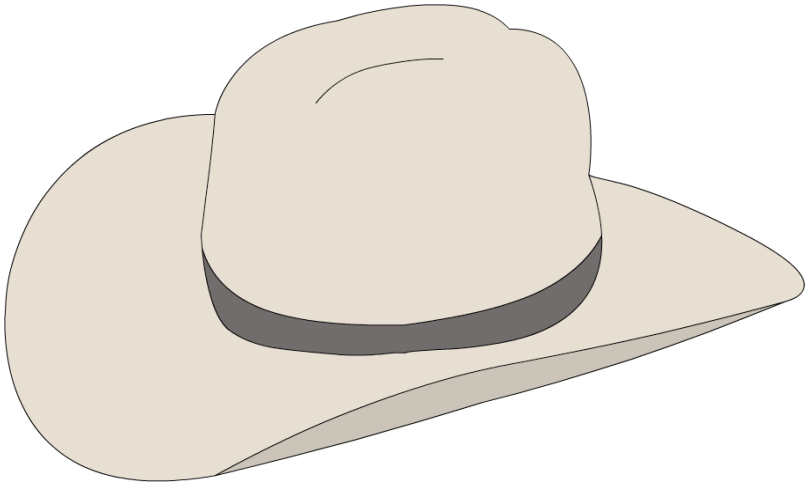

Bonanza

It’s all about the plug-ins..

Table of contents

Introduction

Bonanza is a CMS (Content Management System) written in PHP 5, whose top features are:

- Customizability
- Extensibility
- 99,99% localized
- User-friendliness
- Completely documented for end-users and developers
- Security (Full protection against SQL attacks, JavaScript attacks, SPAM attacks, either vulgar, aggressive or retarded visitors)
- Formulate
- Minimal loading time
- Fast caching framework
- Advanced templates system

Main Configuration

These settings are implemented at system level and cannot be changed without editing code. Most of them can be set upon installation of the core system.

- Database connection string (e.g. 'mysql://root:pw@localhost/bonanza')
- Extension auto-load (e.g. Array [sys_core, sys_linker])
- Path – Working Storage (e.g. 'WorkingStorage')

Core System

The system is a framework managing:

- Installing, activating, de-activating and uninstalling extensions
  - Register pluginables
  - Register plug-ins to the pluginables
  - Run queries, calculate counter-queries
  - Register libraries and make them available
- Loading extensions
  - Load localizations

Working Storage

Any extension gets a directory to store temporary or cache data in the working storage. This is a world writable directory, whose path is specified in the configuration. This directory is preferable not in a web accessible place.

Extensions

What are extensions for, what can they do?

- Register *pluginables* (to the system)
- Register *plug-ins* (to the matching pluginable)
- Offer functions / libraries (for other extensions)

Note: *register* indicates that it only must be done once, meaning during the installation of the extension. The others are on the fly.

How Extensions Work
Upon installation of the extension, directories in the *plug-ins* folder will be submitted to the matching pluginable. The pluginable will process the data the way it needs to be done.

Extension indexes implement from the following abstract class.

```
abstract class Extension
{
        abstract private var requires;
        abstract private var install_queries;

        public function install() { };                  // Runs install queries
        public function activate() { };       // Calls pluginables
        public function deactivate() { };   // Calls pluginables
        public function uninstall() { };      // Runs uninstall queries
}
```

Upon installation the pluginables will be called with an *install* code. Upon uninstalling, the pluginables will be called with an *uninstall* code. The uninstaller also calculates the opposite of the installation queries, to undo them again. Extensions can be deactivated without uninstalling.

index.php contains following structure.

```
class Search implements Extension
{
        private var requires = array("sys_core", "sys_templater");
        private var install_queries = array("");
}
```

If needed the install() and alike functions can be overwritten, if more needs to be done than just running queries.

Pluginables
Each pluginable receives a string (full path to directory) when a new plug-in wishes to join in. This string consists of the directory name of the new plug-in. The system makes sure plug-ins are submitted to the correct pluginable. The pluginable verifies if the new plug-in has all requirements (following his own protocol).

- A new extension is being activated.
- The system looks for pluginables and adds them to the list of available pluginables. If pluginables are found, all currently installed extensions will be checked for plug-ins for this pluginable.
- The system looks for plug-ins and submits them to the correct pluginable.
- If a pluginable receives a new plug-ins it handles it it's own way.

On the fly
Some plug-ins may be created after the extension was already installed. i.e. paginator submits new links to the sys_linker when a new static page is created. There will be a world writable plug-in folder available in the working storage directory.

Extension Rules
When an extension is uploaded, the system calculates a hash and verifies whether the extension is approved by us via a link similar to:

http://www.bonanza.com/certifcate?hash=b7104130175d92728c29d84a22a736fd

The Bonanza server will keep a list of extensions they have validated. People who wants their extension validated must submit them in a special section of the Bonanza site.

Following rules must be obeyed.

- Main language must be English
- Security
- Correct use of system's and other extensions functionalities
    - Localizations (no hard coded languages)
    - Templates (no hard coded html/CSS)
    - No useless or duplicate code

Certificated extensions will be marked positively upon installation; otherwise there will be a warning that there might be risk.

System Extensions
This is a list of extension that are part of the Bonanza Core System.

sys_acp

Requires	sys_core
Pluginables	AdminPanes, AdminPanesCategories
Plug-ins	PermissionFields, LinkerPages (control panel)
Description	Administration panel, allowing easy management of quite everything

sys_blocker

Requires	sys_layout
Pluginables	Blocks
Plugins	AdminPages
Description	Manage blocks for insertion in the layout.

sys_cacher

Requires	sys_debugger
Pluginables	CacheImplementations
Description	A framework allowing caching of any data

sys_core

Requires	sys_debugger, sys_cacher, sys_database, sys_settings, sys_user, sys_permissions

Description An empty extension allowing easy loading of the core extensions.

sys_database
Requires sys_debugger, sys_cacher
Pluginables DatabaseConnectors
Plug-ins CacheImplementations (for caching database queries)
Description Manages your database connection.

sys_debugger
Requires *nothing*
Description Handles debug information, catches errors and handles them in a clean way.

sys_formulate
Requires *nothing*
Libraries class Formulate() – create a form to work with
Description Allow abstract forms, allowing easy validation, inserting, updating and deleting of database records.

sys_layouter
Requires sys_core, sys_templater
Plug-ins AdminPanes (layout management and editing)
Description Layout manager

sys_linker
Requires sys_core
Pluginables LinkerPages
Plug-ins AdminPanes, PermissionFields
Description Link manager, makes sure the correct page is loaded.

sys_permissions
Requires sys_debugger, sys_cacher, sys_database, sys_settings, sys_user
Pluginables PermissionFields
Plug-ins AdminPanes
Description Set permissions per group/user

sys_settings
Requires sys_debugger, sys_cacher, sys_database
Pluginables SettingCategories, SettingFields
Plug-ins AdminPanes
Description Management of various settings from various extensions

sys_templater
Requires sys_cacher
Pluginables Templates
Plug-ins CacheImplementations
Description Templates engine, all html and content go through here

sys_textarea

Requires	*nothing*
Pluginables	TextereaComponents
Plug-ins	AdminPanes (enable/disable and re-order components)
Libraries	class Textarea() – An object to parse a textarea
Description	Library to parse a textarea with all the additional features you wish

sys_url_rewrite

Requires	sys_core
Pluginables	UrlRewrites
Plug-ins	AdminPanes
Description	Allows URL rewriting using the apache mod with the same name

sys_user

Requires	sys_debugger, sys_cacher, sys_database, sys_settings
Pluginables	UserFields, UserPanes, UserPanesCategories
Plug-ins	SearchFields, LinkerPages (member list, login, registration, logout, control panel)

Additional Default Extensions

Extensions that aren't part of the BCS, but will be made by us and shipped by default.

Search

Requires	sys_core, sys_templater
Pluginables	SearchFields
Plug-ins	SettingFields, Blocks, LinkerPages
Description	Search for anything!!

Paginator

Requires	sys_core, , sys_templater
Plug-ins	AdminPanes, LinkerPages
Description	Management of static content

Appendixes

File and directory structure

Files are in *italic*, the other entries are directories.

- Extensions
  - Paginator
  - Search
    - *index.php*
    - Localizations
    - Plugins
      - SettingFields
      - Blocks

            - LinkerPages
        - Pluginables
            - SearchFields
- *index.php*
- Plugins
- System
    - Extensions
        - sys_acp
        - sys_blocker
        - sys_cacher
        - sys_core
        - sys_database
        - sys_debugger
        - sys_formulate
        - sys_layouter
        - sys_linker
        - sys_permissions
        - sys_settings
        - sys_templater
        - sys_textarea
        - sys_url_rewrite
        - sys_user
    - Plugins
        - AdminPanes
        - AdminPanesCategories
        - DatabaseConnectors
        - TextareaComponents
            - BBCode
            - Smilies
            - Attachments
        - UserFields
- Working Storage
    - Plugins
    - Extensions
        - sys_debugger
        - sys_linker
        - *etc…*