

ClickTale PHP Integration Module

The module is PHP5 code, provided with full sources, that allows better integration of a PHP site with ClickTale by enabling accurate caching of the HTML that is sent to the visitor in ClickTale recorded pageviews.

License

The module is subject to a permissive license for ClickTale users. Please see license.txt in the archive for more information.

Specific integration modules - Magento

- For **Magento** integration, please view the installation guide here: http://docs.google.com/Doc?docid=dg68zm2r_2338fbvxzg8&hl=en. (keep this doc open)

Installation Guide

1. The module itself can be found here: http://www.clicktale.com/Integration/ClickTalePHPIntegrationModule_latest.zip
2. Remove the existing ClickTale code and/or disable other integration methods (for example, PHP integration *code* - which supports session state only)
3. Extract the archive to a directory named "ClickTale" in your website. Do not to create a directory named "ClickTalePHPIntegrationModule_X.X" in it. Simply place the files in ".../ClickTale/".
4. Generate the ClickTale tracking code you want to use in your pages (or locate your already generated code)
5. Copy and paste that code (from previous step) into the appropriate parts in ClickTale/ClickTaleScripts.xml file
6. Add

```
ClickTaleFetchFrom="%FetchFromUrl%";
```

just before the line

```
if(typeof ClickTale=='function') ClickTale(#####,###, ###)
```

7. Every page you wish to be recorded should contain the following PHP code:

```
<?php  
require_once("[physical_path_to_clicktale_dir_here]\ClickTale\ClickTaleTop.php");  
?>
```

at the top (before any content is sent, including doctype and html tags)
and

```
<?php  
require_once("[physical_path_to_clicktale_dir_here]\ClickTale\ClickTaleBottom.php");  
?>
```

at the bottom (after all content has been sent)
Do not forget to replace [physical_path_to_clicktale_dir_here] with the relevant folder.

For example, with Joomla you can try adding these lines to the top and bottom of the file index.php (in joomla's root), this will add recording to all pages.

Optionally: visit <http://forum.clicktale.net/viewtopic.php?f=2&t=84> to disable or enable yourself from being recorded. Apply this to everyone who is working with you on the site otherwise they will spend your recording quota, or enable this for testing to make sure all is working correctly.

Troubleshooting

After the files are copied you can navigate your browser to <http://yoursite/ClickTale/Install/index.php> to view some helpful information regarding the caching and other configurations. You can delete the Install folder once you don't need the troubleshooting/debugging assistance it provides. Deleting it prevents some of your configuration setting from being publicly available to others.

Configuring the caching provider

The module caches the content of the pages so it can later provide the content to ClickTale servers for processing. This caching requires some persistent storage and so different caching providers are supported.

The default caching provider is the file system ("cache" folder), it is slower than other methods.

It is recommended to use the MySQLMemory provider. To do so you will need a valid connection to a MySQL database.

There is also the APC module which requires installation. More details about APC can be found here: <http://php.net/apc>.

If you find it difficult to install, we can guide you through the process.

After installing APC or MySQL, you can enable its provider by editing config.php file, and changing:

```
$config['CacheProvider'] = "FileSystem";  
to  
$config['CacheProvider'] = "MySQLMemory";  
or  
$config['CacheProvider'] = "APC";
```

Note: The file-system provider will need write access rights to the "ClickTale/Cache" folder. On the other hand, APC and MySQLMemory will use memory for storage.

Caching provider settings

FileSystem:

- `$config['DeleteAfterPull'] = (true / false)`
If true, the cached page is deleted after clicktale fetches it from your site
- `$config['CacheLocation'] = (path to directory)`
Where the cache files should be saved. This directory should be writable (you can use the `Install/index.php` to check write permissions)
- `$config['MaxFolderSize'] = (integer)`
The size of the cache dir in MegaBytes. If the cache gets larger then that, stale caches are removed

MySQLMemory:

- `$config['DeleteAfterPull'] = (true / false)`
See FileSystem
- `$config['CacheLocation'] = (mysql uri)`
url of the form `http://<db username>:<db password>@<host of mysql>:<port of mysql>/<database to use>.<memory table name to use>`
- `$config['MaxCachedSeconds'] = (integer)`
Will only keep pages in cache for up to `MaxCachedSeconds` seconds. This will make sure that your cache storage will only take a small amount of memory.

APC:

- `$config['DeleteAfterPull'] = (true / false)`
See FileSystem
- `$config['MaxCachedSeconds'] = (integer)`
See MySQLMemory

Use With "Expires" Header And Enable Reuse Of Cached Pages

Some websites use "cache-control" and "expires" headers to cache pages on the client. This improves performance but may be problematic when used with our module. Pages are removed from the module's cache right after they are being accessed (for security and performance reasons). So, if a visitor browses a page more than once without refreshing the content from the server (usually a result of using the back button), **ClickTaleCache.ashx** will be called several times with the same token. This will cause a cache miss for any request beyond the first one.

To overcome this problem, it is possible to use the `DeleteAfterPull="false"` parameter in the configuration. This will disable the removal of cached pages when **ClickTaleCache.ashx** is called. Cached data will be removed after `MaxCachedPages="???"` new pages are cached, so you might want to extend this parameter as well to allow sufficient traffic between the first pageview and the next (duplicate pageview).

Additionally, you can change your HTTP cache control to "no-cache" which will might be more correct for your application logic in the first place.

Currently, this feature is not available when caching to databases.

Q&A

Q: Are cached pages protected from access by third parties?

A: Yes, several layers of protection are in place. Only certain IPs are allowed to request the cached pages (IPs of ClickTale servers) and only processes which already have access to the page have the secret token required to request the cached content.

Q: Is it possible to inject the script in other places rather than after/before body tags?

A: Yes. By default the top script is injected after `<body>` and the bottom before `</body>`, but this can be changed by adding a `InsertAfter` attribute to the `script[name="Top"]` element or adding a `InsertBefore` attribute to the `script[name="Bottom"]` element, both are regular expressions.

Q: Some of my pages already have the ClickTale script, I do not want the script appear twice.

A: You should either remove the script from those pages and let the module handle the insertion or you should use `DoNotReplaceCondition`. See the code step 2 for an example. The default script in Step 2 is already configured to prevent double inclusion.

Q: After installing the module, I tried to watch a recording but I got the following notice instead: "Request from an unauthorized IP." . What should I do?

A: This could be a misconfiguration, a change in our IP addresses or a hacking attempt. Please contact us so we can investigate this further.

Q: I have installed the module but I don't see the tracking code in the source of the page. What is wrong?

A: The module will only inject the code for visitors who are classified as "to-record" (have WRUID cookie with non-zero value) or for those who are not classified (no WRUID cookie). Visitors who are classified as "not-to-record" will get no code.

Q: How can I disable the caching and leave only the script injection?

A: Do note that disabling caching defeats the main purpose of the module and correct page content might not be recorded. The best practice for this case is to put the static tracking code in the website's template. If you still want to use the module for tracking code injection but not for content caching you can add `DisableCache="True"` to the `ClickTale.IntegrationModuleSettings` element in the `web.config` file. When setting `DisableCache` to `True`, you need to remove the `ClickTaleFetchFrom` line from the `ClickTaleScripts.xml` file or else nothing will be cached.

Reference

ASP.NET module documentation (good read as some configuration schema is shared) - http://docs.google.com/Doc?docid=dgkjr2br_3gkpr9&hl=en