

Guide to WHM Plugins - The ACL Object Perl Module

Guide to WHM Plugins

Plugin Files

[Installation Scripts](#)

[Interfaces](#)

[Access Control Lists](#)

[ACL Metadata Perl Module](#)

[ACL Object Perl Module](#)

[ACL Reference Chart](#)

[AppConfig Plugin Registration](#)

[Add Plugins](#)

[Checks](#)

[Configuration File](#)

[Uninstall Plugins](#)

Introduction

The ACL object Perl module configures the ACL's namespace and display information. Create the ACL object Perl module in the `/usr/local/cpanel/Cpanel/Config/ConfigObj/Driver/` directory. This file creates a `Cpanel::Config::ConfigObj::Interface::Config::v1` object with additional information about the object's visibility.

The ACL object Perl module



Important:

To avoid a significant startup cost, we recommend that you load modules dynamically. Load modules at the top of each subroutine; do **not** call modules via the `use` function directly. For more information, see the `init` submodule in the following example.

The following example code creates an object for the `ExampleACL` namespace:

WHM Plugins

About WHM Plugins

- *WHM plugins add custom features to the WHM interface.*
- *In WHM 11.38.1+, you **must** use the AppConfig system to register plugins.*

Compatible with:

- WHM 11.32+

Related Documentation

- [Configure ClamAV Scanner](#)
- [Manage Plugins](#)
- [Guide to WHM Plugins](#) — WHM plugins add new functionality to the WHM interface.
- [Guide to WHM Plugins - AppConfig Registration](#) — Use AppConfig to register plugins with the WHM interface.
- [Guide to WHM Plugins - Access Control Lists](#) — Access Control Lists (ACLs) limit the features that WHM users can access.

```

package Cpanel::Config::ConfigObj::Driver::ExampleACL;

use strict;

use parent qw(Cpanel::Config::ConfigObj::Interface::Config::v1);

our $VERSION = '1.0';

sub init {

    my $class      = shift;
    my $software_obj = shift;

    my $ExampleACL_defaults = {
        'thirdparty_ns' => "ExampleACL",
        'meta'          => {},
    };

    Cpanel::LoadModule::load_perl_module('Cpanel::Config::LoadCpConf');

    my $self = $class->SUPER::base( $ExampleACL_defaults,
    $software_obj );

    return $self;
}

sub info {
    my ($self) = @_;
    my $meta_obj = $self->meta();
    my $abstract = $meta_obj->abstract();
    return $abstract;
}

sub acl_desc {
    return [
        {
            'acl'           => 'software-ExampleACL',
            'default_value' => 0,
            'default_ui_value' => 1,
            'name'          => 'ExampleACL test',
            'acl_subcat'    => 'Third-Party Services',
        },
    ];
}

1;

```

Package the module

```
package Cpanel::Config::ConfigObj::Driver::ExampleACL;
```

This declaration instructs Perl to treat all of the file's functions as part of the `Cpanel::Config::ConfigObj::Driver::ExampleACL` namespace.

For more information, read [perldoc.perl.org's package](http://perldoc.perl.org's%20package) documentation.

Set the strict pragma

```
use strict;
```

This declaration instructs Perl to return errors if the file contains potentially unsafe code.

For more information, read [perldoc.perl.org's strict](http://perldoc.perl.org's%20strict) documentation.

Specify the class's parents

```
use parent qw(Cpanel::Config::ConfigObj::Interface::Config::v1);
```

This declaration instructs Perl to establish an ISA relationship with base classes at compile time.

For more information, read [perldoc.perl.org's perlobj](http://perldoc.perl.org's_perlobj) documentation.

Declare a module version

```
our $VERSION = '1.0';
```

This declaration creates the variable `$VERSION` and sets it to `1.0`. This declaration allows you to differentiate between this and future versions of your module.

Call modules as needed

```
Cpanel::LoadModule::load_perl_module('<MODULE NAME>');
```

To avoid a significant startup cost, you should load modules dynamically. Call modules at the beginning of each subroutine, do **not** call modules via the `use` method.

Initialize the ACL's custom namespace

```
sub init {
    my $class      = shift;
    my $software_obj = shift;

    my $ExampleACL_defaults = {
        'thirdparty_ns' => "ExampleACL",
        'meta'           => {},
    };
    my $self = $class->SUPER::base( $ExampleACL_defaults,
    $software_obj );

    return $self;
}
```

The `init` subroutine initializes the ACL's custom namespace (`ExampleACL`).

Configure the object's information

```
sub info {
    my ($self) = @_;
    my $meta_obj = $self->meta();
    my $abstract = $meta_obj->abstract();
    return $abstract;
}
```

The `info` subroutine configures the ACL's object's information.

Configure the ACL's description

```

sub acl_desc {
    return [
        {
            'acl'           => 'software-ExampleACL',
            'default_value' => 0,
            'default_ui_value' => 1,
            'name'          => 'ExampleACL',
            'acl_subcat'    => 'Third-Party Services',
        },
    ];
}

```

The `acl_desc` subroutine configures the ACL's description. This subroutine must return the following parameters and values:

| Parameter | Type | Description | Possible values | Example |
|-------------------------------|----------------|--|---|---------------------|
| <code>acl</code> | <i>string</i> | The ACL's internal name. | A valid string. | software-ExampleACL |
| <code>default_value</code> | <i>Boolean</i> | The ACL's default setting, to add to the ACL lists for WHM users that already exist. | <ul style="list-style-type: none"> 1 — Enabled. 0 — Disabled. | 0 |
| <code>default_ui_value</code> | <i>Boolean</i> | Whether the ACL displays in WHM's <i>Edit Reseller Nameservers and Privileges</i> interface (<i>WHM >> Home >> Resellers >> Edit Reseller Nameservers and Privileges</i>). | <ul style="list-style-type: none"> 1 — Display. 0 — Do not display. <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note:</p> <p>If you set this value to 0, server administrators can still set the ACL through the WHM API.</p> </div> | 1 |
| <code>name</code> | <i>string</i> | The ACL's display name (<i>Feature List</i> name) in the WHM interface. | A valid string. | ExampleACL |

| | | | | |
|------------|---------------|---|---|----------------------|
| acl_subcat | <i>string</i> | The subcategory name under which the ACL will display in the WHM interface. | An existing category name : <ul style="list-style-type: none">• Account Information• Account Management• Accounts• Advanced Account Management• Clustering• cPanel Management• DNS• Everything• Locales• Package Access• Packages• Packages Creation• Server Information• Services• Third-Party Services• Troubleshooting | Third Party Services |
|------------|---------------|---|---|----------------------|