

WebsitePanel IIS Modules Installation

Version 1.0

Table of Contents

- Introduction 1
- System Requirements 2
- Registering Module 2
- Installing on IIS 7 2
 - For all Web Sites 2
 - For a Single Web Site 3
 - Web Site Settings 4
- Installing on IIS 6 5
 - Enable Wildcard Mapping 7
 - Single Site 8
 - All Web Sites 8
 - Web Site Settings 9
- Configuring in WebsitePanel 10
- Supported Apache Modules and Directives 10
 - .htaccess 10
 - AuthName 11
 - AuthType 11
 - AuthUserFile 11
 - AuthGroupFile 12
 - Require 12

Introduction

WebsitePanel IIS Modules (“module” below in the text) enables secure folders on IIS web sites. It emulates Apache "**mod_auth_basic**", "**mod_authz_user**" and "**mod_authz_groupfile**" modules and works with native .htaccess and .htpasswd files. The module could be used on both IIS 6 and IIS 7.

System Requirements

Operating System

The module supports both IIS 6 and IIS 7 on Windows Server 2003 and Windows Server 2008 respectively. All Windows Server SKUs and both 32-bit and 64-bit operating system editions are supported.

.NET Framework

The module is designed as managed IIS module and it requires Microsoft .NET Framework 2.0 SP1 installed on the server. The installation of .NET framework is only required on Windows Server 2003. Windows Server 2008 and Windows Vista go with pre-installed .NET Framework.

Registering Module

To function properly the module must be installed to Global Assembly Cache (GAC).

If you installed a module with MSI installer it's been already added into GAC.

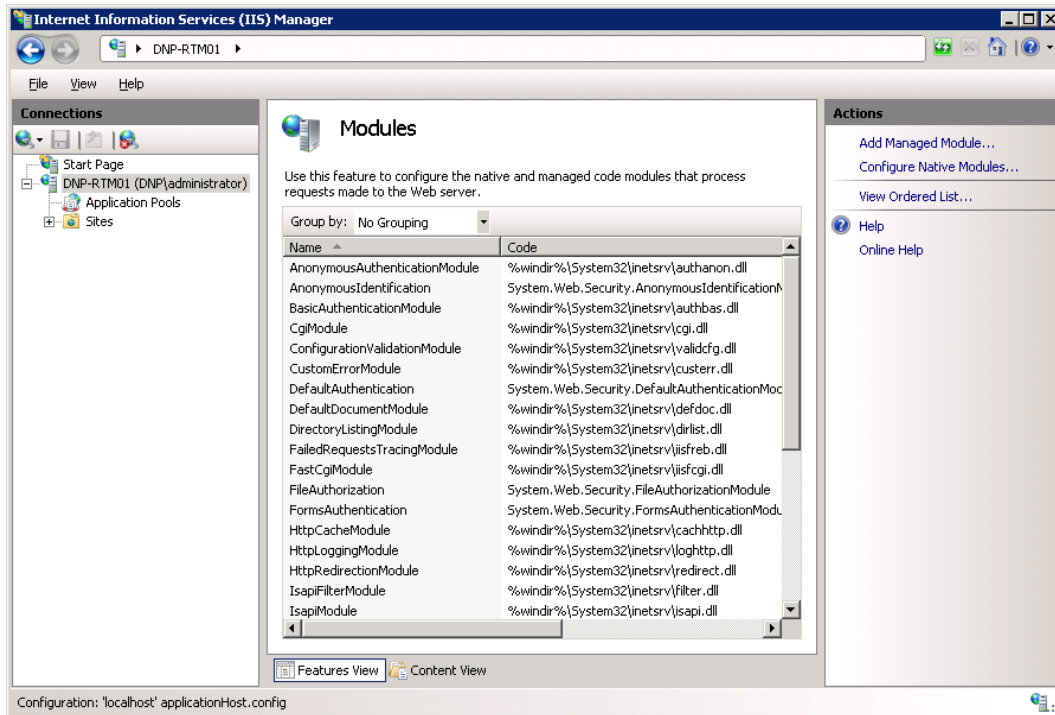
If you are doing manual installation you should use gacutil.exe tool for installing assembly into GAC. Gacutil.exe is a part of Microsoft .NET Framework SDK.

Alternative way to install assembly into GAC is to open c:\Windows\assembly folder in Windows Explorer and then drag assembly file into it.

Installing on IIS 7

For all Web Sites

1. Open Internet Information Services (IIS) Manager MMC snap-in and then open "Modules" snap-in on <computer> level:



2. Click “Add Managed Module...” link on “Actions” pane;
3. Expand “Type” list and select “WebsitePanel.IIsModules.SecureFolders, WebsitePanel.IIsModules, ...” module.
Type “SecureFoldersModule” to the “Name” field.
Leave “Invoke only for requests...” checkbox unchecked.
4. Click “OK” button.

For a Single Web Site

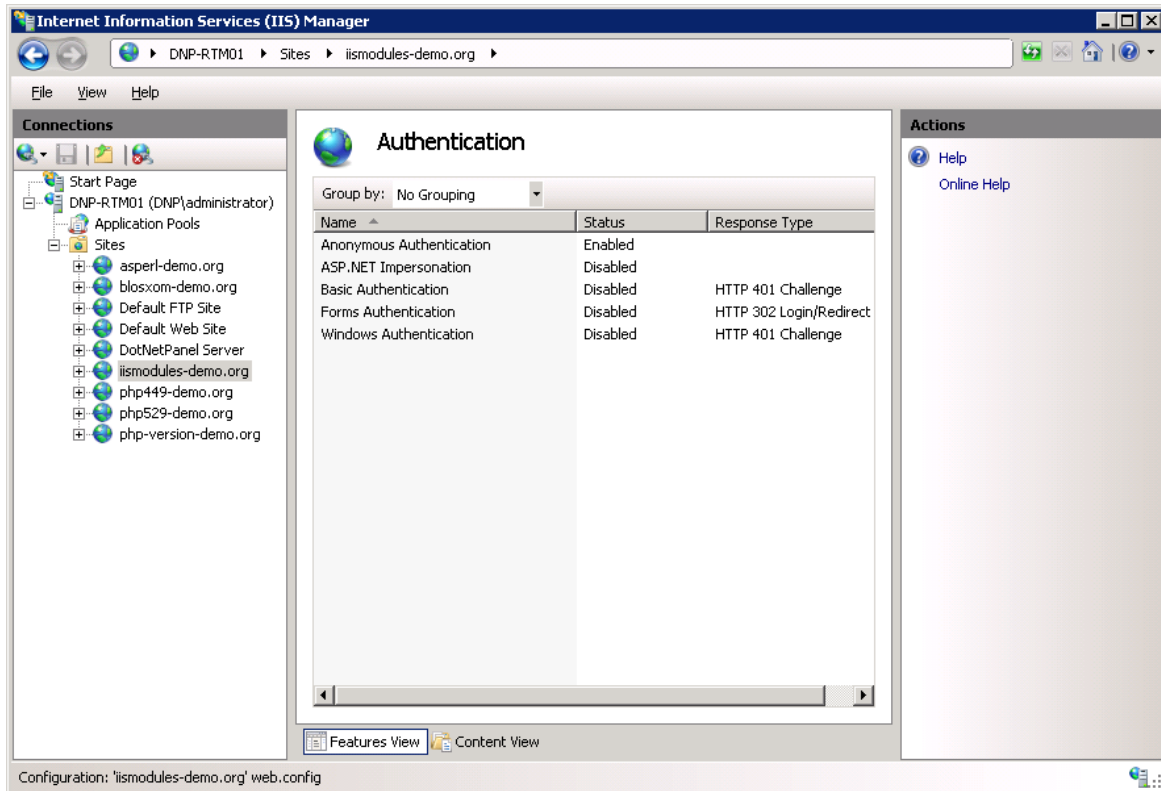
1. To install module for specific web site only click web site node in the left navigation tree and then open web site “Modules” window.
2. Click “Add Managed Module...” link on “Actions” pane;
3. Expand “Type” list and select “WebsitePanel.IIsModules.SecureFolders, WebsitePanel.IIsModules, ...” module.
Type “SecureFoldersModule” to the “Name” field.
Leave “Invoke only for requests...” checkbox unchecked.
4. Click “OK” button.

Web Site Settings

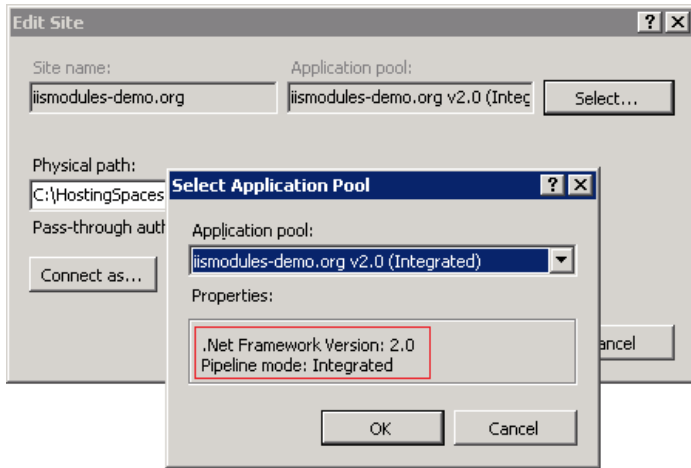
To use module on specific web site two conditions must be met:

1. Windows Authentication is disabled.
2. Web site's application pool is working in "Integrated" mode.

To disable Windows Authentication in IIS Manager click web site node in the left navigation tree and click "Authentication" icon. Make sure "Windows Authentication" module is disable or does not exist (if was not installed as a "Web Server" role feature):

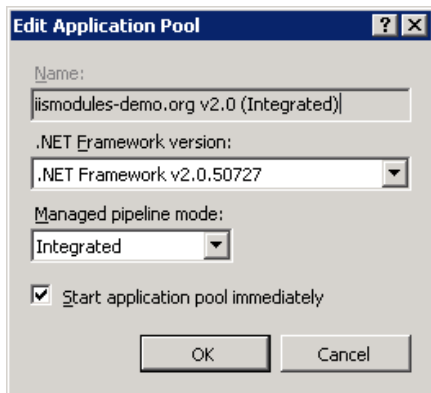


To check/change web site application pool settings click web site node in the left navigation tree and then click "Basic Settings..." link on the right "Actions" pane:



Make sure the pool has “Integrated” pipeline mode.

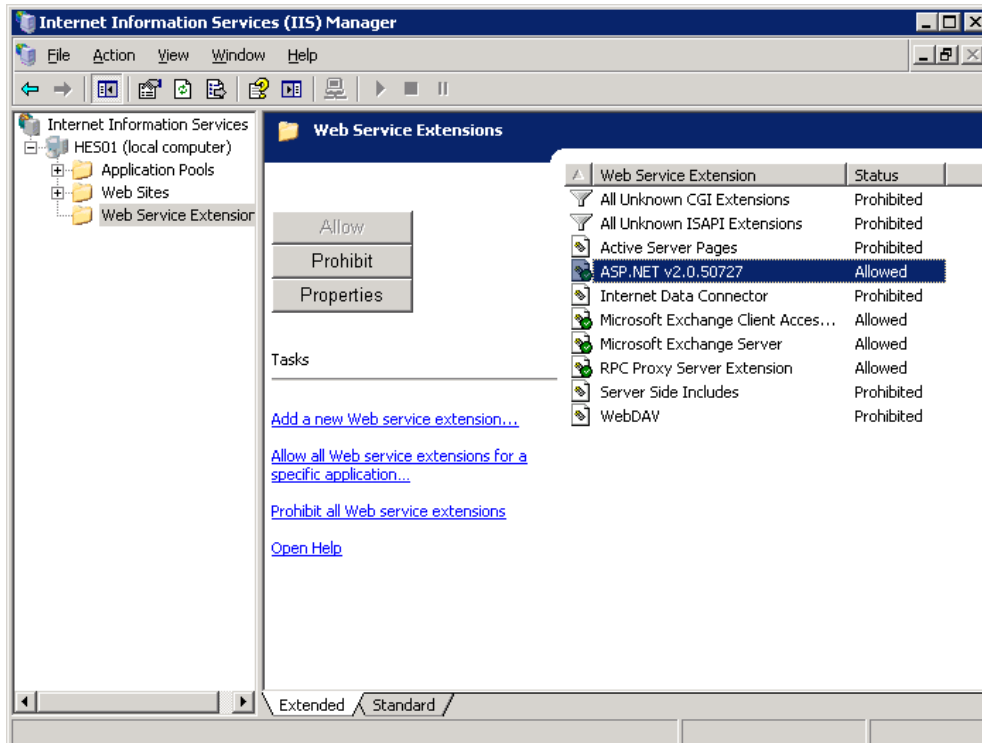
To change pool settings click “Application Pools” node in the left navigation tree and then double-click web site pool to open its properties window:



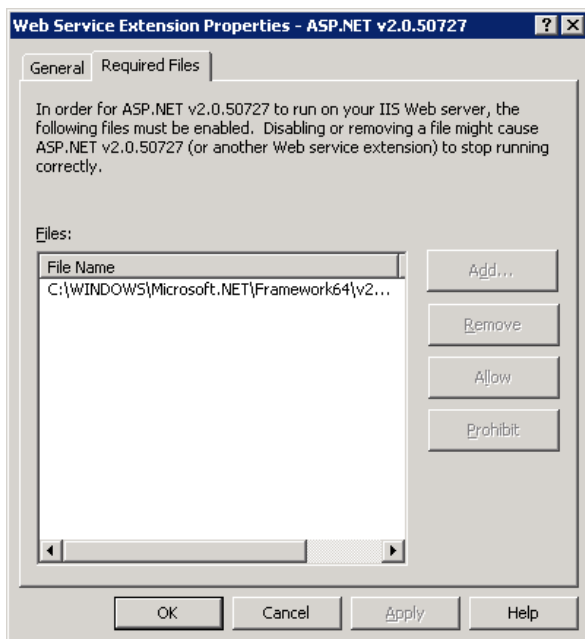
Installing on IIS 6

If you are installing module on Windows Server 2003 x64 then determine which version of .NET framework is being used in IIS.

Open IIS Manager and click “Extensions” node in the left navigation tree:



Then double-click “ASP.NET v2.0.50727” extension to see its properties:

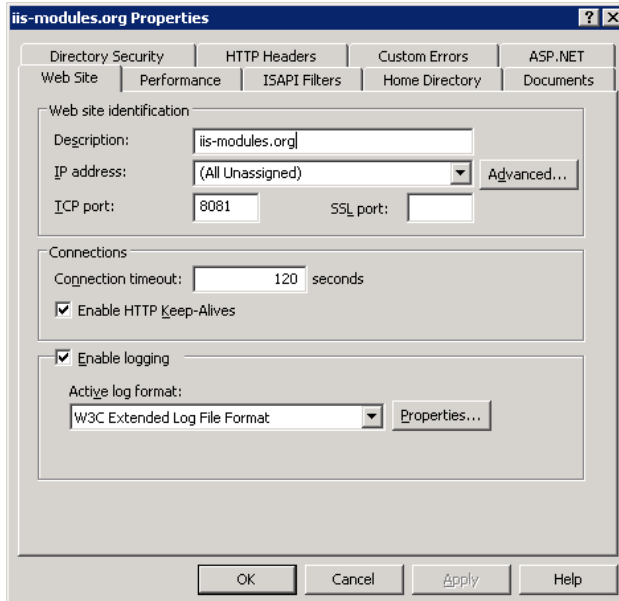


By checking a path of extension required files you could determine the bitness of .NET Framework. If there is “Framework64” in the path IIS uses 64-bit version of .NET Framework 2.0. If there is “Framework” in the path then it is 32-bit extension.

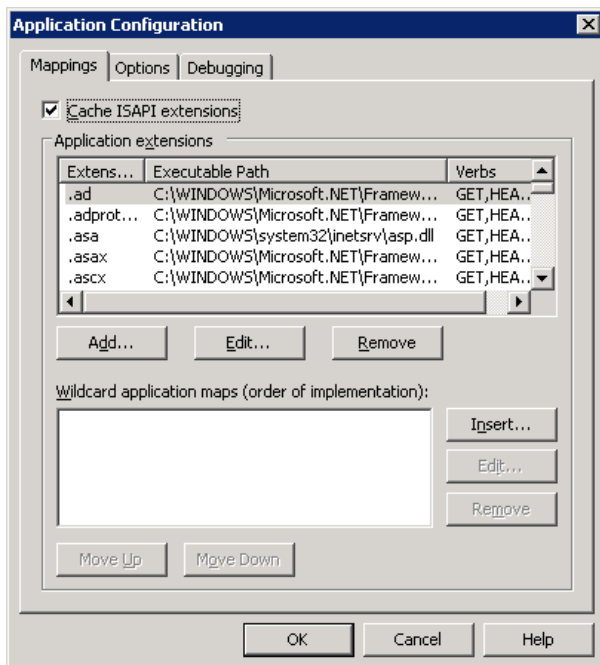
Enable Wildcard Mapping

You could enable wildcard mapping to ASP.NET ISAPI either for all web site or just for particular web site.

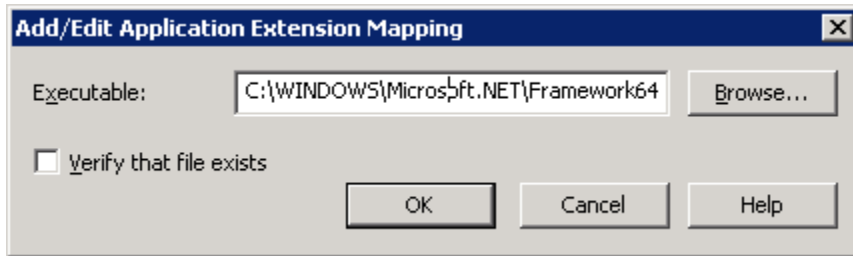
To enable wildcard mapping for all web sites right-click “Web Sites” node in IIS Manager and select “Properties”. To enable it for particular web site right-click web site node and click “Properties”. You will be presented with the following dialog:



Click “Home Directory” tab and then “Configuration...” button:



To add new “Wildcard application map” click “Insert...” button:



Enter path to ASP.NET ISAPI into “Executable” field:

For 64-bit framework: C:\WINDOWS\Microsoft.NET\Framework64\v2.0.50727\aspnet_isapi.dll

For 32-bit framework: C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_isapi.dll

Click “OK” button to save changes.

Single Site

Open global web.config in the following location:

C:\WINDOWS\Microsoft.NET\Framework64\v2.0.50727\CONFIG\web.config

(Change “Framework64” to “Framework” for 32-bit ISAPI).

Add the following code inside “configuration” element:

```
<configuration>
  ...
  <location path="Web-Site-Name">
    <system.web>
      <httpModules>
        <add name="SecureFoldersModule"
type="WebsitePanel.IIsModules.SecureFolders, WebsitePanel.IIsModules,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=37f9c58a0aa32ff0"/>
      </httpModules>
    </system.web>
  </location>
  ...
```

Change Web-Site-Name to the name of the web site (exactly how it appears in the left navigation tree of IIS Manager).

Change version value to the currently installed Module version. You could check assembly version by opening its properties in Windows Explorer or in c:\Windows\assembly folder.

All Web Sites

Open global web.config in the following location:

C:\WINDOWS\Microsoft.NET\Framework64\v2.0.50727\CONFIG\web.config
(Change "Framework64" to "Framework" for 32-bit ISAPI).

Add the following code inside "configuration/system.web/httpModules" element:

```
<configuration>
  ...
  <system.web>
    <httpModules>
      ...
      <add name="SecureFoldersModule"
type="WebsitePanel.IIsModules.SecureFolders, WebsitePanel.IIsModules,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=37f9c58a0aa32ff0"/>
    </httpModules>
  </system.web>
  ...
</configuration>
```

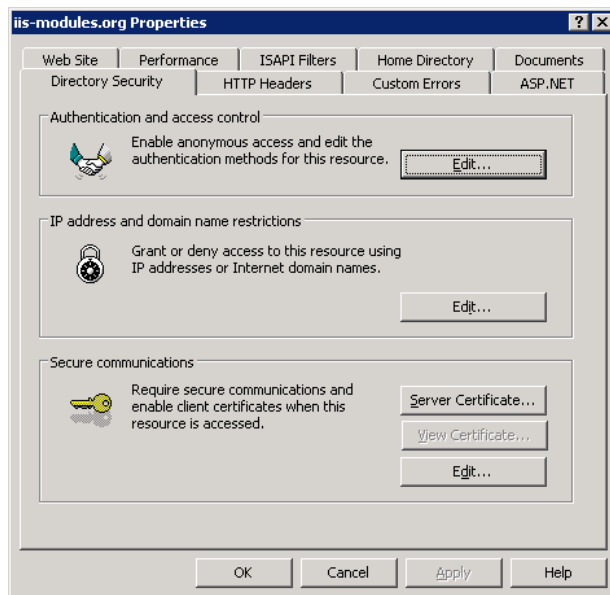
Change version value to the currently installed Module version. You could check assembly version by opening its properties in Windows Explorer or in c:\Windows\assembly folder.

Web Site Settings

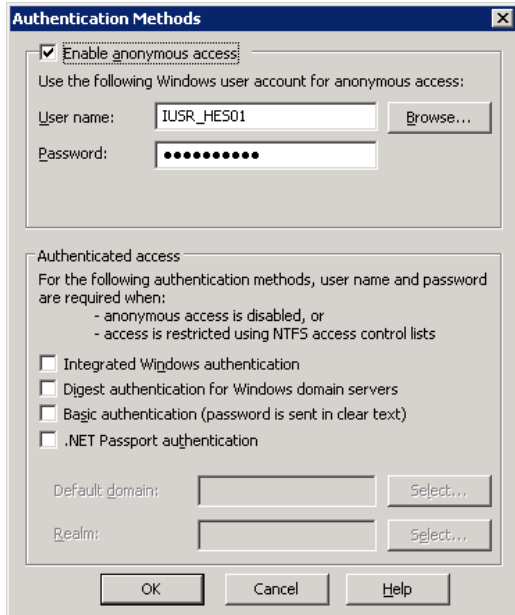
To use module on specific web site the following conditions must be met:

1. Windows Authentication is disabled.

To disable Windows Authentication for IIS 6 web site open its properties and then click "Directory Security" tab:



Click "Edit..." button in "Authentication and access control":



Uncheck “Integrated Windows authentication” checkbox and click “OK” button to save changes.

Configuring in WebsitePanel

To enable Secure Folders module in WebsitePanel open IIS 7 web service properties screen (Configuration -> Servers -> click service properties).

In “Secure Folders” section enter the following value in “Module Assembly” field:

```
WebsitePanel.IIsModules.SecureFolders, WebsitePanel.IIsModules,  
Version=1.0.0.0, Culture=neutral, PublicKeyToken=37f9c58a0aa32ff0
```

Change version value to the currently installed Module version. You could check assembly version by opening its properties in Windows Explorer or in c:\Windows\assembly folder.

Supported Apache Modules and Directives

.htaccess

Module is looking up for .htaccess file in the root directory of the current request. If .htaccess file does not exist in the current directory module is trying to find it in directories above the current one up to web site root directory. For example, if you have the following site structure:

```
/wwwroot  
  /secret  
    /john  
      /myfiles
```

and request /secret/john/myfiles folder in the browser the module will check four directories:

```
\wwwroot\secret\john\myfiles
\wwwroot\secret\john
\wwwroot\secret\
\wwwroot
```

If .htaccess file is not found module does nothing and just returns control back to IIS pipeline (pass-through mode).

AuthName

Directive specifies the name of secure folder. Directive format:

```
AuthName folder_name
```

Folder_name is display name of security folder that will be shown in login dialog of web browser, for example:

```
AuthName Documents
```

If folder_name contains spaces its value must be quoted:

```
AuthName "My secret files"
```

AuthName is mandatory directive.

AuthType

Directive specifies authentication type. Directive format:

```
AuthType {Basic|Digest}
```

Currently, only Basic authentication is supported, for example:

```
AuthType Basic
```

This is optional directive.

AuthUserFile

Directive specifies the path to users-passwords file. Directive format:

```
AuthUserFile <path>
```

where <path> is an absolute path to .htpasswd file, for example:

```
AuthUserFile c:\HostingSpaces\user1\domain.com\wwwroot\.htpasswd
```

<path> could be relative too. This case it must be relative to web site root folder, for example:

```
AuthUserFile \secret_folder\.htpasswd
```

Each line of .htpasswd file represents user-password pair delimited by colon, password is encrypted with Unix crypt() function, for example:

```
john:jz/jJoZNfNmQq  
user:SHLAvMU4ftW0U
```

This is mandatory directive.

AuthGroupFile

Directive specifies the path to user groups file. Directive format:

```
AuthGroupFile <path>
```

where <path> is an absolute path to .htgroup file, for example:

```
AuthGroupFile c:\HostingSpaces\user1\domain.com\wwwroot\.htgroup
```

<path> could be relative too. This case it must be relative to web site root folder, for example:

```
AuthGroupFile \secret_folder\.htgroup
```

Each line of .htgroup file represents group name and its members delimited by colon, group members separated with spaces, for example:

```
Accounting: user1 marry  
Admins: user2
```

This is optional directive.

Require

Directive specifies users and groups allowed to access secure folder. Directive format:

```
Require valid-user | [user|group] user1 user2 ... userN
```

The following directive allows access to any authenticated user:

```
Require valid-user
```

The following directive allows access to users “john” and “marry”:

```
Require user john marry  
  
# or just  
Require john marry
```

The following directive allows access to “Accounting” group:

```
Require group Accounting  
  
# or just  
Require Accounting
```

.htaccess file could have multiple Require directives applied, for example:

```
Require john marry  
Require group Accounting
```

Please note that all user and group names are case-sensitive. That means that “John” and “john” are two different user accounts.

This directive is mandatory.