

# Components & Layouts Documentation

Module Version: 1.003  
09.25.2018

---

## Requirements

Miva Merchant: **9.0006 or higher**

Store User Interface: **Miva Merchant CSSUI**

## Installation

1. Log into your Miva Merchant Admin
2. Navigate to **Modules**
3. Click the **+** icon to add a Module
4. Click **Upload** and upload **tgcomponents.mvc**
5. Press the **+ Add** button
6. Navigate to **Utilities**
7. Click on the **Add/Remove Modules** tab
8. Look for **Components & Layouts** and click **Install**
9. The **Components & Layouts** module is now installed!

## Setting up the tgcomponent Item

By default this should be created. If you do not see the item tgcomponent in your items list, follow the following steps:

1. Navigate to **User Interface**
2. Click on the **Items** tab
3. Click the **+** icon to add an item
4. Set the code to **tgcomponent**
5. Set the module to **TGCOMPONENTS**
6. Click the **+ Add** button
7. Assign this item to any page where you're going to call the item.

The following parameters may be used with the item:

```
<mvt:item name="tgcomponent" param="Layout_Load_Code( 'my_layout_code',  
1.settings:output )"/>
```

The first parameter is the **Layout Code** you are referencing.

The second parameter must be a variable. The structure of the layout will be returned to this variable.

## Components & Layouts Module in the Admin

### Navigating to Component & Layouts

Click the **Menu** button, and navigate to **Utilities**. Click the arrow and there will be a link labeled **Components and Layouts**.

### Adding Components

To add a new component, you will need to fill in the following fields:

- Code
  - This is **required**, and must be unique
- Name
  - This is **Required**
- Description
  - This is used in the admin interface only, and is optional. This could have a description that is used for other users, as a reference.
- Image
  - This is used in the admin interface only, and is optional. This could be a reference image for layout building.
- Allow Nested Components
  - This is used in the admin interface only, and is optional. This determines if you can “drag” elements into this component.

### Adding Attributes/ Options to Components

To add a new attribute to a component, you first need to select your component, and “Go to” that component.

The following fields will be available to fill out:

- Code
  - This is **required** and must be unique to the component
- Prompt
  - This is **required**

- Type
  - This is **required**. This is the type of field it is. The following is a breakdown of each type:
    - Text Field: A Text field
    - Radio Buttons: A list of radio buttons
    - Drop-down List: A list of options
    - Checkbox: A checkbox (Will evaluate as 1/0)
    - Textarea: A textarea
    - Image: An image upload, with options to resize an image (this is available in the pop up)
    - Product: A product popup selector
    - Category: A category popup selector
    - Link: The user will have the "Type" and "Value" available for them in the admin interface.
    - Image Type: The user will have a dropdown list of all Image Types associated with the Store.
    - Date/ Time: The user will have a Date/Time picker.
  - Required
    - If this attribute is required, you are able to specify that here.

When adding options, you will only need to supply the Prompt.

## Adding Layouts

Only two fields are needed to create a layout:

- Code
  - This must be unique
- Name
  - This is a reference for the admin interface only

If you have a layout you like using, and want to duplicate it, you can select the layout you want, and the **Duplicate Layout** button will appear. You will need to specify a new code/ name for the new layout.

## Updating Layouts

To add a new top-level component, you can click on **Add New Component**.

To add a child component, you can click the **+** button on the component you are referencing.

**Please note:** Updates will only be saved when you click the **Update** button in the top right of the screen. If you click Reset, your work will be un-done.

## User Group Privileges

When the module is installed, by default it will add a new **Privilege** labeled **Components & Layouts Module**. If a user is not an administrator, you will need to enable their privileges for the module.

## Import/ Export

### Export

The export is located under **Data Management > Import/Export > Components & Layouts**

For exporting, there are two options:

- Components
  - This will export all components
- Layouts
  - This will export all layouts

### Import

XML Provisioning is the only importing that is available.

The following tags are available (please note these are examples):

```
<Module code="TGCOMPONENTS" feature="util">
  <Layout_Add>
    <Code>Your_Code_Here</Code>
    <Name>Your Name Here</Name>
  </Layout_Add>

  <Layout_Update code="Your_Code_Here">
    <Code>Your_New_Code_Here</Code>
    <Name>Your New Name Here</Name>
  </Layout_Update>

  <Layout_Delete code="Your_Code_Here" />

  <Component_Add>
    <Code>Your_Code_Here</Code>
    <Name>Your Name Here</Name>
    <Descrip>Your Description</Descrip>
    <Image>graphics/00000001/leia.png</Image>
    <Allow_Nest>1</Allow_Nest>
  </Component_Add>
```

```

<Component_Update code="My_Old_Code_Here">
  <Code>Your_Code_Here</Code>
  <Name>Your Name Here</Name>
  <Descrip>Your Description</Descrip>
  <Image>graphics/00000001/leia.png</Image>
  <Allow_Nest>1</Allow_Nest>
</Component_Update>

<Component_Delete code="My_Code_Here" />

<ComponentAttribute_Add component="My_Component">
  <Code>MyAttribute</Code>
  <Prompt>My Prompt</Prompt>
  <Type>image</Type>
  <Required>1</Required>
</ComponentAttribute_Add>

<ComponentAttribute_Update component="My_Component"
code="Attribute_Code">
  <Code>MyAttribute</Code>
  <Prompt>My Prompt</Prompt>
  <Type>image</Type>
  <Required>1</Required>
</ComponentAttribute_Update>

<ComponentAttribute_Delete component="My_Component"
code="Attribute_Code" />

<ComponentAttributeOption_Add component="My_Component"
code="Attribute_Code">
  <Prompt>My Prompt</Prompt>
</ComponentAttributeOption_Add>

<LayoutComponent_Add layout="My_Layout_Code"
component="Component_Code">
  <Name>Name</Name>
  <Active>1</Active>

```



```

    <Parent>Some Parent</Parent>
    <Attributes>
        <Attribute code="MyAttribute">
            <Value>My Value Here</Value>
        </Attribute>
        <Attribute code="MyAttributeLink">
            <LinkType>Product</LinkType>
            <Value>My Value Here</Value>
        </Attribute>
    </Attributes>
</LayoutComponent_Add>
</Module>

```

The following are the available types for component attributes:

| Code Used for <b>Type</b> in XML | Prompt in Admin |
|----------------------------------|-----------------|
| text                             | Text Field      |
| radio                            | Radio Buttons   |
| select                           | Drop-down List  |
| checkbox                         | Checkbox        |
| memo                             | Text Area       |
| image                            | Image           |
| product                          | Product         |
| category                         | Category        |
| link                             | Link            |
| imagetype                        | Image Type      |
| datetime                         | Date/ Time      |

## Item Information (for Developers)

When you use **Layout\_Load\_Code**, the following structure will be returned.

```
[x]:active
[x]:attributes:{{ attribute code }}:value
[x]:children[x]:active=1
[x]:children[x]:attributes:{{ attribute code }}:value=Slide+1+%3A%29
[x]:children[x]:component:allow_children
[x]:children[x]:component:code
[x]:children[x]:component:id
[x]:children[x]:component:name
[x]:children[x]:component_id
[x]:children[x]:disp_order
[x]:children[x]:id
[x]:children[x]:layout_id
[x]:children[x]:name
[x]:children[x]:parent
[x]:children_count
[x]:component:allow_children
[x]:component:code
[x]:component:id
[x]:component:name
[x]:component_id
[x]:disp_order
[x]:id
[x]:layout_id
[x]:name
[x]:parent
```

### Notes

[x] denotes it is part of an array

:children can be nested inside of other :children

{{ attribute code }} would be the attribute code of your attributes which are tied to each component.

Example: I have an attribute with a code of "kitten"= [x]:attributes:kitten:value

## Uninstalling the Module

To uninstall the module, you will need to make sure that the **tgcomponent** item is fully removed.

Navigate to **Utilities**, and click on the **Add/Remove Modules** tab.

Find **Component & Layouts** and click the **Remove** button.