Table of Contents

ZonePlugin **1**
Syntax 2
ADDTOZONE 2
RENDERZONE 2
Perl API 3
Installation Instructions 3

ZonePlugin

Gather content of a page in named zones while rendering it

The primary purpose of ZonePlugin is to streamline the anatomy of a HTML page in a way to allow today's browsers to process it more efficiently. Page loading time has been reported to decrease significantly when all JavaScript files are removed from the HEAD element and appended to the end of the BODY element of a page. That's because the browser will stop processing a page as soon as a JavaScript file is found in the linear order of the page. The browser will only proceed after this file has been downloaded and excecuted. Notably other content like CSS files and image material will *not* be downloaded in parallel as long as the JavaScript interpreter hasn't finished.

Currently, Foswiki uses ADDTOHEAD to place additional CSS and JS files into the HEAD element. It does *not* reorder those files in any way other than specified by the explicit `requires` argument to the macro. Further more, it is only able to add stuff to one specific location of a HTML page, the HEAD element.

By using ADDTOZONE CSS and JS material can be added to the resulting page incrementally while the core engine parses templates and wiki applications. ADDTOZONE's first parameter is the name of the zone to add stuff to. There are two special zones: `body` and `head`. All CSS should be added to the `head` zone while all JS goes into the `body` zone.

The actual location of a zone is specified with an explicit RENDERZONE macro. This macro expands to the content of all material that has been posted to the named zone. Note, that this happens at the very end of the rendering pipeline of Foswiki. That means RENDERZONE is not processed as a normal macro obeying the evaluation order of the TML parser. Instead, all calls to ADDTOZONE are processed by the TML parser first after which all zones are expanded.

If RENDERZONE{head} and RENDERZONE{body} aren't found in the final page explicitly, they are expanded at the appropriate position, that is at `</head>` and `</body>` respectively.

The features of this plugin have been as a core feature for Foswiki to replace the standard ADDTOHEAD with the more generic ADDTOZONE tag. This plugin allows authors of extensions and wiki applications to make use of this advanced feature in a backwards-compatible way. As soon as the ADDTOZONE macro has been released as part of newer Foswiki versions, this plugin won't be of use anymore.

WARNING: Using this plugin can potentially break your installation. If you experience occasional JavaScript errors, switch off `OptimizePageLayout` in configure. This mode will render the HTML page in a non-optimized way similar to how the standard `ADDTOHEAD` mechanism does.

In any case is it recommended to use `ADDTOHEAD` or `ADDTOZONE` to properly add this code to the page. It is *not* recommended to add JavaScript code otherwise.

If you rely on having JavaScript added to the page *without* using `ADDTOHEAD` or `ADDTOZONE` then you have to disable `OptimizePageLayout` mode most probably. In this case the HTML page cannot be optimized.

ZonePlugin will try to detect whether any use of ADDTOHEAD contains `text/javascript` and automatically move its content to the `body` zone while adding anything else to the `head` zone.

Syntax

ADDTOZONE

```
%ADDTOZONE{
  "zone"
  ...
}%
```

- zone: optional, comma-separated list of zones the content should be added to, special zones are `body` and `head`, defaults to `head`
- tag: optional, identificator for the material to be added, to be used in `requires`
- requires: comma-separated list of tags that must occur before the current material, constraints the order of all material in a zone
- text: optional, text to be added to the named zone, mutually exclusive with `topic`
- topic: optional, full qualified web-topic name that contains the text to be added, mutually exclusive with `text`
- section: optional, section of the `topic` to be added, defaults to the default section between STARTINCLUDE and STOPINCLUDE

Note, that using `topic` and `section` is actually a short form of

```
%ADDTOZONE{
   "myzone"
   text="$percntINCLUDE{\"topic\" section=\"section\" warn=\"off\"}$percnt"
}%
```

See also VarRENDERZONE

RENDERZONE

```
%RENDERZONE{"zone" ...}%
```

- zone: required, name of the zone to be expanded
- header: optional, prefix format string
- format: optional, format string for each item added to the zone, defaults to `$item`, any standard escapes like `$percnt`, `$dollar` etc can be used to delay evaluation order of the format parameter
- chomp: removed leading and trailing whitespace in formatted items, useful to pretty-print them one per line
- footer: optional, suffix format string
- separator: optional, put between each item of a zone

Note, that you can create as many zones as you like. The plugin does not restrict you to use it only for `body` and `head`. Interesting use cases in wiki applications:

- create a `sidebar` zone to add widgets
- create a `toolbar` zone to add buttons icons

See also VarADDTOZONE

Perl API

This plugin patches the `Foswiki::Func` API for backwards compatibility.

- New `Foswiki::Func::addToZone($zone, $tag, $text, $requires)`
- Replaces `Foswiki::Func::addToHEAD() to use the =head` namespace of this plugin instead of `Foswiki::_HTMLHEAD`

The latter will try to detect `text/javascrtipt` and move content into the `body` zone. Otherwise, it will be added to the `head`.

Any use of `ADDTOHEAD` or `Foswiki::Func::addToHEAD()` will emit a warning to the log files (must be switched on with the `Warnings` flag in configure). This can be used to hunt down suboptimal use of these APIs.

Installation Instructions

You do not need to install anything in the browser to use this extension. The following instructions are for the administrator who installs the extension on the server.

Open configure, and open the "Extensions" section. Use "Find More Extensions" to get a list of available extensions. Select "Install".

If you have any problems, or if the extension isn't available in `configure`, then you can still install manually from the command-line. See http://foswiki.org/Support/ManuallyInstallingExtensions for more help.

Author(s):	Michael Daum
Copyright:	© 2010 Michael Daum http://michaelaumconsulting.com
License:	GPL (Gnu General Public License)
Release:	2.1
Version:	6966 (2010-03-28)
Change History:	
28 Mar 2010	fix problem where Foswiki 1.0.x installations would fail with "Undefined subroutine &Foswiki::Func::addToZone"
26 Mar 2010	suppressing plugin initialisation on Foswiki engines >= 1.1; renamed BackwardsCompatible switch to OptimizePageLayout (defaults to off)
19 Feb 2010	added `{BackwardsCompatible}` switch
15 Feb 2010	be more careful applying the monkey-patch to the Func API; parsing RENDERZONE properly but finally inserting the zone at the end of the rendering pipeline
12 Feb 2010	initial release
Dependencies:	None
Home page:	Foswiki:Extensions/ZonePlugin
Support:	Foswiki:Support/ZonePlugin

Topic revision: r3 - 15 Feb 2010 - 10:11:15 - ProjectContributor

- . System

- Log In

- **Toolbox**
- Users
- Groups
- Index
- Search
- Changes
- Notifications
- RSS Feed
- Statistics
- Preferences

- **User Reference**
- BeginnersStartHere
- TextFormattingRules
- Macros
- FormattedSearch
- QuerySearch
- DocumentGraphics
- SkinBrowser
- InstalledPlugins

- **Admin Maintenance**
- Reference Manual
- AdminToolsCategory
- InterWikis
- ManagingWebs
- SiteTools
- DefaultPreferences
- WebPreferences

- **Categories**
- Admin Documentation
- Admin Tools
- Developer Doc
- User Documentation
- User Tools

- **Webs**
- Public
- System

Copyright © by the contributing authors. All material on this site is the property of the contributing authors.
Ideas, requests, problems regarding Wiki? Send feedback