Advanced Access Manager

ConfigPress Tutorial

V 2.0

Content

Introduction

ConfigPress has been introduced couple versions before and the main goal is to configure the Advanced Access Manager (AAM) behavior in more specific way. The way you need it!

Think about ConfigPress as a Settings page for AAM but instead of clicking a lot of buttons and checkboxes, you type the configurations. And yes, this may make the configuration process more complicated, but also way more flexible.

The ConfigPress syntax is based on standard Windows INI file. So my suggestion will be to spend 10-15 minutes and understand how it works.

This tutorial is divided on four main sections:

AAM Configurations. Explains all available settings for configuring the plugin's core behavior. It is worth reading if you are advanced WordPress user. As well as it has some useful set of configurations for Multisite support.

Backend Configurations. List of all possible configurations to specify the way how AAM should handle access control for WordPress Backend.

Frontend Configurations. Very similar as Backend Configuration just for Frontend part.

Advanced Configurations. Explains how to use custom functions to extend the AAM functionality.

AAM Configurations

Overview

This section introduces you to the core AAM configurations. I’m suggesting to be careful with these configurations in fact it may cause incorrect WordPress behavior or reduce website performance.

All configurations below should be included in INI section **[aam].** Please make sure that you understand what does it mean INI section before your proceed with tutorial.

Caching

To speed up the AAM execution you can utilize the internal caching mechanism. This is very important when you have large scale website and a lot of visits.

```
[aam]

caching = “true”
```

Admin Menu Restriction

The access to Admin Areas can be controlled in Menu Tab by checking or unchecking the menus and submenus. Each admin menu has it’s unique admin URL, like Pages menu has *edit.php?post_type=page* or AAM Access Control page *admin.php?page=aam.* So each time user accesses any of these URLs the AAM checks if this area is restricted. Currently WordPress does not allow us to have more elegant way to control access rather than checks the URL.

It works perfectly if user does not modify URL. But as soon as URL manually is changed like *admin.php?page=aam* becomes *aam.php?page=aam&test=1*, this is considered as different page and user can get access to Access Control page.

To prevent user from this you can utilize the ConfigPress setting that will deny access to any undefined or modified URL.

```
[aam]

menu.undefined = “deny”
```

Extension Management

There is a way to turn *on* or *off* the Existing Extensions. By default, AAM is scanning **extension** folder inside *wp-content/plugins/advanced-access-manager* folder and load all extensions. To control Extension load you can utilize ***extension*** parameter and Extension’s folder name. As example you can turn off the AAM Activity Log by entering next configuration:

```
[aam]

extension.AAM_Activity_Log = “off”
```

Backend Configurations

Overview

This part of tutorial shows you how to configure the backend access behavior.

Please be sure that all settings below are included inside the [backend] section.

All configurations below should be included in INI section **[backend].** Please make sure that you understand what does it mean INI section before your proceed with tutorial.

Deny Access Behavior

Sometimes it is important to define the default procedure for request or action which is not authorized. Let's say you want to redirect user to the WordPress Dashboard if he is not allowed to see the list of posts.

In this case you may to configure the denial procedure with next lines of configurations:

```
[backend]

access.deny.redirect = "http://yourdomain/wp-admin"
```

In case you want to display just message you may use next configurations:

```
[backend]

access.deny.message = "Your message here"
```

Frontend Configurations

Overview

Frontend access control is pretty much the same as Backend. So I’m not going to repeat myself the only difference is the list of possible access configurations for taxonomies and posts.

All configurations below should be included in INI section **[frontend].** Please make sure that you understand what does it mean INI section before your proceed with tutorial.