

Apache HTTP Server Version 2.4

Apache Module mod_macro

Description:	Provides macros within apache httpd runtime configuration files
Status:	Base
Module Identifier:	macro_module
Source File:	mod_macro.c
Compatibility:	Available in httpd 2.4.5 and later

Summary

Provides macros within Apache httpd runtime configuration files, to ease the process of creating numerous similar configuration blocks. When the server starts up, the macros are expanded using the provided parameters, and the result is processed as along with the rest of the configuration file.

Usage

Macros are defined using `<Macro>` blocks, which contain the portion of your configuration that needs to be repeated, complete with variables for those parts that will need to be substituted.

For example, you might use a macro to define a `<VirtualHost>` block, in order to define multiple similar virtual hosts:

```
<Macro VHost $name $domain>
<VirtualHost *:80>
    ServerName $domain
    ServerAlias www.$domain

    DocumentRoot "/var/www/vhosts/$name"
    ErrorLog "/var/log/httpd/$name.error_log"
    CustomLog "/var/log/httpd/$name.access_log" combined
</VirtualHost>
</Macro>
```

Macro names are case-insensitive, like httpd configuration directives. However, variable names are case sensitive.

You would then invoke this macro several times to create virtual hosts:

```
Use VHost example example.com
Use VHost myhost hostname.org
Use VHost apache apache.org

UndefMacro VHost
```

At server startup time, each of these `Use` invocations would be expanded into a full virtualhost, as described by the `<Macro>` definition.

The `UndefMacro` directive is used so that later macros using the same variable names don't result in conflicting definitions.

A more elaborate version of this example may be seen below in the Examples section.

Tips

Parameter names should begin with a sigil such as \$, %, or @, so that they are clearly identifiable, and also in order to help deal with interactions with other directives, such as the core `Define` directive. Failure to do so will result in a warning. Nevertheless, you are encouraged to have a good knowledge of your entire server configuration in order to avoid reusing the same variables in different scopes, which can cause confusion.

Parameters prefixed with either \$ or % are not escaped. Parameters prefixes with @ are escaped in quotes.

Avoid using a parameter which contains another parameter as a prefix, (For example, \$win and \$winter) as this may cause confusion at expression evaluation time. In the event of such confusion, the longest possible parameter name is used.

If you want to use a value within another string, it is useful to surround the parameter in braces, to avoid confusion:

```
<Macro DocRoot ${docroot}>
    DocumentRoot "/var/www/${docroot}/htdocs"
</Macro>
```

Examples

Virtual Host Definition

A common usage of `mod_macro` is for the creation of dynamically-generated virtual hosts.

```
## Define a VHost Macro for repetitive configurations

<Macro VHost $host $port $dir>
    Listen $port
    <VirtualHost *: $port>

        ServerName $host
        DocumentRoot "$dir"

        # Public document root
        <Directory "$dir">
            Require all granted
        </Directory>
```

```
# limit access to intranet subdir.
<Directory "$dir/intranet">
    Require ip 10.0.0.0/8
</Directory>
</VirtualHost>
</Macro>

## Use of VHost with different arguments.

Use VHost www.apache.org 80 /vhosts/apache/htdocs
Use VHost example.org 8080 /vhosts/example/htdocs
Use VHost www.example.fr 1234 /vhosts/example.fr/htdocs
```

Removal of a macro definition

It's recommended that you undefine a macro once you've used it. This avoids confusion in a complex configuration file where there may be conflicts in variable names.

```
<Macro DirGroup $dir $group>
    <Directory "$dir">
        Require group $group
    </Directory>
</Macro>

Use DirGroup /www/apache/private private
Use DirGroup /www/apache/server admin

UndefMacro DirGroup
```

<Macro> Directive

Description:	Define a configuration file macro
Syntax:	<Macro <i>name</i> [<i>par1</i> .. <i>parN</i>]> ... </Macro>
Context:	server config, virtual host, directory
Status:	Base
Module:	mod_macro

The `<Macro>` directive controls the definition of a macro within the server runtime configuration files. The first argument is the name of the macro. Other arguments are parameters to the macro. It is good practice to prefix parameter names with any of `'$%@'`, and not macro names with such characters.

```
<Macro LocalAccessPolicy>
    Require ip 10.2.16.0/24
</Macro>

<Macro RestrictedAccessPolicy $ipnumbers>
    Require ip $ipnumbers
</Macro>
```

UndefMacro Directive

Description:	Undefine a macro
Syntax:	UndefMacro <i>name</i>
Context:	server config, virtual host, directory
Status:	Base
Module:	mod_macro

The `UndefMacro` directive undefines a macro which has been defined before hand.

```
UndefMacro LocalAccessPolicy
UndefMacro RestrictedAccessPolicy
```

Use Directive

Description:	Use a macro
Syntax:	Use <i>name</i> [<i>value1</i> ... <i>valueN</i>]
Context:	server config, virtual host, directory
Status:	Base
Module:	mod_macro

The `Use` directive controls the use of a macro. The specified macro is expanded. It must be given the same number of arguments as in the macro definition. The provided values are associated to their corresponding initial parameters and are substituted before processing.

```
Use LocalAccessPolicy
...
Use RestrictedAccessPolicy "192.54.172.0/24 192.54.148.0/24"
```

is equivalent, with the macros defined above, to:

```
Require ip 10.2.16.0/24
...
Require ip 192.54.172.0/24 192.54.148.0/24
```

Comments

Notice:
This is not a Q&A section. Comments placed here should be pointed towards suggestions on improving the documentation or server, and may be removed by our moderators if they are either implemented or considered invalid/off-topic. Questions on how to manage the Apache HTTP Server should be directed at either our IRC channel, #httpd, on Libera.chat, or sent to our mailing lists.