

# Tutorial - Create a New WHM Interface in Template Toolkit

## Tutorials

- [Tutorial - Customize the WHM User Interface with CSS](#)
- [Tutorial - Create a New Paper Lantern Interface in PHP](#)
- [Tutorial - Add a Link to the cPanel Interface](#)
- [Tutorial - Use UAPI's Fileman::upload\\_files Function in Custom Code](#)
- [Tutorial - Create an Integration Link](#)
- [Tutorial - Localize Text in cPanel Plugins](#)
- [Tutorial - Create a New WHM Interface in Template Toolkit](#)
- [Tutorial - Register a WHM Plugin with AppConfig](#)
- [Tutorial - Create a New WHM Interface in PHP](#)
- [Tutorial - Create Custom-Branded Login Pages](#)
- [Tutorial - Create a WHM Plugin](#)
- [Tutorial - Integrate Custom Webmail Applications](#)
- [Tutorial - Create a ModSecurity Vendor](#)
- [Guide to Report Receiver APIs for the ModSecurity Rule Reports](#)
- [Tutorial - Create a New Paper Lantern Interface](#)
- [Tutorial - Call UAPI's SSL::install\\_ssl Function in Custom Code](#)
- [Tutorial - Create a Standardized Hook](#)
- [Tutorial - Create a Custom cPanel Style](#)

## Introduction

This tutorial uses Template Toolkit to create a new WHM interface. The examples below use WHM's master template, which creates new interfaces that integrate seamlessly with the look and feel of the main WHM interface.



### Important:

This tutorial **requires** cPanel & WHM version 64 or later. Earlier versions of cPanel & WHM are **not** compatible with this tutorial.

## Create a new WHM interface



### Create a Template Toolkit file with the correct USE statements.

To begin, create a new Template Toolkit file. Generally, this file uses the `.tmpl` file extension.

- Lines 2 and 3 use [Template Toolkit's USE directive](#) to load the `Whostmgr` and `JSON` modules, respectively.
- WHM interfaces **require** these modules in order to load properly.

```
[%  
USE Whostmgr;  
USE JSON;  
%]
```



### Add correct handling for right-to-left locales.



### Note:

By default, cPanel & WHM displays interfaces correctly for left-to-right locales.

If your plugin will support multiple locales, we **strongly** recommend that you configure the interface to correctly display right-to-left **and** left-to-right locales.

- Line 5 uses [Template Toolkit's IF directive](#) to check whether the authenticated user's locale is a right-to-left (`rtl`) locale.
- Line 6 uses [Template Toolkit's SET directive](#) to set the correct minified Bootstrap CSS file if the authenticated user's locale is a right-to-left (`rtl`) locale.

```
[%  
USE Whostmgr;  
USE JSON;  
  
IF locale.get_html_dir_attr() == 'rtl';  
%]
```

## In this tutorial

This tutorial uses the following techniques, languages, tools, or concepts:

- [Template Toolkit](#)
- [Perl](#)
- [JavaScript](#)

Use this tutorial when you develop:

- [WHM Plugins](#)

## Related documentation

- [Configure ClamAV Scanner](#)
- [Tutorial - Customize the WHM User Interface with CSS](#) — This tutorial customizes the appearance of a WHM interface theme.
- [Guide to Template Toolkit](#) — Template Toolkit is the main template system for the cPanel and WHM interfaces.
- [Guide to Template Toolkit - API Calls from Template Toolkit](#) — cPanel & WHM uses template files to construct interfaces that call cPanel's APIs.
- [Guide to Template Toolkit - Localization in Template Toolkit](#) — Use localization in Template Toolkit files to produce dynamic interfaces in multiple languages.

- [Guide to Replacing cPanel API 1 functions with UAPI equivalents](#)

```
SET rtl_bootstrap = Whostmgr.find_file_url('/3rdparty/bootstrap-rtl/optimized/dist/css/bootstrap-rtl.min.css');
END;
%]
```

## 3

### Add WHM's wrappers.

Use [Template Toolkit's WRAPPER directive](#) to apply WHM's master template, title, and theme to the template file.

- Line 9 uses the `WRAPPER` directive to set WHM's master template file.
- Line 10 sets the interface's header text and localizes it. Use the `header` value to set the name at the top of the interface.
- Line 11 applies WHM's Bootstrap theme.

```
[%
USE Whostmgr;
USE JSON;

IF locale.get_html_dir_attr() == 'rtl';
SET rtl_bootstrap = Whostmgr.find_file_url('/3rdparty/bootstrap-rtl/optimized/dist/css/bootstrap-rtl.min.css');
END;

WRAPPER 'master_templates/master.tmpl'
    header = locale.maketxt("Example Plugin")
    theme='bootstrap';
%]

[% END %]
```



#### Important:

You can use additional flags to customize the way in which you include the master template, or the individual header and footer files. For more information and a full list of flags, read our [Guide to WHM Plugins - Interfaces](#) documentation.

## 4

### Add stylesheets.

Add stylesheets to customize the appearance of your interface.

- Line 9 uses [Template Toolkit's SET directive](#) to set the list of stylesheets.
- Line 10 adds the `rtl_bootstrap` stylesheet, which you set in Step 2.
- Line 11 adds the `/libraries/fontawesome/css/font-awesome.min.css` stylesheet, which allows your interface to load [Font Awesome](#) icons.
- Line 16 adds this list of stylesheets to the `WRAPPER` directive.

**Note:**

Enter the CSS files' paths relative to the `/docroot/` directory.

```
[%
USE Whostmgr;
USE JSON;

IF locale.get_html_dir_attr() == 'rtl';
SET rtl_bootstrap = Whostmgr.find_file_url('/3rdparty/bootstrap-
rtl/optimized/dist/css/bootstrap-rtl.min.css');
END;

SET styleSheets = [
    rtl_bootstrap,
    '/libraries/fontawesome/css/font-awesome.min.css'
];

WRAPPER 'master_templates/master.tpl'
    header = locale.maketext("Example Plugin")
    stylesheets = styleSheets,
    theme='bootstrap';
%]

[% END %]
```



## Add interface contents.

Add the contents of your plugin's interfaces. These contents may use any combination of Template Toolkit and other web languages.

```
[%
USE Whostmgr;
USE JSON;

IF locale.get_html_dir_attr() == 'rtl';
SET rtl_bootstrap = Whostmgr.find_file_url('/3rdparty/bootstrap-
rtl/optimized/dist/css/bootstrap-rtl.min.css');
END;

SET styleSheets = [
    rtl_bootstrap,
    '/libraries/fontawesome/css/font-awesome.min.css'
];

WRAPPER 'master_templates/master.tpl'
    header = locale.maketext("WHM Example Plugin")
    stylesheets = styleSheets,
    theme='bootstrap';
%]

<div class="callout callout-info">
Interface contents go here.
</div>

[% END %]
```

