Boost the performance of your website

The Joomlatwork cache component isavailable for Mambo 4.6x, Joomla 1.0x series and the Joomla 1.5.

We are proud to release the new 3.0 release of the cache component. The cache component is now also available for Joomla 1.5 and Mambo 4.6.x. The 3.0 release is build from scratch again based on an own code base and libraries and is available for Mambo and Joomla websites Version 1.0b of the cache component is still available for Joomla 1.0x serie only or higher.

Standard out of the box Joomla and Mambo websites generates dynamicly webpages on the user request. This means that for each request the specific page must be composed by excecuting PHP code and SQL queries. On normal "hosting" environment the generation of a Joomla pages can take up to 0,4 - 1,4 seconds depending on the hosting environment and the number of components used. For large volume visitor sites this can result into slow performance of the website. When enabled this component can "generates" pages within 0,02 - 0,001 seconds (depending on your hosting environment)

NEW release: d.d. 29 june 2008
The new release has the ability to compress, combine and compact Javascript and CSS files.

This can reduce the load time of the page if you have multiple CSS and Javascripts files included (which is currently normal with the web 2.0 layouts). Read on the next page about these new features!

If you want to boost your performance of your Joomla / Mambo website this component can help you. Before buying the component read the next coming paragraphs:

The component actual creates an cache environment around Joomla and Mambo. Please notice that this is a different caching mechanism that is default available within Joomla and Mambo: The default caching engine still needs to execute a lot of PHP code and SQL code and is therefore slow or will not function correctly.

Important configuraiton settings for boosting the cache engine for Joomla 1.5

How is this performance component working?

The basics of the caching engine is simple. When a page is requested by a user the cache components first checks if the specific page request is within the cache repository. If the page is present within the repository it checks if the cache is still valid, based on the time to live configuration parameter. If the page within the cache is vallid it is retrieved from the cache repository and sent to the user.
If the page is not present within the cache or if its no longer valid Joomla will generate the page. At the end of the page generation process the output of the request is written to the cache repository (if it's allowed by the filters which can be set within the cache configuration).

Please notice that this cache component is working with other core and third party component as long as these components doesn't alter the index.php file.

The caching mechanism will not work if a user logs in to your website. The cache is then disabled for the specific user until the user environment. If a users logs into the Joomla

environment (frontend) the cache for the specific user is disabled until the users logs out. Actual you can configure the cache engine so that based on specific actions the cache can be disabled and you can enable the cache again also based on specific actions. For example the default config is when a user logs in (= POST action) the cache is disabled for the user until the user logs out (also an action)

The cache component functions almost like a static HTML page, although some php code must be executed to check and validate the cache. This can save you a heavily load on your SQL backend environment for Joomla / Mambo

Before we have launched this component we have tested this component throughly within several websites. For example the component is now running on the KiKa website (http://www.kika.nl) and has drammaticly increased the performance of the website. With sometimes more then 3.000 visitors a day bad performance is no longer an issue. Standard page generation time was arround 0,5 - 5 seconds before the cache component was implemented and after the implementation the generation time is arround 0,02 - 0,004 seconds.

Combine, compress and compact Javascript and CSS files in Joomla!

The new release 3.1 (release date 29 june 2008) has the ability to combine, compress and compact Javascript and CSS files beside the caching mechanism of the JRE cache. When your website has multiple CSS and Javascript files you can reduce the load time by enabling the optimizing settings for Javascript and CSS files. Not only the files are combined into 1 CSS or Javascript file but the content of the CSS and Javascript can be compressed also. This results into fast load time of the CSS and Javascript functionality within your webpages.
The functionality of this is simple. If you look into your HTML head code you can see that multiple Javascript files and CSS files are linked within the header. Each CSS / Javascript link must be requested at your server which causes traffic between the webbrowser and your server. The more links you have in your HTTP header the more request are needed to your webserver. When you combine these files into 1 javascript or CSS file there is only one request needed which results into faster load times (there is only one request). Beside this you can also compress (zip) the Javascript and CSS content to the webbrowser which reduce the loadsize with around 15 - 30% of the orginal size, which again speed up the load time.

NOTICE at activation:

When you active the optimalisation of the Javascript and CSS please test this ***good***! This because Javascript files can have internal link to other Javascript files which can be broken once they are included into one Javascript file. The solution is then to put these files that causes the errors into the dissallow optimalisation filter for the Javascript or CSS settings. Always clear the cache after you have changed the settings at the desktop of the cache component. This because the Javascript file is cached also and included into the cached files.
Test also your layout of your website! Although the optimalisation settings for the CSS takes care of rewriting relative URL image links within the CSS it could be that this is not function well.

Notice about the inclusion of CSS and Javascript files:
Only physical Javascript and CSS files can be combined and compressed into on one file. Dynamicly created Javascript and/or CSS files will not be included into the cached Javascript / CSS file (for example this is the case at the Virtuemart component for CSS output).

You can test the savings and reduction of the page size with the Firefox developers plugin. Install the plugin (http://downloads.chrispederick.com/work/web-developer/web-developer.xpi) into firefox browser and checked the information -> view document size.

What happens for the users that are loggin in?

The cache is not used when a user logs into the system and will be bypassed automaticly. The logged in user is automaticly detected and the cache is bypassed until the user logs out the system. You don't have to define the actions any longer since the latest cache release (from 21 juni 2008) automaticly detects when a user is logged on.

What are the performance gains of the cache component?

The performance gains are based on the generation time of the page when no cache is used. The generation time with no cache used is based on the number of online users and the number of third partie components that you are using. The more user and the more components you are using you will face perfomance issues in regards of page generation. When you are using the cache engine a performance gain of a factor 80-100 can easily be achieved for pages that are retrieved for 'public' pages (meaning that a page is request by a user that is not logged in). When a user logs into your frontend the cache is no longer used. See the example below of a debug output file of the cache component for a local Joomla installation (meaning no online visitors) and no third party components, only the basic Joomla installation with the cache component on:

URL request: /
Saving compressed cache file
Generated cache file: 460bac323d9baa1aafa3a85e7aa3e76355edbe01
File size: 6.2 Kb
Generation time:0.400991 sec.

URL request: /
Output compressed encoding: gzip
File **retrieved** from cache: 460bac323d9baa1aafa3a85e7aa3e76355edbe01
File size: 6.2 Kb
Generation time:0.0329899 sec.

URL request: /
Specific cache page for for HTTP user-agent filter: MSIE 6.0
User agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; InfoPath.2; .NET CLR 1.1.4322)
Saving compressed cache file
Generated cache file: f82f8f1cf140bf3b7346e5112e13c6b016aa0314
File size: 6.2 Kb
Generation time:0.4084651 sec.

URL request: /
Specific cache page for for HTTP user-agent filter: MSIE 6.0
User agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; InfoPath.2; .NET CLR 1.1.4322)
Output compressed encoding: gzip
File **retrieved** from cache: f82f8f1cf140bf3b7346e5112e13c6b016aa0314
File size: 6.2 Kb
Generation time:0.0405559 sec.

URL request: /index.php?option=com_content&view=article&id=19&Itemid=27
Output compressed encoding: gzip
File **retrieved** from cache: b2354c028bba592c78bc5aaca4444590e2583c42
File size: 3.9 Kb
Generation time:0.0328001 sec.

User is **logged in**: Administrator
URL request: /index.php?option=com_content&view=category&layout=blog&id=1&Itemid=50
Cache is **bypassed**
Generation time:0.57641 sec.

General these figures of the page generation time of Joomla are normally much higher since there are more components installed and more online visitors (not users that are logged in).

Can I use different style's for different browsers based on PHP logic?

Yes, you can! Within the configuration it's possible to define for which browser types a specific cached page must be generated for that type of browser. For example if you have special features for the internet explorer 6.0 browser you can define this within the configuration section of the cache that for MSIE 6.0 a seperate cache file must be generated.

What are the *"site"* effects of the component?

When you apply the component at your website make sure that you are aware of the following:

If you are using 'dynamicly changing content parts' that change on each page request this will not work when you apply the cache component. For example when you display the date and time, the number of visitors online or dynamicly rotating banners will not displayed with the correct values. The reason for this is that the page is retrieved from the cache repository which represents the state when it was generate and stored within the cache repository. If the page is retrieved from the cache be aware that no interaction with the databases is made. If you have performance issues the only solution is to remove these dynamic displays or keep the time to live (= how long a cache page is valid) as low as possible. Also with the use of so called Ajax modules and components the content can be updated as will and the page will be retrieved from the cache.

Compatibility with SEF Patch extended version

The accelerate component is fully compatibel with the SEF Patch extended version. To make both components work first install the SEF Patch extended version and secondly the accelerate component.

Compatibility with templates and components using cookies

The cache component also support cookies within the URL cache repository unique key. For example Joomlafish is using a cookie to set the language for display at the end user. In this case the URL is the same but the difference in display is made by the cookie. In this case the cookie can be included with the URL to generate a cache page specific for the value of the cookie in combination with the URL. Also templates make use of cookies, for example the RocketTheme templates are using cookies for the display settings (larger fonts, etc...). If you have questions about these cookies please check or ask it in our forum.

Installation of the Joomla accelerate component

Installation of the component is simple. You can install the component through the Joomla installer, but make sure that the **/index.php** and the **/administrator/index2.php** files of your webserver has the permissions to be overwritten by the patch file (set the permission on 666) and make also sure that your root directory of your website (the location where the index.php is) and the root of the administrator directory is also writeable.
In the installation process the orginal **index.php** file in your webserver root is copied to **jindex.php** and the **administrator/index2.php** is copied to **administrator/jindex2.php** then the cache component will install the cache index.php file and the administrator/index2.php file. These files will included the original index.php file and the administrator/index2.php file.

If you ***get a permission error*** at installation time you can install the component simply by your self with a FTP program. Therefor perform the following steps:

1. rename your **/index.php** file into **/jindex.php**
2. copy the **/administrator/components/com_jrecache/install_files/index.php** to **/index.php** file
3. rename your **/administrator/index2**.php file **/administrator/jindex2.php**
4. copy the **/administrator/components/com_jrecache/install_files/index2.php** to **/administrator/index2.php** file

After installation you will need to "enable" the cache through the configuration panel. When you enable the cache make sure that the 'default' settings are correct. For example the component uses the default location of your website configuration file, but if this is located on a different

location due to security reasons correct the path in the configuration setting.

ithin the Administration panel you can administrate the behauviour of the cache repository.

Configuration settings:

Within the component you can set the following configuration parameters:

Configuration parameter	Description / usage
Host configuration file location:	The location of your Joomla or Mambo configuration file setting. The component will assign the default
Enable cache:	With this paranmeter you can enable / disable the cache function. Default value when you install the c
Cache directory:	The location of where the cache files should be stored. Default is this the joo
Cache time to live:	The time that a cached page is vallid within the cache repository in seconds
Enable automatic garbage collection	Enables the automatic garbage collection. If enable the cache will be cleaned up periodicly according t
Garbage collection time schedule	Number of seconds that the automatic garbage collection should be initiated (if enabled). Garbage coll
Enable compressed cache storage:	This parameter allows you to store compressed (zipped) pages within the cache repository (recommend Note: make sure your website supports zip within PHP.
Use GZIP output page Compression:	When enabled this parameter will sent the cached content zipped to the requested browser if the client/b Notice that when you have a high visitors rate this could save you bandwith at your hosting provider (abo
Enable POSTS to be cached:	Default posts are disabled from caching. In this case you can prevent that form submissions are cache

Create unique cache pages for selected types:	When selected you can define unique HTTP user agents for which an uniqu
Browser that should be have the http cache page:	Enter the user agents (browser types) for which an unique cache page
Enable debug:	You can enable the debug function for the cache component. When enable
IP -address for debug:	If you enable the debug functionality you will need to enter the ip-address w
Location debug file:	The location of the debug file. Default the debug file will be stored within the
Drop http hack try requests:	

The 2.0 cache component release has the ability to drop so called hack requests to your website if the re

/products/free_products_for_joomla/?mosConfig_absolute_path=http://www.xxxx.com/shop/data/id.txt?

Cookies to support

Several templates make use of cookies to display the content on different way. For examle to change th
This also works on Joomfish which uses cookies for the display of the correct language within the fronte

Known cookies:

Joomfish:
mbfcookie[lang]

Template based cookies:
Rockettheme:
c-colorstyle
c-fontstyle
c-widthstyle

(notice that rocket theme sometimes uses other cookie names)

The best way to check which cookies are used is to use the firefox browser with the cookie viewer extension installed. With the extension you can s

Disable cache if the following cookie is used:	You can disable the cache if certain cookies are used by adding the cookie
Set the bypass cache cookie:	This setting is used to disable the cache for a specific visitor when a specif
Clean the bypass cache cookie:	This setting is used to enable the cache again for a specific user when a spe

Enter URL parts that may not be cached: With this field you can enter part of URL's that may not be included within
Disable for cache if the page content contains

You can disable certain pages from the cache if the content of the pages contains specific "content". Sta

To make this simple: just edited your Joomla language file and find the text for the page not found error.

(avaible from release 1.0b, 20 april 2007)

Enable automatic cache deletion for Joomla backend

When enabled the cache will automaticly be cleared if specific actions in the backend are executed. For

Notice: the automatic cache deletion is default disabled.

(avaible from release 1.0b, 20 april 2007)

Removes the cache if the following joomla admin actions are performed

In this field you can apply the parameters / actions on which the cache must be cleared when a specific
option=[joomla option] and task=[joomla task]

Each line represents a Joomla action which can exists from multiple parameters and to clear the cache

option=com_content|task=save
option=com_content|task =remove

In this example if the above "rule" is met the cache will be deleted when a content item is saved and the

With the installation of the component the most common Joomla backend action are default present in th

(avaible from release 1.0b, 20 april 2007)

Enable the automatic cache deletion for the frontend
The same kind of parameter as the backend admin. When enabled the cache will automaticly be cleare
Removes the cache if the following joomla frontend actions are performed

Same as the	*Removes the cache if the following joomla admin actions are performed*	

(avaible from release 1.0b, 20 april 2007)

Cleaning up the cache repository

You can clean up the cache repository through the administration panel->Clean up the Cache repository. When you select this option you can choose "to delete all files within the cache repository" and "delete the outdated and invallid cache files"
within the cache repository. With the first option all the files and the records within the SQL database are removed either the cache entry is vallid or not. The second option only deletes cache entries that are no longer vallid (e.q. purpose is to make diskspace available from cache files that are no longer used).

From version 1.0b and later it's also possible to cleanup the cache when specific actions are performed. Within the configuration you can apply "rules" for the backend and for the frontend if specific actions are performed.
For example for the backend: if a Joomla content item is saved, deleted or published the cache can be cleared automaticly so that the frontend is always up-to-date. When you apply this rule you can set the TTL of the cache to 0 since every updated in the backend (or frontend) will clear the cache automaticly (if the rule is defined in the cache config).

Overview of the cache repository

Within this option you can get an overview of the records that are stored within the cache repository. Each file that is located within the cache repository is stored within a SQL table which enables you to manage specific pages within your cache repository.

Within this view you can add a new "cache" entry and don't allow the specific URL to be cached.

You can also edited an existing URL / cache entry and disables the caching for this spefic URL.

You can also remove a specific URL from the cache repository if needed.

Notice: disabled URL's are not removed when you clean up the cache repository within the administration pannel.

Reinstall the index.php file

When the first installation was not succesfull you can reinstall the needed index.php with this option. This is ussualy the case if the index.php could not be replaced at the installation time of the component.

Restore the original index.php file

At installation time a backup of the orginal index.php file is made. With this option you can restore the "orginal" index.php when needed.

Deinstallation of the component.

When you deinstall the component the orginal files are restored.