RC Publisher 2

# Specifications

RC Publisher 2 is to be a Wiki type web experience, though not in the traditional sense as it is not intended for public modification. It is meant to be an easy way for website administrators to add and delete content.

The site is divided into two main parts, with some additional features. The two main features are pages and files. The additional features include a news feed.

## Pages

Pages are the brunt of the website. A page is defined by text which contains a basic markup language. Pages may imbed certain types of files (such as images). A complete history of each page is stored in case a previous revision needs to be restored. Each page may be commented on. Pages may be deleted, but this merely makes them inaccessible, the entire history is still maintained and they may be restored. Comments may be deleted and otherwise moderated. Deleted comments cannot be restored.

## Files

Any type of file may be uploaded. Most likely image files will be uploaded. A file may also be pulled from another site and then stored locally. A file may also be replaced by a new version of the file or deleted. A history of each file is not maintained. Images and other file types may be embedded into a page. Files are stored in a special files directory, the path to a file is given by the /f/i/filename.ext (first two letters for path, followed by the full name of the filename). Files must be at least three characters long before the extension. All filenames are converted to lower case when they are uploaded. Any references to files is also converted to lower case. All files are referred to by their filename, the actual path is irrelevant and exists only to speed up loading a file from the file system.

## News Feed

The news feed is meant to be an aggregate of news stories. New news stories may be posted, and a list of news stores may be displayed on a page.

# Database Tables

## Page Table

The page table is a list of all pages that are available or ever have been available. It does not actually contain page data, instead it contains a reference to the page ID as well as the “current” version of the page. This contains the page slug.

### tblPages

* id
* txtSlug
* idPage
* idVersion\_Current
* txtTextHTML (This is a formatted version of tblPageHistory::txtPage, that is convered from RC markup to html).

## Page History Table

This contains all the actual page data and page title for the entire history of all the pages.

### tblPageHistory

* id
* idPage (matches tblPages::idPage)
* idVersion (current version of page is tblPages::idVersion\_Current)
* txtTitle
* txtPage
* dt (date this revision was posted)

### tblNews

* id
* dt
* txtTitle
* txtArticle

### tblFiles

* id
* txtSlug
* txtDesc
* dt
* txtType (Predefined types img, pdf, txt, etc…)
* txtFileLocation (location of file should always be txtName[0]/txtName[1]/txtName)

# RC Markup

The idea behind RC markup is to create something capable of many of the features of HTML, but limited to the needs of an RC site. It is also meant to be typed more naturally without the need of tags. Many of the features of RC markup will be inspired by MediaWiki markup. Some items include:

* Italicized text is given by ‘’two single quotes’’.
* Bold text is given by ‘’’three single quotes’’’.
* Bullet points are declared with a “:\*” (That is colon star). And the newline break. Generally you’ll want to declare bullets at the beginning of a line, but other possibilities are allowed.
* Links are done with [[page Brackets for an internal link.]]
* External links are done with [[site:{{address}} Description.]]
* Files are embedded with [[file:slug {{parameters}}]]. Parameters may be different depending on the type of file. Different parameters are separated by pipes “|”. Parameters should be unique enough that it doesn’t matter what order they are in. Currently image files resolve to display the image. All other files turn into a link.
* Blog links. Links to the associated blog are given by [[blog:{{slug-url}} Description.]] The description is optional and it will be the text for the link.