Theme Options

- Multiple option pages
- Localizing the page title
- Menu icon
- Menu title
- Menu position
    - NB! You can specify custom icons only to parent theme options pages.
- Accessing field values

Theme option containers are used to add pages with options in the back-end. Field data is stored as options. Note that by default only users with the `manage_options` capability have access to this container. In order to override this behavior please refer to the `carbon_fields_theme_options_container_admin_only_access` filter in the Hooks section.

By default, theme options containers automatically create main page in the admin area menu named **“Theme Options”**. In most cases these default settings are sufficient, but if you need to change the title or the location of a page in the menu, read the **“Multiple option pages”** section below.

```
use Carbon_Fields\Container;
use Carbon_Fields\Field;

Container::make( 'theme_options', __( 'Theme Options' ) )
    ->add_fields( array(
        Field::make( 'text', 'crb_facebook_url', __( 'Facebook URL' ) ),
        Field::make( 'textarea', 'crb_footer_text', __( 'Footer Text' ) )
    ) );
```

Multiple option pages

It is sometimes needed to create more than one option page. At other times you need to place different pages in different sections in the admin menu. For example, you might have extensive list of settings for the background that you would want to place on a separate Theme options page

under Appearance.

To change the location of your Theme Options page, you use `set_page_parent( $parent )`, where `$parent` is either:

- A reference to a top level Theme Options container.
- The menu slug of a top level Theme Options container.
- The menu slug of any top level adminsitration page. This corresponds to the `$parent_slug` parameter of add_submenu_page(). You can see all predefined page parents here.

Below you see sample code for creating three theme option containers:

```
use Carbon_Fields\Container;
use Carbon_Fields\Field;

// Default options page
$basic_options_container = Container::make( 'theme_options', __( 'Basic Options' ) )
    ->add_fields( array(
        Field::make( 'header_scripts', 'crb_header_script', __( 'Header Script' ) ),
        Field::make( 'footer_scripts', 'crb_footer_script', __( 'Footer Script' ) ),
    ) );

// Add second options page under 'Basic Options'
Container::make( 'theme_options', __( 'Social Links' ) )
    ->set_page_parent( $basic_options_container ) // reference to a top level container
    ->add_fields( array(
        Field::make( 'text', 'crb_facebook_link', __( 'Facebook Link' ) ),
        Field::make( 'text', 'crb_twitter_link', __( 'Twitter Link' ) ),
    ) );

// Add third options page under "Appearance"
Container::make( 'theme_options', __( 'Customize Background' ) )
    ->set_page_parent( 'themes.php' ) // identificator of the "Appearance" admin section
    ->add_fields( array(
        Field::make( 'color', 'crb_background_color', __( 'Background Color' ) ),
        Field::make( 'image', 'crb_background_image', __( 'Background Image' ) ),
    ) );
```

For detailed information on managing admin pages, see Administration_Menus.

Localizing the page title

By default, the URL of the option page is generated from the title passed to the `Container::make` method. In the example above, the URL will be `admin.php?page=theme-options`. If you use a `gettext` function to localize the title, for example:

```
Container::make( 'theme_options', __( 'Theme Options', 'my-textdomain' ) )
```

then the URL of your page will be different for different languages. This will cause issues with multilingual plugins, such as WPGlobus or WPML, that allow for admin language switching.

To resolve this, you need to call the `set_page_file` method, like this:

```
use Carbon_Fields\Container;
use Carbon_Fields\Field;

Container::make( 'theme_options', __( 'Theme Options', 'my-textdomain' ) )
    ->set_page_file( 'theme-options' )
    ->add_fields( array(
        Field::make( 'text', 'crb_facebook_url' ),
        Field::make( 'textarea', 'crb_footer_text' ),
    ) );
```

Menu icon

To change the icon of your Theme Options page, you can use `set_icon( $icon )`, where `$icon` can be one of the values, supported in the `$icon_url` parameter of the add_menu_page() function.

Below you see an example for setting up the icon of the theme options page:

```
use Carbon_Fields\Container;
use Carbon_Fields\Field;

Container::make( 'theme_options', 'Basic Options' )
    ->set_icon( 'dashicons-carrot' )
    ->add_fields( array(
        Field::make( 'text', 'crb_test_field' ),
    ) );
```

Menu title

To change the menu button title of the container, you can use `set_page_menu_title( $title )`, where `$title` is your desired title. For more information see the add_menu_page() function.

```
Container::make( 'theme_options', 'A very long option page title' )
    ->set_page_menu_title( 'Custom Options' )
    ->add_fields( array(
        // ...
    ) );
```

Menu position

To change the position priority of the page's menu button, you use `set_page_menu_position( $position )`, where `$position` is a number. For more information see the add_menu_page() function.

```
Container::make( 'theme_options', 'Basic Options' )
    ->set_page_menu_position( 80 )
    ->add_fields( array(
        // ...
    ) );
```

NB! You can specify custom icons only to parent theme options pages.

Accessing field values

To retrieve field values from a theme options container, you need to use the function `carbon_get_theme_option( $name )`, where:

Parameter	Description
`$name`	The field name pattern of the field to be retrieved.

```
<p>Copyright <?php echo carbon_get_theme_option( 'crb_copyright' ); ?></p>
<p>
    Office locations:
    <?php
    $address_lines = carbon_get_theme_option( 'crb_addresses' );
```

```
    foreach ( $address_lines as $line ) {
        echo $line . '<br/>';
    }
    ?>
<p>
```

After saving, the `carbon_fields_theme_options_container_saved` hook is called, which allows you to hook additional functionality after saving.

□□□ #2
□ □□□ □□□ 2026-08-29 16:13:57 UTC
□ □□□ □□□ 2026-08-29 16:13:57 UTC