

A Basic Guide to Your Joomla! 1.5.9+ Website

New Local Media
newlocalmedia.com

February 2009



CONTENTS

1. Security and Upgrades
2. Performance Optimization
3. The ACL
4. User Management
5. Menus, Templates, Pages and Modules
6. Creating, Placing, and Managing Ads and other Modules
7. Search Engine Friendly (SEF) URLs
8. Handling Images in Articles
9. Typography

1. Security and Upgrades

The parameters listed below are important for securing you website, especially on shared server hosting. Actual configuration will vary from site to site depending on individual needs and various requirements or limitations. As a standard practice, New Local Media will specify the settings in use within the documentation provided for each site built by New Local Media. Subsequent changes to these settings should be justified and documented.

Many of these security-hardening measures are based on best practices recommended by the international Joomla user community, the [Joomla Security Checklist](#), the [Joomla Security and Performance FAQ](#) and Tom Canavan's [Joomla! Web Security \(2008\)](#). New Local media keeps a detailed compendium of security basics and recommended practices online at <http://www.newlocalmedia.com/learn/17-dont-be-the-dumb-cow.html>

Server and PHP Environment (php.ini and/or .htaccess settings)

- Default Joomla **core SEF redirects** active
- Default Joomla **exploit prevention redirects** active
- **No remote database access.** If needed, only restricted IPs allowed.
- **Nonstandard database names**, prefixes, etc.
- **.htaccess** locked from public view, if possible
- **disable_functions = show_source, system, shell_exec, passthru, exec, phpinfo, popen, proc_open** (Note that prohibiting phpinfo means it cannot be accessed from the Joomla backend either, so System Information→PHP Information will appear blank.
- **open_basedir = NOT USED / NO EXCLUSIONS** if backups are saved above the webroot. If you do set tighter restrictions with open_basedir, you will not be able to generate backups above the webroot from within Joomla's backup/restore extensions. You will need to find a workaround or else download backups immediately after creating them. *Do not leave backup files on or below the webroot.*
- **magic_quotes_gpc =**
- **allow_url_fopen = 0 / OFF**
- **short_open_tag = 0 / OFF**
- **safe_mode = 0 / OFF**
- **register_globals = 0 / OFF**
- **tg_emulation = 0 / OFF**
- **allow_url_fopen = 0 / OFF**
- **display_errors = 0 / OFF**
- **IP banning:** In some situations you may want to use selective blocks on known sources of trouble.

Joomla:

- **Remote database connections:** If not needed, OFF
- **XML-RPC:** If not needed, OFF
- **SSL for Admin Login:** ON for whole site if needed at all
- **SSL Login Mandatory:** YES if SSL needed at all on site
- **Restricted IPs for Admin Login:** Site dependent
- **Special link to Admin login panel:** YES
- **Log file location:** Stored above web root, if possible.
- **Joomla FTP layer:** Disabled
- **Joomla Legacy mode:** Disabled
- **File and folder permissions (root):** 755 for folders and 644 for files, or lower when write access is no longer necessary.
- **configuration.php** stored above webroot, if possible.
- **Nonstandard name and number for initial superadmin.**
- Default **robots.txt** for Joomla with additional restrictions, as needed.
- **Restricted editor and upload access** for users as needed.
- **Administrator login panel** double password protected, restricted by IP, and/or accessible only by a nonstandard URL.
- Keep a list all the software you have installed on your site, including Joomla extensions, and note the current version numbers. Keep the software and this list updated. Uninstall anything you don't use.
- Add security hardening and monitoring extensions available for Joomla as needed.

Maintaining Security Discipline: It's On You

Don't be an easy target. Make security a discipline and a habit. This is all on you, because only site owners (and the people they authorize) know how to access the major entry points: Email accounts, domain registrar/DNS services accounts, FTP accounts, databases, and web-based applications you've installed (e.g. Joomla) all have windows and doors you need to keep locked.

- **Use strong passwords and hard-to-guess login id names.** Do not share them or write them down anywhere.
- **Change all passwords on a regular, routine basis.**
- **Don't casually create new accounts** or give out more access to users than they need when you are able to be restrictive.
- **Eliminate old accounts** when they are no longer needed.
- **Be careful with mail server access.** If users with email accounts on your server have their accounts compromised by physical theft or infection by malware, your mail system may be used to push spam resulting in the blacklisting of your domain and IP, which will affect all your mail users. It may also cause you site and/or internet access to be terminated.

Keeping Your Site Secure Over Time

This is all on you too unless you are retaining New Local media or another entity to do it for you.

- **Monitor Security News for Critical Updates and Make Timely Updates.** *This is the number one priority for security maintenance and is fully the responsibility of website owners/operators, especially with regard to critical core updates to Joomla.* The Joomla Security Strike Team (JSST) actively seeks out security vulnerabilities and immediately acts to secure them when they are found. This process, along with the regular development cycle for new Joomla releases, is one you should follow at Joomla.org. You may also subscribe [by RSS and/or email to JSST alerts](#) which are conveniently displayed on this page to the right, with other Joomla news. Every extension added to Joomla that is provided by a third-party developer should also be monitored for critical security releases. New extension releases can be monitored via the [Joomla Extension Directory \(JED\)](#) and the relevant extension developers' websites. Many extensions have internal notifications or version/update checkers on their backend admin screens. Third-party extension developers often have RSS feeds and/or email newsletters to help you keep abreast of their current release status. Use these resources. Secunia Advisories (free by subscription from Secunia.com) list the latest known public exploits for all software, including Joomla.
- **Take regular backups of the Joomla database and file system, download and delete them from the server.** Use the **JoomlaPack** backup component, and make sure it saves backup files in a folder above the webroot where they can't be accessed by any browser. Then download backup files from the server to local storage in case of server failure, even though the host keeps backups as well. Leaving no backup files on the server will make it impossible for attackers to exploit them should they break into the server itself.

Upgrading Joomla! and its Extensions

Some Joomla! extensions have auto-update functions and can update themselves at the click of a button. There are also some recently developed extensions that add this capability for the Joomla! core and certain third-party extensions. These capabilities may become more common in the future. Currently most upgrades must be performed manually.

Typical Upgrade Process:

1. Take a fresh backup of the site files and database.
2. Read any instructions for the upgrade files, possibly contained within the compressed (.zip/.tar/.gzip/.tgz) upgrade file package.
3. Follow the instructions. Usually they will involve:
 1. Uninstalling an old extension and then installing a new version from the admin backend.
 2. Uploading new extension or core Joomla! files over the old versions. In rare occasions you may need to make changes to the database as well.

4. Test everything to make sure it works.
 1. Menu items for components that are uninstalled and reinstalled will need to be reconfigured.

Keep copies and keep track of modified Joomla! core and extension files—they may be overwritten during an upgrade. Commonly modified files are the default Joomla language files (usually EN-US and EN-GB) and others added by third-party.

Your Security Baseline

The settings discussed in the previous section are important, because in their totality they comprise a baseline or reference point. This is your “optimally secure” model, indicating what is being protected when all is well and as it should be. You need this model in the following scenarios:

- ▶ **Your actual settings appear to have been changed.** Have they? Check your baseline settings to be sure. If there’s been a change, either an authorized person has made a mistake, or you have an intruder.
- ▶ **You need to change or restore the original settings.** In that case, where you started is crucial knowledge.

Unless you are doing *a lot* of sales online or have a site that (for whatever reason) you can’t afford to have taken down, even temporarily, you’re probably not going to bother with active security monitoring. But if you do active monitoring, a solid baseline is essential in the following scenario:

- ▶ **You are examining your server logs or using automated security monitoring.** You observe an uptick in attempted exploits. Many seem to pertain to a particular area or areas of potential vulnerability. Is it a weak spot? Are you covered? What should you do? First, check your baseline. In situations like this is also important to have additional baseline information, like the ratio of all site traffic to attempted exploits. A certain (small) percentage of your weekly/monthly traffic will be script kiddies bouncing off your shields or fishing for a vulnerability in an extension you don’t have installed on your site. How much of this activity defines a “normal” range? If that activity increases a lot, you know something is up. Time to be extra-vigilant. Have you missed an important patch or upgrade? Should you just block the IPs of persistent exploit attempts, or is there more you need to do? Your baseline lets you come up with an informed response.

2. Performance Optimization

Within Joomla’s global configuration settings, the following settings improve page load speed if you activate them: gzip page compression and caching. Page caching is also controlled through a system plugin, and most modules can be cached or left uncached based on their individual settings.

For Joomla's dynamically generated pages, caching will improve page load speed and decrease the server processor load by not generating pages dynamically when there is likely no reason to do so. When a page request is made, active caching means a copy of that page will be stored as a file for a designated period of time before generating a new version. (The typical default is 15 minutes.) Any further traffic to that page will be given the cached file, instead of querying the database to generate a fresh page.

Most modules can have their cache turned on or off individually. By default most modules will follow the global cache settings in Joomla's global configuration. If it's on, the modules will be cached. Modules typically allow you to specify the time delay before the module cache is refreshed.

Keep in mind that not all modules can or should be cached, such as those that are supposed to show constantly changing content that refreshes with every page load.

In addition to the core caching features, there are a variety of 3rd party extensions that can be added to Joomla to enhance the speed of its administrative backend, such as the **AJAX toggler** and **Google Gears** support.

Typical NLM Joomla sites will make use of all these core optimization features and possibly some additional third party enhancements.

3. The ACL

Joomla's **Access Control List (ACL)** offers fixed roles-based permissions for 7 user types or levels: **Registered Users, Authors, Editors, Publishers, Managers, Administrators, and Super Administrators**. These levels are described in detail [here](#). The first three have frontend access only. The last three have access to the administrative backend but only Super Administrators have access to everything on the backend. Only Administrators and Super Administrators have access to user account management functions.

Content articles, menu items, modules, and plugins can all be designated as visible and accessible to anyone (**public**), only to **registered** users (anyone with an account), or only to "**special**" users, i.e. those with backend access. More complex and granular ACL functions require additional extensions as of Joomla 1.5.x. Joomla 1.6.x will be distributed with a more customizable ACL.

It is important to understand the ACL since it is the key part of creating a workflow process for administration of your website and protecting some content and functionality from being accessed by unauthorized users.

4. User Management

Joomla's User Manager (Site→User Manager) should *not* be used by if the **Community Builder (CB)** or a similar component is in use. CB integrates with and expands the Joomla user manager. Use CB's User Management feature instead under Components→Community Builder→User Management.

Primary user management tasks include: creating new user accounts, approving and rejecting newly registered user accounts, deleting or blocking user accounts, or editing their account information.

*In some extensions that allow user-submitted content, there are also access and user management controls where administrators can approve, deny, edit or block user-contributed content. For example, the **Joomla Content Editor (JCE)** and **EventList** calendar component has its own extensive user management functions on the backend to handle events, venues and other material submitted by users; it also allows the creation of special groups of users that have special privileges with regard to the calendar's functions.*

5. Menus, Templates, Pages and Modules

Core Concept: What Makes a Page in Joomla?

In database-driven websites like those created with Joomla, pages are generated "on the fly." The URLs Joomla generates do not refer to files on the webserver. They are actually commands that are interpreted as database queries asking for X content from component Y (and its own taxonomies) under menu item A.

There are no web pages in the form of static HTML files that sit on the webserver, except when you have Joomla's cache feature turned on, and even they are not truly static. How does Joomla know what to send to browsers then? It's rather simple, but it can quickly become very complex:

Menu items determine page identities.

Since menus are the navigational and organizational backbone of any website, Joomla identifies every menu item as having a corresponding page or pages and lets you assign modules (and templates) to those pages. You can create pages that have no tie to any menu, but you will not be able to assign any modules to them except for modules that are assigned to all pages. One way to work around this is to assign pages to a hidden menu whose module position is non-existent or never actually visible.

Here is an example of how page generation works. This page you are looking at now is currently generated with the following URL:

index.php?option=com_content&view=article&id=60:templates-and-modules&catid=28:documentation&Itemid=133

If you have SEF turned on, this is the "real" URL behind the simplified, more readable SEF URL. This URL will always generate this page as long as the items it references exist in Joomla.

Here how the URL is "read" by Joomla. Index.php is the base page display file--it's an actual file with program code that manages page creation. What follows next (after the question mark) instructs index.php to run the core content component in Joomla, instructing it to display the article with the ID number of 60. It also notes the alias ("templates-and-modules") for this article, its article category ID and the category's alias ("documentation"), which is helpful for SEF URL functionality. Finally our page URL conveys the itemid, which refers to a menu item. Since there is no direct link to this page from a menu item currently, the item ID for this page is associated with the nearest menu item ID, which is the "Documentation" link under "Staff Guidebook" under "My Account." Consequently the settings for any of those menu items can affect the display and behavior for this page, as they are hierarchically linked, and many settings in a hierarchy within Joomla cascade down unless overridden at a lower level.

Template Basics

Templates are installed like other extensions. Any number of templates can be installed. For instance, you might want a secondary template that mobile devices use. (There are extensions for Joomla that will identify mobile device browsers and route them to a special template.) Yet one and only one template must be the default template. The default template is used to determine how pages are rendered unless additional templates are assigned to specific pages.

Templates consist of a master index file composed of PHP rendered X/HTML, any number of stylesheets, associated graphics, possibly [template overrides](#) that provide sub-templating for core and third-party components, modules, and plugins. Certain JavaScript libraries and custom code (e.g. built-in menus) may be included in some templates. The master index file and stylesheets can be edited from the Joomla backend. There are often settings as well for governing options coded into a template, such as a choice of menus, the width of the template, color schemes, etc.

Making Modules Appear *Where You Want Them*

Installed modules are assigned to module *positions* in the **Module Manager** and given an access level, just like menu items, articles, and plugins. If there are more than one module in the same position, they will appear in the order determined in the Module Manager.

Module positions are written into site *templates*. If template "A" has a module position "alpha" and module B is assigned to "alpha," module B will appear on the site as rendered for browser requests *if the following conditions are met*: 1) the module is published (turned on), 2) the requesting browser/client has the requisite access level; 2) the module is assigned to the currently viewed page or all pages; 3) the module position

is actually invoked by the template when the page is rendered and served to browser requests. If a module position is not invoked at the template level then the modules in that position will not be displayed *regardless* of their published state and page assignments.

Regarding the last point, most templates are written so that all their module positions are potentially executable on every possible page. However, template code logic can be written to perform checks and determine whether to invoke a module position based on certain conditions--e.g., only invoke this module position if the requesting browser is on the home page, or only if another module is active, etc. Such checks are performed to generate a unique home page layout on this site. Additionally, some modules invoke one or more other modules that may be assigned to positions that may never be invoked by the template, and these modules may have their own conditions for doing this. E.g., module "alpha" displays all modules in module positions "beta" and "gamma" which are never invoked by the template.

Making Modules Appear As You Want Them

The location, size and other visual attributes of the module *positions* are defined in the template HTML/CSS, which may also cascade to affect module output, particularly when module CSS class suffixes are used on modules within the module manager, allowing them to be styled individually.

The master template's default module styling can be cut off simply by adding a bogus module suffix (in the module's settings) that does not correspond to any CSS class or corresponds to a class that you define.

The module output itself is determined by the module's code and its own stylesheets and/or templates and template overrides, if it has them. Joomla 1.5's "[module chrome](#)" offers some built-in styling possibilities if it's not overridden by templates and other CSS.

Joomla 1.5 also implements a [model-view-controller \(MVC\)](#) architecture that modules and other extensions should and increasingly do adhere to, but this is not always the case. Following an MVC architecture means extensions' program logic--the primary code determining raw content output--is separated from design/layout specifications, which are governed by external templates, template overrides, and/or stylesheets.

Tip: Use the **jPosition** component to see a list of all module positions and what's currently in them--this is a little more visually accessible than the Module Manager. You can also click the "Preview" icon within the template manager to see the module positions outlined over the current output of the default template.

6. Creating, Placing and Managing Ads and other Modules

The Banner Component

The **Banners** component is a simple, basic, yet fairly robust advertising management extension that is part of the Joomla core/default installation. It does not allow automated ad purchasing, more than basic statistical reporting (impressions, clicks, and CPM), or user control of the ad system from the frontend, but it is easy and quick to use. To use Banners, you will need to set up **clients** to own the ads, **categories** of ads based on their size and position, and the **ads** themselves.

Ads must be allocated a certain number of impressions (time they appear) before expiring or have an unlimited number of impressions. There is no date tracking for banners, but there is a note field you can use as an ad management aid.

Ads can be assigned tags/keywords. This corresponds to a setting in the ad display module that, if turned on, will only display ads with tags that match other tags/keywords assigned to a page, i.e. article metadata settings.

Ads can also be designated as "sticky." If one or more banners in a category are sticky, they will take priority over banners that are not sticky. For example, if two banners in a category are sticky and a third banner is not sticky, the third banner will not display if the module setting is 'Sticky, Randomize'. Only the two sticky Banners will display.

Banner Modules

There are numerous third-party ad display modules you can install that work with Banners. The Joomla! core banner display module will randomly or sequentially display ads from one designated banner category. A keyword matching function can be turned on, if used, to match ads with relevant content.

Some pages on your site may not be able take ads (or any modules) or shouldn't have them because will make the template look bad or break. Generally it's not good to have ads or other modules in the sidebar columns if their vertical length routinely exceeds the vertical length of the main content area. (Likewise, the main content area's vertical length shouldn't radically exceed the vertical length of an active side column, but that doesn't look as bad as the other way around.)

7. Search Engine Friendly (SEF) URLs

Search engines interpret URLs as part of their effort to understand the subject, authority, and relevance of the content tied to that URL. The kind of URLs generated by Joomla and other CMS applications used to confuse search engines because they offer little semantic content, and there is more than one possible URL that will display approximately the same page content. An option to generate "Search Engine Friendly"

URLs in the Joomla core and through various third-part extensions was developed to address this problem.

That said, search engines have adjusted to dynamic URLs, and SEF functionality is no longer a significant aid to search engine optimization (SEO). SEF does offer some SEO benefit if words in the URL match keywords in the corresponding page's content and metadata. But for the most part, SEF just looks good and makes page addresses more memorable to people. "Human-readable URLs" might be a better classification for SEF.

SEF works by redirecting the still always valid normal URL to an aliased URL that's simpler and uses words and numbers that are meaningful to people. Multi-lingual translations are also possible. Even with automatic caching of SEF URLs, SEF takes up more of the server's resources and can cause performance problems in some cases. However there are some security benefits to SEF as well, and the leading third-party SEF extension, **sh404SEF**, has some security and SEO enhancements as part of its capabilities.

Tip: *SEF can be tricky with page redirects and modules that use them, like third-party login modules. Avoid making changes to the login system if possible.*

8. Handling Images in Articles

Joomla's built-in **Media Manager** (under "Site" in the Administrator backend menu) allows you to upload files and organize them on the server for use in Joomla content. Images are stored in /images/stories and its subfolders. Use consistent folder and file naming practices to keep accumulated material easy for you and others to find.

You can access the media manager from within rich text editors like **TinyMCE** and **JCE** to upload and insert images in the manager into articles. Without working directly with the underlying HTML/CSS, you're limited to doing static image insertion and placement- unless you use the **JCE rich text editor's** advanced features, or specialized image effects . Many extensions for specialized image handling exist for Joomla and can be installed to take advantage of their features.

Further documentation about the Joomla Media Manager and JCE can be found at joomla.org, (see the [documentation wiki](#)) and joomlacontenteditor.net.

9. Typography

Certain typographical enhancements are part of a good master template's style sheets and can be used by coding references to their appropriate IDs and Classes into your page elements. They can also be made to appear as options in the 'styles' selection box in the installed rich text editors (TinyMCE, JCE), so they can be used without directly manipulating the underlying code, though often that is necessary for proper output results.

There are several third-party plugins, like **xTypo**, which offers some additional typographical features that can be inserted into articles via bracketed commands like any other plugin.