ultimo theme

For Magento 2

User Guide

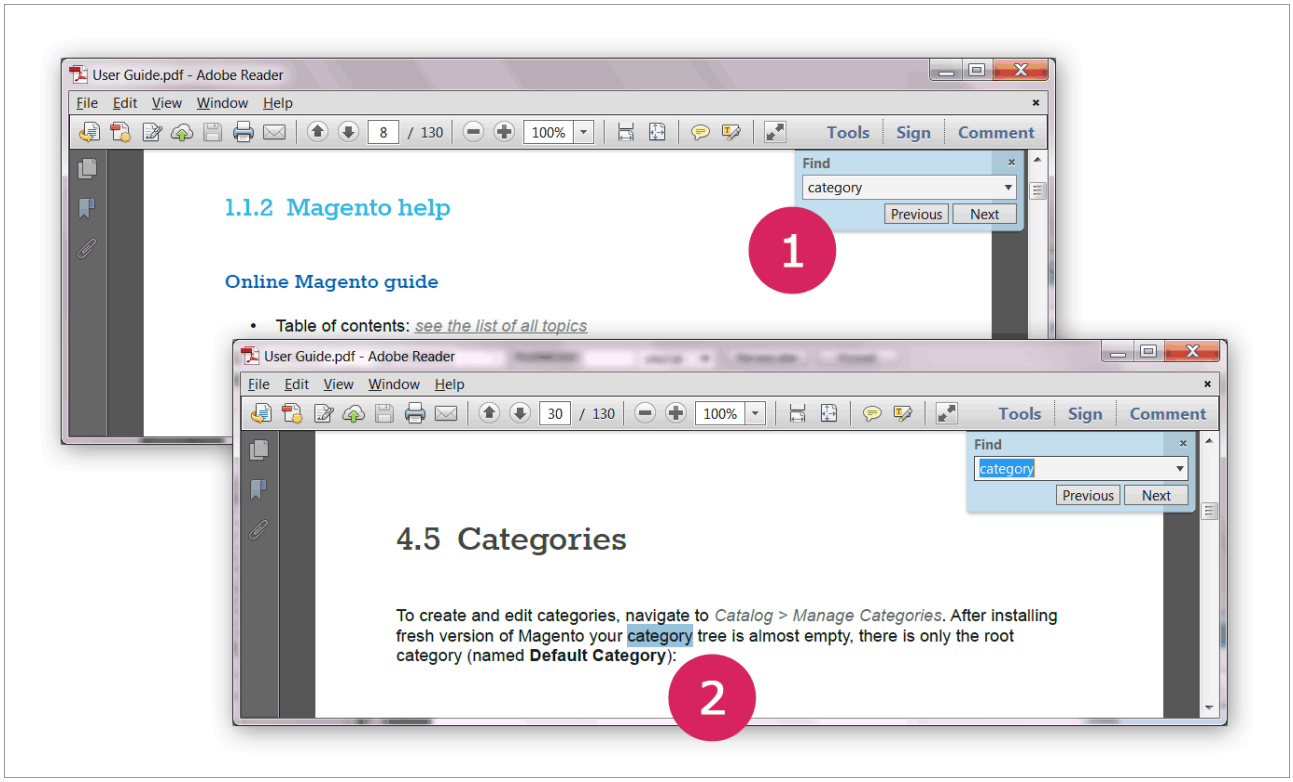
Copyright © 2012-2017 Infortis. All rights reserved.

Document updated: September 03, 2017

How to use this document

Please read this user guide carefully, it will help you eliminate most of potential problems with incorrect configuration of the theme and Magento.

Use the search tool in your PDF reader (click **CTRL+F** in most readers) to quickly find all the keywords which you are looking for:



This document is organized as follows:

- Chapters **1. FAQ + Quick Start** and **2. Help & Support** provide an overview of this theme, some basic information about support and links to many useful websites about Magento.
- Chapter **3. Installation** describes theme installation process.

- Chapter **4. Magento configuration** describe selected Magento features.

- Chapter **5. Customization** shows how to customize the theme. Subchapter **5.2** describes Magento themes structure and some basic concepts which you need to know when starting to work with Magento.

- Chapter **6. Theme features and elements** provides a description of selected elements of this theme, such as product sliders, home page, built-in static blocks, useful CSS classes etc.

- Chapters from **7.** to **12.** – specify a list of all theme settings available in the theme admin panel. **Each chapter refers to specific section of theme admin panel.**

- Chapter **13. Troubleshooting** describes the most common problems which you may encounter after installation or during theme and Magento configuration.

All chapters are listed in the **Table of Contents** below.

Table of Contents

1. FAQ + Quick Start....11
1.1 Help & support....11
1.2 License....11
1.3 Installation....12
1.4 After installation....12
2. Help & Support....14
2.1 References....14
2.1.1 Documentation....14
2.1.2 Magento community....15
2.2 Support policy....16
2.3 Third-party extensions....18
3. Installation....19
3.1 Theme upgrade....19
3.2 Theme installation....19
3.2.1 Installation troubleshooting....20
3.2.2 Preparing your Magento....21
3.2.3 Copying and uploading theme files....22
3.2.4 Enabling the Ultimo modules....24
3.2.5 Generating Magento's static files....27
3.2.6 Enabling the theme....28
3.2.7 Installing CMS blocks and pages....30
3.3 Problems after installation....31
4. Magento configuration....33
4.1 Caching system....34
4.1.1 Flush cache....34
4.1.2 Disable cache....34
4.1.3 Enable cache....35
4.1.4 Refresh cache....35
4.2 Index management....36
4.3 How to enable the theme....37
4.4 Logo....38
4.4.1 Sticky logo....38
4.5 Categories....39
4.5.1 Category info....40
4.5.2 Find category ID....41
4.5.3 Landing pages....41
4.5.4 Depth of the menu....41
4.5.5 Layered Navigation ("Shop By" filters)....41
4.5.6 Categories inside Layered Navigation....42
4.6 Products....44
4.6.1 "Only X left" stock indicator....44

4.7 Attributes....45
4.7.1 Attribute Management....45
4.7.1.A Examples of attributes....46
4.7.1.A.1 Attribute: manufacturer....46
4.7.1.A.2 Attribute: sale....48
4.7.1.A.3 Attribute: new....50
4.7.1.A.4 Attribute: custom_label....52
4.7.2 Attribute Sets....55
4.8 Content Management System (CMS)....57
4.8.1 Add blocks to selected pages....57
4.8.2 Static blocks....60
4.8.3 CMS markup tags....61
4.8.3.A Store URL....61
4.8.3.B Media URL....62
4.8.3.C Block ID....63
4.8.4 WYSIWYG editor....64
4.9 Translation / Localization....65
4.9.1 Interface translation....65
4.9.2 Language flag....66
5. Customization....68
5.1 How to modify the theme....69
5.1.1 Override default CSS styles....69
5.1.1.A Use _custom.less file....69
5.1.1.B Debugging....69
5.1.1.C Add CSS via Magento admin....70
5.1.1.D How to find out which part of CSS to override?....71
5.1.2 Modify template files....72
5.1.3 Create custom sub-theme....73
5.2 Magento theme customization....74
6. Theme features and elements....75
6.1 Home page and CMS pages....76
6.1.1 Default pages....76
6.1.1.A Selecting a home page....76
6.1.2 Creating a page....76
6.1.2.A Page layout....77
6.1.2.B Page content....77
6.1.2.C Full width page....78
6.1.2.C.1 Maximum width of the content....78
6.1.2.C.2 Content with full width background....79
6.1.3 Home page blocks....81
6.1.3.A Image slideshow....81
6.1.3.B Banners....81
6.1.3.C Grid/List of Products....81
6.1.4 Featured Products Slider....84
6.1.4.A List of parameters....84
6.1.4.B Examples of shortcodes....88
6.1.4.C Display any set of products using the slider....90
6.2 Built-in blocks and containers....92
6.2.1 Add custom content....92
6.2.1.A Built-in static blocks....93
6.2.1.B Custom static blocks assigned to containers....93
6.2.2 Import static blocks....93
6.2.3 List of built-in static blocks and containers....95
6.2.3.A Page....95
6.2.3.B Header....98
6.2.3.C Footer....102

6.2.3.D Product page....105
6.3 Pages and elements....108
6.3.1 Drop-down cart....108
6.4 Grid system....109
6.4.1 Bootstrap grid overview....109
6.4.2 Grid examples....111
6.4.3 Additional utility classes for grid system....112
6.4.4 Old 12-column grid system (deprecated)....113
6.5 CMS components....118
6.5.1 CSS utility classes....118
6.5.1.A General utility classes....118
6.5.1.B Hide/show content depending on screen size....120
6.5.1.C Old classes to hide elements (deprecated)....122
6.5.2 Collapsible blocks....123
6.5.3 Icons....124
6.5.3.A Basic icons....124
6.5.3.A.1 Add CSS styles to icons....125
6.5.3.A.2 Size....125
6.5.3.A.3 List icons....126
6.5.3.A.4 Pulled icons and borders....126
6.5.3.A.5 Spinning icons....126
6.5.3.A.6 Rotated and flipped....126
6.5.3.A.7 Icons inside buttons....127
6.5.3.B Iconbox (icon with background color)....127
6.5.3.B.1 Change color on mouse hover....128
6.5.3.B.2 Size....128
6.5.3.B.3 Shape....128
6.5.3.B.4 Iconbox effects....129
6.5.3.B.5 Image file iconbox....129
6.5.3.C Social icons....130
6.5.3.D Letters, numbers and other characters as icons....132
6.5.3.E Font Awesome icons....132
6.5.4 Icon + text....134
6.5.4.A Simple block with icon....134
6.5.4.B Change iconbox color on mouse hover....135
6.5.4.C Indented block....135
6.5.4.D Centered block....136
6.5.5 Banners....137
6.5.5.A Image banner....137
6.5.5.A.1 How to get the image....137
6.5.5.B Banner with hyperlink....138
6.5.5.C Banner with image hover effect....139
6.5.5.D Banner with color overlay....139
6.5.5.E Banner with caption....141
6.5.5.E.1 Inline styles....141
6.5.5.E.2 Remove caption background color....142
6.5.5.E.3 Caption with fade in/out effect....142
6.5.5.E.4 Center caption....143
6.5.5.E.5 Center vertically, full cover....143
6.5.5.E.6 Caption positioning....144
6.5.5.E.7 Precise positioning....144
6.5.5.E.8 Caption effects....145
6.5.5.E.9 Caption with text background....146
6.5.5.E.10 Old classes to create captions for slideshow (deprecated)....147
6.5.5.F Groups of banners....148
6.5.6 Animations....150
6.5.6.A Basic usage....150
6.5.6.B List of animations....151
6.5.6.C Additional options....151
6.5.6.C.1 Animation delay....151
6.5.6.C.2 Animation duration....152

6.6 Theme customization....154
7. Theme settings....155
7.1 Header....156
7.1.1 Header structure....156
7.1.2 Header block positions....158
7.1.3 Header configuration options....159
7.2 Category View....162
7.3 Category View (Grid Mode)....163
7.4 Category View (List Mode)....168
7.5 Layered Navigation....169
7.6 Product Page....170
7.6.1 Product page structure....170
7.6.2 Product page block positions....178
7.6.3 Product page configuration options....179
7.7 Product Page Gallery....184
7.8 Product Labels....185
7.9 Footer....188
7.10 Product sliders....190
7.11 Data Import....191
7.11.1 Theme Configuration Import/Export....192
7.11.2 Import Static Blocks and Pages From Selected Demo....196
8. Theme Design....199
8.1 Colors....200
8.2 Font....201
8.3 Effects....201
8.4 Page....201
8.5 Header....202
8.6 Header – Top....203
8.7 Header – Primary....203
8.8 Main Menu....203
8.9 Main....203
8.10 Category View....203
8.11 Product Page....203
8.12 Slideshow....204
8.13 Footer....204
8.14 Footer – Top....204
8.15 Footer – Primary....204
8.16 Footer – Secondary....204
8.17 Footer – Bottom....205
9. Theme Layout....206
9.1 Responsive Layout....207
10. Slideshow....210
10.1 Slideshow on home page....210
10.1.1 Add home page slideshow through the module admin panel....210
10.1.2 Add home page slideshow through the shortcode....211

10.2 Slideshow on any page....211
10.2.1 List of parameters....212
10.3 Create slides....214
10.3.1 Basic structure of the static block with slides....215
10.3.2 Content of a slide....217
10.3.3 Animated content of a slide....218
10.3.4 Size of slides and banners....220
10.3.4.A Slideshow with additional side banners....221
10.4 Create additional side banners....222
10.5 Import sample slides and shortcodes....224
10.5.1 Import static blocks with sample slides....224
10.5.2 Import page with examples of shortcodes....224
10.6 Full width slideshow on home page....226
10.7 Slideshow configuration....227
10.7.1 General....227
10.7.2 Additional Banners....230
10.7.3 Navigation....231
11. Brands....232
11.1 Basic information....232
11.1.1 Brand attribute....232
11.1.2 Brand images....233
11.1.3 Brand names and image file names....233
11.2 Brands configuration....236
11.2.1 General....236
11.2.2 Brand on Product Page....241
11.2.3 Brand List....242
11.2.4 Brand Slider....243
12. Menu....246
12.1 Menu configuration....247
12.1.1 Top Menu....247
12.1.2 Mobile Menu....249
12.1.3 Sidebar Menu....250
12.1.4 Category Labels....255
12.2 Drop-down boxes....256
12.2.1 Drop-down box configuration options....256
12.2.2 Submenu types....261
12.2.3 Category blocks....264
12.2.3.A Category blocks in mobile menu....267
12.2.4 Sample content for category blocks....269
12.3 Custom links in the menu....272
12.3.1 Custom links inside a static block....272
12.4 Custom drop-down boxes in the menu....274
13. Troubleshooting....275
13.1 General issues after installation....275
13.1.1 Some features don't work or give errors....275
13.1.2 Blank page or 404 not found – when trying to open theme admin pages....276
13.1.3 There is no theme admin tab in the admin panel, or I get a blank page, or Access Denied error....276
13.1.4 File permissions....276
13.1.5 Theme features don't work after installation or upgrade....276
13.1.6 "Package with this name does not exist" message is displayed when trying to

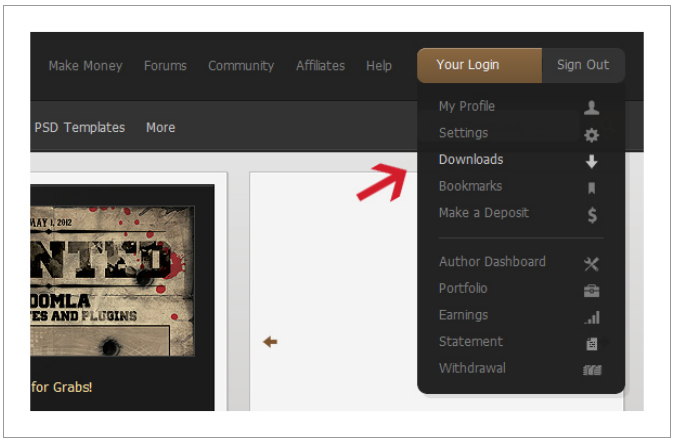
enable the theme....277
13.1.7 After theme installation the store gives blank page or an error....277
13.1.8 Tabs, sliders or other interactive elements don't work....277
13.1.9 Images are not displayed....277
13.1.10 Sliders: previous/next buttons don't work in some browsers....278
13.1.11 Slideshow is not displayed on the home page....278
13.2 Configuration....279
13.2.1 Theme configuration changes do not appear on the frontend....279
13.3 Third-party extensions....280
13.3.1 One of my extensions doesn't work with this theme....280
13.3.2 One of the features stopped working after third-party extension was installed 280

Introduction

Ultimo is extremely customizable responsive Magento theme. It is suitable for every type of store. Based on Ultimo you can easily create your custom sub-themes (in Magento 2 called "child themes").

Rate this theme

Don't forget to rate this theme on *ThemeForest* – it's good to know what others think about our work. You can rate it in the download area of your account on ThemeForest:



More info

To find out more about theme updates, new features and about our upcoming themes, you can follow us on Twitter and on ThemeForest:

themeforest Twitter

1. FAQ + Quick Start

1.1 Help & support

1. **Theme support**

   Please read this user guide carefully, it will help you eliminate most of potential problems with incorrect configuration of the theme and Magento. If you don't find the answer to your questions, please refer to this chapter for more information about support policy: **2.2 Support policy**.

2. **Magento support**

   Magento configuration, installation, maintenance, customization etc. is beyond the scope of our support. But since Magento is very popular platform you can find tons of useful tutorials and articles on the web. Here are some resources for start: **2. Help & Support**.

1.2 License

With **ThemeForest regular license** you are licensed to use this theme to create one single End Product (the final website customized with your content) for yourself or for one client. In short:

What is allowed with single regular license:

1. **You can create one website** for yourself or for your client and you can transfer that single website to your client **for any fee**. This license is then transferred to your client.

2. To sell the same website to another client, you will need to purchase **another regular license**.

3. You can install the theme on your **test server** for testing/development purposes (that installation shouldn't be available to the public).

4. You can modify or manipulate the theme, you can **combine it with other works** to create the End Product.

5. Theme **updates** for that single End Product (website) are free.

What is not allowed:

1. With single regular license it is not allowed to create **multiple websites**. For multiple websites you will need multiple regular licenses.

2. **Multi-store Magento installations** are not allowed with single regular license. Each regular license entitles you to use the theme in only one store (store/domain/subdomain). For multiple stores/domains/subdomains you will need multiple regular licenses.

For more information about licenses please refer to ThemeForest:

- License FAQ: *http://themeforest.net/licenses/faq*
- License comparison table: *http://themeforest.net/licenses*
- Regular license details: *http://themeforest.net/licenses/regular*
- Extended license details: *http://themeforest.net/licenses/extended*

1.3 Installation

If you encounter any problems after installation, refer to chapter **13. Troubleshooting**.

1.4 After installation

To start using the theme after installation you need to take a few quick steps to make sure all is configured properly. Also, not all of the features are enabled by default.

1. **Enable the theme in a correct way**

2. **Flush Magento cache**

Remember to flush Magento cache after making any changes in the configuration of theme and Magento, after editing content of CMS pages and blocks, after editing products etc. And after flushing the cache, always refresh your web browser to make sure that you can see the most recent files.
See for more details: **4.1 Caching system**.

3. ### Import configuration settings (optional)

   After the theme was installed and enabled, you can import theme configuration settings (colors, font, page width etc.) from one of the demo stores. For more details refer to **7.11 Data Import**. Of course, **configuration import is optional** and you can as well leave the default configuration (by default theme looks the same as demo 1).

   > **Note:**
   >
   > if you decide to import configuration from one of the demo stores, it is recommended to import to the **"Default Config"** scope.

4. ### Logo

   Upload your store logo: **4.4 Logo**.

5. ### Home page

   Configure your home page: **6.1 Home page and CMS pages**.

6. ### Home page slideshow

   Configure the home page slideshow: **10. Slideshow**. Each slide is a static block which can contain any custom images or HTML. After blocks import, your slideshow will have a few sample slides filled with some dummy images and captions. To edit captions, see this chapter for details: **6.5.5.E.10 Old classes to create captions for slideshow (deprecated)**.

2. Help & Support

2.1 References

You may find these resources useful while installing and configuring your Magento store:

2.1.1 Documentation

- **Basic information about Magento:**

  https://magento.com/developers/magento2

  https://magento.com/products/community-edition

- **System requirements:**

  http://devdocs.magento.com/guides/v2.0/install-gde/system-requirements.html

- **Magento documentation:**

  http://docs.magento.com/m2/ce/user_guide/getting-started.html

- **Magento installation guide:**

  http://docs.magento.com/m2/ce/user_guide/magento/installation.html

- **Magento developer documentation:**

  http://devdocs.magento.com/

2.1.2 Magento community

One of the most important things with Magento is that it's had a big community for many years. Which means that you can find articles and tutorials about almost anything you need. If you have a problem in Magento, you can be sure that someone has already solved it. So in most cases all you need to do is to find it on the forums. There's no need to reinvent the wheel.

These forums are here to provide you with access to a network of dedicated Magento enthusiasts sharing tips, tricks and support.

- **Magento Stack Exchange (Q&A):**

  http://magento.stackexchange.com/questions?sort=frequent

- **Stack Overflow (Q&A):**

  http://stackoverflow.com/questions/tagged/magento2?sort=newest

- **Magento's official forums:**

  https://community.magento.com/

2.2 Support policy

This user guide was created to help you quickly configure the theme – please read it carefully, it will help you eliminate most of potential problems with incorrect configuration.

> **Important:**
>
> please note that as authors of this theme we are able to provide support only for the features which we created and for the issues related strictly to this theme. We do not provide support for Magento configuration, installation, maintenance, customization etc.

Theme support

If you have found any bugs or have some other problems with this theme, please refer to chapter **13. Troubleshooting**. If the problem is not covered there, you can contact us via the contact form on ThemeForest. The most up-to-date information about support channel can always be found at:

- *http://themeforest.net/item/ultimo-fluid-responsive-magento-theme/3231798/support*

We will respond as soon as possible (within 24 – 48 hours, usually faster). Please make sure you take a look at the available resources before submitting a support request.

Questions about Magento

Elements such as *category management*, *product management*, *checkout process*, *payment methods* (and many others) are Magento's core functionalities not related to the theme and they are beyond the scope of our support. How to get Magento support? If you have some questions about Magento itself, please refer to the following websites, almost every detail of Magento configuration was discussed there so you will find answers for most of your questions:

- **Magento documentation:**

  The best place to start searching is the online Magento guide in which you can find description of all the basic Magento settings and many other useful information:

http://docs.magento.com/m2/ce/user_guide/getting-started.html

- **Magento Stack Exchange (Q&A):**

  http://magento.stackexchange.com/questions?sort=frequent

- **Stack Overflow (Q&A):**

  http://stackoverflow.com/questions/tagged/magento2?sort=newest

- **Magento's official forums:**

  https://community.magento.com/

Theme and Magento customization/modification support

Customization/modification of Magento and themes is a very large subject. So unfortunately custom modifications in the theme and in Magento are beyond the scope of our support. For more information about how you can customize this theme, please refer to chapter **5. Customization**.

2.3 Third-party extensions

Unfortunately we're unable to guarantee that all third-party extensions will work "out of the box" with our theme. It’s impossible for theme authors to make the theme compatible by default with all existing extensions because there are thousands of available extensions.

This is extension author's responsibility. Each extension modifies theme/Magento functionality in different way, so it's not up to us how it will work.

When you want to install an extension, the best option is to ask about details the authors of that extension – they will know better how the extension works, what elements of themes it overrides and how to integrate it with custom themes.

Only the “default” Magento theme is compatible with all third-party extensions, because all extensions are designed to work with it. But authors of the extension should provide detailed instruction about how to integrate the extension with custom themes, since nobody really uses “default” theme, everybody uses custom themes.

Generally all extensions can work with all themes but they sometimes have to be integrated manually. That’s because sometimes the extension and the theme override the same Magento template file so there can be a conflict. In such cases you need to customize the theme and merge two conflicting template files into one template file.

This isn't something specific to our themes, this is general rule for all Magento themes and extensions.

Support for extensions

Unfortunately third-party extensions are beyond the scope of our support. When you want to install an extension, the best option is to ask about details the authors of that extension. They should provide detailed instructions about their extension.

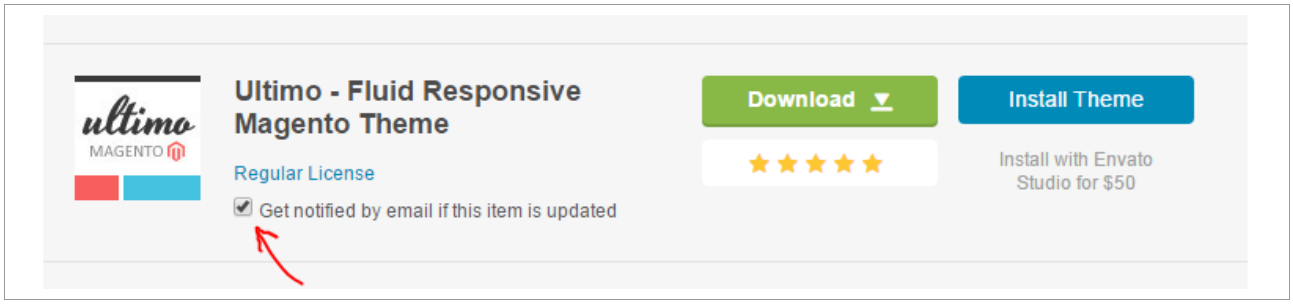
3. Installation

To install or upgrade the theme, please follow the instructions below.

3.1 Theme upgrade

To upgrade the theme to the latest version, follow the same procedure as for initial installation (which is described in the next chapter). Except the step in which you enable the theme in your Magento (of course if it's already enabled).

Always test the upgrade on a copy of your Magento before you upgrade your live store. Never go live with the upgrade without testing.



To get the latest version, re-download the theme from your ThemeForest account (from "Downloads" section). There's always the latest version.

3.2 Theme installation

These instructions will guide you through installing the Ultimo theme (and the accompanying Magento modules) into your Magento 2 system.

Ultimo is distributed as a ZIP archive, and installed directly into your system's `app` folder. Unzip your archive, and you'll see two folders:

```
• app
• pub/media
```

The **Magento module** files are located in `app/code`, and the **Magento theme** files are located in `app/design`. Folder `pub/media` contains additional sample images.

Installing the theme is as simple as:

- Uploading the module and theme files
- Running a few Magento CLI commands
- Configuring the theme in *`Stores > Configuration`*

The rest of this document will describe these steps in great detail.

Also, if you're partnered with a Magento agency, please consult with them for adapting Ultimo into their custom continuous integration/deployment systems.

3.2.1 Installation troubleshooting

Before we begin, there's one important thing to mention. Magento 2, despite its long development cycle, is new software. Magento 2 is also new software under rapid, ongoing development by the Magento core team and other third party extension developers and providers. While the broad strokes of these installation instructions should remain stable for the foreseeable future, small specific details may change.

If you run into a problem installing the theme, first, **don't panic**. :)

Step one to troubleshooting anything with computers is to go back, and reread the step you're currently on. Often, as human beings, there's small details that we don't realize are important until we've tried something. Often, re-reading the instructions can help you get to the bottom of your problem.

Step two is to use the internet to find a solution to your problem. Chances are if you've run into a new problem, someone else has as well. The following sites are great resources for finding up-to-date information on the latest developments in Magento 2, and asking for technical help with specific problems.

- **Magento Stack Exchange (Q&A):**

  http://magento.stackexchange.com/questions?sort=frequent

- **Magento's official forums:**

  https://community.magento.com/

- **Magento developer documentation:**

  http://devdocs.magento.com/

If you're still having trouble, step three is to get in touch with us via support. While there's a lot of things about a Magento system that are out of our control (server host configuration, third party extensions, change to the system), we're always happy to help out with some advice.

3.2.2 Preparing your Magento

To avoid unexpected problems after installation, you should first prepare your Magento. So before you install the theme in your Magento store, please strictly follow the steps described in this chapter.

1. It is highly recommended to install the theme on a **test server** before you install it on your live store.

> **Important:**
>
> always test the theme (and any other extension) on a copy of your Magento store before you install it on your live store. Never go live without testing.

2. **Backup Magento** files and database. Creating backup before installation of any themes or extensions in Magento is extremely important, especially if you are working on a live store.

> **Important:**
>
> as always, when updating your Magento system, confirm your backup system is correctly backing up your system files and the contents of your database, and that your backup restore process is working as expected.

3. **Disable the following options**: `Merge CSS Files`, `Enable Javascript Bundling`, `Merge JavaScript Files` and `Minify JavaScript Files` under `Stores > Configuration > Developer`.

You can enable these options again after installation.

4. **Remove all possible custom modifications** of Magento's Blank theme (one of the default themes in Magento 2). Ultimo inherits files from the Blank theme, so any modifications of the Blank theme may change the default behavior of Ultimo and break some functionality.

   You should never edit Magento's core files and default themes. Custom modifications should always be made via custom themes and modules.

5. Before you install the theme, make sure your store is disabled during the installation process. Enable Magento **maintenance mode**. You can find more details in Magento documentation: *http://devdocs.magento.com/guides/v2.0/install-gde/install/cli/install-cli-subcommands-maint.html#instgde-cli-maint*

   To enable the maintenance mode you can create a simple empty file named `.maintenance.flag` (**note the dot at the beginning of the file name**) in your Magento's `var` directory. Magento will be offline after that.

   You will need to disable the maintenance mode after installation by removing this file.

6. **Log out** from Magento admin panel.

> **Important:**
>
> do not just close the browser window, you need to click the *`Log Out`* link in the admin panel.

3.2.3 Copying and uploading theme files

Before you install the theme you need to prepare your Magento, so please do not omit the previous step: **3.2.2 Preparing your Magento**

1. Extract the theme package on your computer and navigate inside `Magento2/Theme Files/` folder.

2. You will see there a few folders. Navigate inside the folder which is named the same as **your version of Magento**. For example, if you are using *Magento 2.1.0,* navigate inside the folder named `Magento 2.1.x`.

   Inside selected folder you will find `Theme.zip` archive which contains theme files for your version of Magento.

3. Using your FTP client upload `Theme.zip` file **into the root directory of your Magento** installation (the root directory of Magento is the folder which contains files, such as `index.php` and the directories `app`, `bin`, `lib`, `pub` and more).

4. Unzip `Theme.zip` file. It contains two directories: `app`, `pub`. Magento already contains the same directories – **do not remove those existing directories**. Directories extracted from `Theme.zip` have to be **merged** with existing directories of Magento. Approve, if your system asks you whether you want to merge directories.

> **Note:**
>
> this will not overwrite any core files of Magento, this will only add the theme to Magento. So if you see a message that those directories already exist on the server, you can confirm that you want to merge directories.

In case you can't unzip files on your server:

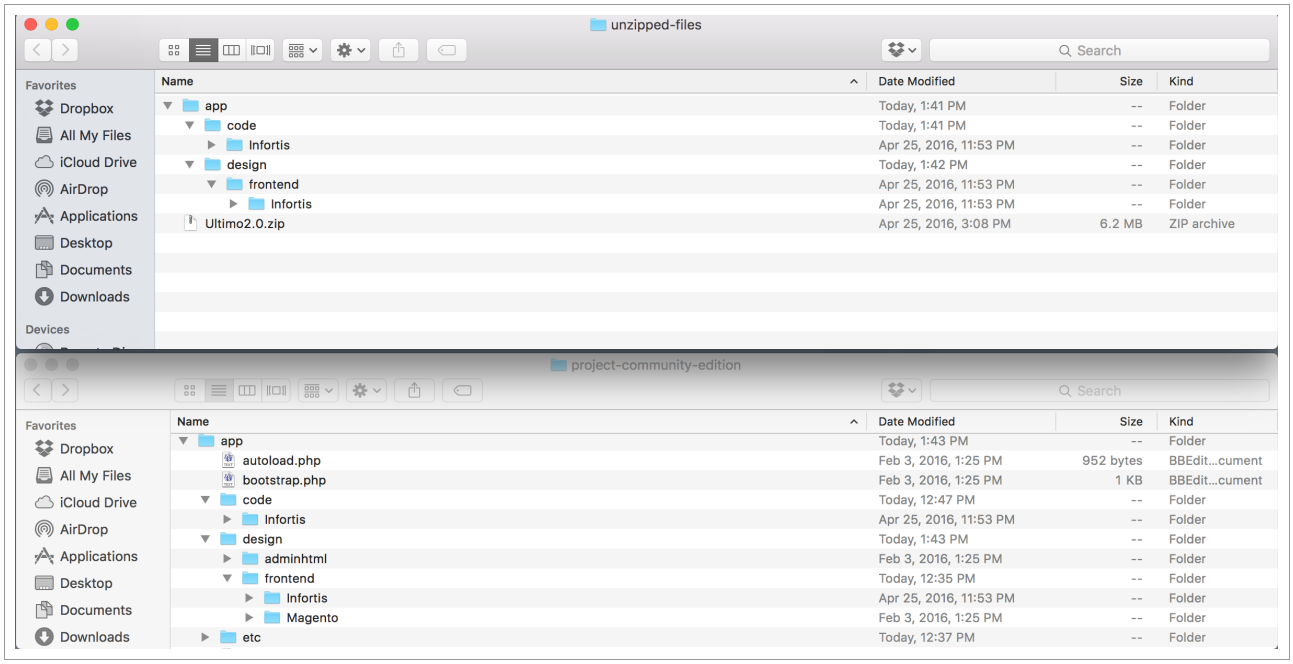
Alternatively, you can unzip `Theme.zip` archive on your computer, and upload its content (two directories: `app`, `pub`) to your Magento's root directory. But note that files can get broken if you send them via FTP without zipping. So **it is highly recommended to upload zipped files**.

5. **Set correct file permissions and ownership** for all the theme files and folders which you uploaded to your server.

> **Note:**
>
> file system ownership and permissions may differ depending on server. In case of any doubts, please contact your hosting provider. Refer to Magento documentation for more details: *http://devdocs.magento.com/guides/v2.0/install-gde/prereq/file-sys-perms-over.html*

6. Verify if all folders are on their place. Inside `Theme.zip`, you can see two top level folders: `app/code/Infortis` and `app/design/frontend/infortis`.

   After extracting `Theme.zip` in the root directory of your Magento, the `app/code/Infortis` folder should be visible inside the `app/code` folder of your Magento installation. And the `app/design/frontend/infortis` folder should be visible inside the `app/design/frontend` folder of your Magento installation:



When you've complete this step, you'll be ready to move on to the next step.

3.2.4 Enabling the Ultimo modules

Next, we need to enable the modules. In Magento 2, this requires the use of the `bin/magento` command line application.

> **Note:**
>
> This is the same command which you have used to install Magento (here's the link to Magento documentation: *http://devdocs.magento.com/guides/v2.0/install-gde/install/cli/install-cli-subcommands.html*).

1. **Enable Ultimo modules**

   Run the following command to enable each Ultimo module:

```
bin/magento module:enable Infortis_Base Infortis_Brands
Infortis_Cgen Infortis_Dataporter Infortis_Infortis
Infortis_Ultimo Infortis_UltraMegamenu
Infortis_UltraSlideshow
```

> **Important:**
>
> place the entire command in **one line** before pasting it in the command line.
>
> When you copy the command from this PDF document, the copied command will be divided in multiple lines (because of text formatting in PDF). So you need to edit it properly before pasting it in the command line.
>
> The easiest way: copy the command, paste it in a text editor, remove new lines to make sure the entire command is in one line. Then you can paste it to your command line and execute.

After executing the command, you will see messages similar to the following:

```
$ php bin/magento module:enable Infortis_Base Infortis_Brands Infortis_Cgen Infortis_Dataporter
Infortis_Infortis Infortis_Ultimo Infortis_UltraMegamenu Infortis_UltraSlideshow

The following modules have been enabled:
- Infortis_Infortis
- Infortis_Brands
- Infortis_Cgen
- Infortis_Dataporter
- Infortis_Base
- Infortis_UltraSlideshow
- Infortis_UltraMegamenu
- Infortis_Ultimo

To make sure that the enabled modules are properly registered, run
'setup:upgrade'.
Cache cleared successfully.
Generated classes cleared successfully. Please run the 'setup:di:compile' command to generate
classes.
Info: Some modules might require static view files to be cleared. To do this, run
'module:enable' with the --clear-static-content option to clear them.
```

2. Run setup upgrade scripts

Then, after enabling the modules, you need to tell Magento it should run its setup upgrade scripts. Just run the following standard Magento command:

```
bin/magento setup:upgrade
```

You will see messages similar to the following:

```
$ php bin/magento setup:upgrade

Cache cleared successfully
File system cleanup:
/path/to/magento/var/generation/Composer
/path/to/magento/var/generation/Magento
/path/to/magento/var/generation/Symfony
Updating modules:
Schema creation/updates:
//...

Module 'Infortis_Infortis':
Module 'Infortis_Brands':
Module 'Infortis_Cgen':
Module 'Infortis_Dataporter':
Module 'Infortis_Base':
Module 'Infortis_UltraSlideshow':
Module 'Infortis_UltraMegamenu':

//...

Schema post-updates:
Module 'Infortis_Infortis':
Module 'Infortis_Brands':
Module 'Infortis_Cgen':
Module 'Infortis_Dataporter':
Module 'Infortis_Base':
Module 'Infortis_UltraSlideshow':
Module 'Infortis_UltraMegamenu':
Module 'Infortis_Ultimo':

//...

Data install/update:
Module 'Infortis_Infortis':
Module 'Infortis_Brands':
Module 'Infortis_Cgen':
Module 'Infortis_Dataporter':
Module 'Infortis_Base':
Module 'Infortis_UltraSlideshow':
Module 'Infortis_UltraMegamenu':
Module 'Infortis_Ultimo':

//...

Please re-run Magento compile command
```

After performing the above two steps, you'll be ready to move on to the next step – generate Magento's static files.

3.2.5 Generating Magento's static files

Magento 2 introduces the concept of "static", or "generated/compiled" front end files. Magento modules and themes store their CSS, Javascript, and other front end related files in the module folders, and generating/compiling them will move these files to Magento's web accessible folder.

Generating static content is simple – just run the following Magento command:

```
bin/magento setup:static-content:deploy
```

You will see messages similar to the following:

```
$ php bin/magento setup:static-content:deploy

Requested languages: en_US
=== frontend -> Magento/blank -> en_US ===
...
Successful: 2396 files; errors: 0
---

=== frontend -> Magento/luma -> en_US ===
...
Successful: 2469 files; errors: 0
---

=== frontend -> Infortis/base -> en_US ===
...
Successful: 2495 files; errors: 0
---

=== frontend -> Infortis/ultimo -> en_US ===
...
Successful: 2499 files; errors: 0
---

=== adminhtml -> Magento/backend -> en_US ===
...
Successful: 1854 files; errors: 0
---

=== Minify templates ===
...
Successful: 875 files modified
---

New version of deployed files: 1462825470
```

After the process finishes running, you can see the generated files in the following folder:

```
pub/static/frontend/Infortis/
```

You're now ready to move on to the next step.

3.2.6 Enabling the theme

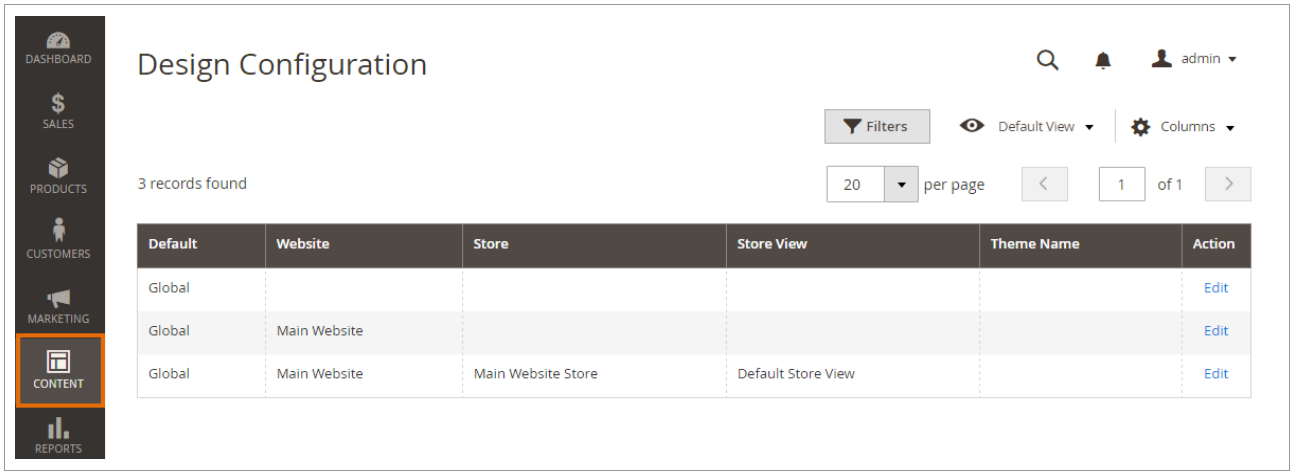
Ultimo is now installed on your Magento system.

> **Note:**
>
> at this point, you need to disable **maintenance mode** in your Magento (if you enabled it earlier). To do this you need to remove file `.maintenance.flag` which you created inside Magento `var` folder. After that, you will be able to log in to the admin panel again.

The next, and final step, is to configure Magento to use the Ultimo theme.

If you're using Magento 2.1 or later:

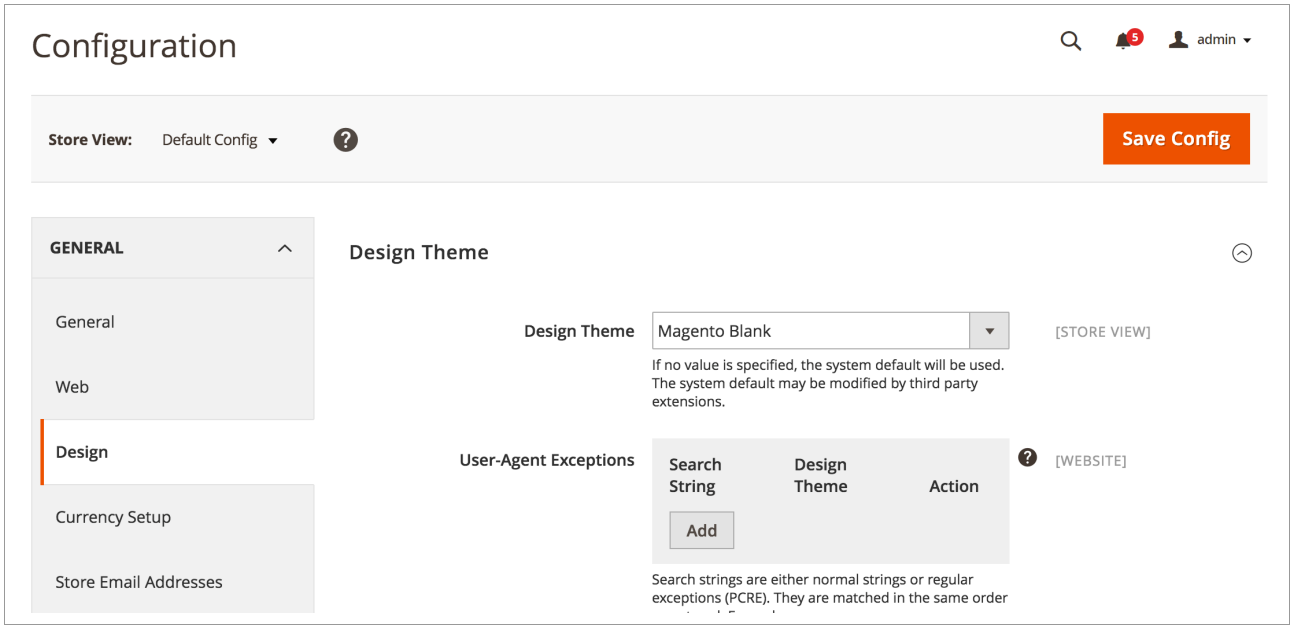
1. In Magento's backend, navigate to *`Content > Design > Configuration`*. You will see a grid with the available configuration scopes. For example:



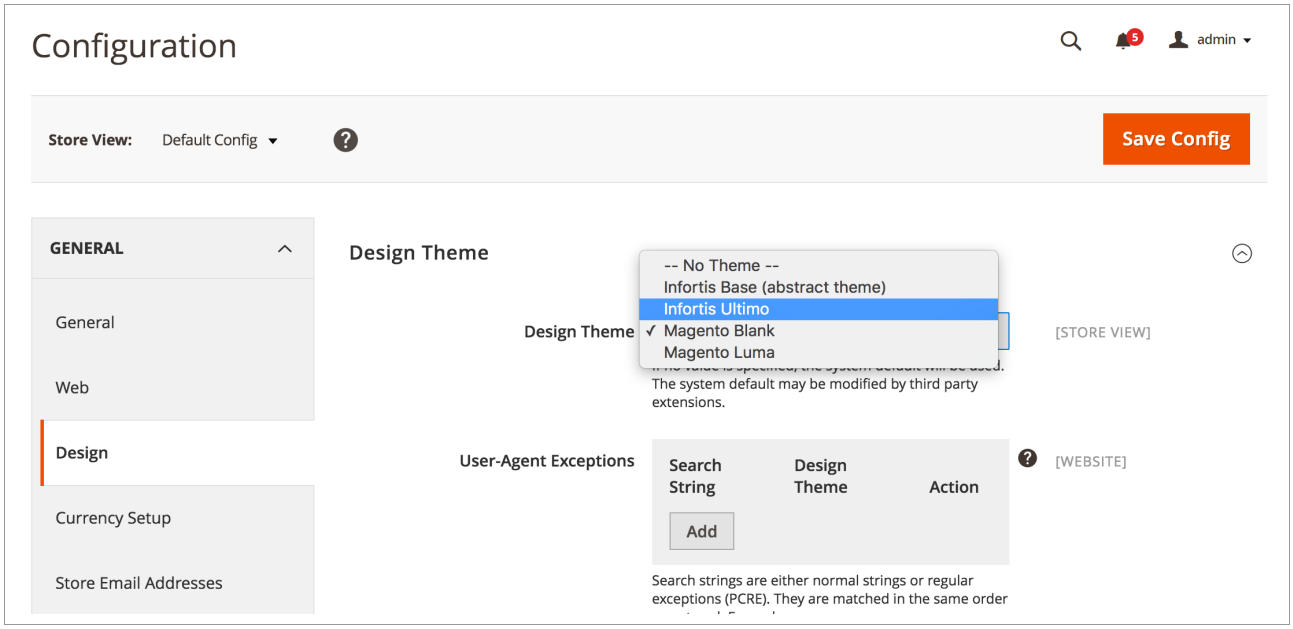
2. In the configuration record corresponding to your store view, click *Edit*.
3. On the *Default Theme* tab, in the *Applied Theme* drop-down, select `Infortis Ultimo`.
4. Click *Save* button.
5. If caching is enabled, navigate to *System > Cache Management* and **flush Magento cache**.
6. To see your changes applied, reload the store front pages.

If you're using Magento 2.0.x:

1. In Magento's backend, navigate to *Stores > Configuration > General > Design*:



2. Select `Infortis Ultimo` from the *Design Theme* drop down menu:



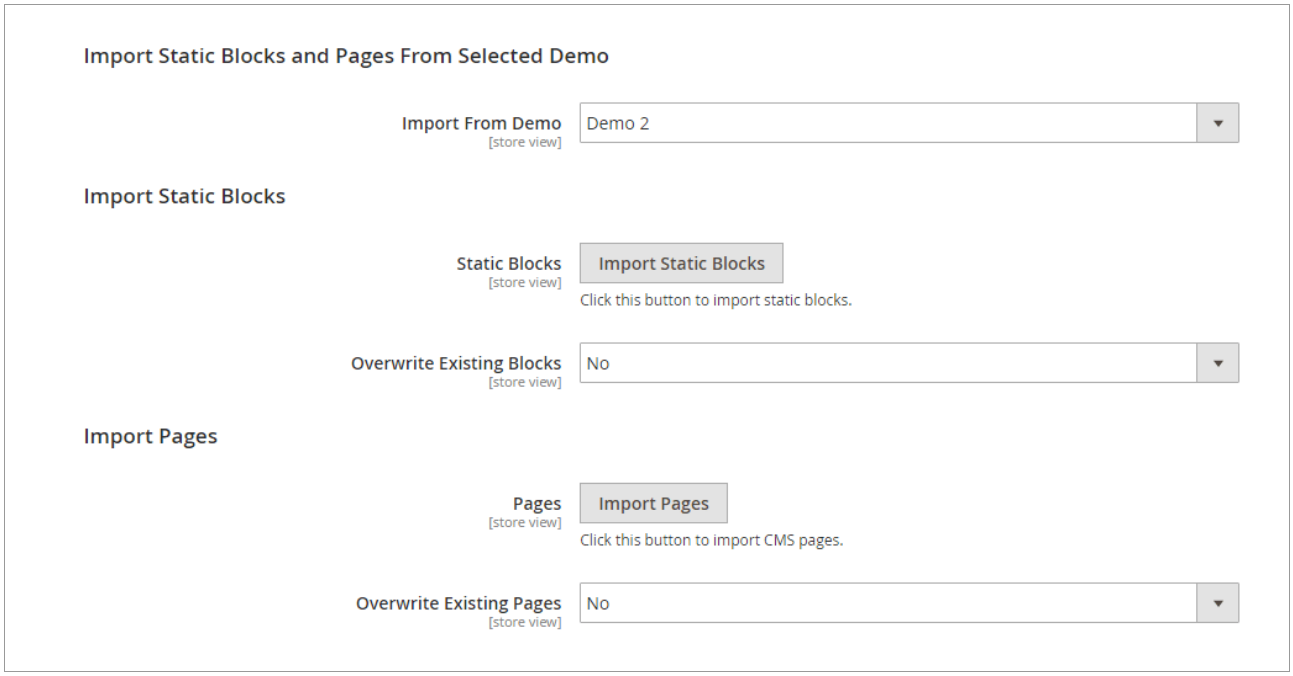
3. and click the `Save Config` button.

4. If caching is enabled, navigate to `System > Cache Management` and **flush Magento cache**.

After performing the above steps, Ultimo will be installed and enabled on your system. To see your changes applied, reload the store front pages.

3.2.7 Installing CMS blocks and pages

Ultimo ships with a number of useful CMS pages and blocks. To import these items into your system, navigate to `Stores > Configuration > Infortis Themes - Configuration > Theme Settings` and open the **Data Import** tab:



Refer to chapter **7.11 Data Import** for more details about the import procedure.

3.3 Problems after installation

> **Important:**
>
> you can **disable Magento cache** before you start configuring your store. Otherwise you will need to flush the cache in order to see any changes you have made in the configuration.

If you encounter any problems after installation: flush Magento cache, log out from the admin panel, and log in again. Check again the installation steps and make sure you didn't omit any steps.

What can cause problems:

a) Magento cache (or external cache) was not flushed after installation of the theme.

b) Some of the theme files were not uploaded or are broken. Re-upload original theme files to your server (overwrite existing files) and flush the cache. Note that files can get broken if you send them in large numbers via FTP without zipping. So it is highly recommended to upload zipped files.

c) File permissions issue – refer to **13.1.4 File permissions** for more details.

d) You have some third-party extensions which change/break theme's default behavior or cause some conflicts. Disable all extensions and flush the cache, then enable extensions one by one to check which one is causing problems. Refer to **2.3 Third-party extensions** for more details about third-party extensions.

e) You edited theme files and accidentally removed or changed something in the files. Re-upload original theme files to your server (overwrite existing files) and flush the cache.

f) You have customized the theme by creating custom child theme and your modifications override or break the default behavior of the theme.

g) You have some conflicts with custom extensions or with other themes installed in your Magento.

Also check your log files in `var/log/` for more information about possible errors.

Please refer to the following chapter: **13. Troubleshooting** for more information about the most common problems.

4. Magento configuration

The following chapter describes some of the basic Magento settings and features.

Please note that as authors of this theme we provide support only for the issues related strictly to the theme. Support for Magento configuration/installation/maintenance etc. is beyond the scope of our support. This chapter should be only treated as a starting point, more information about Magento can be found in Magento documentation: **2.1.2 Magento community**

4.1 Caching system

Magento's cache management system is an easy way to improve the performance of your site. Whenever a cache needs to be refreshed, a notice appears at the top of the workspace to guide you through the process. Follow the link to **Cache Management**, and refresh the invalid caches.

To access the cache management screen, you can also use the link in the admin panel menu: `System > Cache Management`.

Always refresh the cache after making changes in the configuration of your Magento/themes/extensions or after making any custom changes in the files.

> **Note:**
>
> for all details about cache, refer to Magento documentation:
> *http://docs.magento.com/m2/ce/user_guide/system/cache-management.html*

4.1.1 Flush cache

Go to `System > Cache Management`. Click the following buttons to completely clear the store cache:

1. Flush Magento Cache
2. Flush Cache Storage

4.1.2 Disable cache

Go to `System > Cache Management`. Click **Select All** link, select **Disable** in **Actions** field

and click **Submit** button.

4.1.3 Enable cache

When the store is configured, you can enable caching system again. Go to `System > Cache Management`. Click **Select All** link, select **Enable** in **Actions** field and click **Submit** button.

4.1.4 Refresh cache

If the cache is enabled, after each configuration change, extension installation, upgrade or uninstallation always refresh the cache. To do this, go to `System > Cache Management`. Click **Select All** link, select **Refresh** in **Actions** field and click **Submit** button.

4.2 Index management

Indexing is how Magento transforms data such as products, categories, and so on, to improve the performance of your storefront. As data changes, the transformed data must be updated or reindexed. Magento has a very sophisticated architecture that stores lots of merchant data (including catalog data, prices, users, stores, and so on) in many database tables. To optimize storefront performance, Magento accumulates data into special tables using indexers.

You can reindex in any of the following ways:

- using a **cron job** (preferred)
- using the **command**: `bin/magento indexer:reindex`, which reindexes selected indexers, or all indexers, one time only

> **Note:**
>
> for more details, refer to Magento documentation:
>
> - *http://docs.magento.com/m2/ce/user_guide/system/index-management.html*
>
> - *http://devdocs.magento.com/guides/v2.0/extension-dev-guide/indexing.html*
>
> - *http://devdocs.magento.com/guides/v2.0/config-guide/cli/config-cli-subcommands-index.html#config-cli-subcommands-index-reindex*

To access the index management screen, go to `System > Index Management`.

> **Important:**
>
> when you edit your categories, products, attributes etc., and your changes are not reflected of frontend (even after flushing the cache), you may need to wait until your data will be reindexed (if you configured your indexes to be refreshed by **cron job**) or you may need to reindex manually from the **command line**.

4.3 How to enable the theme

In Magento you can enable the theme on many levels. You can set one theme for the entire store, but you can also set different themes for different parts of the store (even for selected categories, products or CMS pages).

If you want all your store pages to look the same (to use the same theme), your theme should only be enabled in `Stores > Configuration > Design` (the same as it was described in the installation steps **3. Installation**). You need to remove all theme overrides from categories, products and CMS pages. There are many places where a theme can be enabled. If you have just installed a fresh copy of Magento, those places will be empty. Otherwise, please take a look at these sections of the admin panel:

- These fields should be left empty, do not select any theme here:
  - `Catalog > Manage Categories > [category] > Custom Design > Custom Design`
  - `Catalog > Manage Products > [product] > Design > Custom Design`
  - `Content > Pages > [page] > Design > Custom Theme`

Remember to **flush the cache** after you change design settings.

4.4 Logo

In Magento 2.1 go to: *Content > Design > Configuration > Header*.

In the **Logo Image** field upload your logo image. Magento stores logo images in the `pub/media/logo/` folder. If you have multiple websites and store views in your Magento, you will see there more subfolders.

If logo image is not uploaded via Magneto admin, Magento will display logo imagewhich is located here: `app/design/frontend/Infortis/ultimo/web/images/logo.png`. You can replace that default file with your own file.

4.4.1 Sticky logo

You can also display site logo in the sticky header. To display sticky logo:

1. Enable sticky logo under *Stores > Configuration > Theme Settings > Header*.

2. Upload your sticky logo image to **the same folder where your main logo is stored**.

3. Name your sticky logo file the same as the main logo file, but add `_sticky` suffix to the file name.

   For example:
   if your main logo file is `logo.png`, then your sticky logo file has to be named like this: `logo_sticky.png`

4. Flush the cache and refresh the web browser cache.

4.5 Categories

> **Note:**
>
> All information about categories can be found in Magento documentation: *http://docs.magento.com/m2/ce/user_guide/catalog/categories.html?Highlight=categories*

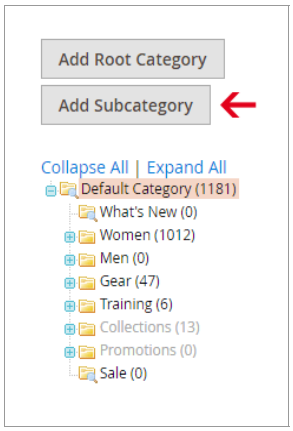
To create and edit categories in Magento, navigate to `Products > Categories` section of the admin panel. After installation of Magento your category tree is almost empty, there is only the **root category** (named `Default Category`).

The root category is not displayed in the frontend of the store – **that's why your main menu is empty** just after installing Magento. You need to add new top-level categories (also called subcategories because they are descendants of the root category) to the root category.

> **Important:**
>
> all categories have to be **descendants of the root category**. Do not add categories outside of the root category tree. A category which is not a descendants of the root category will not be displayed in the menu on frontend.

To add a new category:

1. Select the root category by clicking on it. By selecting the root category you're "telling" Magento to add your new category to the existing catalog.
2. Click the `Add Subcategory` button.
3. A new form will be displayed – fill it in.

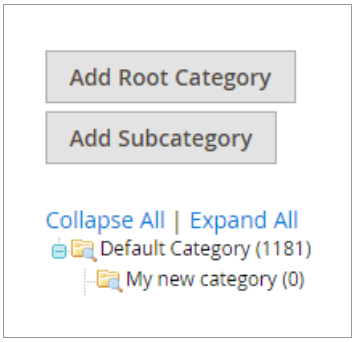


Fill in all the required fields. Especially enable fields `Enable Category` and `Include in Menu`, and then save the category.

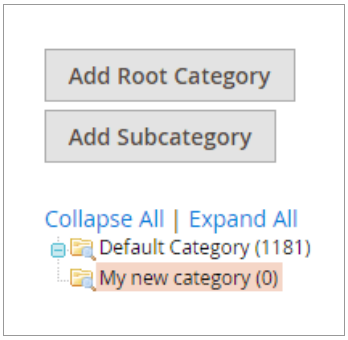
This will create a new category which will be displayed as a top-level category in the main menu on frontend.

Remember that this new category **has to be a descendant of the root category** – it has to be one level lower than the root category:

Correct:



Not correct:



If the category was added correctly but it's not displayed on frontend, you may need to flush the cache (**4.1 Caching system**) and reindex your Magento (**4.2 Index management**).

4.5.1 Category info

For each category you can add some basic information which will be displayed on category page (above the products). Navigate to `Products > Categories` and select a category:

- `Description` – category description. You can enter text or HTML.
- `Category Image` – main category image.

4.5.2 Find category ID

How to find out an internal identifier (ID) of a category through Magento admin panel? Navigate to *Products > Categories* and click desired category. Look at the URL of the page in your web browser. It can look like this:

```
http://example.com/admin/catalog/category/edit/id/21/
```

You can find the category ID at the end of the URL. In this example it's **21**.

4.5.3 Landing pages

You can show your customers a landing page instead of the standard product listing page when they select the category. You can do so by enabling static block on your category page (block will be displayed above category products). Navigate to *Products > Categories*, select a category, open **Display Settings** tab and select appropriate value in the **Display Mode** field. In the drop-down with CMS blocks, select which block you would like to display for selected category. The list will include all existing static blocks.

There's also another (more flexible) method to add custom static blocks to categories. Go to *Content > Widgets* and add a new widget instance. Through widgets you can add your static blocks not only to selected categories but also **to selected areas (containers) of the page**. For details refer to chapter **4.8.1 Add blocks to selected pages**.

4.5.4 Depth of the menu

In the main menu you can limit number of displayed category levels. To do that, go to *Stores > Configuration > Catalog > Category Top Navigation* and specify the depth in **Maximal Depth** field.

For example, if you want to display only three levels of categories, enter `3` in **Maximal Depth** field and save the configuration. From now on, only three levels of categories will be displayed in the main menu.

4.5.5 Layered Navigation ("Shop By" filters)

Magento's Layered Navigation allows your customers to filter down products using any attribute that is set as filterable for Layered Navigation. The Layered Navigation menu will display in the left column of your category pages, and will contain all of the attributes which are filterable.

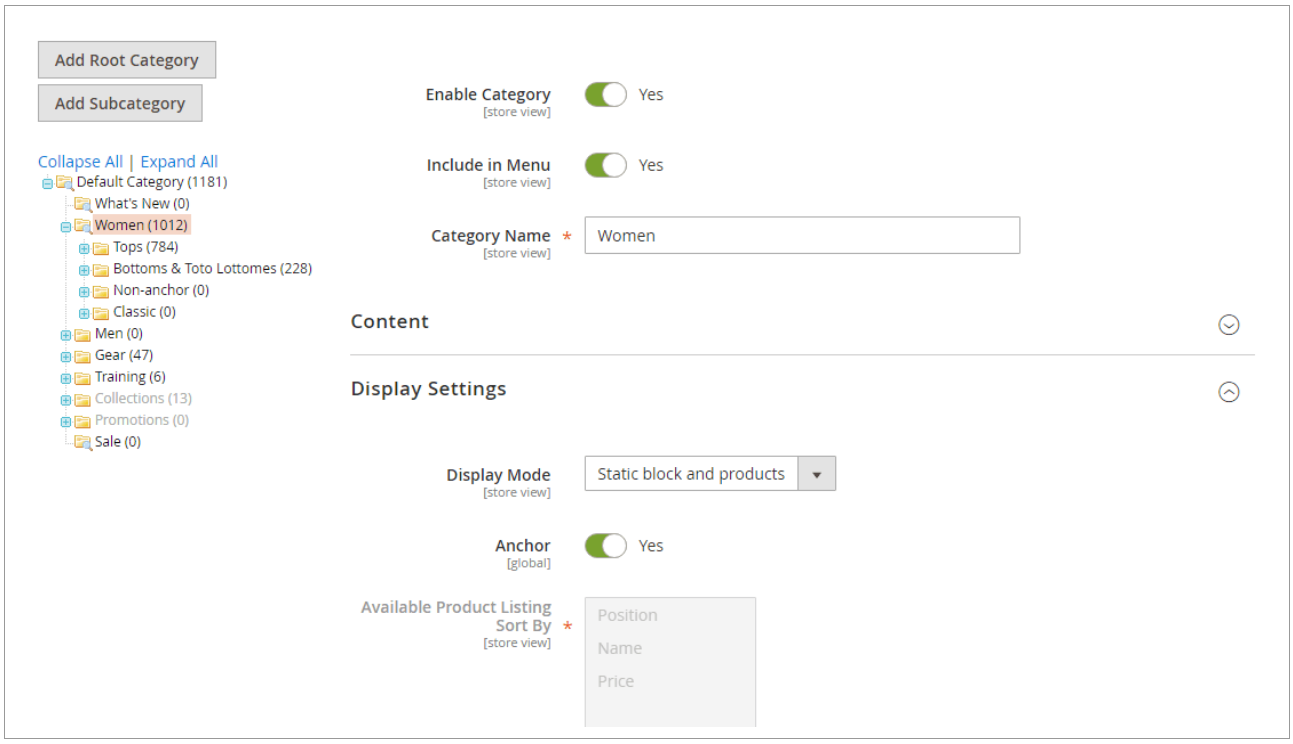
You can configure each attribute to be displayed in Layered Navigation: Use in Layered Navigation field in `Stores > Attributes > Products > [select attribute]`. Please refer to Magento documentation for more details:

- *http://docs.magento.com/m2/ce/user_guide/catalog/navigation-layered.html*

4.5.6 Categories inside Layered Navigation

> **Note:**
>
> This theme adds ability to display a list of categories in the category view (independent from Magento's Layered Navigation). For more details refer to **12.1.3 Sidebar Menu**.

You can display categories **inside** Magento's Layered Navigation block ("Shop by" block) at the top of the left sidebar in category view. To display selected category in the Layered Navigation, go to `Products > Categories`, click selected category and set **Anchor** field (in **Display Settings** tab) to **Yes:**



From now on this category will be displayed in the Layered Navigation. If the category is not displayed in the frontend, you may need to flush the cache (**4.1 Caching system**) and

reindex your Magento (**4.2 Index management**).

4.6 Products

4.6.1 "Only X left" stock indicator

There is a quantity indicator displayed on the product page. It shows the quantity of items left in stock if the stock level is equal or below the threshold specified in the admin panel. The threshold can be set in **Only X left Threshold** field in `Stores > Configuration > Inventory`.

> **Note:**
>
> The quantity indicator is displayed only for simple products, it's not displayed for configurable products.

For example, if you set the option to 20, all **simple products** with quantity equal or less than 20 will have the `Only X left` (where `X` is the quantity of products) displayed on product page.

4.7 Attributes

Attributes are the building blocks of your product catalog, and describe specific characteristics of a product. Attributes determine the type of input control that is used for product options, provide additional information for product pages, and are used as search parameters and criteria for layered navigation (filters), product comparison reports, and promotions.

> **Note:**
>
> all the details about attributes in Magento can be found in Magento documentation:
>
> - *http://docs.magento.com/m2/ce/user_guide/stores/attributes.html*
>
> - *http://docs.magento.com/m2/ce/user_guide/catalog/product-attributes.html*

You can create as many attributes as you need to describe the products in your catalog. Refer to chapter **4.7.1 Attribute Management** for more details.

Attributes are grouped in **attribute sets**. Refer to chapter **4.7.2 Attribute Sets** for more details.

4.7.1 Attribute Management

To create or edit attributes, go to `Stores > Attributes > Products`.

- To edit existing attribute, select an attribute from the list.
- To create a new attribute, click the `Add New Attribute` button.

There are many types of attributes. Each attribute has many properties to configure, but in most cases you will want to leave the default values. If you're not sure how to configure each property, then all the necessary information can be found in Magento documentation: *http://docs.magento.com/m2/ce/user_guide/stores/attribute-product-create.html*

> **Important:**
>
> after you create an attribute, you need to add it to the **attribute set** which you use for your products. Otherwise your attribute will not be visible on the product edit page. Refer to chapter **4.7.2 Attribute Sets**.

> **Important:**
>
> after making any changes in attributes and attribute sets, you will need to flush the cache (**4.1 Caching system**) and reindex your Magento (**4.2 Index management**).

In the chapters below you can find a few examples of how to create attributes of different types.

4.7.1.A Examples of attributes

4.7.1.A.1 Attribute: manufacturer

One of the most useful attribute types is the **dropdown** type which lets you define a list of values which will be available when creating new products. Here's an example of the `manufacturer` attribute which stores a list of product brands.

Go to *`Stores > Attributes > Products`* and click the *`Add New Attribute`* button. Fill in the form as shown on the screenshots below.

- Enter `manufacturer` in the *`Attribute Code`* field.
- The *`Catalog Input Type for Store Owner`* field needs to be set to `Dropdown`.
- In the *`Manage Options (Values of Your Attribute)`* table you need to define values of the attribute – in this case the names of brands which you want to assign to your products.

ATTRIBUTE INFORMATION

Properties

Manage Labels

Storefront Properties

Attribute Properties

Default Label * Manufacturer

Catalog Input Type f or Store Owner: Dropdown

Values Required: No

Manage Options (Values of Your Attribute)

Is Default	Admin*	Default Store View	
	Sample		Delete
	Apple		Delete
	Samsung		Delete

Add Option

Advanced Attribute Properties

Attribute Code: manufacturer

This is used internally. Make sure you don't use spaces or more than 30 symbols.

Scope: Global

Declare attribute value saving scope.

Unique Value: No

Not shared with other products.

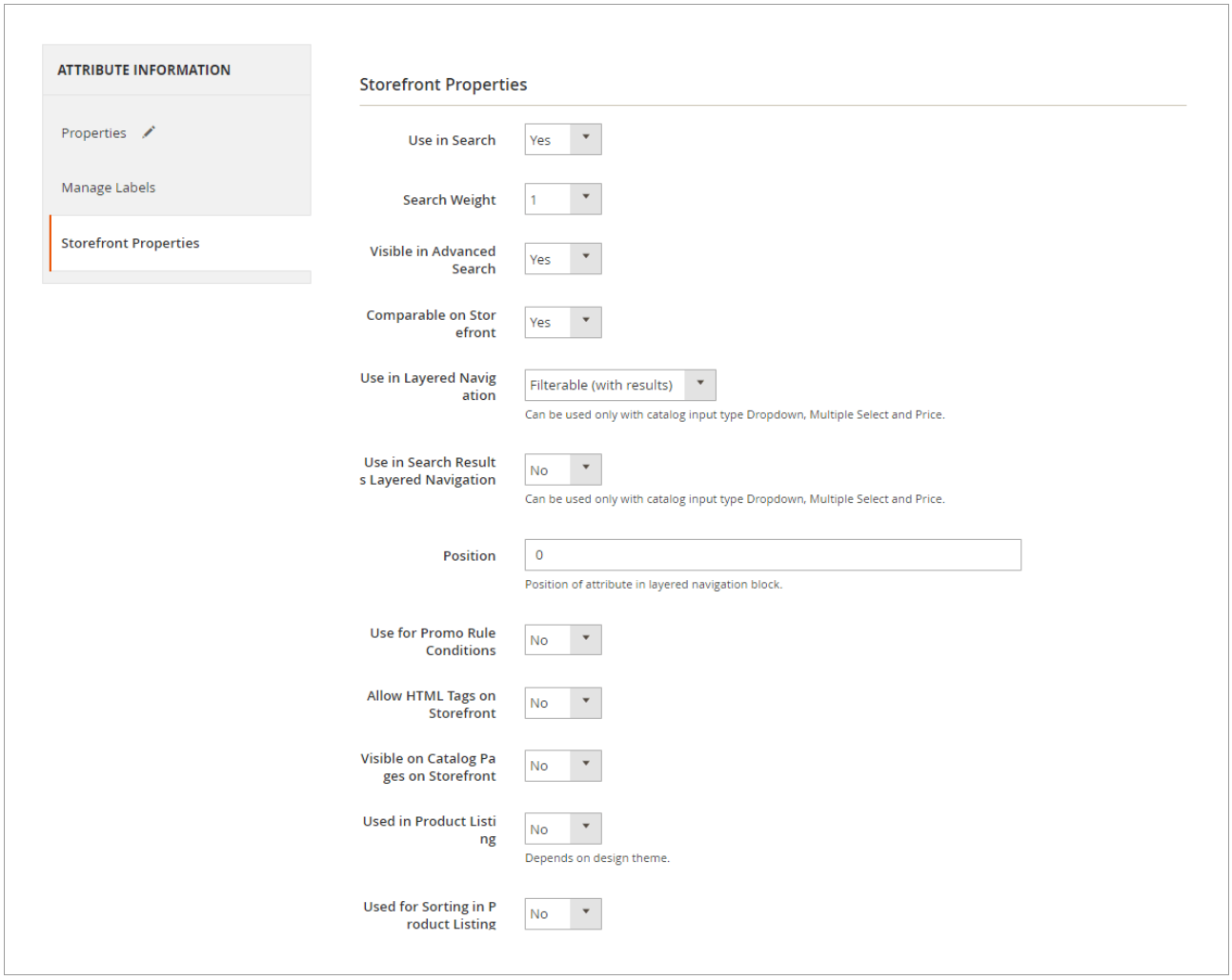
Input Validation for Store Owner: None

Add to Column Opti ons: Yes

Select "Yes" to add this attribute to the list of column options in the product grid.

Use in Filter Options: Yes

Select "Yes" to add this attribute to the list of filter options in the product grid.



Remember that after you create the attribute, you need to add it to the **attribute set** which you use for your products. Otherwise the attribute will not be visible on the product edit page.

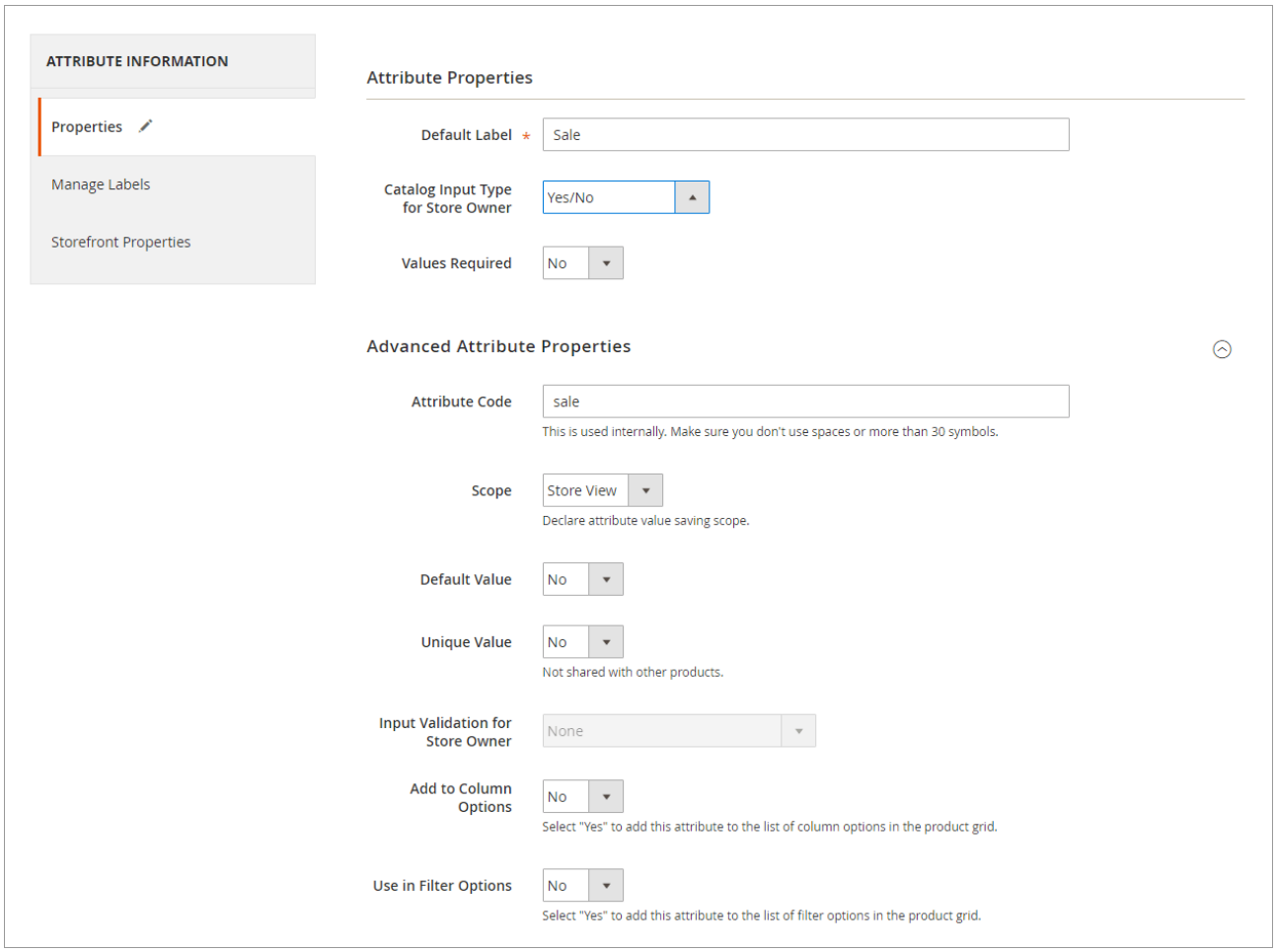
4.7.1.A.2 Attribute: sale

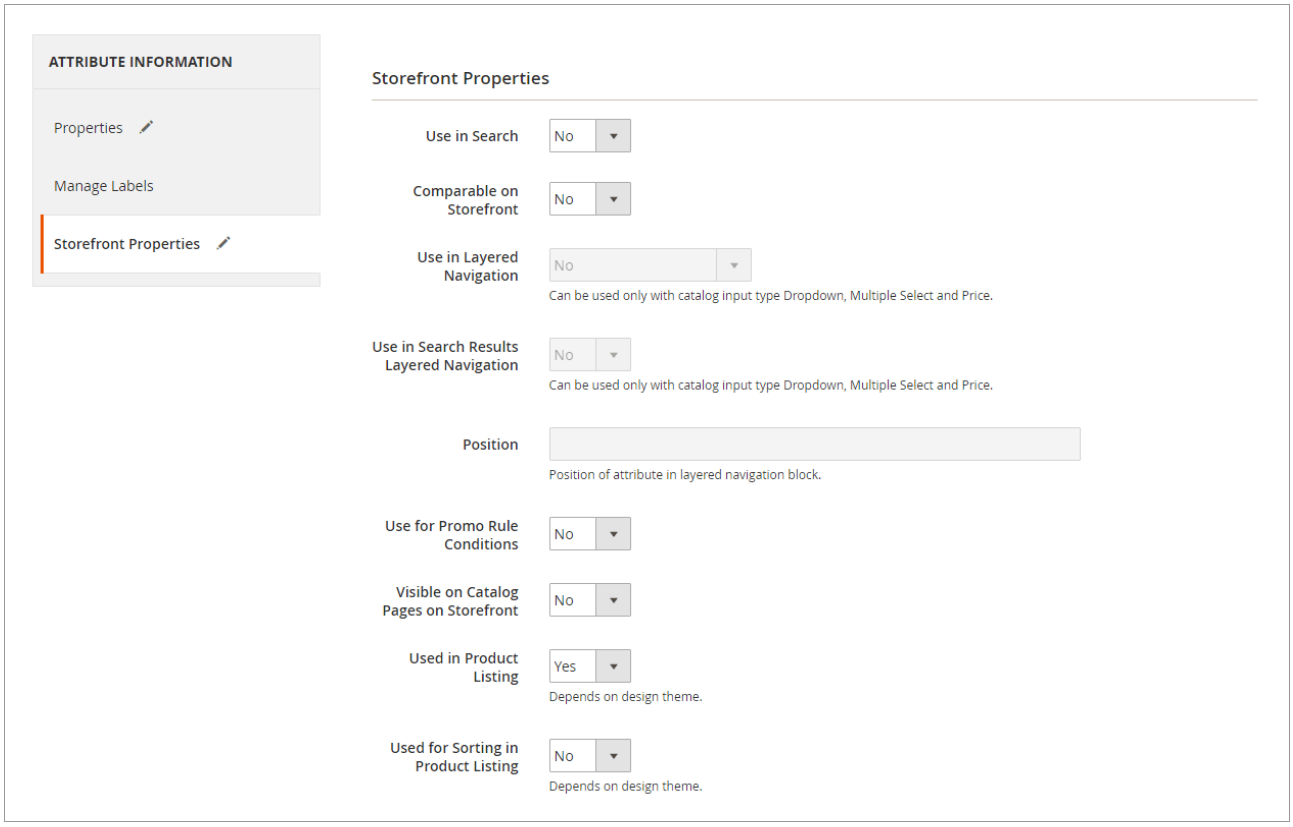
In this example we will create the `sale` attribute which will be used to mark products which should have the "sale" label in product listings. For more details about product labels refer to **7.8 Product Labels**.

Go to *`Stores > Attributes > Products`* and click the *`Add New Attribute`* button. Fill in the form as shown on the screenshots below.

- Enter `sale` in the *`Attribute Code`* field.
- The *`Catalog Input Type for Store Owner`* field needs to be set to `Yes/No`.

- The *Used in Product Listing* field needs to be set to `Yes`. Otherwise the "sale" label wouldn't be visible in product listings.





Remember that after you create the attribute, you need to add it to the **attribute set** which you use for your products. Otherwise the attribute will not be visible on the product edit page.

4.7.1.A.3 Attribute: new

In this example we will create the `new` attribute which will be used to mark products which should have the "new" label in product listings. For more details about product labels refer to **7.8 Product Labels**.

Go to *`Stores > Attributes > Products`* and click the *`Add New Attribute`* button. Fill in the form as shown on the screenshots below.

- Enter `new` in the *`Attribute Code`* field.
- The *`Catalog Input Type for Store Owner`* field needs to be set to `Yes/No`.
- The *Used in Product Listing* field needs to be set to `Yes`. Otherwise the "new" label wouldn't be visible in product listings.

ATTRIBUTE INFORMATION

Properties

Manage Labels

Storefront Properties

Attribute Properties

Default Label * New

Catalog Input Type for Store Owner: Yes/No

Values Required: No

Advanced Attribute Properties

Attribute Code: new

This is used internally. Make sure you don't use spaces or more than 30 symbols.

Scope: Store View

Declare attribute value saving scope.

Default Value: No

Unique Value: No

Not shared with other products.

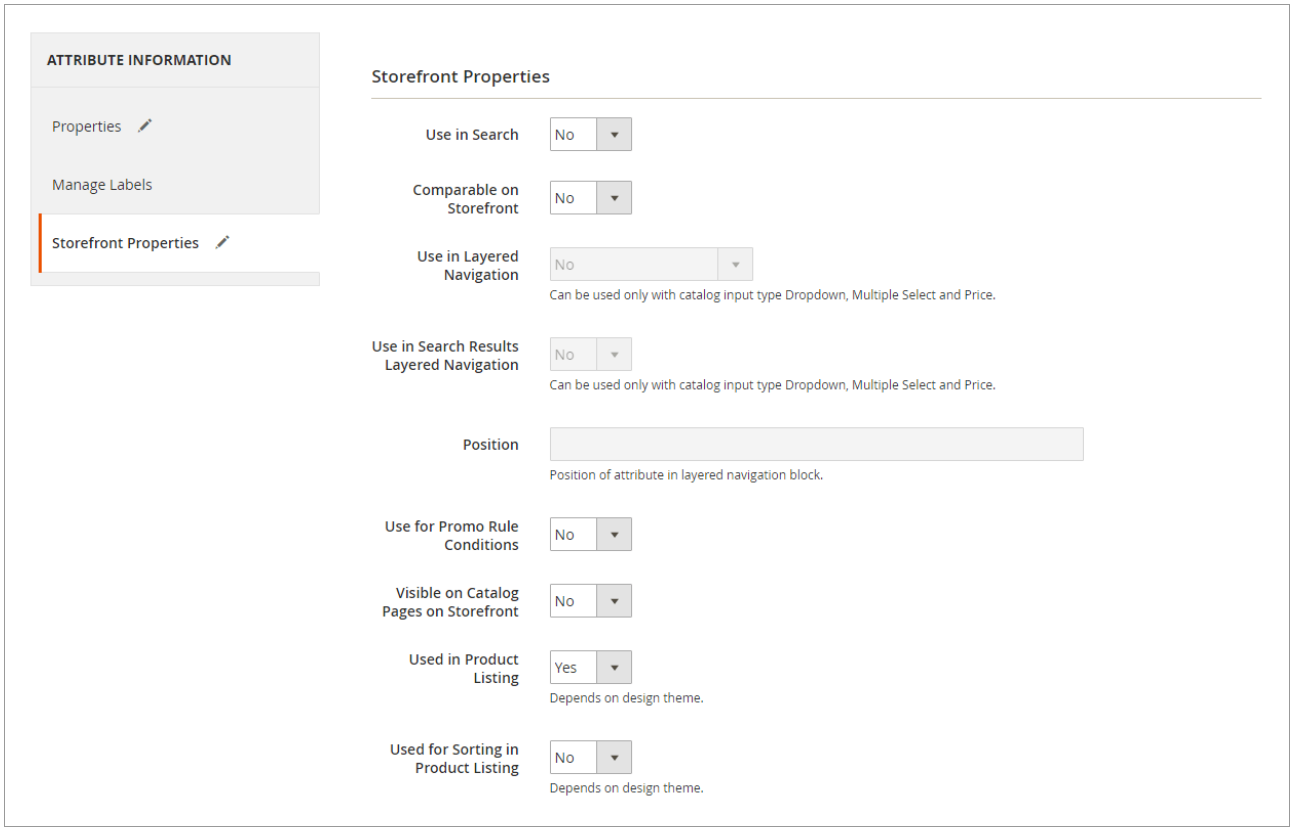
Input Validation for Store Owner: None

Add to Column Options: No

Select "Yes" to add this attribute to the list of column options in the product grid.

Use in Filter Options: No

Select "Yes" to add this attribute to the list of filter options in the product grid.



Remember that after you create the attribute, you need to add it to the **attribute set** which you use for your products. Otherwise the attribute will not be visible on the product edit page.

4.7.1.A.4 Attribute: custom_label

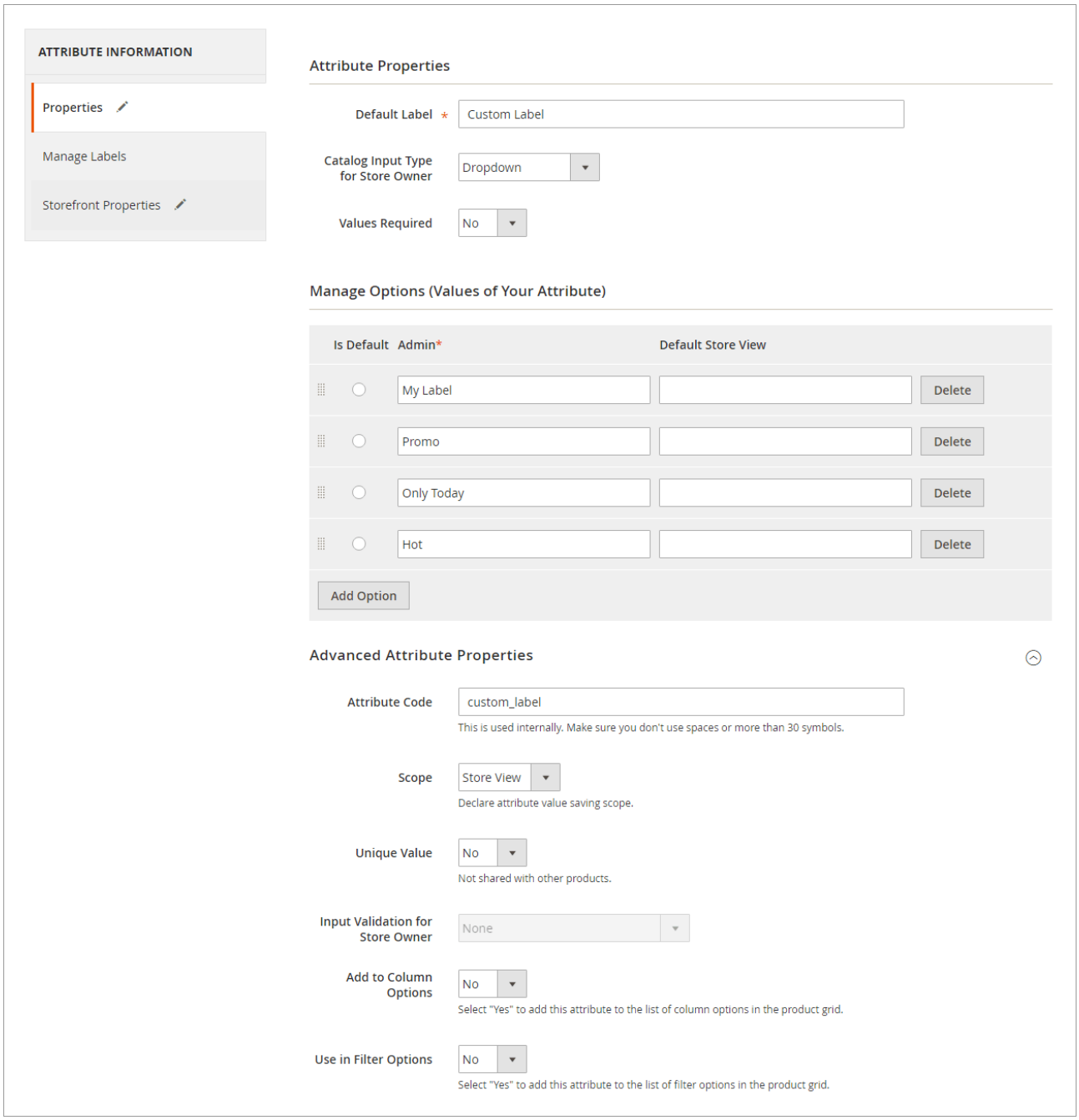
In this example we will create the `custom_label` attribute which will be used to add custom labels to products. For more details about product labels refer to **7.8 Product Labels**.

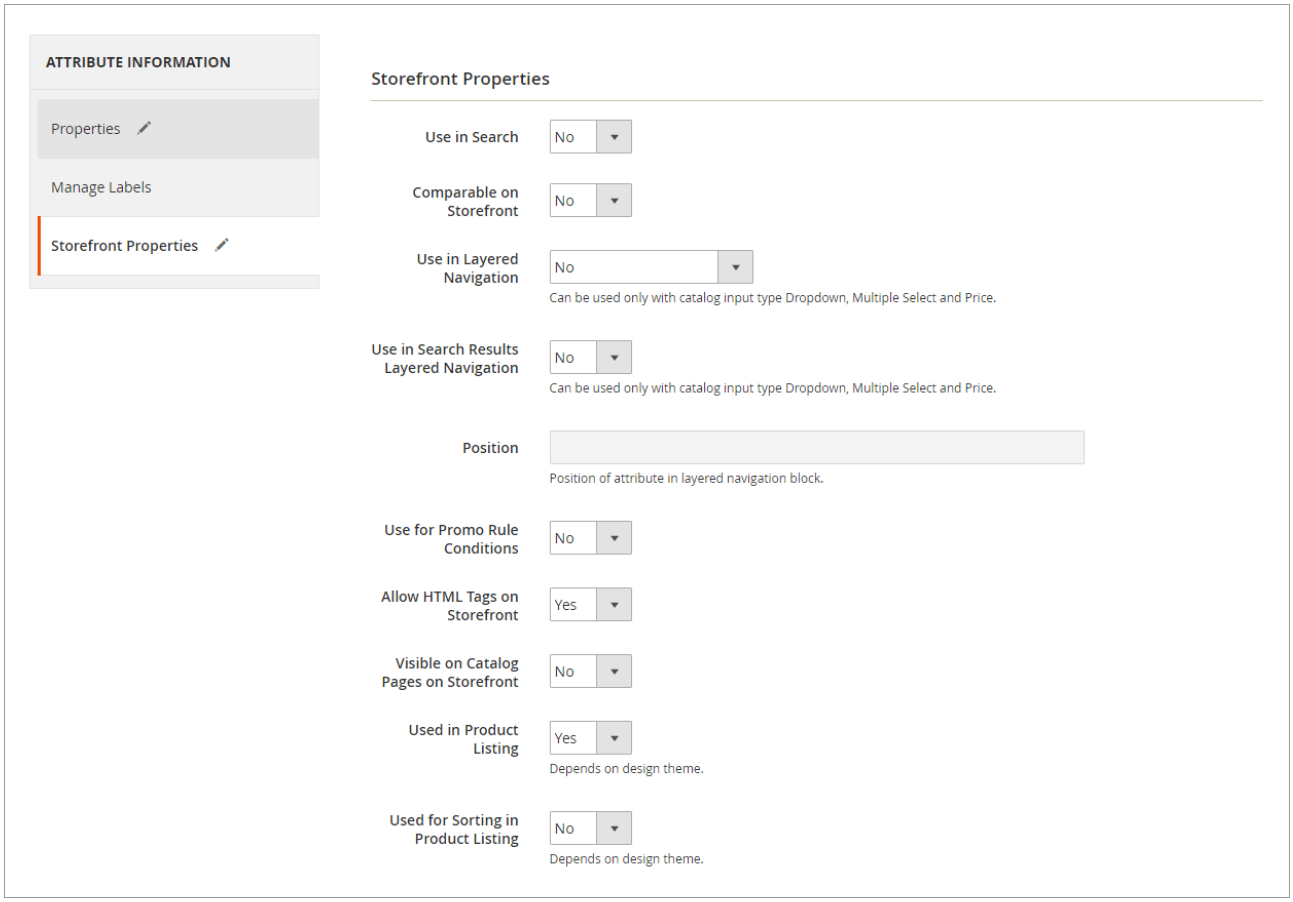
Go to *`Stores > Attributes > Products`* and click the *`Add New Attribute`* button. Fill in the form as shown on the screenshots below.

- Enter `custom_label` in the *`Attribute Code`* field.
- Set the *`Catalog Input Type for Store Owner`* field to `Dropdown`.
- Set the *`Values Required`* field to `No`. Otherwise it wouldn't be possible to not select any value on product edit page.
- In the *`Manage Options (Values of Your Attribute)`* table define values of the

attribute. These values will be displayed on the product labels, define as many as you want. Through this attribute you will be able to assign custom labels to selected products.

- In the `Is Default` column, do not select any value as the default.
- The *Used in Product Listing* field needs to be set to `Yes`. Otherwise the label wouldn't be visible in product listings.





Remember that after you create the attribute, you need to add it to the **attribute set** which you use for your products. Otherwise the attribute will not be visible on the product edit page.

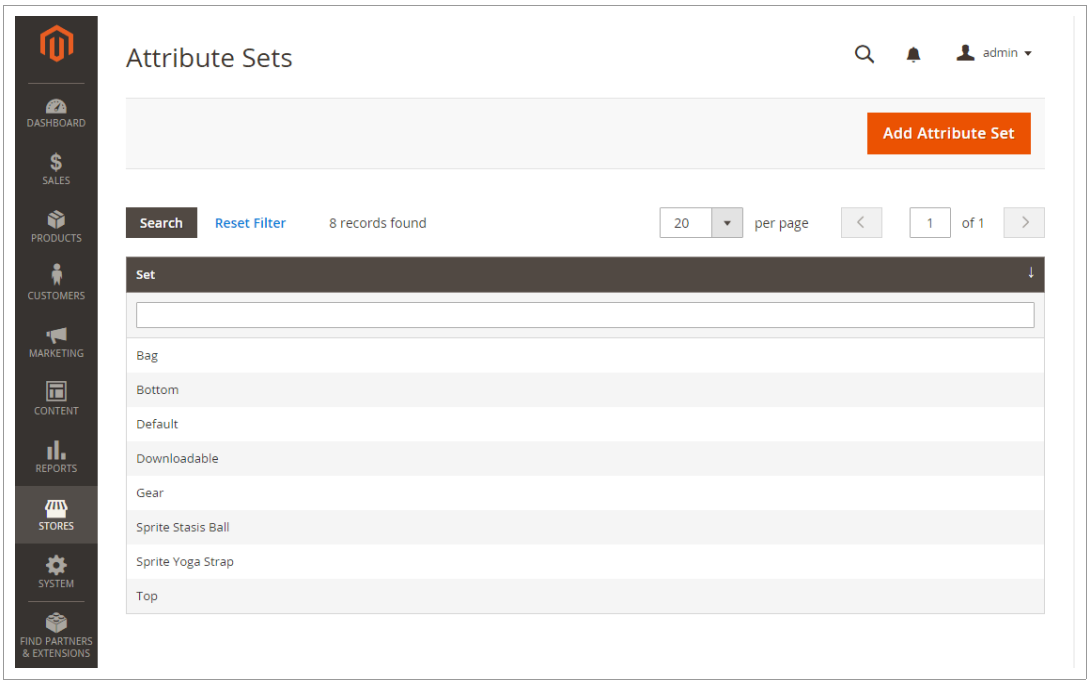
4.7.2 Attribute Sets

One of the first steps when creating a product is to choose the **attribute set** that is used as a template for the product record. The attribute set determines the fields that are available during data entry, and the values that appear to the customer. To manage attribute sets, go to `Stores > Attributes > Attribute Set`.

The attributes are organized into groups that determine where they appear in the product record. Your store comes with an initial attribute set called `default` which includes a set of commonly-used attributes. If you would like to add only a small number of attributes, you can add them to the default attribute set. However, if you sell products that require specific types of information, such as cameras, it might be better to create a dedicated attribute set that includes the specific attributes that are needed to describe the product.

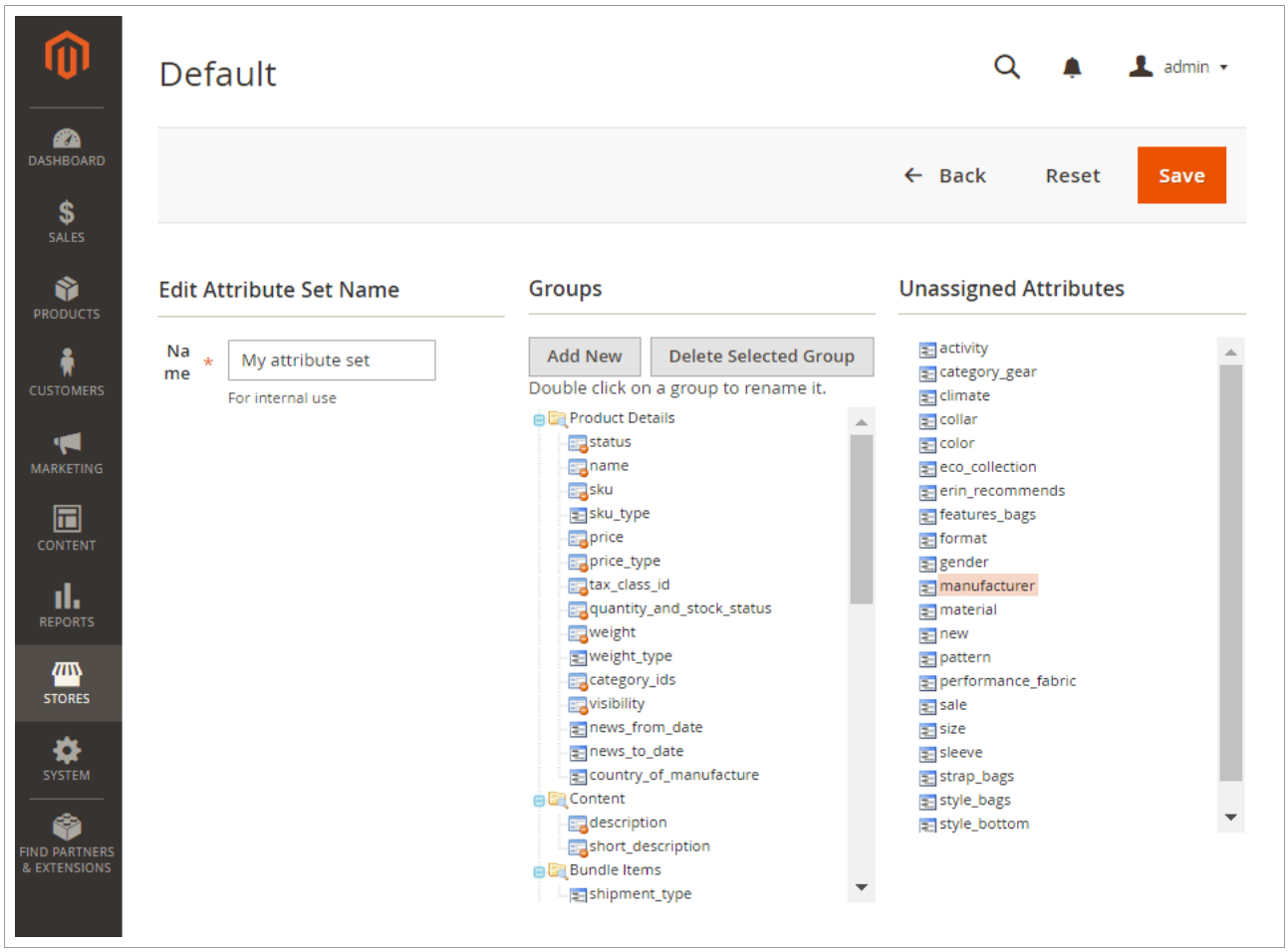
> **Important:** after making any changes in attributes and attribute sets, you will need to flush the cache (**4.1 Caching system**) and reindex your Magento (**4.2 Index management**).

Let's continue an example with the `manufacturer` attribute from the previous chapter. To add your attribute to an attribute set, go to `Stores > Attributes > Attribute Set`:



From the list, select an attribute set. You will see `Unassigned Attributes` section with a

list of available attributes. You can associate attributes to your attribute set by clicking and dragging an attribute into the `Groups` section. To disassociate an attribute, simply drag it back to the `Unassigned Attributes` list.



Click the `Save` button after you finish. You can add your custom attributes to many attribute sets.

Then, go to `Products > Catalog` and open a product. If the product is based on the attribute set which contains your `manufacturer` attribute, then on the product edit page you will see the field with the list of manufacturers.

If you correctly added the `manufacturer` attribute to that attribute set, but on the product edit page you don't see the field with the list of manufacturers, you may need to flush the cache (**4.1 Caching system**) and reindex your Magento (**4.2 Index management**).

4.8 Content Management System (CMS)

4.8.1 Add blocks to selected pages

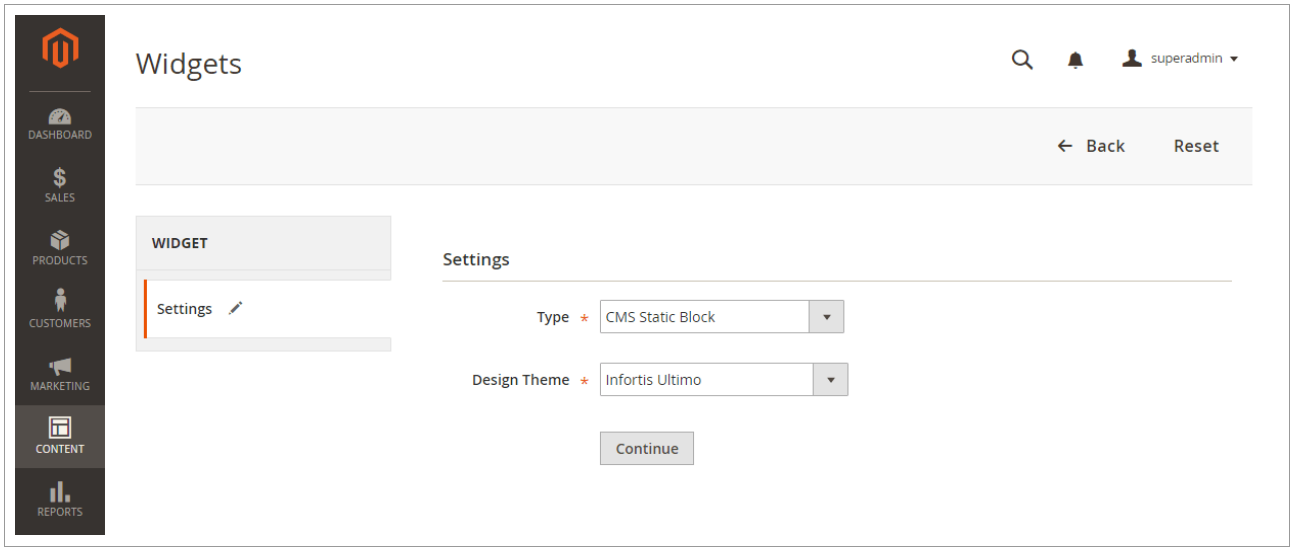
Magento Widget tool (located under `Content > Widgets`) allows users to easily add dynamic content to selected pages. This allows for greater control and flexibility in creating informational and marketing content through administrator tools.

Example 1:

Below you can see an example in which we will add a simple block of content to a product page.

To add custom content to selected place, first create a static block. Static blocks in Magento are simple portions of content that can be displayed throughout the site. For more info about creating a static block, refer to next chapter: **4.8.2 Static blocks**.

When your static block with custom content is ready, go to `Content > Widgets` and add a new widget instance. Widget will add your static block to a selected place on the frontend of your store. Click the **Add New Widget Instance** button. On the next page, you will see the following options:

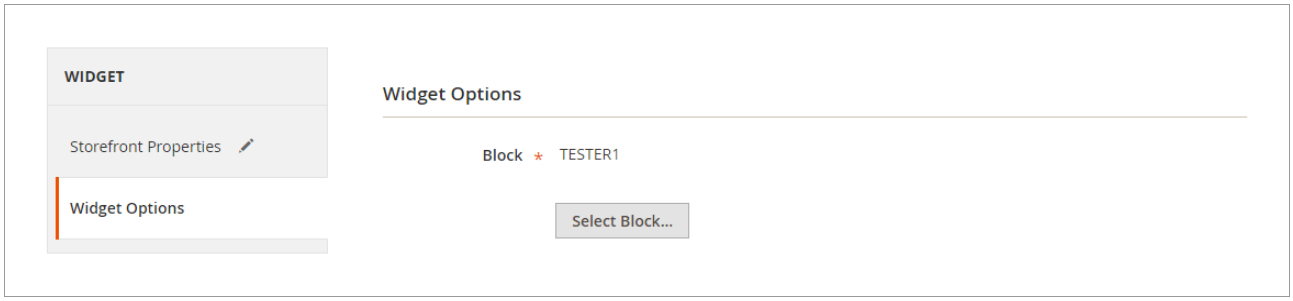


1. Choose the widget type: `CMS Static Block`.

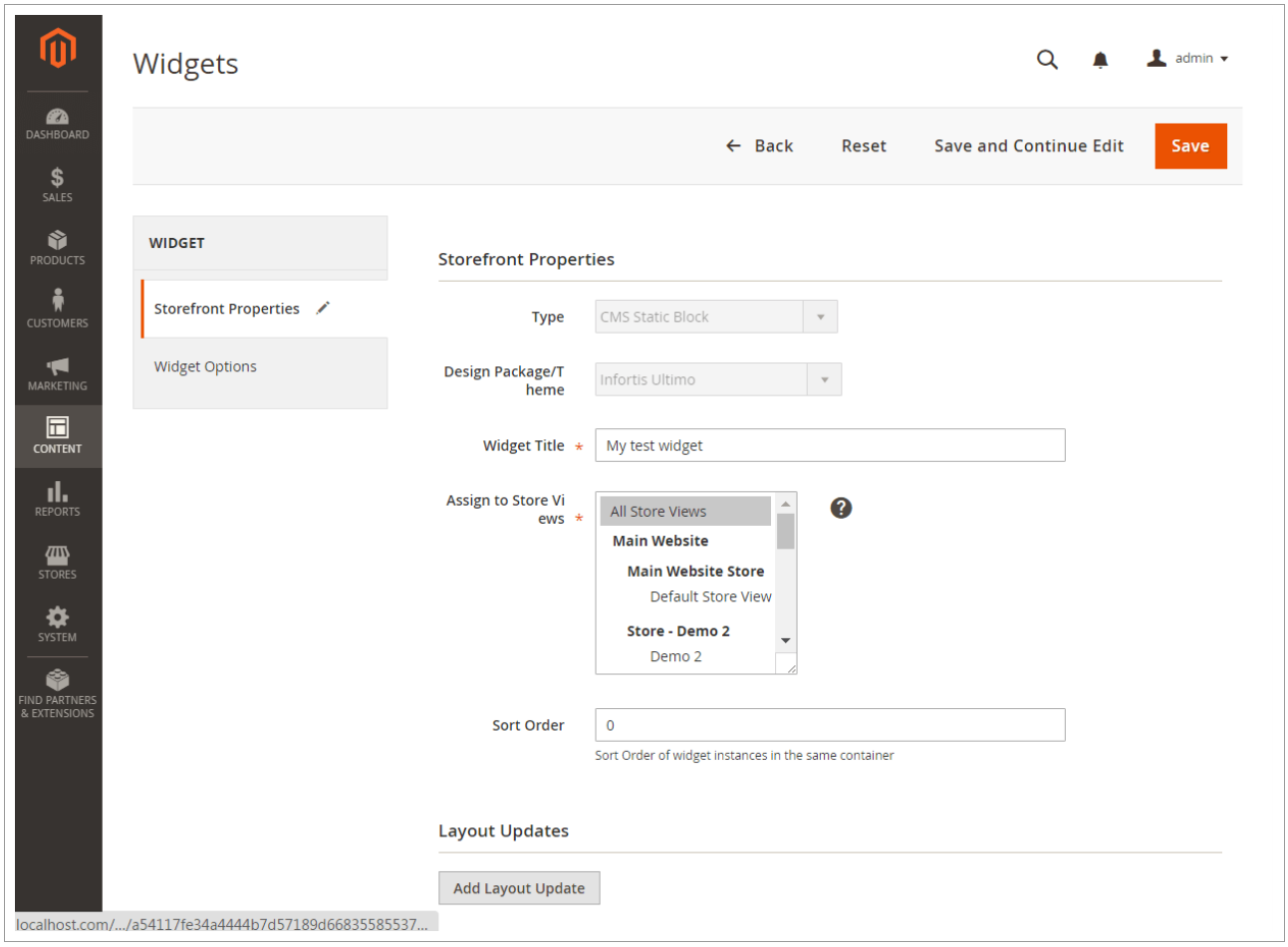
2. Then you need to select the design in which the widget will be displayed. Select: `Infortis Ultimo`.

3. Then, click **Continue** button.

4. On the next page, goto the **Widget Options** tab and click the **Select Block...** button to choose the static block. You will see the list of all static block. Find and click the one which you want to display on the frontend:



5. Next, go to the **Storefront Properties** tab.

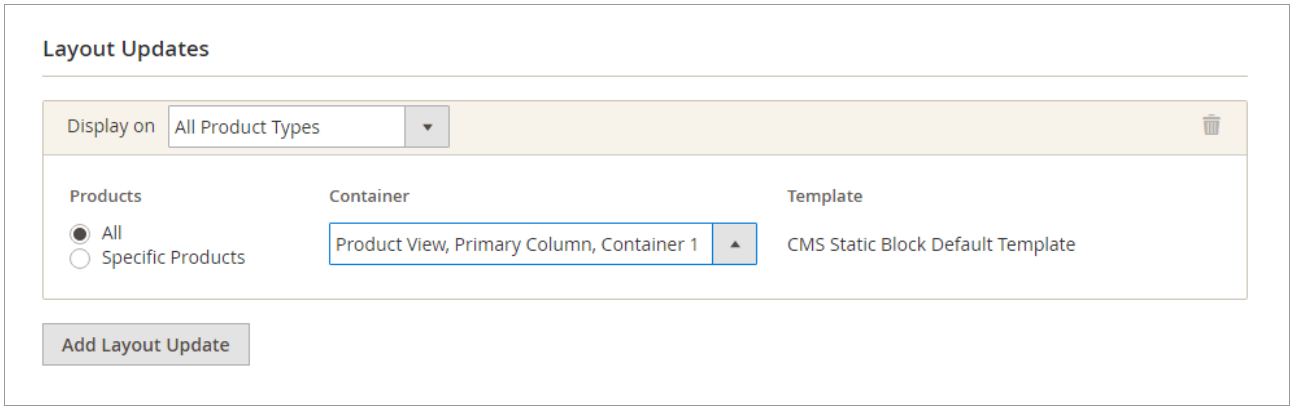


6. In the **Widget Title** field give the widget a title – you can choose any name you like (try to make it informative).

7. If you have a multiple store views, in the **Assign to Store Views** field select in which store views you want to display the widget.

8. The **Sort Order** is optional, it will be used to sort widgets if more than one widget was assigned to the same container on the frontend.

9. Click on the **Add Layout Update** button – new options will be displayed inside the **Layout Updates** section:



10. In the **Display On** drop-down select a general type of pages on which the widget will be displayed. Select `All product Types`.

11. Again, new options will be displayed. You will see the contextual options for your selection. In this case, you can select for which products you want to add your widget: `All` or `Specific Products`. Select `All`.

12. In the **Block Reference** field select where exactly on the page you want to display your custom static block. Select one of the product page containers, for example: `Product View, Primary Column, Container 1`.

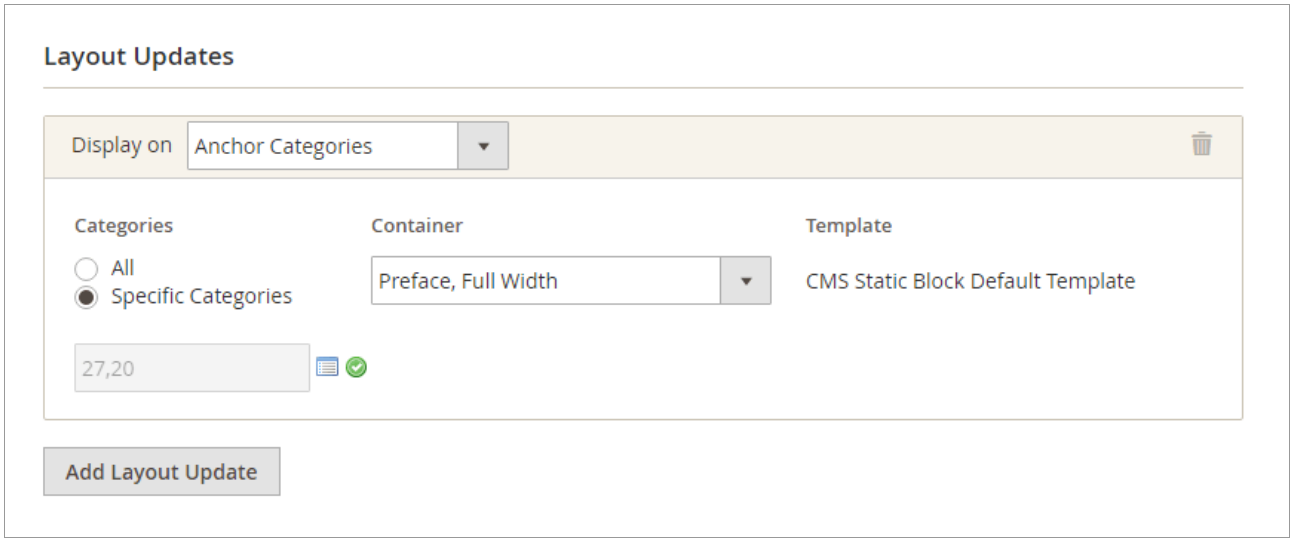
    List of available containers on product page can be found in this chapter **6.2.3.D Product page**.

    You will notice some additional containers, the list may differ in your specific case – additional containers may have been added by any additional extensions you have installed in your Magento.

13. Finally, save the widget and then flush Magento cache to see the effect on the frontend.

Example 2:

Here's another example in which we will add a block to category view but only for selected categories (see the **Display On** list). Block will be displayed in the `Preface, Full Width` container (see the **Block Reference** field), which is a full width container just below the main menu and above the page content area:



List of all containers available on each page can be found in **6.2.3.A Page**.

4.8.2 Static blocks

Static blocks in Magento are simple portions of content that can be displayed throughout the site. To create a static block, go to `Content > Blocks`, click `Add New Block` button and follow these steps:

1. In the **Block Title** field, enter the title of the block for internal reference.
2. In the **Identifier** field, assign a unique identifier to the block. Use all lowercase characters, with underscores instead of spaces.
3. Select the **Store View** to which this block will apply.
4. Select `Enabled` in the **Status** field. `Disabled` means that the block is not visible in the frontend.
5. Insert your **Content** (it can be text or HTML) and click **Save Block**.

> **Note:**
>
> when you want to edit static blocks, it's recommended to disable Magento's WYSIWYG so that it doesn't process the content of your block before you start editing it. Turn it on only when it is needed. Go to `Stores > Configuration > Content Management` and set WYSIWYG as **Disabled by Default**.

> **Important:**
>
> in order to see your changes, you need to refresh Magento cache after making any changes in your static blocks.

For more details on creating static blocks, refer to Magento documentation:
http://docs.magento.com/m2/ce/user_guide/cms/blocks.html

4.8.3 CMS markup tags

There are some special markup tags – a bit of text surrounded by double curly braces – which have a special meaning in Magento. They are also called "CMS tags", "CMS directives", but for simplicity, in this document we will call them "shortcodes".

Shortcodes can be used either in static blocks or CMS pages. For example, rather than hard-coding the full path to a page, you can use a shortcode to represent the store URL. Magento will automatically process all shortcodes during page rendering, so on the frontend instead of shortcodes you will see content generated by each shortcode.

In this chapter you can find examples of the most useful shortcodes available in Magento. For more information, refer to Magento documentation:
http://docs.magento.com/m1/ce/user_guide/cms/markup-tag-syntax.html

4.8.3.A Store URL

The **Store URL** shortcode represents the base URL of your website, and is used as a substitute for the first part of a full URL, including the domain name.

```
{{store url=""}}
```

This shortcode typed into the content of the page will be replaced with the base URL of your website.

You can use this shortcode to dynamically build all of your links (to avoid hard-coding the domain name). Thanks to that you will not need to rebuild your links after you move your store to a new domain.

For example, if your store domain is `www.example.com`, the following shortcode typed into the page content:

```
{{store url="path/to/page/about-us"}}
```

will be replaced with this URL:

```
http://www.example.com/path/to/page/about-us/
```

4.8.3.B Media URL

The **Media URL** shortcode represents the location and file name of an image. The shortcode can be used to place an image on a page, static block, or email template.

```
{{media url="image.png"}}
```

This shortcode typed into the content of the page will be replaced with the URL of the image.

The image path is relative to Magento's `media` directory, so you need to store your

images inside that directory.

For example, if this is the path of your image: `pub/media/wysiwyg/path/to/image.png`, you can get the URL of that image with the following shortcode:

```
{{media url="wysiwyg/path/to/image.png"}}
```

Here's how you can use that shortcode in the `src=""` attribute of `<img>` element to display your image on a page:

```
<img class="image" src="{{media url="wysiwyg/path/to/image.png"}}" alt="My image" />
```

4.8.3.C Block ID

The **Block ID** shortcode is one of the easiest to use, and can be used to place a static block directly on a CMS page, or even nested inside another static block. You can use this technique to modify a block for different promotions or languages.

```
{{block id="my-block-id"}}
```

The Block ID shortcode references a static block by its identifier. In the `id` attribute, enter an identifier of the static block which you want to display. For example:

```
{{block id="tester01"}}
```

This shortcode typed into the content of the page will be replaced with the content of the static block `tester01`.

4.8.4 WYSIWYG editor

WYSIWYG (*What You See Is What You Get*) editor is a convenient way to add content to pages and static blocks in Magento.

> **Important:**
>
> WYSIWYG should only be used to edit text, **never use it to edit HTML**.
>
> Always turn off WYSIWYG editor when you want to edit HTML or Magento's CMS tags in the page/block content. Otherwise WYSIWYG editor can break your content.

But the best way to use WYSIWYG is to **turn it on only when it is needed**. Go to `Stores > Configuration > Content Management` and set it as **Disabled by Default.**

After that, if needed, WYSIWYG can always be turned on with **Show/Hide Editor** button above the page/block content field.

4.9 Translation / Localization

4.9.1 Interface translation

You can download language packages for Magento from Magento Marketplace: *https://marketplace.magento.com/extensions/content-customizations/translations-localization.html*

For more details about translations and languages please refer to the official documentation of Magneto:

- *http://docs.magento.com/m2/ce/user_guide/stores/store-language-add.html*
- *http://devdocs.magento.com/guides/v2.0/frontend-dev-guide/translations/xlate.html*

Ultimo theme adds a few additional interface labels. All the new labels are located in this file: `app/design/frontend/Infortis/base/i18n/en_US.csv`

This file can be used to:

- translate the interface labels into other languages.
- replace one phrase with another phrase. For example, when you want to use "Compare" instead of "Add to Compare".

Example:

To translate the interface into Spanish language, follow these instructions:

- Go to folder `app/design/frontend/Infortis/base/i18n/`
- Make a copy of `en_US.csv`
- Rename that copy to `es_ES.csv`
- Open your new file in Open-office Calc (or some good text editor like Notepad++, PSPad) and make all required changes in your translation.

> **Note:**
>
> The editor should be capable to save file in UTF-8 encoding. Do not use Excel, it can break file structure. Read more about useful tools:
>
> *http://www.magentocommerce.com/wiki/groups/166/useful_tools_to_work_with_translations*

- If you open your file in Open-office Calc it might look like this:

My Wish List	Wish List
My Account	Account
Create an Account	Sign Up
Add to Wish List	Wish List
Add to Compare	Compare
…	

File opened in a text editor might look like this:

```
"My Wish List", "Wish List"
"My Account", "Account"
"Create an Account", "Sign Up"
"Add to Wish List", "Wish List"
"Add to Compare", "Compare"
...
```

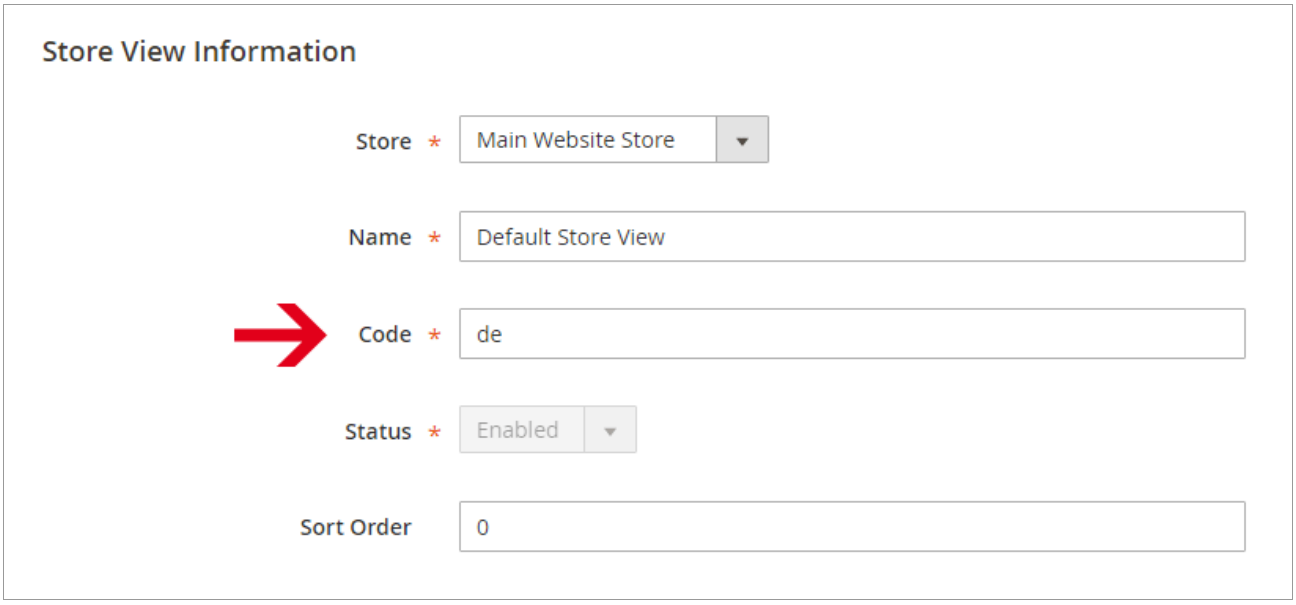
The first column contains the original character strings. It should be **left intact**. In the second column you can place your translation of each string.

4.9.2 Language flag

If you have more than one store view in your store, the store view switcher will be displayed at the top of the page. You can enable different language for each store view.

For each available language a flag is displayed in the store view switcher (we can also call it “language switcher”). Flag images (16x12 pixels, PNG format) should be uploaded to `app/design/frontend/Infortis/base/web/images/flags` folder. Image names should be the same as the store view codes. E.g. if you have a store view with the store code `de`, you will need to upload a flag image `de.png` to the `app/design/frontend/Infortis/base/web/images/flags` folder.

To check what is the store view code, go to *`Stores > All Stores`* and click on the store view name. You can find the code in the *`Code`* field:



Remember that if you change the store view code, you will also need to change the name of the corresponding flag image.

5. Customization

> **Important:**
>
> please note that Magento theme customization is far beyond the scope of this user guide. This chapter is only for informational purposes, you should treat it as a starting point. Before you start to customize the theme you need to be sure that you know what you're doing.

Magento is based on a theming concept referred to as parent/child theming – changes in the theme can be implemented without modifying the original.

The correct way to customize the theme in Magento is to create your own sub-theme of the default theme in the design package. **You should never edit original files of the design package**. Files that need to be modified can be copied from the default theme to your custom sub-theme inside the design package. But don't copy all files, only the files which you're going to modify.

This way you can override any file of the theme. And you can be sure that when something goes wrong you can simply delete your custom sub-theme without breaking the original theme files.

In this chapter you can find basic information that will help you to modify the theme.

5.1 How to modify the theme

5.1.1 Override default CSS styles

5.1.1.A Use _custom.less file

When you want to make some non-standard design changes (for which you can't find any settings in the theme admin panel) you can do it by adding custom CSS styles. In Ultimo 2 you can add custom CSS styles directly inside this file:

```
<YOUR_MAGENTO_INSTALL_DIRECTORY>/app/design/frontend/Infortis/ultimo/web/css/_custom.less
```

Or, preferably, create your own child-theme (with Ultimo as its parent theme). And then, create `_custom.less` file in the following location:

```
<YOUR_MAGENTO_INSTALL_DIRECTORY>/app/design/frontend/<YOUR_VENDOR_NAME>/<YOUR_THEME_NAME>/web/cs
s/_custom.less
```

Like always in Magento 2: when you make changes in CSS, JavaScript, interface images, icons etc., **you need to rebuild Magento static files**. Generating static files is simple – just run the following Magento command:

```
bin/magento setup:static-content:deploy
```

If you correctly refresh Magento static files, your custom styles will be applied to the frontend of your store.

5.1.1.B Debugging

If your custom styles are not applied to the frontend of your store, then there are a few things that could go wrong:

1. This may be a problem with generating static files in your Magento.

2. The cache in your Magento was not flushed or you didn't clear the cache in your web browser.

3. Your `_custom.less` file may be not accessible for Magento because of incorrect file permissions on your server.

4. If file is loaded and is accessible for Magento, but CSS rules from that file are not visible in Firebug then it means that your CSS rules are incorrect (or formatting of the CSS code in that file is incorrect).

5. If CSS rules are visible in Firebug, but they don't take any effect, then it means that your CSS rules are not specific enough and they are being overridden by theme's default CSS rules.

   CSS rules need to have proper specificity to override default CSS rules. You can read more about it here:

   - *https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity*
   - *http://www.smashingmagazine.com/2007/07/27/css-specificity-things-you-should-know/*
   - *http://specificity.keegan.st/*

5.1.1.C Add CSS via Magento admin

There is an alternative way to add custom CSS styles. Custom CSS can be added through Magento admin panel.

In Magento 2.1, go to: *`Content > Design > Configuration > HTML Head`*. In the *`Scripts and Style Sheets`* field add the following line:

```
<link  rel="stylesheet" type="text/css"  media="all" href="{{MEDIA_URL}}my_styles.css" />
```

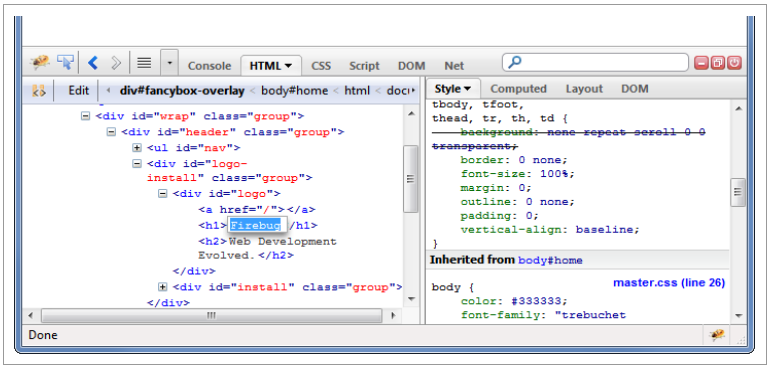
File `my_styles.css` containing your custom CSS styles needs to be placed in the following folder in your Magento:

```
pub\media\my_styles.css
```

5.1.1.D How to find out which part of CSS to override?

How to find out which part of the theme's or Magento's CSS you need to override or extend? The best way is to use **Firebug** (an add-on for Firefox: *www.getfirebug.com*), **Chrome Developer Tools** (available in Google Chrome) or other similar tools.

See the short video tutorial about **Firebug**: *http://getfirebug.com/video/Intro2FB.mp4*



How to use CSS?

Here you can find more information about CSS:

- *https://developer.mozilla.org/en-US/learn/css*
- *http://www.w3schools.com/css/*
- *http://reference.sitepoint.com/css/syntax*

5.1.2 Modify template files

Magento consists of dozens of template files, each one is used to render a block of content in the frontend of the store. When you want to customize a theme, you can display the name of template file of every block in the frontend by enabling Magento's **Template Path Hints**.

> **Note:**
>
> to enable Magento's **Template Path Hints** go to `Stores > Configuration > Developer`. Select your current website under **Current Configuration Scope** in the top left corner of the admin panel, otherwise Template Path Hints settings will not be visible.
>
> See this short video tutorial for more details: *http://vimeo.com/1067069*

This way you can easily check which template files you need to edit to customize specific sections of Magento. **Don't modify the core files. Files which you want to modify needs to be copied to your custom sub-theme.**

5.1.3 Create custom sub-theme

The correct way to customize the theme in Magento is to create custom sub-theme. You should never edit the original files. Files that need to be modified can be copied to your custom sub-theme. Refer to **5.2 Magento theme customization** for more details.

> **Important:**
>
> when you're creating a sub-theme in Magneto you should only copy the files which you want to modify. **Do not copy all files.** Otherwise you will have tons of additional work in the future with any theme upgrade.

To check which template file you need to edit to customize specific section of Magento you can enable Magento's **Template Path Hints**. Refer to chapter **5.1.2 Modify template files** for more details.

If the file which you selected with **Template Path Hints** doesn't exist in Ultimo package (a theme doesn't need to override all template files from Magento's “base” theme), you can copy that file from Magento's “base” theme to your custom sub-theme inside Ultimo package.

By creating custom sub-theme:

- You can override any file.
- You can create upgrade-proof modifications.
- You can be sure that when something goes wrong you can simply delete your sub-theme without breaking the original files.

5.2 Magento theme customization

Note: Chapter requires updating, the structure of themes has changed completely in Magento 2.

6. Theme features and elements

In this chapter you can find description and configuration details of selected elements of this theme, such as home page, built-in static blocks, CSS classes etc.

> **Note:**
>
> The list of all settings available in the theme admin panel in `Stores > Configuration` can be found in the next chapters.

6.1 Home page and CMS pages

To manage content of CMS pages in Magento, go to `Content > Pages` section in Magento admin panel.

6.1.1 Default pages

To configure the default pages for your store, go to `Stores > Configuration > General > Web > Default Pages` tab.

The Default Pages configuration determines the landing page that is associated with the base URL, and the corresponding CMS home page. It also determines which CMS page appears when a "404 Page Not Found" error occurs. It also determines if a breadcrumb trail appears at the top of CMS pages.

6.1.1.A Selecting a home page

To configure the default home page for your store, go to `Stores > Configuration > General > Web > Default Pages` tab, select the desired page in the `CMS Home Page` field and save configuration.

> **Important:**
> after the theme is installed, you can import sample versions of home pages from the demo. For more details about import procedure refer to **7.11 Data Import**. Imported pages will be visible in the `Content > Pages` section.

6.1.2 Creating a page

The process of adding a new CMS page to your store is essentially the same for any type of page you might want to create. You can include text, images, blocks of content, variables, and widgets. Most content pages are designed to be read by search engines first, and by people second. Keep the needs of each of these two very different audiences in mind when choosing the page title and URL, composing the meta data, and writing the

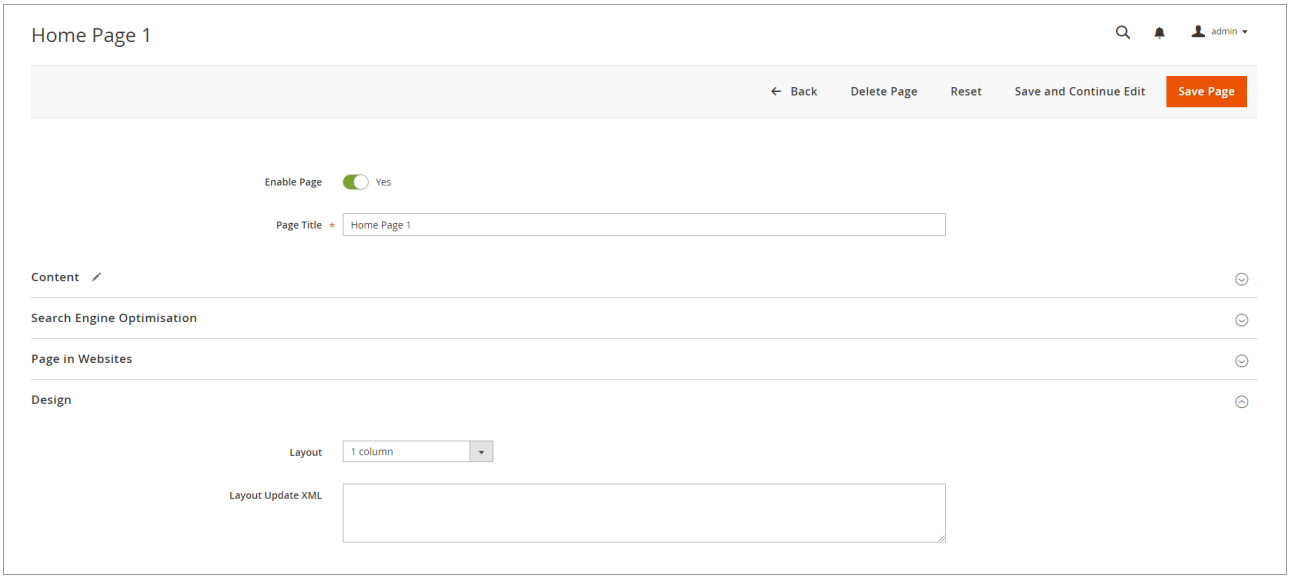
content.

To create a new page, go to `Content > Pages` section.

6.1.2.A Page layout

For every CMS page in Magento you can select page layout – the number of columns: one, two, or three columns.

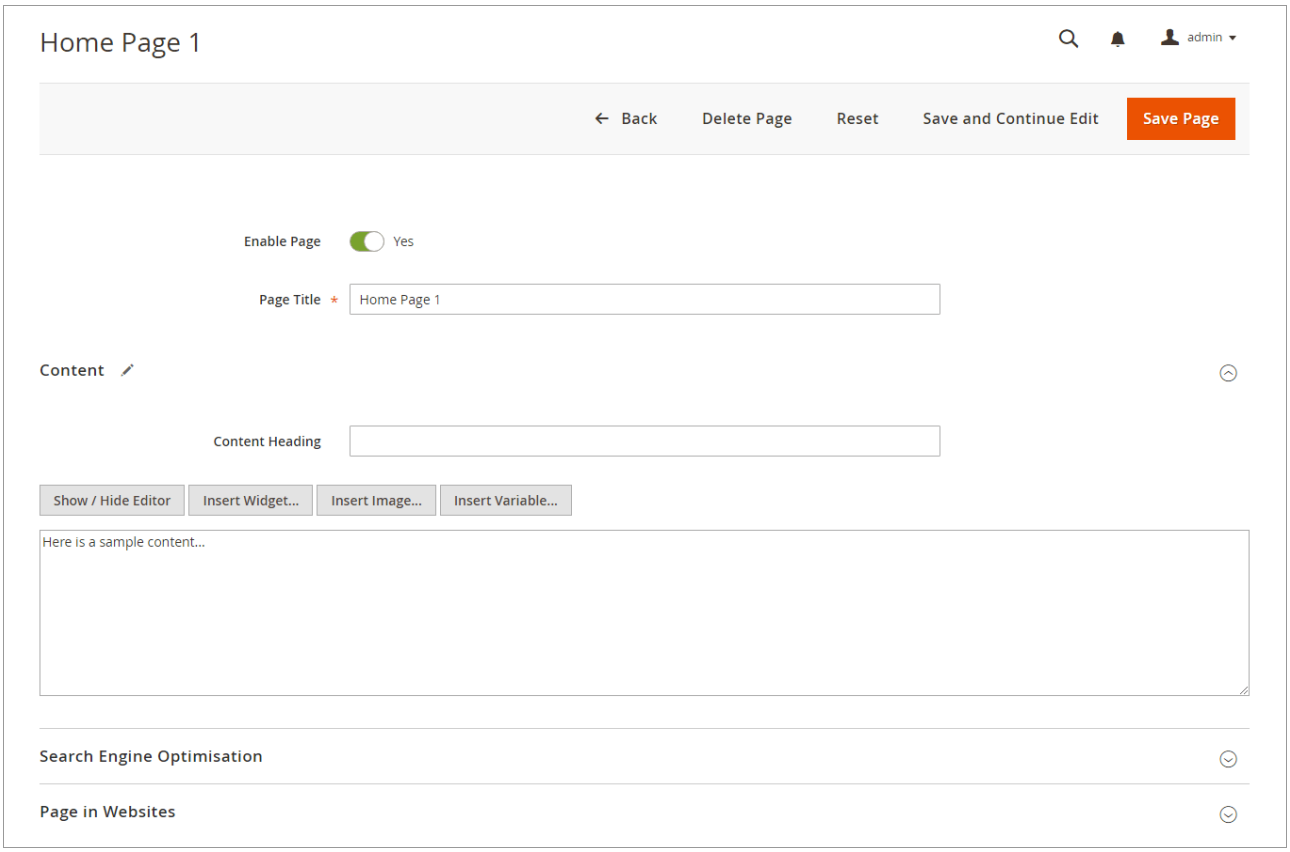
To change the layout of the page, go to `Content > Pages`, open desired page, open the `Design` tab, then select one of the options in the `Layout` field and click the `Save Page` button:



6.1.2.B Page content

Page content can include text, images, blocks of content, variables, and widgets.

To edit a content of a page, go to `Content > Pages` and open desired page. Content can be found in the main text field of the `Content` tab:



> **Important:**
>
> always turn off WYSIWYG editor when you want to edit HTML of the page. Otherwise WYSIWYG editor will break the content. Refer to **4.8.4 WYSIWYG editor** for more details.

6.1.2.C Full width page

To create a full width page, go to `Content > Pages`, select a page (or create a new one), open the *`Design`* tab, select `1 column, full width` in the *`Layout`* field and click the *`Save Page`* button. From now on, the content area of the page will be stretched to full width of the browser viewport.

6.1.2.C.1 Maximum width of the content

If you want the content area of that page to keep the same width as a standard page has,

you need to wrap the content in an additional `div` with the `container` class, like on the example below. The content inside this wrapper will be aligned with the other sections of the page (e.g. with the header) and will not exceed the maximum width which is defined in the *`Maximum Breakpoint`* field under `Stores > Configuration > Theme Layout`.

Here's the code example:

```
<div class="container">
     <div class="inner-container">
             This is a sample text
     </div>
</div>
```

The `container` class is responsible for applying the maximum width of the content area. And the `inner-container` class is responsible for additional side padding of the content area.

6.1.2.C.2 Content with full width background

To create full width blocks on the page, wrap your content in an additional div and apply a background color using inline CSS (using a style attribute in HTML elements), for example:

```
<div style="background-color: pink;">
     <div class="container">
             <div class="inner-container">
                     This is a sample text 1
             </div>
     </div>
</div>
```

More details about the `bacground-color` property can be found here: https://developer.mozilla.org/en-US/docs/Web/CSS/background-color

You can add multiple blocks to divide a page content into multiple sections, each section in different color:

```
<div style="background-color: pink;">
     <div class="container">
             <div class="inner-container">
                     This is a sample text 1
             </div>
     </div>
</div>
<div style="background-color: lightblue;">
```

```
    <div class="container">
            <div class="inner-container">
                    This is a sample text 2
            </div>
    </div>
</div>
```

6.1.3 Home page blocks

The following chapter describes blocks which can be displayed inside the content area of the home page.

> **Important:**
>
> it's highly recommended to turn off the WYSIWYG editor so that it doesn't process the code before you start editing it. Turn it on only when it is needed. Refer to **4.8.4 WYSIWYG editor** for more details.

6.1.3.A Image slideshow

To add a slideshow to the home page, refer to chapter **10. Slideshow**.

6.1.3.B Banners

To create image banners, refer to chapter **6.5.5 Banners**.

6.1.3.C Grid/List of Products

To display a block of products (grid or list mode) similar to the block of products displayed in the category view – use the following shortcode:

```
{{block
class="Magento\Catalog\Block\Product\ListProduct"
template="product/list.phtml"
category_id="3"
}}
```

Enter the shortcode in the content field of your page, Magento will **automatically replace the shortcode with a block** during page rendering.

The shortcode can take a few parameters. All available parameters are presented below.

Required parameters:

- `category_id` – identifier of the category which you want to present in the block. Make sure you enter identifier of the category which actually exists in your store (and which is active) – otherwise the block will not appear.

> **Note:**
>
> to find out an identifier of a selected category, refer to chapter **4.5.2 Find category ID**.

Optional parameters:

- `grid_column_count` – determines the number of products displayed in a single row.

  If you omit this parameter, the grid will have the number of columns defined in the admin panel in section *Stores > Configuration > Theme Settings > Category View (Grid Mode) > Number of Columns*.

- `mode` – choose the mode in which the block displays products. Available values:
  - `grid` – grid mode. This is the default mode. If you want to display grid, you can completely omit this parameter.
  - `list` – list mode
- `size` – changes the size of the elements of the grid, such as product name, price. Available values:
  - `XL` – elements bigger than the standard size
  - `L` – elements a little bit bigger than the standard size
  - `M` – this is the default size. If you want elements to have the default size, you can completely omit this parameter.
  - `S` – elements a little bit smaller than the standard size
  - `XS` – elements much smaller than the standard size
- `hide_toolbar` – enter `1` to hide the toolbar

- `img_width` – enter product image width. If width is not specified, default width will be used (208 pixels).

- `img_height` – enter product image height. To **keep the original aspect ratio** of product images, specify the width of images using the `img_width` parameter and **omit** the `img_height` parameter. Thanks to that the height of each image will be calculated automatically based on its width.

6.1.4 Featured Products Slider

Featured products slider can show products from any category (only single category is allowed). Slider can be displayed on any CMS page (and inside any static block). To add a slider to a page, use the following shortcode:

```
{{block
class="Infortis\Base\Block\Product\ProductList\Featured"
template="product/list_featured_slider.phtml"
category_id="3"
block_name="My Products"
}}
```

Enter the shortcode in the content field of your page, Magento will **automatically replace the shortcode with a slider** during page rendering.

Later in this chapter you can find examples of the shortcodes: **6.1.4.B Examples of shortcodes**.

> **Note:**
>
> make sure that in the `category_id` parameter you specified an identifier of a category which actually exists in your store (and which is active) – otherwise the slider will not appear. Other parameters are described later in this chapter.

> **Important:**
>
> in order to display the slider correctly, you may need to flush the cache (**4.1 Caching system**) and reindex your Magento (**4.2 Index management**) after making any changes in your products or categories.

6.1.4.A List of parameters

The shortcode which creates a slider can take a few parameters. Only the `category_id`

parameter is required. The other parameters are optional, they can be used to override the global settings of sliders (global settings can be found in **7.10 Product sliders**). If you create multiple sliders on a single page (by adding multiple shortcodes), then in each shortcode you can have different values of parameters.

All available parameters are presented below.

Required parameters:

- `category_id` – identifier of the category which you want to present in the slider. Make sure you enter identifier of the category which actually exists in your store (and which is active) – otherwise the slider will not appear.

> **Note:**
>
> to find out an identifier of selected category, refer to chapter **4.5.2 Find category ID**.

Optional parameters:

- `block_name` – the name of the block displayed as a heading.
- `product_count` – the total number of products presented in the slider.
- `thumbnails` – display products as thumbnails (small image with name and price displayed at the side of the image). Available values:
  - `L` – image size: 200 x 200 pixels
  - `M` – image size: 150 x 150 pixels
  - `S` – image size: 100 x 100 pixels
  - `XS` – image size: 80 x 80 pixels
- `rows` – products can be displayed vertically in multiple rows (one below another). In this parameter you can specify the number of rows of products.
- `breakpoints` – with this parameter you can specify the number of columns (products) in a row visible with a particular browser viewport width. Use the following format (this is the default value, it will be used if this parameter is not specified in the block code):

```
breakpoints="[0, 1], [320, 2], [480, 3], [768, 4], [992, 5], [1200, 6]"
```

Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers `[A, B]` number `A` indicates the web browser viewport width, number `B` indicates the number of columns visible if browser viewport width is greater than the width specified in `A`.
In other words: display `B` columns if viewport width is above `A` pixels.

Below you can find some examples:

- **Example 1:**

  ```
  breakpoints="[0, 2], [480, 3], [992, 5]"
  ```

  This value can be translated as: display 2 columns if viewport width is between 0px and 480px, display 3 columns if viewport width is above 480px, display 5 columns if viewport width is above 992px.

- **Example 2:**

  ```
  breakpoints="[0, 1], [480, 2], [768, 1]"
  ```

  This value is useful if you want to place the slider **inside a narrow container**, e.g. inside a sidebar, or inside a narrow column created with the grid system.

  This value can be translated as: display 1 column if viewport width is between 0px and 480px, display 2 columns if viewport width is above 480px, display 1 column if viewport width is above 768px.

  This means that **if the screen is wider than 768px** only 1 column will be visible. Thanks to that, the slider can be displayed inside narrow containers.

- **Example 3:**

  ```
  breakpoints="[0, 4]"
  ```

  which can be translated as: always display 4 columns.

- `img_width` – enter product image width. If width is not specified, default width will be used (170 pixels).

- `img_height` – enter product image height. To **keep the original aspect ratio** of

product images, specify the width of images using the `img_width` parameter and **omit** the `img_height` parameter. Thanks to that the height of each image will be calculated automatically based on its width.

- `timeout` – to animate the slider automatically, specify the time (in milliseconds) between transitions. Note that 1000 milliseconds = 1 second.
- `move` – number of products that should move on animation. Allowed values:
  - enter `1` to move one product.
  - enter `0` to move all visible products. This is also the default value which is used if the parameter is not specified.
- `loop` – enter `1` if the slider should loop.
- `pagination` – enter `1` to display pagination.
- `centered` – enter `1` to align elements of the slider to the center.
- `size` – with this parameter you can change the size of the elements of the slider, such as product name, price. Available values:
  - `XL` – elements bigger than the standard size
  - `L` – elements a little bit bigger than the standard size
  - `M` – this is the default size. If you want elements to have the default size, you can completely omit this parameter.
  - `S` – elements a little bit smaller than the standard size (the old value `size-s` was replaced with `S`).
  - `XS` – elements much smaller than the standard size (the old value `size-xs` was replaced with `XS`).
- `is_random` – enter `1` to display random products from the selected category.
- `sort_by` – with this parameter you can change the order of products. Available values:
  - `position` – default order of products in category. This is the default sort order if you omit this parameter in the shortcode.
  - `name` – sort by name

  - `price` – sort by price

- `sort_direction` – with this parameter you can change the direction of sorting. Available values:
  - `ASC` – ascending (default)
  - `DESC` – descending

- `is_responsive` – enter `0` if the slider **should not be responsive**. If this parameter is not specified, it is always assumed that the slider is responsive.

> **Important:**
>
> If the slider **is not** responsive, the number of visible products should be specified with the additional `show_items` parameter (see below).

  - `show_items` – the number of products in a row in the non-responsive slider. Use this parameter only if responsive behavior was disabled with parameter `is_responsive` (see above)

- `hide_button` – enter `1` to hide "Add to cart" button
- `hide_labels` – enter `1` to hide product labels (such as "sale")
- `hide_addto_links` – enter `1` to hide "Add to Wishlist" and "Add to Compare" links
- `hide_name` – enter `1` to hide product name
- `hide_rating` – enter `1` to hide product ratings
- `hide_price` – enter `1` to hide product price

6.1.4.B Examples of shortcodes

Sort by name, automatic scrolling

To change the default sort order, use `sort_by` and `sort_direction` parameters.

To add the automatic scrolling, use the `timeout` parameter. For example, `timeout="8000"` will animate the slider every 8 seconds (8000 milliseconds equals 8 seconds).

```
{{block
class="Infortis\Base\Block\Product\ProductList\Featured"
template="product/list_featured_slider.phtml"
category_id="3"
product_count="12"
block_name="My Products"
sort_by="name"
sort_direction="ASC"
timeout="8000"
hide_button="1"
}}
```

Large products with large images

To make the products larger, you need to change the value of the `breakpoints` parameter to change the number of products in a row.

Also, you need to increase the size of product images using `img_width` and `img_height` parameters. To **keep the original aspect ratio** of images, specify only the width and **omit** the `img_height` parameter. Thanks to that the height of each image will be calculated automatically based on its width.

```
{{block
class="Infortis\Base\Block\Product\ProductList\Featured"
template="product/list_featured_slider.phtml"
category_id="3"
product_count="12"
block_name="Our Featured Products"
breakpoints="[0, 1], [480, 2], [640, 3], [992, 4]"
img_width="265"
centered="1"
hide_button="1"
}}
```

Vertical thumbnails

To display products as thumbnails, add the `thumbnails` parameter to specify the size of the thumbnails.

To display products vertically in multiple rows (one below another), specify number of rows using the `rows` parameter.

Also, if you want to place the slider with vertically displayed thumbnails inside a narrow column (e.g. inside a sidebar, or inside a narrow column created with the grid system), it's

recommended to set the `breakpoints` parameter
to `[0, 1], [480, 2], [768, 1]`, so that the slider displays only one column of products when screen is wide (on large tablets and desktops).

```
{{block
class="Infortis\Base\Block\Product\ProductList\Featured"
template="product/list_featured_slider.phtml"
category_id="3"
product_count="12"
block_name="Vertical Thumbnails"
thumbnails="M"
breakpoints="[0, 1], [480, 2], [768, 1]"
rows="4"
hide_button="1"
}}
```

Small vertical thumbnails

This example is similar to previous examples. What's different is that you can additionally make the thumbnails smaller by changing the value of the `thumbnails` parameter to `S`, and make the elements of products (name, price) smaller by adding `size="S"` parameter.

You can also hide elements such as: "Add to cart" buttons, product labels, "Add to..." links, ratings etc.

```
{{block
class="Infortis\Base\Block\Product\ProductList\Featured"
template="product/list_featured_slider.phtml"
category_id="3"
product_count="12"
block_name="Small Thumbnails"
thumbnails="S"
breakpoints="[0, 1], [480, 2], [768, 1]"
rows="4"
size="S"
hide_button="1"
hide_labels="1"
hide_addto_links="1"
hide_rating="1"
}}
```

6.1.4.C Display any set of products using the slider

You can use the slider to display any set of products in your store. Slider can display products **from any category**, so you can simply create categories which will group any type of products you need, for example: bestsellers, products from specific brand, new products, promotional products etc.

You can, for example, display "on sale" products on the home page:

1. Create a new category. Name it "Special Offer".

2. Set the *Include in Menu* field to No – thanks to that the category will not be displayed in the main menu. The category will be only used to group selected products for slider.

3. Add all products with special price (or any other products you want) to the new "Special Offer" category.

4. Make sure Magento **indexes** are refreshed after you added products to the new category.

5. Create a slider (using a shortcode) which will display products from the "Special Offer" category:

   a) Create a shortcode which will define the slider. Examples of shortcodes and description of all available parameters can be found earlier in this chapter.

   b) Specify the identifier of the "Special Offer" category in the shortcode. Make sure you use an identifier of a category which really exists in your store – otherwise the slider won't appear.

   > **Note:**
   >
   > to find out an identifier of selected category, refer to chapter **4.5.2 Find category ID**.

   c) Add the shortcode to the content of your page.

6. Flush Magento cache (and any other cache you use).

6.2 Built-in blocks and containers

In Magento, you can create custom blocks of content without writing any code, and assign them to appear in a specific place in the page layout. Custom content can be added to the frontend through **static blocks**.

A **static block** is a modular unit of content that can be easily managed through Magento admin panel. Static blocks are sometimes referred to as **CMS blocks**, or **content blocks**, and can be used to display fixed information such as text, images, and embedded video, as well as dynamic information from widgets.

To create a static block, go to `Content > Blocks` section of admin panel. For more information, refer to chapter **4.8.2 Static blocks**.

> **Note:**
>
> when you want to edit static blocks, it's recommended to disable Magento's WYSIWYG so that it doesn't process the content of your block before you start editing it. Turn it on only when it is needed. Go to `Stores > Configuration > Content Management` and set it as **Disabled by Default**.

You can also import **predefined static blocks** included with this theme – refer to **7.11 Data Import** for more details about import.

6.2.1 Add custom content

There are two main ways to add custom content to the frontend:

- **built-in static blocks** – import (or create) a static block which will be automatically displayed in the predefined area of the page.
- **custom static blocks assigned to selected containers** – create a custom static block and then assign it (through Magento's Widget tool) to one of the available containers in the page layout.

Full list of identifiers of all **built-in static blocks** and **containers** available in this theme can be found later in this chapter (**6.2.3 List of built-in static blocks and containers**).

6.2.1.A Built-in static blocks

There is a number of **built-in static blocks** in Ultimo theme. A built-in static block is a block that is **automatically** displayed in the predefined area of the page on frontend.

For example, `block_footer_payment` is an identifier of the static block which is displayed in the footer of every page. This means that if you create and enable a block with this identifier (`block_footer_payment`), the content of that block will be automatically displayed in the footer of every page.

This particular block is being used to show information about payment methods but of course you can use it for any other purpose. All built-in static blocks (with sample content) are available for import right after theme installation.

6.2.1.B Custom static blocks assigned to containers

When you create a custom static block, you can assign it to appear in a specific place in the page layout. To do that, you need to assign your static block to a **container**.

There are many containers available to choose from. Containers exist for the sole purpose of assigning content structure to a page. A container has no additional content except the content of included blocks.

To **assign a static block to a selected container**, you need to create a **widget**. Widget will bind your static block with selected container – it will "tell" Magento where you want to display your block and on which pages it should be displayed. To create a widget, go to *`Content > Widgets`* section of admin panel. Refer to chapter **4.8.1 Add blocks to selected pages** for more details.

An example of a container is the sidebar on category pages. This container is named `Sidebar Main`. You can assign static blocks to the sidebar to display some additional information on selected pages, e.g. information about discounts for products from selected categories.

6.2.2 Import static blocks

There is a number of predefined static blocks included with Ultimo theme. You can import these blocks with one click – refer to **7.11 Data Import** for more details about the import procedure.

> **Note:**
>
> content of all static blocks is also available in XML format in case you need to restore the original content of some blocks. XML can be found in the following directory:
>
> `app/code/Infortis/Base/etc/importexport/cms/`

There are two types of imported blocks:

1. **built-in static blocks** – these are blocks that are automatically displayed in predefined area of the page. Identifiers of all built-in blocks start with `block_` prefix. For example: `block_footer_payment`, `block_footer_links`.

> **Note:**
>
> you **can not** change the identifiers of built-in static blocks. If identifier is changed, block will not be displayed in predefined area any more.

2. **sample static blocks** – sample blocks which you can use as a starting point when creating your custom content. You can assign these blocks to selected containers using the **Widget** tool. Identifiers of all sample blocks start with `sample_` prefix. For example: `sample_slideshow_1`, `sample_slideshow_2`.

> **Note:**
>
> in contrast to built-in static blocks, you can change the identifiers of sample static blocks to whatever you want.

After importing, some of the blocks are **disabled**. You will need to enable them in order to show their content in your store. You can replace the default content of each block with your own content. Remember that you need to flush Magento cache after making any changes in your static blocks.

Sample content

Most of the blocks already contain sample content. You can replace it with any custom content, but using provided samples will help you to keep the design consistent.

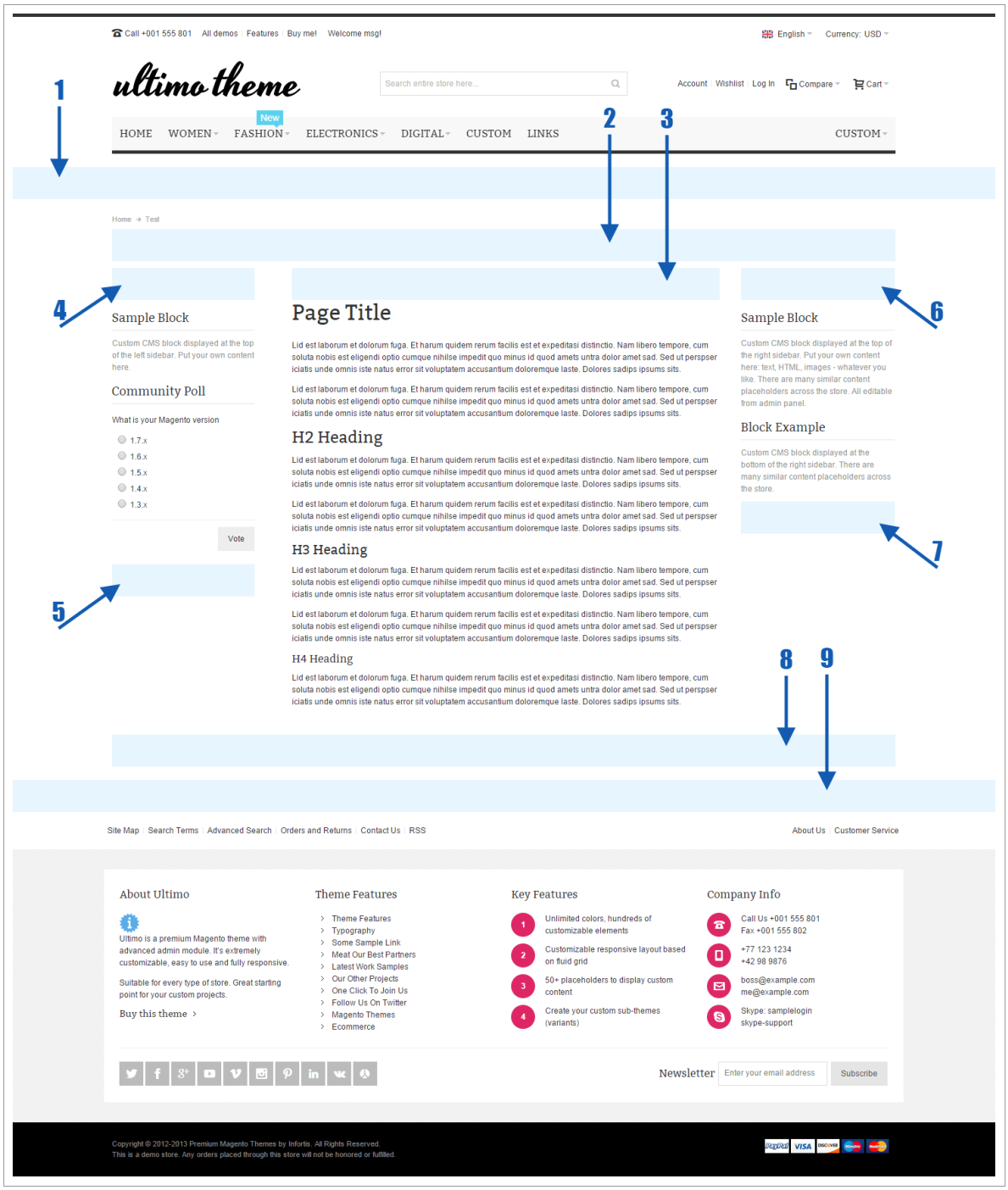
6.2.3 List of built-in static blocks and containers

In the next subchapters you can find identifiers of **built-in static blocks** and **containers** available in Ultimo theme.

6.2.3.A Page

The screenshot below presents positions of containers which are available on every page of the frontend, including home page, product pages, category pages etc.

> **Important:**
>
> Please note that this user guide is not yet fully updated. In Magento 2 some containers may have slightly different names than in Magento 1.



You can assign any number of custom **static blocks** to each container. If block is assigned to a container, you can choose to display it on all pages, but you can also choose to display it only on selected pages. For details about assigning static blocks to containers refer to chapter **4.8.1 Add blocks to selected pages**.

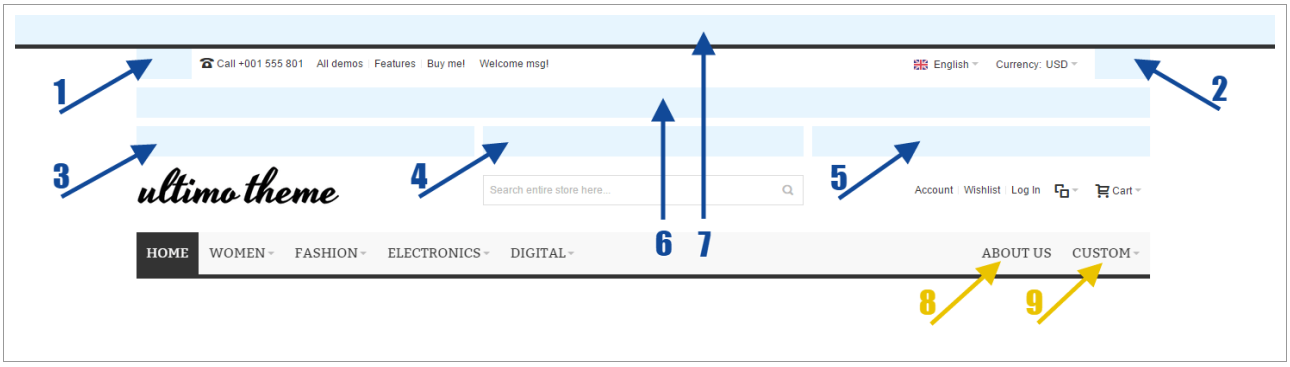
Containers are suitable for any kind of additional information:
discounts and promotions, image banners, social services bookmarks etc.

List of containers:

1. **Preface, Full Width** – **container** above the content area and above sidebars. It is stretched to full width.
2. **Preface** – **container** above the content area and above sidebars.
3. **Main Content Area** – **container** above the content area of the page. It is as wide as the content area. It is displayed between the sidebars (if page has sidebars).
4. **Left Column, Top** – **container** at the top of the left sidebar. Blocks added to this container will be visible only if page has the left sidebar.
5. **Left Column** – **container** at the bottom of the left sidebar. Blocks added to this container will be visible only if page has the left sidebar.
6. **Right Column, Top** – **container** at the top of the right sidebar. Blocks added to this container will be visible only if page has the right sidebar.
7. **Right Column** – **container** at the bottom of the right sidebar. Blocks added to this container will be visible only if page has the right sidebar.
8. **Postscript** – **container** below the content area and below sidebars.
9. **Postscript, Full Width** – **container** below the content area and below sidebars. It is stretched to full width.

6.2.3.B Header

The screenshot below presents positions of block containers which are available in the page header:



You can assign any number of custom **static blocks** to each container. If your block is assigned to a container, you can choose to display it on all pages, but you can also choose to display it only on selected pages. For details about assigning static blocks to containers refer to chapter **4.8.1 Add blocks to selected pages**.

List of containers which are available in the page header:

1. Page Header, Top, Left – **container** at the left side of the top header. Blocks added to this container are floating from left to right.

   There is also an additional **built-in static block** block_header_top_left. Content of this block will be automatically displayed just next to this container (if the block exists and is active). To use it, go to `Content > Blocks`, create a static block with identifier block_header_top_left and insert your content. Note that this block is also available (with sample content) if you import predefined static blocks provided with this theme.

2. Page Header, Top, Right – **container** at the right side of the top header. Blocks added to this container are floating from right to left.

   There is also an additional **built-in static block** block_header_top_right. Content of this block will be automatically displayed just next to this container (if the block exists and is active). To use it, go to `Content > Blocks`, create a static block with identifier block_header_top_left and insert your content.

3. Page Header, Primary, Left Column – **container** displayed above the content of the left column in the primary header.

4. **Page Header, Primary, Central Column** – **container** displayed above the content of the central column in the primary header.

5. **Page Header, Primary, Right Column** – **container** displayed above the content of the right column in the primary header.

6. **Page Header** – **container** displayed above the columns of the primary header.

7. **Page Top** – full-width **container** displayed above the header.

Additionally you can also display two predefined static blocks in the menu bar:

8. **block_nav_links** – **built-in static block** for the main menu. To display custom links in the main menu bar, create a static block with identifier **block_nav_links**. The content of this block will be automatically displayed directly in the menu bar so the content needs to have proper structure (it's described in this chapter: **12.3.1 Custom links inside a static block**).

9. **block_nav_dropdown** – **built-in static block** for the main menu. To display a custom drop-down box in the main menu, create a static block with identifier **block_nav_dropdown**. The content of this block will be automatically displayed in a drop-down box and the title of this block will be displayed as the name of the menu item.

There's also an additional container displayed only in mobile view:

10. **Page Mobile Header, Top** – **container** in the top header. Custom static blocks assigned to this container via Magento widgets will be automatically displayed in the top header only in mobile view.

    For quick start you can use sample static block `sample_mobile_header_top` which is provided with the theme (you just need to import static blocks first). Simply create a new widget and inside that widget assign the static block `sample_mobile_header_top` to the container named **Page Mobile Header, Top**.

    For details about assigning static blocks to containers, refer to chapter **4.8.1 Add blocks to selected pages**.

    You can add any content inside your static block. Below you can find some examples.

Example 1:

A short text with a background color:

```
<div style="background-color: #e4fbf6; padding: 10px;">
     Text example
</div>
```

To align the content to the left side, put your content inside a `<div>` with the following classes:

```
<div class="item item-left">
...
</div>
```

So the content of your block can look like this:

```
<div class="item item-left">
     <div style="background-color: #e4fbf6; padding: 10px;">
             Text example
     </div>
</div>
```

To align the content to the right side, replace the `item-left` class with the `item-right` class, like this:

```
<div class="item item-right">
     <div style="background-color: #e4fbf6; padding: 10px;">
             Text example
     </div>
</div>
```

To make the content centered, remove the `item-left` and `item-right` classes and use an inline CSS `style="text-align: center;"` on the containing `<div>`:

```
<div class="item" style="text-align: center;">
```

```
    <div style="background-color: #e4fbf6; padding: 10px;">
            Text example
    </div>
</div>
```

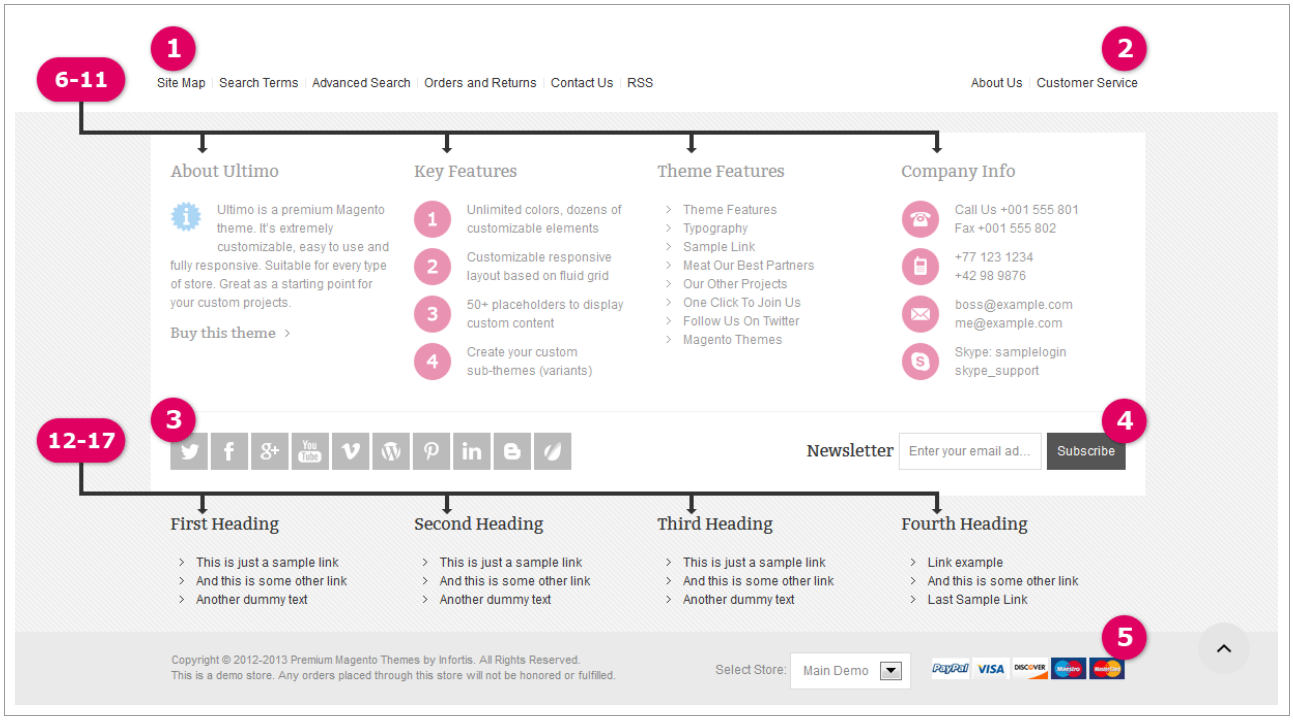
Example 2:

A few simple links aligned to the left side:

```
<div class="item item-left">
    <ul class="links">
            <li>
                    <a href="#">Mobile</a>
            </li>
            <li>
                    <a href="#">Links</a>
            </li>
    </ul>
</div>
```

6.2.3.C Footer

Static blocks with the following identifiers will be displayed in the footer section of the page:



1. block_footer_links – this **built-in static block** can be used to substitute Magento's default Footer Links (a set of links: *Contact Us*, *Advanced Search*, *Orders and Returns* etc.). Default Footer Links can be disabled in the admin panel, refer to **7.9 Footer** for more details.

2. block_footer_links2 – additional **built-in static block** for footer links, floating right.

3. block_footer_primary_bottom_left – **built-in static block** with icons and links to social services.

4. block_footer_primary_bottom_right – this **built-in static block** can substitute Magento's default newsletter subscription form. Newsletter form can be disabled in the admin panel, refer to **7.9 Footer** for more details.

5. block_footer_payment – **built-in static block** suitable for information about available payment methods, credit cards, SSL certificates etc.

6. block_footer_column1 – **built-in static block** which can be used as a column of

links in the main area of the footer section, suitable for any custom content: social services links, shipping information, promotions, company address etc. You can use up to six similar static blocks.

7. block_footer_column2 – see above
8. block_footer_column3 – see above
9. block_footer_column4 – see above
10. block_footer_column5 – see above
11. block_footer_column6 – see above
12. block_footer_row2_column1 – obsolete, this block was removed in Magento 2
13. block_footer_row2_column2 – obsolete, this block was removed in Magento 2
14. block_footer_row2_column3 – obsolete, this block was removed in Magento 2
15. block_footer_row2_column4 – obsolete, this block was removed in Magento 2
16. block_footer_row2_column5 – obsolete, this block was removed in Magento 2
17. block_footer_row2_column6 – obsolete, this block was removed in Magento 2

Note that the columns inside the footer don't need to be equal. You can arrange your content in many different ways, you have full control over those blocks.

Disable all static blocks with ID `block_footer_column[x]` (where `[x]` is a number from 1 to 6) and leave only one block enabled, e.g. this one: `block_footer_column6`.

Thanks to that, your block `block_footer_column6` will be stretched inside the footer and it will take the entire available space.

Now, inside `block_footer_column6` organize your content using built-in grid classes, for example:

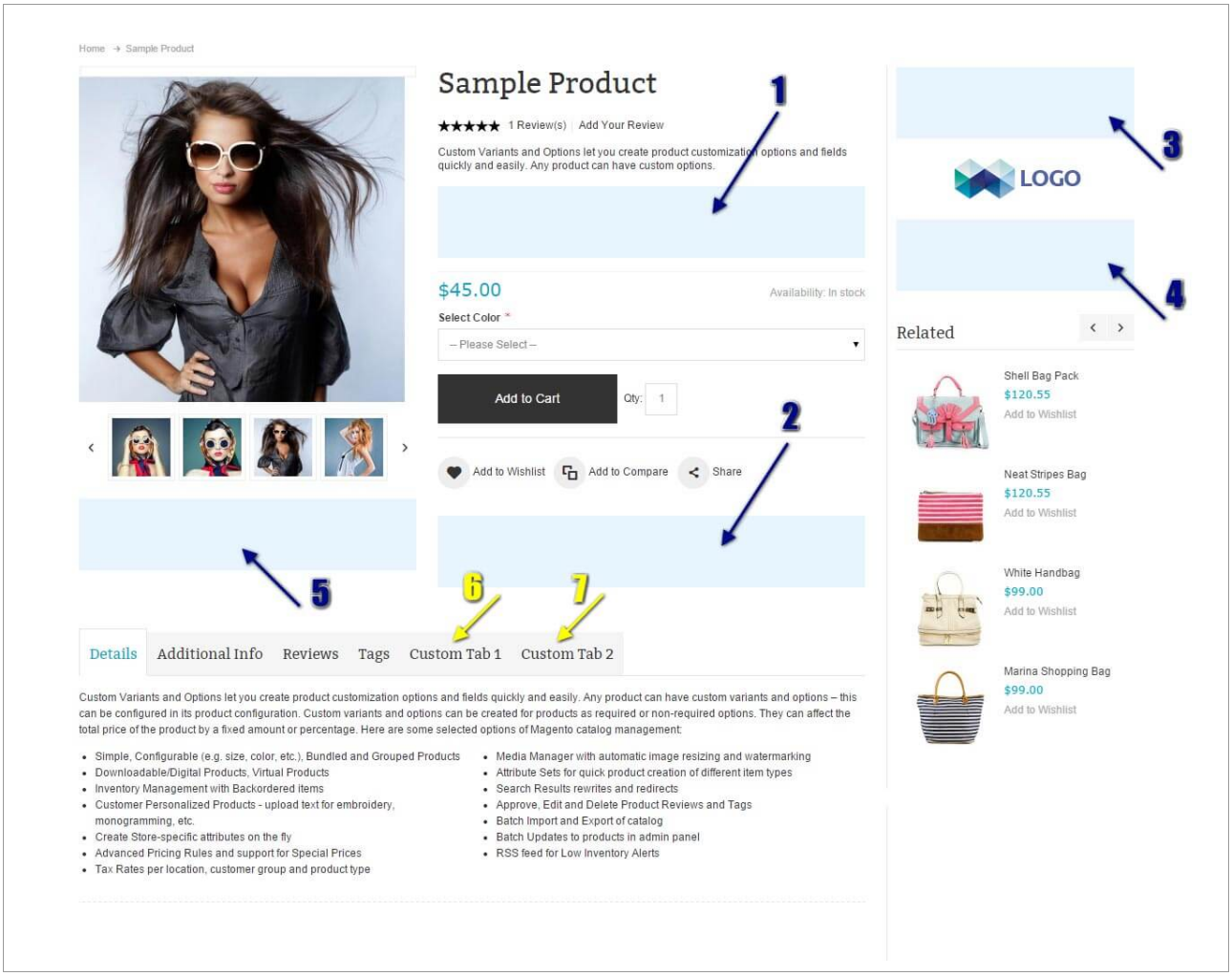
```
<div class="grid-container">
     <div class="grid12-3">Column1...</div>
     <div class="grid12-2">Column2...</div>
     <div class="grid12-2">Column3...</div>
     <div class="grid12-2">Column4...</div>
     <div class="grid12-3">Column5...</div>
```

```
</div>
```

More info about grid system can be found in **6.4 Grid system**.

6.2.3.D Product page

The screenshot below presents positions of block containers which are available on product page:



You can assign any number of custom **static blocks** to each container. If block is assigned to a container, you can choose to display it on all pages, but you can also choose to display it only on selected pages. For details about assigning static blocks to containers refer to chapter **4.8.1 Add blocks to selected pages**.

Containers are suitable for any kind of additional information:
discounts and promotions, available shipping methods, available payment methods, promotional image banners, social services bookmarks (such as AddThis bookmarks) etc.

List of containers which are available on product page:

1. **Product View, Primary Column, Container 1** – **container** displayed below product's short description inside the primary column.

2. **Product View, Primary Column, Container 2** – **container** at the bottom of the primary column.

> **Note:**
> the static block with ID ***block_product_primary_bottom*** was removed in Magento 2.

3. **Product View, Secondary Column, Container 1** – **container** at the top of the secondary column.

4. **Product View, Secondary Column, Container 2** – **container** at the bottom of the secondary column.

   There is also an additional **built-in static block** `block_product_secondary_bottom`. Content of this block will be automatically displayed below the brand logo inside the secondary column. To use it, go to *Content > Blocks*, create a static block with identifier `block_product_secondary_bottom` and insert your content. Note that this block is also available (with sample content) if you import sample static blocks provided with this theme.

5. **Product View, Image Column** – **container** displayed just below the image gallery.

> **Note:**
> the static block with ID ***block_product_img_bottom*** was removed in Magento 2.

Additionally you can also display two static blocks as tabs on product page. Content of these tabs will be displayed on the product page of every product in the store.

6. **block_product_tab1** – to display custom tab on product page, create a static block with identifier `block_product_tab1`. The content of that block will be displayed inside the panel of the tab and the title of that block will be displayed as the name of the tab.

7. **block_product_tab2** – to display second custom tab on product page, create a static block with identifier `block_product_tab2`. The content of that block will be displayed inside the panel of the tab and the title of that block will be displayed as the name of the tab.

There are also two other tabs on product page for **product-specific** info:

- **Details** – tab which displays product description. You can add there any custom content, it can be a simple text as well as a complex HTML with images, lists etc. You can also embed videos from services such as YouTube, Vimeo etc.

- **More Information** – tab which displays product-specific attributes. You can enable/disable attribute's visibility for product page in Magento's attribute manager.

6.3 Pages and elements

6.3.1 Drop-down cart

This theme is using Magento's default sidebar cart to display drop-down cart at the top of the page. In order to show drop-down cart, option **Display Shopping Cart Sidebar** in `Stores > Configuration > Checkout > Shopping Cart Sidebar` has to be enabled.

6.4 Grid system

In version 2.4.0 of Ultimo theme we added the grid system from Bootstrap (Bootstrap version 3.3.7). This new *Bootstrap grid system* will replace the old deprecated grid system which we have used in Ultimo theme since the very beginning.

> **Note:**
>
> we didn't add a complete Bootstrap library, but only our custom build of Bootstrap. Only these components were compiled: "Grid system" (*grid.less*) and "Responsive utilities" (*responsive-utilities.less*).

You can still use the old grid (the code will be still available), but **we recommend to use the Bootstrap grid** – especially when you create custom content (e.g. on CMS pages) or when you override (or create new) template files in your custom child themes. The new grid system is more powerful so we think you will enjoy the change.

6.4.1 Bootstrap grid overview

All the information about **Bootstrap grid system** can be found on the official website: *http://getbootstrap.com/css/#grid*
In this chapter you can find only a quick introduction.

> **Important:**
>
> Bootstrap documentation states that rows must be placed within an element marked with the `container` class. You can ignore this information, **don't use the `container` class in your custom content**. Ultimo theme already uses the `container` class to build the base structure of the page so this class can't be used anywhere else, especially not in the custom content of CMS pages.
>
> The only exception is when you want to create a full width page. In that case you may need to wrap the content in an additional div with the `container` class. See for more details: **6.1.2.C Full width page**

Grid systems are used for creating page layouts through a series of rows and columns that house your content. Bootstrap grid system is based on **twelve columns**. Grid

columns are created by specifying the number of twelve available columns you wish to span. This means that for every block you can specify how many columns it should span (from 1 to 12). The number at the end of the class name indicates the number of spanned columns. For example, class `col-md-3` will create a block which is 3-columns wide. Which means: the width of the block equals 3/12 of the entire available space.

Here's how the Bootstrap grid system works:

- Use rows to create horizontal groups of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like `row` and `col-xs-4` are available for quickly making grid layouts.
- Columns create gutters (gaps between column content) via padding attribute. That padding is offset in rows for the first and last column via negative margin on elements with the `row` class.
- The negative margin on rows makes the rows outdented. It's so that content within grid columns is lined up with non-grid content.
- Grid columns are created by specifying the number of **twelve** available columns you wish to span. For example, three equal columns would use three `col-xs-4`.
- If more than 12 columns are placed within a single row, each group of extra columns will, as one unit, wrap onto a new line.
- Grid classes apply to devices with screen widths greater than or equal to the breakpoint sizes, and override grid classes targeted at smaller devices. Therefore, e.g. applying any `col-md-*` class to an element will not only affect its styling on medium devices but also on large devices if a `col-lg-*` class is not present.

See how aspects of the grid system work across multiple devices with a handy table.

	Extra small devices Phones (<768px)	**Small devices** Tablets (≥768px)	**Medium devices** Desktops (≥992px)	**Large devices** Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Class prefix	`col-xs-`	`col-sm-`	`col-md-`	`col-lg-`
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			

Look to the examples for applying these principles to your code.

6.4.2 Grid examples

Using a single set of `col-md-*` grid classes, you can create a basic grid system that starts out stacked on mobile devices and tablet devices (the extra small to small range) before becoming horizontal on desktop (medium) devices. Place grid columns in any `row`.

Two equal columns

Get two equal-width columns starting at desktops and scaling to large desktops. On mobile devices, tablets and below, the columns will automatically stack.

```
<div class="row">
     <div class="col-md-6">.col-md-6</div>
     <div class="col-md-6">.col-md-6</div>
</div>
```

Three equal columns

Get three equal-width columns starting at desktops and scaling to large desktops. On mobile devices, tablets and below, the columns will automatically stack.

```
<div class="row">
     <div class="col-md-4">.col-md-4</div>
     <div class="col-md-4">.col-md-4</div>
     <div class="col-md-4">.col-md-4</div>
</div>
```

Three unequal columns

Get three columns starting at desktops and scaling to large desktops of various widths. Remember, grid columns should add up to twelve for a single horizontal block. More than that, and columns start stacking no matter the viewport.

```
<div class="row">
     <div class="col-md-3">.col-md-3</div>
     <div class="col-md-6">.col-md-6</div>
     <div class="col-md-3">.col-md-3</div>
</div>
```

Two unequal columns

Get two columns starting at desktops and scaling to large desktops.

```
<div class="row">
     <div class="col-md-8">.col-md-8</div>
     <div class="col-md-4">.col-md-4</div>
</div>
```

6.4.3 Additional utility classes for grid system

Row without outdent

Rows in Bootstrap's grid system are outdented (with negative margin at both sides of a row). Thanks to that the content within grid columns is lined up with non-grid content. But sometimes (e.g. on CMS pages) you may need a row with no outdent so that the edges (not the content inside) of the grid columns are lined up with non-grid content. You can force this behavior with the additional class `row-no-outdent` which will remove the outdent from the elements with the `row` class:

```
<div class="row row-no-outdent">
     <div class="col-md-6">Column example</div>
     <div class="col-md-6">Column example</div>
</div>
```

Row with bottom gutter

To add a bottom gutter to all grid columns inside a row, add the `row-bottom-gutter` class to the row element. This is an easy way to add equal spaces between rows. The bottom gutter will have the same size as the gutter between columns. Here's an example:

```
<div class="row row-bottom-gutter">
     <div class="col-md-4">Column example</div>
     <div class="col-md-4">Column example</div>
     <div class="col-md-4">Column example</div>
</div>
<div class="row">
     <div class="col-md-4">Column example</div>
     <div class="col-md-4">Column example</div>
```

```
    <div class="col-md-4">Column example</div>
</div>
```

If the bottom gutter should be narrower, use the `row-bottom-gutter-half` class – the bottom gutter will be half the size of the gutter between columns:

```
<div class="row row-bottom-gutter-half">
    <div class="col-md-4">Column example</div>
    <div class="col-md-4">Column example</div>
    <div class="col-md-4">Column example</div>
</div>
<div class="row">
    <div class="col-md-4">Column example</div>
    <div class="col-md-4">Column example</div>
    <div class="col-md-4">Column example</div>
</div>
```

Column without gutter

Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on rows.

But sometimes you may need to remove the gutter from selected grid columns. You can remove the gutter by adding the `no-gutter` class to the columns:

```
<div class="row">
    <div class="col-md-6 no-gutter">Column example</div>
    <div class="col-md-6 no-gutter">Column example</div>
</div>
```

To remove only the gutter at the left side, use the `no-left-gutter` class. To remove only the gutter at the right side, use the `no-right-gutter` class.

6.4.4 Old 12-column grid system (deprecated)

This grid system is used only for theme interface elements. If you want to use a grid in

your custom content (e.g. on CMS pages), we recommend to use the **Bootstrap grid system** which we added to the theme in version 2.4.0 – it is described in chapter **6.4.1 Bootstrap grid overview**.

Below you can find description of the old deprecated grid system.

Theme interface is based on 12-column grid system. This means that for every element you can specify width in grid units (from 1 unit to 12 units) using grid classes. These are available classes:

```
grid12-1
grid12-2
grid12-3
grid12-4
grid12-5
grid12-6
grid12-7
grid12-8
grid12-9
grid12-10
grid12-11
grid12-12
```

The number at the end of the class name indicates number of grid units. For example, class `grid12-3` will create a block which is 3-units wide.

As this is a 12-column grid, each grid class spans a number of those 12 columns, and should always add up to 12 for each row.

Simple grid examples

2 columns

If you want to display content on your custom page in two equal columns, create two blocks (`div`) and add appropriate class with number of grid units. In this case use `grid12-6` class to create two 6-units wide blocks. And 6 + 6 equals 12, so you will create a row of two equal columns.

Then wrap the blocks inside another block with class `grid-container` – this class is important to clear floats of the columns (all columns float to the left side) so that the content of the grid doesn't overlap with other content below and above the grid. Here's an example:

```
<div class="grid-container show-grid">
```

```
    <div class="grid12-6">Sample text...</div>
    <div class="grid12-6">Sample text...</div>
</div>
```

When developing the grid, you can also temporarily add class `show-grid` to visualize the grid (like in the above example).

3 columns

```
<div class="grid-container">
    <div class="grid12-4">Sample text...</div>
    <div class="grid12-4">Sample text...</div>
    <div class="grid12-4">Sample text...</div>
</div>
```

4 columns

```
<div class="grid-container">
    <div class="grid12-3">Sample text...</div>
    <div class="grid12-3">Sample text...</div>
    <div class="grid12-3">Sample text...</div>
    <div class="grid12-3">Sample text...</div>
</div>
```

3 columns (not equal)

Of course columns doesn't have to be equal, you can use grid units in different proportions. But the sum of all units in one row should be equal to 12, otherwise some columns may drop to another row. Here is an example of three columns which are not equal:

```
<div class="grid-container">
    <div class="grid12-7">Sample text...</div>
    <div class="grid12-3">Sample text...</div>
    <div class="grid12-2">Sample text...</div>
</div>
```

Multiple rows

To create multiple rows of columns simply add more units to the container:

```
<div class="grid-container">
     <div class="grid12-6">Sample text...</div>
     <div class="grid12-6">Sample text...</div>
     <div class="grid12-4">Sample text...</div>
     <div class="grid12-4">Sample text...</div>
     <div class="grid12-4">Sample text...</div>
</div>
```

Space between rows

To add even vertical spaces between rows, replace `classgrid-container` with class `grid-container-spaced`:

```
<div class="grid-container-spaced">
     <div class="grid12-6">Sample text...</div>
     <div class="grid12-6">Sample text...</div>
     <div class="grid12-4">Sample text...</div>
     <div class="grid12-4">Sample text...</div>
     <div class="grid12-4">Sample text...</div>
</div>
```

Grid on mobile devices

If the web browser's viewport width is **below 768 px**, all grid columns are stretched (they are as wide as the container) and stack vertically. If you want to avoid this and want your columns to stay narrow on narrow screens, you can add special class `mobile-grid` to grid classes. It can be added to the following grid classes: `grid12-1`, `grid12-2`, `grid12-3`.

On narrow screens columns could be too narrow to display content properly, so thanks to the `mobile-grid` class, each column will be 2 times wider than the actual number of units in the class name, i.e. `grid12-3` will behave like `grid12-6`.

When the viewport width goes **below 480 px**, all grid columns with the `mobile-grid` class will also be stretched and stacked vertically, just like all the other columns.

The following example will create 4 equal columns:

```
<div class="grid-container-spaced">
     <div class="grid12-3 mobile-grid">Sample text...</div>
     <div class="grid12-3 mobile-grid">Sample text...</div>
     <div class="grid12-3 mobile-grid">Sample text...</div>
     <div class="grid12-3 mobile-grid">Sample text...</div>
</div>
```

which will look like this:

When the viewport is below 768 px, the 4 columns will be automatically transformed to 2 rows of 2 columns:

6.5 CMS components

In this chapter you can find description of selected components which can be used to build custom content for CMS pages and static blocks.

6.5.1 CSS utility classes

This chapter presents available utility classes (also called helper classes). Utility classes are simple CSS classes which can be used to modify HTML content of any page or static block. You can use utility classes as an alternative for inline CSS styles.

6.5.1.A General utility classes

Here you can find list of single responsibility classes – each class is responsible for doing one job.

For example, to add a bottom margin below selected text, add the `margin-bottom` utility class to the paragraph element:

```
<p class="margin-bottom">This is a sample text.</p>
Second line of text.
```

If the text is not placed inside any element, wrap the text inside a `<div></div>` element and add the class to that `<div>`:

```
<div class="margin-bottom">
     This is a sample text.
</div>
Second line of text.
```

You can use many classes on a single element. For example, to make the text aligned to the center, add the `text-center` class:

```
<div class="margin-bottom text-center">
```

```
    This is a sample text.
</div>
Second line of text.
```

All available classes are presented below.

Floating

- `pull-left` – float element to the left. This class is an equivalent of the CSS property `float: left`. It specifies that an element should be placed along the left side of its container, where text and inline elements will wrap around it. This way, the element is taken from the normal flow of the web page, though still remaning a part of the flow.
- `pull-right` – float element to the right. This class is an equivalent of the CSS property `float: right`.
- `clearfix` – clear floats. Add this class to the container (the parent element) of the floating elements.

Margin

- `margin-top` – add margin to the top.
- `margin-bottom` – add margin to the bottom.
- `no-margin` – remove all margin.

Padding

- `padding-top` – add padding to the top.
- `padding-bottom` – add padding to the bottom.
- `no-padding` – remove all padding.

Text

- `text-left` – align text to the left.
- `text-right` – align text to the right.
- `text-center` – align text to the center.

Design

- `no-bg` – remove background image from the element.

6.5.1.B Hide/show content depending on screen size

In version 2.4.0 of Ultimo theme we added new utility classes for showing and hiding content depending on the width of the screen. These classes come from the Bootstrap library (Bootstrap version 3.3.7). They will replace the old deprecated classes which we have used in Ultimo theme since the very beginning.

You can still use the old classes, but we highly recommend to use the new classes when you create custom content, e.g. on CMS pages. The new classes offer much more flexibility.

All the information about new utility classes from Bootstrap can be found on the official website: *http://getbootstrap.com/css/#responsive-utilities*
In this chapter you can find only a quick introduction.

Available classes

Try to use these on a limited basis and avoid creating entirely different versions of the same site. Instead, use them to complement each device's presentation.

Use a single or combination of the available classes for toggling content across viewport breakpoints.

	Extra small devices Phones (<768px)	**Small devices Tablets** (≥768px)	**Medium devices** Desktops (≥992px)	**Large devices** Desktops (≥1200px)
`.visible-xs-*`	Visible	Hidden	Hidden	Hidden
`.visible-sm-*`	Hidden	Visible	Hidden	Hidden
`.visible-md-*`	Hidden	Hidden	Visible	Hidden
`.visible-lg-*`	Hidden	Hidden	Hidden	Visible

`.hidden-xs`	Hidden	Visible	Visible	Visible
`.hidden-sm`	Visible	Hidden	Visible	Visible
`.hidden-md`	Visible	Visible	Hidden	Visible
`.hidden-lg`	Visible	Visible	Visible	Hidden

The `.visible-*-*` classes for each breakpoint come in three variations, one for each CSS `display` property value listed below.

Group of classes	CSS display
`.visible-*-block`	`display: block;`
`.visible-*-inline`	`display: inline;`
`.visible-*-inline-block`	`display: inline-block;`

So, for extra small (`xs`) screens for example, the available `.visible-*-*` classes are: `.visible-xs-block`, `.visible-xs-inline`, and `.visible-xs-inline-block`.

Example 1

To hide an element, we can add a class to that element. For example, with the `.hidden-xs` class, this element will be hidden on **extra small** devices – when viewport width is below 768 pixels:

```
<div class="hidden-xs">
     This is sample text. It will be hidden on extra small devices.
</div>
```

Example 2

This element will be hidden on **small** devices – when viewport width is equal or larger than 768 pixels but below 992 pixels:

```
<div class="hidden-sm">
     This is sample text. It will be hidden on small devices.
</div>
```

Example 3

In this example we combine two classes: `.hidden-xs` and `.hidden-sm`. This element will be hidden on **small** and **extra small** devices – when viewport width is below 992 pixels:

```
<div class="hidden-xs hidden-sm">
     This is sample text. It will be hidden on small and extra small devices.
</div>
```

6.5.1.C Old classes to hide elements (deprecated)

These are old **deprecated** classes. If you want to hide elements below specific screen resolution in your custom content (e.g. on CMS pages), we recommend to use the new utility classes which we added to the theme in version 2.4.0 – the classes are described in previous chapter: **6.5.1.B Hide/show content depending on screen size**.

Below you can find description of the old deprecated classes.

These classes can hide elements below specific screen resolution. Number at the end of the class name indicates the resolution:

```
hide-below-1680
hide-below-1440
hide-below-1360
hide-below-1280
hide-below-960
hide-below-768
hide-below-480
hide-below-320
```

For example, add class `hide-below-1280` to the HTML element which needs to be hidden if the screen is less than 1280 pixels wide.

```
<div class="hide-below-1280">
     This is just a sample text
</div>
```

6.5.2 Collapsible blocks

With collapsible block, your content can be shown and hidden on button click. You can build components similar to accordions. Blocks like this are used in the footer static blocks in the demo:

```
<div class="collapsible active">
     <h6 class="block-title">My block title</h6>
     <div class="block-content">Sample content</div>
</div>
```

If you remove the `active` class, the block will be collapsed on page load.

If you want the block to be collapsed **only** on mobile devices (when viewport width is below 768 pixels), use the `mobile-collapsible` class **instead of** the `collapsible` class in the main wrapper:

```
<div class="mobile-collapsible">
     <h6 class="block-title">My block title</h6>
     <div class="block-content">Sample content</div>
</div>
```

6.5.3 Icons

Icons are a simple and effective way to draw users into the content of your website. They can help you structure content and separate different sections of the page. The primary goal of using icons should be to help the user find information on the page.

In this chapter you can find description and examples of all kinds of icons. You can also **import a sample CMS page which contains dozens of ready-to-use code examples**. Refer to chapter **7.11 Data Import** for more details about the import procedure. Identifier of the page with examples of icons is: `magento-icons`.

6.5.3.A Basic icons

You can place icons just about anywhere using simple markup. We are going to use an inline HTML element such as `<span>` and add appropriate classes to it. These are required classes: `ic` and the icon's name prefixed with `ic-`, for example `ic-star`. Here's the full example of the code which will add star icon:

```
<span class="ic ic-star"></span>
```

If you change the font-size of the icon's container, the icon gets bigger. Same things goes for color, drop shadow, and anything else that gets inherited using CSS.

> **Note:**
>
> list of names of all available icons can be found in `Resources/Icons_Demo/demo.html` inside the theme package.

6.5.3.A.1 Add CSS styles to icons

Now you can start having more fun with icons. By default all icons have the same color as text, but if you want to change the color of selected icon, you can do it with **inline styles**. Add `style` attribute to the icon element and specify the color. For example, this will make the icon red:

```
<span class="ic ic-star" style="color:red;"></span>
```

You can add inline styles to icons the same way as to any other HTML elements in a HTML document. The `style` attribute can contain any CSS property, such as font-size, text-shadow etc. List of available CSS properties can be found on *https://developer.mozilla.org/en-US/docs/Web/CSS/Reference*

6.5.3.A.2 Size

If you change the font-size of the icon's container, the icon gets bigger. To increase icon size relative to the font-size of the icon's container, use the following classes: `ic-lg` (increases the size of the icon by 33%), `ic-2x`, `ic-3x`, `ic-4x`, `ic-5x`, `ic-6x`, `ic-7x`, `ic-8x`.

```
<span class="ic ic-star ic-lg"></span>
<span class="ic ic-twitter ic-2x"></span>
<span class="ic ic-heart-o ic-3x"></span>
```

To have more control over the size of the icon, you can also change the font size of the icon element by adding inline styles:

```
<span class="ic ic-star" style="font-size:32px;"></span>
```

6.5.3.A.3 List icons

Use `ic-ul` and `ic-li` to build unordered lists with icons displayed as bullets.

```
<ul class="ic-ul">
     <li><span class="ic ic-star ic-li"></span>Some text example</li>
     <li><span class="ic ic-tag ic-li"></span>Some text example</li>
     <li><span class="ic ic-arrow-right ic-li"></span>Some text example</li>
     <li><span class="ic ic-letter ic-li"></span>Some text example</li>
     <li><span class="ic ic-bell ic-li"></span>Some text example</li>
</ul>
```

6.5.3.A.4 Pulled icons and borders

Use `ic-border` and `ic-pull-left` or `ic-pull-right` for easy pull article icons.

```
<span class="ic ic-alarm ic-3x ic-border pull-left "></span> Some sample text. Lid est laborum
et dolorum fuga. Et harum quidem rerum facilis est et expeditasi distinctio.
```

6.5.3.A.5 Spinning icons

Use the `ic-spin` class to get any icon to rotate.

```
<span class="ic ic-reload ic-spin"></span>
```

6.5.3.A.6 Rotated and flipped

To rotate and flip icons, use the `ic-rotate-*` and `ic-flip-*` classes.

```
<span class="ic ic-twitter"></span> normal<br>
```

```
<span class="ic ic-twitter ic-rotate-90"></span> ic-rotate-90<br>
<span class="ic ic-twitter ic-rotate-180"></span> ic-rotate-180<br>
<span class="ic ic-twitter ic-rotate-270"></span> ic-rotate-270<br>
<span class="ic ic-twitter ic-flip-horizontal"></span> ic-flip-horizontal<br>
<span class="ic ic-twitter ic-flip-vertical"></span> ic-flip-vertical
```

6.5.3.A.7 Icons inside buttons

Font icons work great inside buttons and other elements.

```
<a class="button" href="#"><span class="ic ic-user"></span>  My Account</a>
```

6.5.3.B Iconbox (icon with background color)

To display an icon inside a box with background color (we call it an **iconbox**), add the `ib` class to the icon element:

```
<span class="ic ic-star ib"></span>
```

Background color will be automatically added to the icon element. Make sure to leave the `<span>` tag empty – otherwise the contents of the tag will be displayed together with the icon inside the iconbox and any additional space can dislocate the icon.

> **Note:**
>
> The default colors of the iconbox can be configured in the admin panel:
> *Stores > Configuration > Theme Design > Colors > **Iconbox***

6.5.3.B.1 Change color on mouse hover

To change the color on mouse hover over the iconbox, add the `ib-hover` class to the iconbox.

```
<span class="ic ic-star ib ib-hover"></span>
```

6.5.3.B.2 Size

To increase iconbox size, use the following classes:
`ib-size-l`, `ib-size-xl`, `ib-size-xxl`, `ib-size-xxxl`. For example:

```
<span class="ic ic-star ib ib-size-xxl"></span>
```

The size of the icon is independent of the iconbox and can be increased with classes which were described earlier in this chapter. For example, add class `ic-2x` to make the icon a little bit bigger:

```
<span class="ic ic-star ic-2x ib ib-size-xxl"></span>
```

6.5.3.B.3 Shape

To change the shape of the iconbox, use one of the following classes:
`ib-circle`, `ib-rounded`, `ib-square`. For example:

```
<span class="ic ic-star ib ib-square"></span>
```

By default the iconbox is circular. If you prefer square icons, you can wrap multiple icons with an element (it can be `<div>`) with class `ib-wrapper-square`:

```
<div class="ib-wrapper-square">
     <span class="ic ic-star ib"></span>
     <span class="ic ic-heart ib"></span>
     <span class="ic ic-letter ib"></span>
</div>
```

6.5.3.B.4 Iconbox effects

To add eye-catching hover effects to the iconbox, use one of the following combinations of classes:

- `ib-ef-1 ib-ef-1a`
- `ib-ef-1 ib-ef-1b`
- `ib-ef-2 ib-ef-2a`
- `ib-ef-2 ib-ef-2b`
- `ib-ef-3 ib-ef-3a`
- `ib-ef-3 ib-ef-3b`

Note that in each case the combination consists of two classes. For example:

```
<span class="ic ic-star ib ib-ef-1 ib-ef-1a"></span>
```

6.5.3.B.5 Image file iconbox

You can also use image files to create iconboxes with custom icons. With the following markup: `<span>` tag with classes `ic ic-img ib` and `<img>` tag inside the `<span>` tag, you can quickly create custom icon:

```
<span class="ic ic-img ib">
     <img src="{{media url="wysiwyg/infortis/other/image-icons/example.png"}}" alt="My icon" />
</span>
```

To get the URL of the image, you need to use Magento's CMS tag `media`. Magento will replace the tag with the URL of the image, for example: `{{media url="wysiwyg/path/to/image.png"}}`. The path is relative to Magento's `media` directory, so you need to store your images inside that directory.

Recommended size of an image: 30 x 30 pixels. An image can be larger but it will be scaled down to fit the size of the iconbox. So it is better to use images of correct size to keep good quality.

6.5.3.C Social icons

To add icons of social networking services you need to use exactly the same markup as for other icons described earlier in this chapter. Name of the service needs to be prefixed with `ic-`, for example `ic-twitter`. Here's the full example of the code which will add a Twitter icon:

```
<span class="ic ic-twitter"></span>
```

> **Note:**
>
> list of names of all available icons can be found in `Resources/Icons_Demo/demo.html` inside the theme package.

Icons can be wrapped inside links. This is very useful when you want to create links to your profiles on social networking services. Here's an example:

```
<a href="https://twitter.com/infortis">
     <span class="ic ic-twitter"></span>
</a>
```

You can also add iconbox classes (`ib` and `ib-hover`, refer to **6.5.3.B Iconbox (icon with background color)** for more details) to create an icon with background color, for example:

```
<a href="https://twitter.com/infortis">
     <span class="ib ib-hover ic ic-twitter"></span>
</a>
```

If you additionally wrap the icon inside a container with the `social-links` class, you will be able to configure colors through the admin panel. The default colors for social icons can be configured under:
*Stores > Configuration > Theme Design > Colors > **Iconbox (Social Icons)***

In the example below, replace `#` with URLs of your profiles on social networking services:

```
<div class="social-links">
     <a href="#">
             <span class="ib ib-hover ic ic-lg ic-twitter"></span>
     </a>
     <a href="#">
             <span class="ib ib-hover ic ic-lg ic-facebook"></span>
     </a>
     <a href="#">
             <span class="ib ib-hover ic ic-lg ic-googleplus"></span>
     </a>
     <a href="#">
             <span class="ib ib-hover ic ic-lg ic-youtube"></span>
     </a>
     <a href="#">
             <span class="ib ib-hover ic ic-lg ic-vk"></span>
     </a>
     <a href="#">
             <span class="ib ib-hover ic ic-lg ic-renren"></span>
     </a>
     <a href="#">
             <span class="ib ib-hover ic ic-lg ic-instagram"></span>
     </a>
</div>
```

6.5.3.D Letters, numbers and other characters as icons

By adding class `ic-char` to the icon, you can display any characters (such as letters, numbers and other special characters) as icons. The code is almost the same as for standard font icons, but inside the `<span>` tag you need to add a single character. For example:

```
<span class="ic ic-char ic-2x ib">1</span>
<span class="ic ic-char ic-2x ib">2</span>
<span class="ic ic-char ic-2x ib">A</span>
<span class="ic ic-char ic-2x">@</span>
<span class="ic ic-char ic-2x">$</span>
<span class="ic ic-char ic-2x">%</span>
```

6.5.3.E Font Awesome icons

Font Awesome is a font and icon toolkit based on CSS. It offers a collection of more than 600 vector icons which can be easily customized (the same as other font icons available in the theme).

All the information about Font Awesome can be found on the official website:
http://fontawesome.io/examples/
In this chapter you can find only a quick introduction.

How to disable Font Awesome icons

Font Awesome icons are enabled by default. To disable Font Awesome:

1. Open this file: `<YOUR_MAGENTO_INSTALL_DIRECTORY>/app/design/frontend/Infortis/ultimo/web/css/_custom.less`

2. Set the `@fontawesome-enable` variable to `false`, like this:

```
@fontawesome-enable: false;
```

3. Rebuild Magento static files – just run the following Magento command:

```
bin/magento setup:static-content:deploy
```

4. Flush the cache.

If you correctly rebuild Magento static files, the CSS styles will be recompiled and the Font Awesome will be disabled.

For more details about editing CSS styles, refer to chapter **5.1.1.A Use _custom.less file**.

Basic Font Awesome icons

To create an icon, use the `fa` class and the icon's name with an inline HTML element `<span>`. Here's an example of the code which will create a flag icon:

```
<span class="fa fa-flag fa-3x"></span>
```

Use Font Awesome icons with other icon classes

You can use Font Awesome icons together with other icon classes described in this document. Here's an example of an **iconbox** element (the `ib` class) with Font Awesome icon inside a block:

```
<div class="feature feature-icon-hover indent indent-size-l">
     <span class="ic ic-2x ib ib-size-l left fa fa-flag"></span>
     <h4>Heading Example</h4>
     <p>This is a short paragraph of sample text inside a block.</p>
</div>
```

6.5.4 Icon + text

Icons can help you structure content and separate different sections of the page. The primary goal of using icons should be to help the user find information on the page and with icons you can quickly sum up what your text is about. For example, when you build lists, instead of using standard bullets, you can use icons to draw attention to paragraphs and other blocks of content.

In this chapter you can find information about creating blocks of text with icons. You can also **import a sample CMS page which contains dozens of ready-to-use code examples**. Refer to chapter **7.11 Data Import** for more details about the import procedure. Identifier of the page with examples is: `magento-icons`.

6.5.4.A Simple block with icon

To create a simple block of text with an icon, wrap your text inside a `<div>` element with the `feature` class. Here's a minimal example:

```
<div class="feature">
     <span class="left ic ic-star"></span>
     <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
</div>
```

If you add `left` or `right` class to the icon element, the icon will be taken from the normal flow and placed along the left or right side of its container, and text will wrap around it.

Using this markup you can quickly build all kinds of blocks (with text paragraphs, headings, links etc.) enhanced with icons. For more information about creating icons, refer to chapter **6.5.3 Icons**.

Here's another, more complex example with additional headings and a bigger icon inside an iconbox:

```
<div class="feature">
```

```
     <span class="left ic ic-star ic-2x ib ib-size-xl"></span>
     <h6 class="above-heading">Above heading</h6>
     <h3>Block Heading Example</h3>
     <h6 class="below-heading">Text below heading</h6>
     <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
     <p>Example of another text paragraph inside a block. Icons are an effective way to draw
users into the content of your store.</p>
     <a href="#">Read more...</a>
</div>
```

6.5.4.B Change iconbox color on mouse hover

To change the background color of the **iconbox** on mouse hover over the entire block, add the `feature-icon-hover` class to the block element:

```
<div class="feature feature-icon-hover">
     <span class="left ic ic-star ib"></span>
     <h3>Block Heading Example</h3>
     <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
</div>
```

The default colors of the iconbox can be configured in the admin panel.
Refer to **6.5.3.B Iconbox (icon with background color)** for details.

6.5.4.C Indented block

To display a block with indentation on the left side, add the `indent` class to the block element:

```
<div class="feature feature-icon-hover indent">
     <span class="left ic ic-star ib"></span>
     <h3>Block Heading Example</h3>
     <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
</div>
```

If you increase the iconbox size (by adding a class such as `ib-size-xxl`), you will also need to add corresponding class (in this case: `indent-size-xxl`) to the block element. It will adjust the size of the indentation, for example:

```
<div class="feature feature-icon-hover indent indent-size-xxl">
     <span class="left ic ic-star ib ib-size-xxl"></span>
     <h3>Block Heading Example</h3>
     <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
</div>
```

To change the size of the indentation, use the following classes: `indent-size-l`, `indent-size-xl`, `indent-size-xxl`, `indent-size-xxxl`.

6.5.4.D Centered block

To align elements of the block to the center, use class `centered`. Like this:

```
<div class="feature centered">
     <span class="ic ic-star ib ib-size-xl"></span>
     <h3>Block Heading Example</h3>
     <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
</div>
```

6.5.5 Banners

Image banners can be useful in many different areas of your store. You can place banners just about anywhere using simple markup. With additional utility classes you can easily add captions and multiple hover effects to your banners.

In this chapter you can find description and examples of all kinds of banners. You can also **import a sample CMS page which contains dozens of ready-to-use code examples**. Refer to chapter **7.11 Data Import** for more details about the import procedure. Identifier of the page with sample banners is: `magento-banners`.

6.5.5.A Image banner

The image and the entire content of the banner is wrapped inside an element with the `ban` class. The image element needs to be marked with the `image` class. Here's a minimal code example to create a simple banner:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
</div>
```

6.5.5.A.1 How to get the image

To get the URL of the image, you need to use the **Media URL shortcode**. Magento will automatically replace it with the image URL during page rendering.

The image path is relative to Magento's `media` directory, so you need to store your images inside that directory.

For example, if this is the path of your image: `media/wysiwyg/path/to/image.png`, you can get the URL of that image with the following shortcode:

```
{{media url="wysiwyg/path/to/image.png"}}
```

Here's how you can use that shortcode in the `src=""` attribute of the `<img>` element to display your image on a page:

```
<img class="image" src="{{media url="wysiwyg/path/to/image.png"}}" alt="My image" />
```

For more details about the **Media URL shortcode**, refer to chapter: **4.8.3.B Media URL**.

6.5.5.B Banner with hyperlink

To create a banner with a hyperlink to other pages (or to locations within the same page), wrap the banner inside the anchor element `<a>`. Inside the `href=""` attribute put a URL that the hyperlink points to. You can wrap the entire banner, or only the caption.

To get the URL of the page, you need to use the **Store URL shortcode**. For example:

```
<a href="{{store url="about-us"}}">
     <div class="ban">
             <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
     </div>
</a>
```

Replace `about-us` with the **URL Key** of any page in your store. Path is relative to the store URL. For example, if you want to create a link to the category page which have the following URL in your store: `www.example.com/cell-phones`, simply replace `about-us` with `cell-phones`:

```
<a href="{{store url="cell-phones"}}">
     <div class="ban">
             <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
     </div>
</a>
```

For more details about the **Store URL shortcode**, refer to chapter: **4.8.3.A Store URL**.

6.5.5.C Banner with image hover effect

To add a hover effect to the banner, add an effect class to the banner element (to the element with the `ban` class). The following classes are available: `ban-effect-1`, `ban-effect-2`, `ban-effect-3`, `ban-effect-fade-out`, `ban-effect-grayscale-in`, `ban-effect-grayscale-out`, `ban-effect-blur-in`, `ban-effect-blur-out`.

```
<div class="ban ban-effect-1">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
</div>
```

6.5.5.D Banner with color overlay

To add a color overlay to the banner, add a caption but without any text. Captions are described with more details in chapter **6.5.5.E Banner with caption**, but in this section we will only use an empty caption element.

To add an empty caption to the banner, simply add an element with the `cap` class inside the banner: `<div class="cap"></div>`. The empty caption element will be displayed as a color overlay. Here's an example:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
     <div class="cap"></div>
</div>
```

Note:

The default color of the overlay can be configured in the admin panel:

> Stores > Configuration > Theme Design > Colors

Change the color of the overlay

To change the color of the overlay, you can also add an inline CSS to the caption element to override the default value of its `background-color` property. You can add semi-transparent color using the `rgba()` functional notation allowing specification of the opacity of a color. For example:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"  />
     <div class="cap" style="background-color: rgba(135, 195, 30, 0.45);"></div>
</div>
```

Color overlay with fade in/out effect

By default the color overlay (the caption element) covers the banner all the time. But you can change it with two useful classes. The `ban-caption-fade-out` class will make the overlay fade out on mouse hover over the banner. And the `ban-caption-fade-in` class will make the overlay fade in only on mouse hover over the banner. Add these classes to the banner element, **not** to the caption element.

```
<div class="ban ban-caption-fade-out">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample01.jpg"}}"
alt="Banner1" />
     <div class="cap" style="background-color: rgba(245, 88, 86, 0.75);"></div>
</div>

<div class="ban ban-caption-fade-in">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample01-c.jpg"}}"
alt="Banner2" />
     <div class="cap" style="background-color: rgba(0, 0, 0, 0.85);"></div>
</div>
```

6.5.5.E Banner with caption

To create a caption for the banner, add an element with the `cap` class inside the banner. Inside the caption you can add any text elements, such as headings or paragraphs of text. The caption (with a color overlay) will be displayed over the image.

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
     <div class="cap">
             <h3>Caption Example</h3>
             <p>This is a sample text</p>
     </div>
</div>
```

> **Note:**
>
> The default color of the overlay can be configured in the admin panel:
> *Stores > Configuration > Theme Design > Colors*

You can extend this simple example with additional features:

- by adding more CSS classes (described in this chapter) to the banner and to the caption element
- by adding more HTML (e.g. icons) inside the caption

6.5.5.E.1 Inline styles

If you want to change the style of the caption, e.g. change the font size or the color, you can do it with **inline CSS styles**. Add the `style` attribute to the caption element and specify values of selected properties such as `font-size`, `font-family`, `color`, `background-color` etc.

In the example below, we override the default values of the `color` and the `background-color` properties of the caption element and the font-size property of `<h2>` and `<p>` elements. To make the background color semi-transparent, use the `rgba()` functional notation.

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
     <div class="cap" style="color: #000; background-color: rgba(255, 254, 84, 0.6);">
             <h2 style="font-size: 32px;"><span class="ic ic-lightbulb"></span> Caption
Example</h2>
             <p style="font-size: 18px;">Create responsive banners in your Magento with
ease.</p>
     </div>
</div>
```

6.5.5.E.2 Remove caption background color

To get rid of the background color from the caption, add the `cap-no-bg` class to the caption element. This will make the caption element transparent, only the text will be visible over the banner:

```
<div class="ban ban-effect-1">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
     <div class="cap cap-no-bg">
             <h2 style="font-size: 32px;"><span class="ic ic-star"></span> Caption Example</h2>
             <p style="font-size: 18px;">Create responsive banners in your Magento with
ease.</p>
     </div>
</div>
```

6.5.5.E.3 Caption with fade in/out effect

By default the caption element covers the banner all the time. But you can change it with two useful classes. The ban-caption-fade-out class will make the caption fade out on mouse hover over the banner. And the ban-caption-fade-in class will make the caption fade in on mouse hover over the banner.

> **Note:**
>
> Add these classes to the banner element, **not** to the caption element.

Here's the code example of the caption with fade out effect:

```
<a href="#">
     <div class="ban ban-caption-fade-out">
             <img class="image" src="{{media url="wysiwyg/infortis/banners/sample02.jpg"}}"
alt="Banner" />
             <div class="cap">
                     <h4>This caption will fade out on hover.</h4>
             </div>
     </div>
</a>
```

6.5.5.E.4 Center caption

To center the caption horizontally, add the `cap-center-horizontally` class to the caption element. To center the caption vertically, add the `cap-center-vertically` class to the caption element. Both classes can be used together.

Here's the code example of the caption centered horizontally and vertically:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample02.jpg"}}"
alt="Banner" />
     <div class="cap cap-center-vertically cap-center-horizontally">
             <h2>Caption Example</h2>
             <p>Create responsive banners in your Magento with ease</p>
     </div>
</div>
```

6.5.5.E.5 Center vertically, full cover

When the caption is centered vertically, if you want the caption to cover the entire banner, you need to use **two nested caption elements**. The **inner** caption element needs to have the `cap-center-vertically` class. The additional `cap-no-bg` class will remove the background color from the inner caption element:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample02.jpg"}}"
alt="Banner" />
     <div class="cap">
             <div class="cap cap-center-vertically cap-no-bg">
                     <h2>Caption Example</h2>
                     <p>Create responsive banners in your Magento with ease</p>
             </div>
```

```
    </div>
</div>
```

6.5.5.E.6 Caption positioning

To set the position of the caption, use the following classes: `cap-top`, `cap-bottom`, `cap-left`, `cap-right`, `cap-top-left`, `cap-top-right`, `cap-bottom-left`, `cap-bottom-right`.

Here's a code example of a caption placed at the bottom of the banner:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample02.jpg"}}"
alt="Banner" />
     <div class="cap cap-bottom">
             <h3 class="no-margin">Caption Example</h3>
     </div>
</div>
```

6.5.5.E.7 Precise positioning

To position the caption with more precision, you can use the classes presented below. The number at the end of each class name indicates the value (percentage) by which the caption will be moved in the selected direction. In the **vertical direction** the percentage is calculated with respect to the height of the entire banner. In the **horizontal direction** the percentage is calculated with respect to the width of the entire banner.

For example, the `cap-push-down-10` class will push the caption down by 10%. Here's the list of all available classes:

Push down	Push up	Push left	Push right
`cap-push-down-5`	`cap-push-up-5`	`cap-push-left-5`	`cap-push-right-5`
`cap-push-down-10`	`cap-push-up-10`	`cap-push-left-10`	`cap-push-right-10`
`cap-push-down-15`	`cap-push-up-15`	`cap-push-left-15`	`cap-push-right-15`
`cap-push-down-20`	`cap-push-up-20`	`cap-push-left-20`	`cap-push-right-20`

cap-push-down-25	cap-push-up-25	cap-push-left-25	cap-push-right-25
cap-push-down-30	cap-push-up-30	cap-push-left-30	cap-push-right-30

Below you can see an example of a caption which is pushed up by 20% and pushed right by 15%.

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/wide01.jpg"}}"
alt="Banner" />
     <div class="cap cap-push-up-20 cap-push-right-15">
             <h2>Example of Banner Caption</h2>
             <p>Create custom banners with captions</p>
     </div>
</div>
```

6.5.5.E.8 Caption effects

To add eye-catching hover effects for the caption, add classes listed below to the banner element:

Fade	Slide	Push	Hinge
ban-caption-fade-out	ban-caption-slide-up	ban-caption-push-up	ban-caption-hinge-up
ban-caption-fade-in	ban-caption-slide-down	ban-caption-push-down	ban-caption-hinge-down
	ban-caption-slide-left		ban-caption-hinge-left
	ban-caption-slide-right		ban-caption-hinge-right

Note:

> Add these classes to the banner element, **not** to the caption element.

Here's a code example of a banner with caption "hinge up" effect:

```
<div class="ban ban-caption-hinge-up">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample01.jpg"}}"
alt="Banner" />
     <div class="cap">
             <h3>Caption Example</h3>
             <p>Create custom banners with captions</p>
     </div>
</div>
```

6.5.5.E.9 Caption with text background

To create a caption with background color behind each block of text, you need to make a few steps:

1. Mark each block of text inside the caption element with the `text` class.
2. Add these two classes to the caption element:
   - `cap-no-bg` class which removes the background color from the caption element
   - `cap-text-bg` class which adds the background color to text blocks
3. Add one of these classes to the caption element to choose the tone of the text background - dark (semi-transparent black) or light (semi-transparent white):
   - `cap-text-bg-dark-1`
   - `cap-text-bg-dark-2`
   - `cap-text-bg-dark-3`
   - `cap-text-bg-light-1`
   - `cap-text-bg-light-2`
   - `cap-text-bg-light-3`

Here's an example of a caption with dark text background:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/sample.jpg"}}"
alt="Banner" />
     <div class="cap cap-no-bg cap-text-bg cap-text-bg-dark-3">
             <h2 class="text">Caption Example</h2>
             <p class="text">Create customizable banners with captions</p>
             <p class="text">Responsive Magento theme</p>
     </div>
</div>
```

6.5.5.E.10 Old classes to create captions for slideshow (deprecated)

These are old deprecated classes. If you want to create feature-rich banners with captions, we recommend to use the new system of banners which was added to the theme in version 2.5.0. Refer to chapter **6.5.5 Banners** and **6.5.5.E Banner with caption** for more details.

Below you can find description of the **old deprecated classes**.

Presented code can be used to create captions inside the homepage Slideshow (inside the main slides and inside small banners at the side of the Slideshow).

To set background color, use one of the following classes: `light1`, `light2`, `light3`, `dark1`, `dark2`, `dark3`. For example:

```
    <div class="caption light1">
        <p>Lorem ipsum dolor sit amet</p>
    </div>
```

To position caption, use the following classes (by default captions is positioned in the bottom left corner): `top`, `top-right`, `bottom-right`. For example:

```
    <div class="caption light1 top-right">
```

```
    <h2 class="heading">Big Heading</h2>
    <p>Lorem ipsum dolor sit amet</p>
</div>
```

Additionally, the paragraph of text inside the caption can be aligned right with class: `right`, or centered with class: `centered`. For example:

```
<div class="caption dark1 centered">
    <h2 class="heading">Big Heading</h2>
    <p>Lorem ipsum dolor sit amet</p>
</div>
```

To stretch the caption, use class: `full-width`. To make it narrower, use: `narrow`. For example:

```
<div class="caption dark3 full-width">
    <p>Lorem ipsum dolor sit amet dictum sit altum.</p>
</div>
```

On narrow screens elements of the caption will be hidden. To keep selected elements visible, use class: `permanent`. For example:

```
<div class="caption light1">
    <h2 class="heading permanent">Heading always visible</h2>
    <p>This paragraph will be hidden on mobile</p>
</div>
```

6.5.5.F Groups of banners

To create groups of banners and various layouts (with multiple rows and columns) you can use the grid system which is described in this chapter: **6.4 Grid system**.

For example, to create two equal-width banners displayed side by side, create two columns using the grid system. Here's the markup for columns:

```
<div class="row">
     <div class="col-md-6">
             ...Column 1...
     </div>
     <div class="col-md-6">
             ...Column 2...
     </div>
</div>
```

Now, add a banner inside each column:

```
<div class="row">
     <div class="col-md-6">
             <div class="ban">
                     <img class="image" src="{{media
url="wysiwyg/infortis/banners/sample02.jpg"}}" alt="Banner 1" />
             </div>
     </div>
     <div class="col-md-6">
             <div class="ban">
                     <img class="image" src="{{media
url="wysiwyg/infortis/banners/sample02.jpg"}}" alt="Banner 2" />
             </div>
     </div>
</div>
```

6.5.6 Animations

This chapter presents Animate.css – a cross-browser library of CSS animations. With simple CSS classes this library will let you add some life to the content of your store. All the information about the library can be also found in the official documentation: https://github.com/daneden/animate.css

You can use the animations, for example, to create eye-catching icons on promotional banners, but the animations are especially useful with the **Slideshow** module – all details can be found in chapter **10.3.3 Animated content of a slide**.

6.5.6.A Basic usage

To add an animation, you need to follow these steps:

1. Add the `animated` class to the element you want to animate.
2. You may also want to include the optional `infinite` class to create an infinite loop. Without it, the animation will stop after the first cycle is finished (most animations take 1 second to complete one cycle).
3. Add class which indicates the name of the animation, for example: `pulse`. Full list of available animation can be found later in this chapter: **6.5.6.B List of animations**.

Let's test it on a simple icon. Here's the code of the icon before adding animations:

```
<span class="ic ic-heart-o ic-5x"></span>
```

And here's how it should look like with our animation classes. The `pulse` class is the name of the animation, it will add pulsing effect to the icon:

```
<span class="ic ic-heart-o ic-5x animated infinite pulse"></span>
```

Animations are triggered on page load and last only a short moment so in most cases it's good idea to use the `infinite` class to create an infinite loop.

6.5.6.B List of animations

Here's the list of class names of all available animations. All the animations can be also tested on this website:
https://daneden.github.io/animate.css/

- `bounce`
- `flash`
- `pulse`
- `rubberBand`
- `shake`
- `headShake`
- `swing`
- `tada`
- `wobble`
- `jello`
- `bounceIn`
- `bounceInDown`
- `bounceInLeft`
- `bounceInRight`
- `bounceInUp`
- `bounceOut`
- `bounceOutDown`
- `bounceOutLeft`
- `bounceOutRight`
- `bounceOutUp`
- `fadeIn`
- `fadeInDown`
- `fadeInDownBig`
- `fadeInLeft`
- `fadeInLeftBig`
- `fadeInRight`
- `fadeInRightBig`
- `fadeInUp`
- `fadeInUpBig`
- `fadeOut`
- `fadeOutDown`
- `fadeOutDownBig`
- `fadeOutLeft`
- `fadeOutLeftBig`
- `fadeOutRight`
- `fadeOutRightBig`
- `fadeOutUp`
- `fadeOutUpBig`
- `flipInX`
- `flipInY`
- `flipOutX`
- `flipOutY`
- `lightSpeedIn`
- `lightSpeedOut`
- `rotateIn`
- `rotateInDownLeft`
- `rotateInDownRight`
- `rotateInUpLeft`
- `rotateInUpRight`
- `rotateOut`
- `rotateOutDownLeft`
- `rotateOutDownRight`
- `rotateOutUpLeft`
- `rotateOutUpRight`
- `hinge`
- `rollIn`
- `rollOut`
- `zoomIn`
- `zoomInDown`
- `zoomInLeft`
- `zoomInRight`
- `zoomInUp`
- `zoomOut`
- `zoomOutDown`
- `zoomOutLeft`
- `zoomOutRight`
- `zoomOutUp`
- `slideInDown`
- `slideInLeft`
- `slideInRight`
- `slideInUp`
- `slideOutDown`
- `slideOutLeft`
- `slideOutRight`
- `slideOutUp`

6.5.6.C Additional options

6.5.6.C.1 Animation delay

Animations are triggered immediately on page load. If you want to delay the start of the animation, you can use one of the additional `delay-*` classes presented below. The number at the end of each class name indicates the value (in seconds) by which the animation will be delayed.

For example, class `delay-3` will delay the animation start by **3 seconds**:

```
<span class="ic ic-star ic-3x animated infinite flash delay-3"></span>
```

If we add animations to more than one element, we can use different delay classes on each element to get more interesting effects. In this example, the first icon will start animation after **0.5 seconds** and the second icon will start after **1 second**:

```
<span class="ic ic-heart-o ic-3x animated infinite pulse delay-0-5"></span>
<span class="ic ic-heart-o ic-3x animated infinite pulse delay-1"></span>
```

Here's the list of all available classes:

- `delay-0-5`
- `delay-1`
- `delay-1-5`
- `delay-2`
- `delay-2-5`
- `delay-3`
- `delay-3-5`
- `delay-4`
- `delay-4-5`
- `delay-5`
- `delay-6`
- `delay-7`
- `delay-8`
- `delay-9`
- `delay-10`

6.5.6.C.2 Animation duration

Most animations take **1 second** to complete one cycle. To make an animation last longer, you can use one of the additional `duration-*` classes presented below. These classes specify the length of time that an animation should take to complete one cycle. The number at the end of each class name indicates the time (in seconds).

For example, with the `duration-3` class the animation will take **3 seconds** to complete one cycle:

```
<span class="ic ic-star ic-3x animated infinite flash duration-3"></span>
```

While with the `duration-0-5` class the animation will take only **0.5 seconds** to complete one cycle:

```
<span class="ic ic-star ic-3x animated infinite flash duration-0-5"></span>
```

Here's the list of all available classes:

- `duration-0-5`
- `duration-1`
- `duration-2`
- `duration-3`
- `duration-4`
- `duration-5`

6.6 Theme customization

This chapter was moved to: **5. Customization**

7. Theme settings

This chapter refers to the following section of the theme admin panel:
Stores > Configuration > Infortis Themes - Configuration > Theme Settings

> **Important:**
>
> beginning from this chapter, each chapter refers to a specific sections of theme admin panel. Path of the admin panel section is visible at the top of each page.

7.1 Header

This chapter refers to the following section of the theme admin panel:
Stores > Configuration > Theme Settings > Header

List of all settings available in that section can be found later in this chapter: **7.1.3 Header configuration options**.

7.1.1 Header structure

Width of the header sections (Left Column, Central Column, Right Column) is specified in grid units. As this is a 12-column grid, each section spans a number of those 12 columns, and should always add up to 12.

Example 1:

If you want to display three equal sections (Left Column, Central Column, Right Column), sum of these three sections has to be equal to 12 grid units. For example:

```
Left Column = 4, Central Column = 4, Right Column = 4
4 + 4 + 4 = 12
All the three sections will have equal width.
```

Example 2:

If you want to get rid of one section, e.g. Central Column, leave the **Central Column** field empty and make the Right Column wider. For example:

```
Left Column = 4, Central Column = leave empty, Right Column = 8
4 + ... + 8 = 12
```

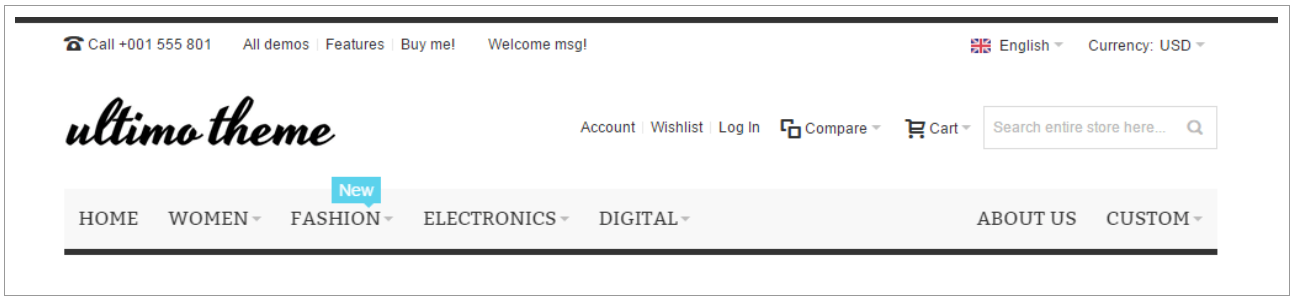
Thanks to that the Central Column will disappear and the Right Column will be 2 times wider than the Left Column:

Now you need to assign the search box to the Right Column instead of the default Central Column. The Central Column is not displayed so any blocks assigned to the Central Column will also not be displayed.

> **Important:**
>
> if given header section is not displayed, then any blocks assigned to that section will also not be displayed.

To assign search box to the Right Column, in the **Position** field under `Stores > Configuration > Theme Settings > Header > Search` select “Inside User Menu...” and then in the **Position Inside User Menu** field select where the search box should be

displayed. The final effect can look like on the following screenshot:



7.1.2 Header block positions

In section `Stores > Configuration > Theme Settings > Header` of admin panel you can select where to display some of the header blocks. Below you can see a list of all available positions:

1. Top, Left
2. Top, Right
3. Primary, Left Column
4. Primary, Central Column
5. Primary, Right Column
6. Inside User Menu

7.1.3 Header configuration options

In this chapter you can find a list of options available in the following section of the theme admin panel: `Stores > Configuration > Theme Settings > Header`.

Width of Primary Header Columns

- **Left Column** – width of the left section. Specify value in grid units (between 2 and 12). Important: Leave empty to remove this column.
- **Central Column** – width of the central section. Specify value in grid units (between 2 and 12). Important: Leave empty to remove this column.
- **Right Column** – width of the right section. Specify value in grid units (between 2 and 12). Important: Leave empty to remove this column.

Logo

- **Position** – select position of the site logo. Screenshot with all available positions can be found in this chapter: **7.1.2 Header block positions**.

Search

- **Position** – select position of the search box. Screenshot with all available positions can be found in this chapter: **7.1.2 Header block positions**. Select `Inside User Menu...` to show another option (see below).
- **Position Inside User Menu** – this field is visible only if `Inside User Menu...` was selected in the field above. Select position of the search box **inside** the User Menu.

Account Links

This block is a container that groups Magento's Top Links (a set of links: *Account*, *Wish List*, *Log In/Log Out*) and optional custom links. You can add custom links by creating a static block with identifier `block_header_top_links`. The content of this block will be displayed just next to the Top Links. Also, note that the Top Links can be disabled in theme settings, refer to TODO or more details.

- **Position** – select position of the Account Links.
- **Include Top Links** – disable to remove the Top Links from the Account Links

block.

User Menu

This block is a container that groups a few elements. By default it contains the following blocks: Mini Cart drop-down, Account Links.

- **Position** – select position of the User Menu. Screenshot with all available positions can be found in this chapter: **7.1.2 Header block positions**.
- **Position of Additional Line Break** – e.g. add a line break after the Mini Cart to force other elements of the User Menu to be displayed in the new line (below the Mini Cart). Note that elements of the User Menu are rendered in the following order: Mini Cart, Top Links.

Main menu

- **Position** – select position of the main menu. Screenshot with all available positions can be found in this chapter: **7.1.2 Header block positions**. If `Full Width Menu Container` was selected, menu will be displayed at the bottom of the header.

Mini Cart Drop-down

- **Position** – select position of the Mini Cart. Screenshot with all available positions can be found in this chapter: **7.1.2 Header block positions**.
- **Show Text Label "Cart"** – if set to `No`, label "Cart" will be hidden in the Mini Cart drop-down block.

> **Note:**
>
> that in order to display Mini Cart drop-down block, option **Display Shopping Cart Sidebar** under *Stores > Configuration > Checkout > Shopping Cart Sidebar* has to be enabled in your Magento.

Currency Switcher

- **Position** – select position of the Currency Switcher. If you configure more than one currency in your Magento, the Currency Switcher switcher will appear automatically. You just need to configure currencies in your Magento and then clear Magento cache.

Language Switcher

- **Position** – select position of the Language Switcher.

  The Language Switcher is just a store view switcher which is a standard feature of Magento. You can, for example, configure 3 store views (each with different language) in your Magento and the store view switcher will work as a language switcher.

Sticky Header

- **Sticky Header** – if enabled, the main menu will not disappear when the user scrolls down the page.

- **Full Width Sticky Header** – if set to `Yes`, content of the sticky header will be stretched to full width.

- **Show Logo in Sticky Header** – if set to `Yes`, additional logo image will be displayed in the sticky header (if the additional logo image was configured correctly). All details about configuration of the logo in Magento can be found in chapter **4.4 Logo**.

Mobile Header

- **Move Switchers Next To Logo** – if set to `Yes`, if mobile header is enabled, language and currency switchers will be moved next to the logo (but only in mobile header) to make the header smaller.

7.2 Category View

In this tab, you can configure the options related to the category view. Category view presents the list of products from the selected category.

Alternative Image

- **Alternative Image** – show alternative image on mouse hover over the main image.
- **Select Alternative Image By Attribute** – which attribute of the product image gallery will be used to select an alternative image. Available attribute:
  - `Label`
  - `Sort Order`

  (see examples below)
- **Alternative Image Column Value** – specify the attribute value which will mark alternative images of your products.

Example 1:

If `Select Alternative Image By Attribute` is set to `Sort Order`, specify the number of product image which will mark the alternative image in the gallery. For example, if you enter `2`, the second image in each product will be used as the alternative image.

Example 2:

If `Select Alternative Image By Attribute` is set to `Label`, specify label which will mark the alternative image in the gallery. The label should be a short single word, e.g. `alt`.

7.3 Category View (Grid Mode)

In this section of admin panel you can configure the options related to the **grid mode** of the category view.

Images

- Image Width – maximum width of the product image. If width is not specified, default width will be used.

  Remember that in the category view the visual size of the product image depends on the number of columns. For example, if you increase the number of columns, columns will be narrower and there will be less space for the image in each column, so the visual size of the image will be smaller than the actual dimensions of the image.

- Image Height – maximum height of the product image.

  To **keep the aspect ratio** of the product image intact, leave this field empty. If this field is empty, height of the image will be calculated automatically based on width.

Number of Columns

In the grid mode, products are displayed in columns. In this section of admin panel you can specify number of columns (products) in a row visible with a particular browser viewport width (breakpoint).

Below is the list of all available breakpoints for which you can specify the number of columns. Later in this chapter you can also find examples and explanation how exactly to use these fields.

- Number of Columns Above 1920 px (breakpoint 3XL) – number of columns displayed if browser viewport width is greater than 1920 px. The default value is 6.

- Number of Columns Above 1680 px (breakpoint 2XL) – number of columns displayed if browser viewport width is greater than 1680 px. The default value is 6.

- Number of Columns Above 1440 px (breakpoint XL) – number of columns displayed if browser viewport width is greater than 1440 px. The default value is 5.

- Number of Columns Above 1200 px (breakpoint L) – number of columns

displayed if browser viewport width is greater than 1200 px. The default value is 4.

- **Number of Columns Above 992 px (breakpoint M)** – number of columns displayed if browser viewport width is greater than 992 px. The default value is 3.
- **Number of Columns Above 768 px (breakpoint S)** – number of columns displayed if browser viewport width is greater than 768 px. The default value is 3.
- **Number of Columns Above 640 px (Breakpoint XS)** – number of columns displayed if browser viewport width is greater than 640 px. The default value is 3.
- **Number of Columns Above 480 px (Breakpoint 2XS)** – number of columns displayed if browser viewport width is greater than 480 px. The default value is 2.
- **Number of Columns Above 320 px (Breakpoint 3XS)** – number of columns displayed if browser viewport width is greater than 320 px. Below 320 px products are automatically displayed in a single column. The default value is 2.

> **Important:**
>
> note that you can specify the number of columns **only** for breakpoints (viewport widths) which are **less than or equal** to the maximum breakpoint which is configured in the *Maximum Breakpoint* field (under *Stores > Configuration > Theme Layout*).
>
> Refer to chapter **9.1 Responsive Layout** for more details about configuration of the maximum breakpoint.

Example 1:

If the *Maximum Breakpoint* field (which defines the maximum breakpoint for the entire store) is set to `1200 px`, then you can specify the number of columns in the following fields:

- *Number of Columns Above 1200 px (breakpoint L)*
- *Number of Columns Above 992 px (breakpoint M)*
- *Number of Columns Above 768 px (breakpoint S)*
- and in all the fields for lower breakpoints.

But you can't specify the number of columns in the *Number of Columns Above 1440 px (breakpoint XL)* field (or in the fields for larger breakpoints) because that breakpoint (1440 px) is **larger** than the value configured in the *Maximum Breakpoint* field (1200 px).

So on the frontend even when browser viewport width is greater than 1200 px (which is the maximum breakpoint in this example), the grid will have only as many columns as configured in the *Number of Columns Above 1200 px (breakpoint L)* field. Values from all the fields for larger breakpoints:

- *Number of Columns Above 1920 px (breakpoint 3XL)*
- *Number of Columns Above 1680 px (breakpoint 2XL)*
- *Number of Columns Above 1440 px (breakpoint XL)*

will be ignored.

Example 2:

If the *Maximum Breakpoint* field (which defines the maximum breakpoint for the entire store) is set to 1680 px, then you can specify the number of columns in the following fields:

- *Number of Columns Above 1680 px (breakpoint 2XL)*
- *Number of Columns Above 1440 px (breakpoint XL)*
- *Number of Columns Above 1200 px (breakpoint L)*
- and in all the fields for lower breakpoints.

But you can't specify the number of columns in the *Number of Columns Above 1920 px (breakpoint 3XL)* field because that breakpoint (1920 px) is **larger** than the value configured in the *Maximum Breakpoint* field (1680 px).

Product Grid Display

- **Equal Height of Grid Items (Products)** – if enabled, all grid items will have the same height and "Add to cart" button will be displayed at the bottom of the item. This option is limited to product grid in category view and will not be applied to product sliders.

- **Product Hover Effect** – enable/disable effect which is visible on mouse hover over the product: block is emphasized with frame and shadow.

> **Note**:
>
> enable in order to use option "Display On Hover" in **Display Selected Elements**. Otherwise displaying elements on mouse hover over the product will not work properly.

- **Disable Hover Effect Below...** – product Hover Effect will be disabled if width of the browser viewport is below selected value.
- **Hide "Add to" Links Below...** – "Add to wishlist" and "Add to compare" links will be hidden if width of the browser viewport is below selected value.
- **Align Center** – align to the center elements of the grid item: product name, price, button etc.
- **Size of Elements** – control the size of the elements of the grid item, such as: product name, price.

Display Selected Elements

In this section, you can configure the display of selected elements of the grid items (products). Each of the elements can be: **a)** disabled and not displayed in the grid, **b)** displayed in a standard way, **c)** displayed only on mouse hover over the product.

Below is the list of the elements which can be configured:

- **Name** – name of the product
- **Display Name in Single Line** – If set to "Yes": if product name is too long to be displayed in a single line, it will be clipped and an ellipsis (…, U+2026 HORIZONTAL ELLIPSIS) will be appended at the end of the line to represent the clipped text. This will only take effect if the field **Name** is set to "Display".
- **Price** – price of the product
- **Ratings** – product ratings (stars and number of reviews)
- **Add To Cart** – "Add to cart" button

- **"Add to" Links** – "Add to wishlist" and "Add to compare" links
- **Show "Add to" Links as Icons Above the Image** – if set to `No`, "Add to wishlist" and "Add to compare" links will be displayed next to the "Add to Cart" button. Otherwise, links will be displayed as simple icons over the product image.

> **Note**:
>
> this option also affects "Add to wishlist" and "Add to compare" links in the **Featured Products Slider**.

7.4 Category View (List Mode)

In this section of admin panel you can configure the options related to the list mode of the category view.

Images

- **Image Width** – maximum width of the product image. If width is not specified, default width will be used.

> **Note:**
>
> note that in the list mode, for layout purposes **the column with product images has a fixed width** which equals 240 pixels. This is also the default width of product images. To increase the width of product images and exceed this fixed width you need to specify **both dimensions** of product images – width and height (in `Image Width` and `Image Height` fields). Otherwise, if you specify only the width (even if the specified value is very big), the column with images will never exceed the default width of 240 pixels.

- **Image Height** – maximum height of the product image.

  To **keep the aspect ratio** of the product image intact, leave this field empty. If this field is empty, height of the image will be calculated automatically based on width.

Product List Display

- **Product Hover Effect** – enable/disable effect which is visible on mouse hover over the product: block is emphasized with frame and shadow.

7.5 Layered Navigation

In this section of admin panel you can find additional options for Magento's **layered navigation** (the "Shopping Options" block which is displayed in the sidebar in category view).

The basic settings of the layered navigation can be found under `Stores > Configuration > Catalog > Layered Navigation` and `Stores > Attributes > Products`. All the details about the layered navigation can be found in Magento documentation, for example: *http://docs.magento.com/m2/ce/user_guide/catalog/navigation-layered-filterable-attributes.html*

- **Accordion Mode** – in the accordion mode only one filter can be expanded at the same time. If it's disabled, multiple filters can be expanded at the same time.
- **Filter Expanded On Load** – specifies which filters should be expanded when the layered navigation gets initialized on page load. By default, all filters are collapsed.
- **Category Filter** – set to `No` to remove the category filter from Magento's layered navigation block.

  You may want to remove the category filter if the sidebar menu (which also displays categories) was already enabled for the category view. More details about the sidebar menu can be found in chapter **12.1.3 Sidebar Menu**.

7.6 Product Page

This chapter refers to the following section of the admin panel: `Stores > Configuration > Theme Settings > Product Page`, in which you can configure the options related to the product page.

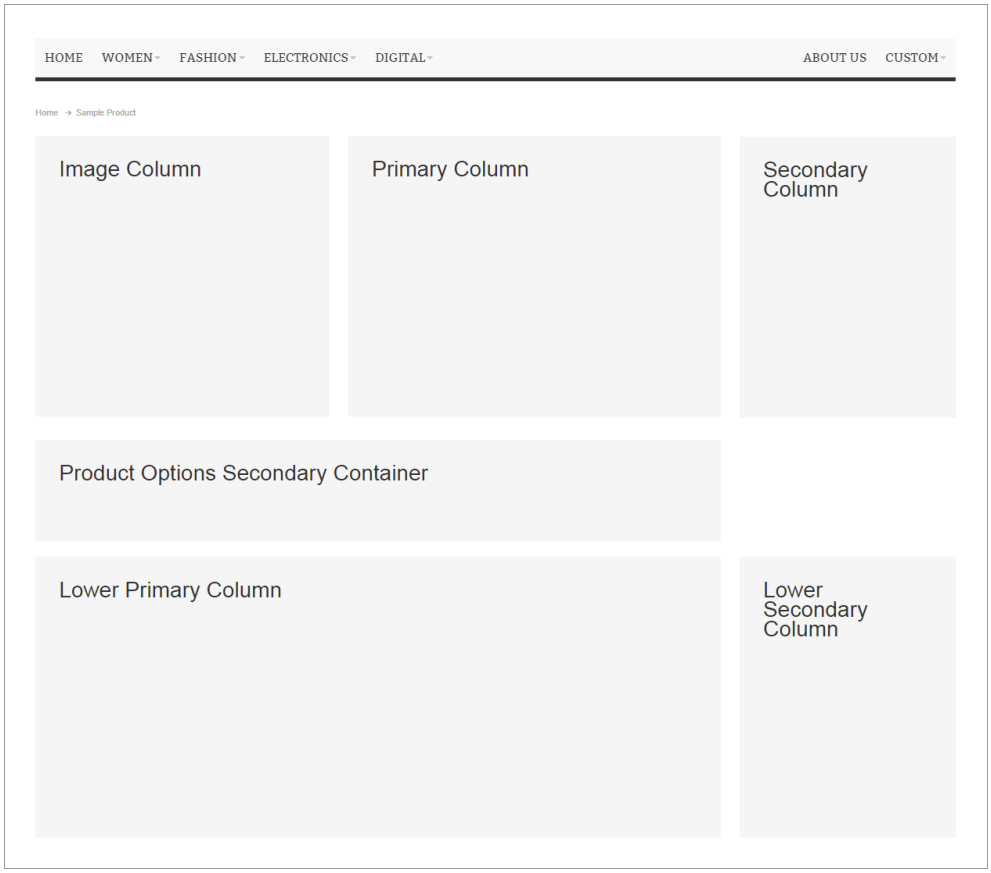
First part of the chapter describes basics of the configuration of the product page.

In the last part you can find the list of all available options:
7.6.3 Product page configuration options.

7.6.1 Product page structure

Width of the product page sections is specified in grid units. As this is a 12-column grid, each section spans a number of those 12 columns, and should always add up to 12 for each row.

Below you can see all available sections:

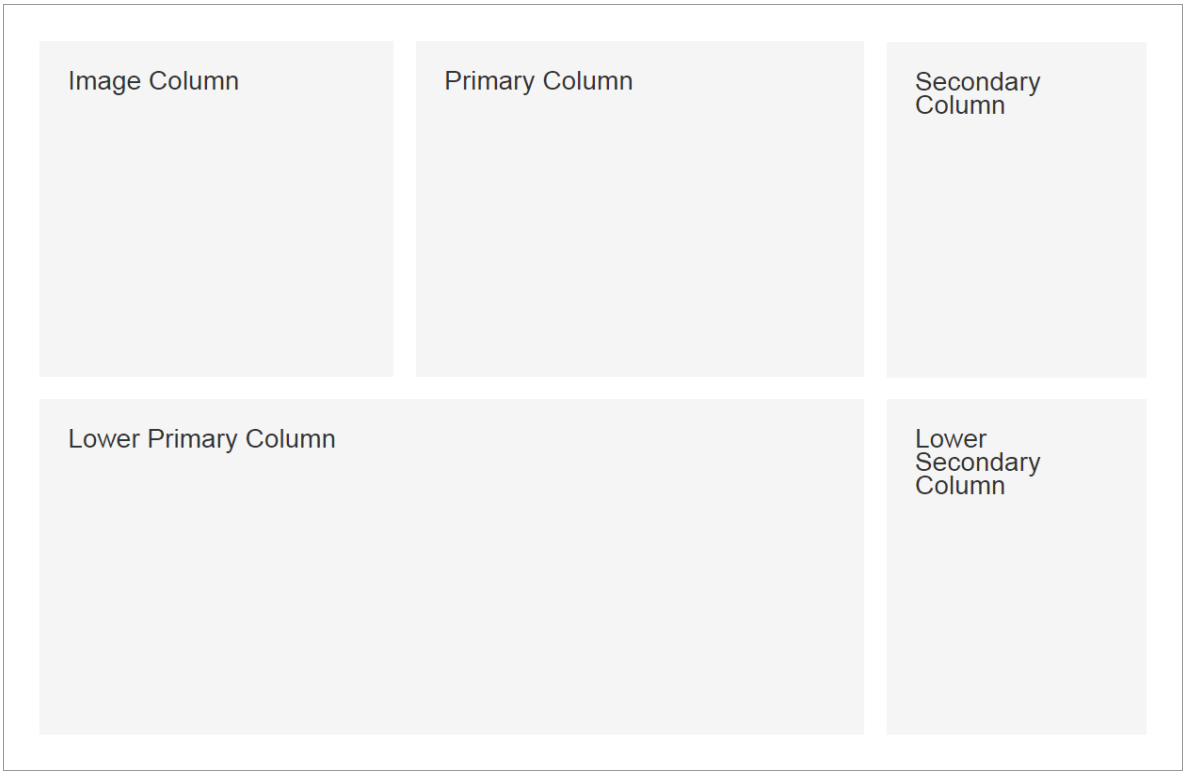

HOME
WOMEN
FASHION
ELECTRONICS
DIGITAL
ABOUT US
CUSTOM
Home → Sample Product
Image Column
Primary Column
Secondary Column
Product Options Secondary Container
Lower Primary Column
Lower Secondary Column

Example 1:

If you want to display three main sections (Image Column, Primary Column, Secondary Column) in a single row, sum of these three sections has to be equal to **12 grid units**. For example:

```
Image Column = 4, Primary Column = 5, Secondary Column = 3
4 + 5 + 3 = 12
All the main three sections will be displayed in one row.

Lower Primary Column = 9, Lower Secondary Column = 3
9 + 3 = 12
Lower sections will be displayed in another row, below the main sections.
```

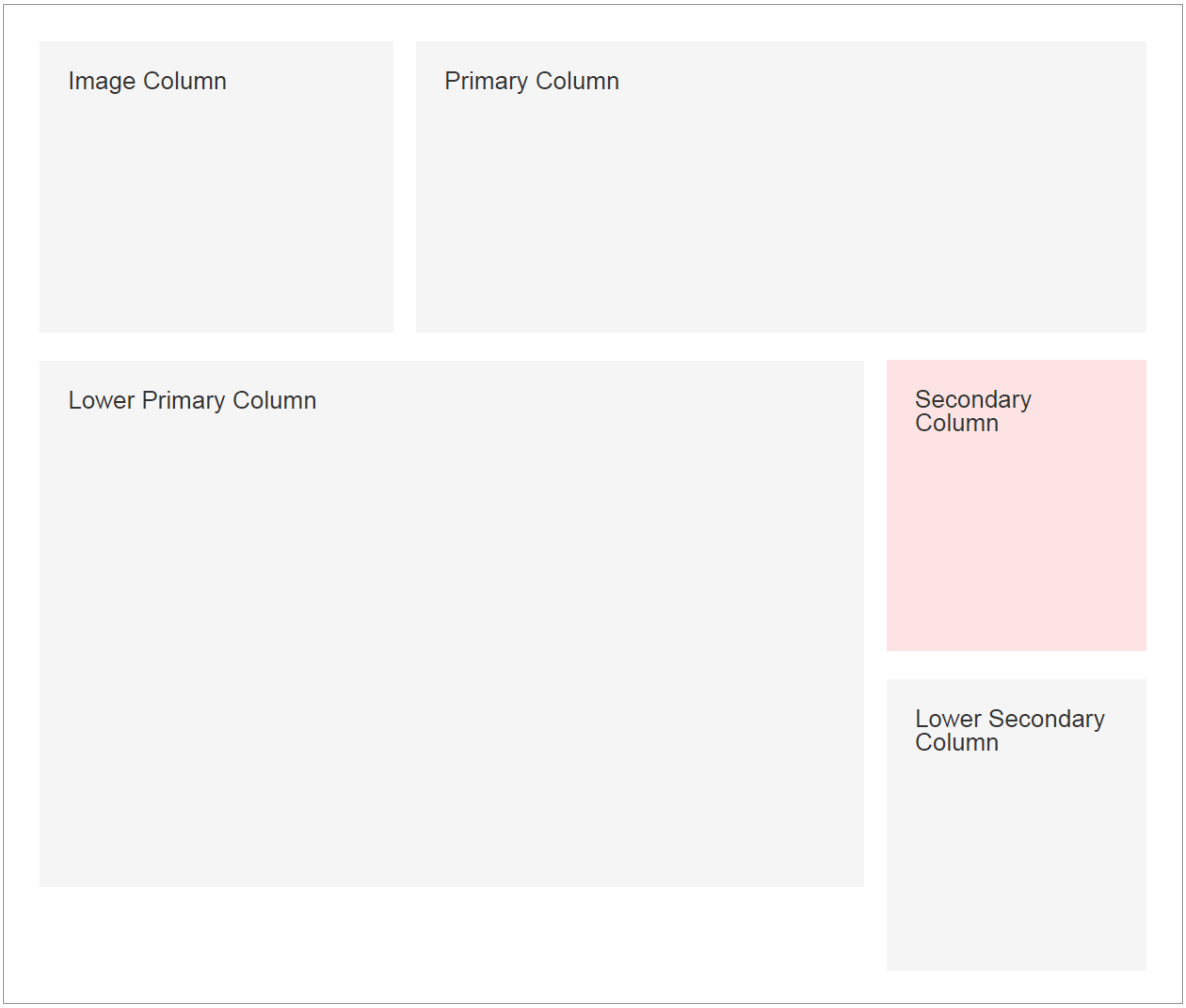
Note that the Product Options Secondary Container is optional (see more info later in this chapter) and will be displayed only if product options were assigned to "Block after Info Column" in product's settings. Otherwise, options container will not be displayed:



But if sum of these three sections is greater than 12, the last section will drop to the next row. For example:

```
Image Column = 4, Primary Column = 8, Secondary Column = 3
4 + 8 + 3 = 15
Image Column and Primary Column will be displayed in one row, and Secondary Column will drop to
the next row.

Lower Primary Column = 9, Lower Secondary Column = 3
9 + 3 = 12
```

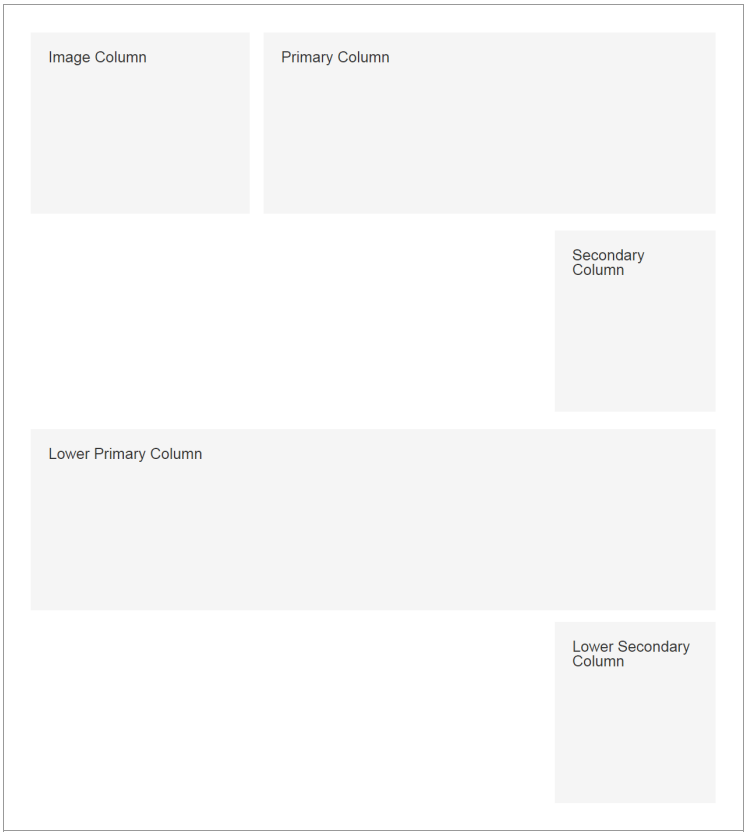
Note that the Secondary Column always **floats to the right side** so it will be placed above the Lower Secondary Column and at the side of the Lower Primary Column:



If the Lower Primary Column is wider, then it will be displayed in another row, below the Secondary Column. For example:

```
Image Column = 4, Primary Column = 8, Secondary Column = 3
4 + 8 + 3 = 15
Secondary Column will drop to the next row.

Lower Primary Column = 12, Lower Secondary Column = 3
12 + 3 = 15
```

The Lower Primary Column takes entire available space so it will drop to the next row, below the Secondary Column. And the Lower Secondary Column will drop to another row, below the Lower Primary Column.



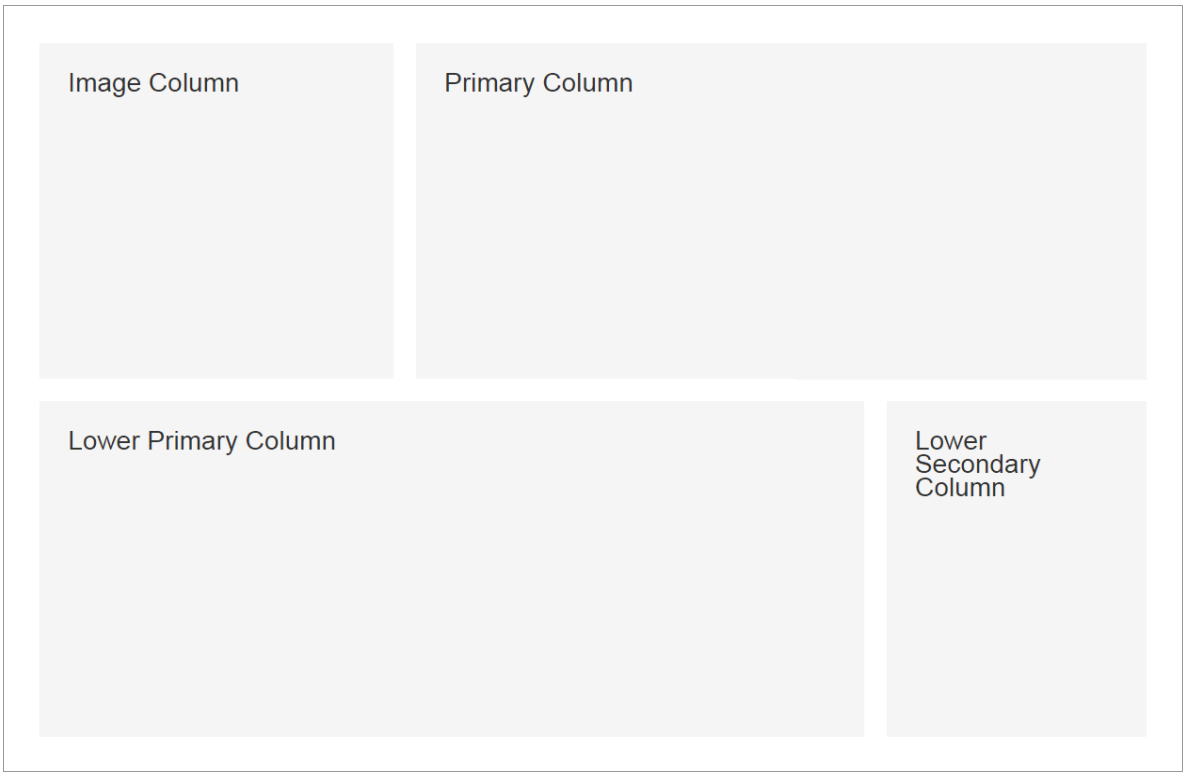
In this case it's recommended to make the Lower Primary Column narrower so that it fits the empty space at the left side of the Secondary Column.

Example 2:

If you want to get rid of the Secondary Column, leave the **Secondary Column Width** field empty and make the Primary Column wider. For example:

```
Image Column = 4, Primary Column = 8, Secondary Column = leave empty
4 + 8 + ... = 12

Lower Primary Column = 9, Lower Secondary Column = 3
9 + 3 = 12
```

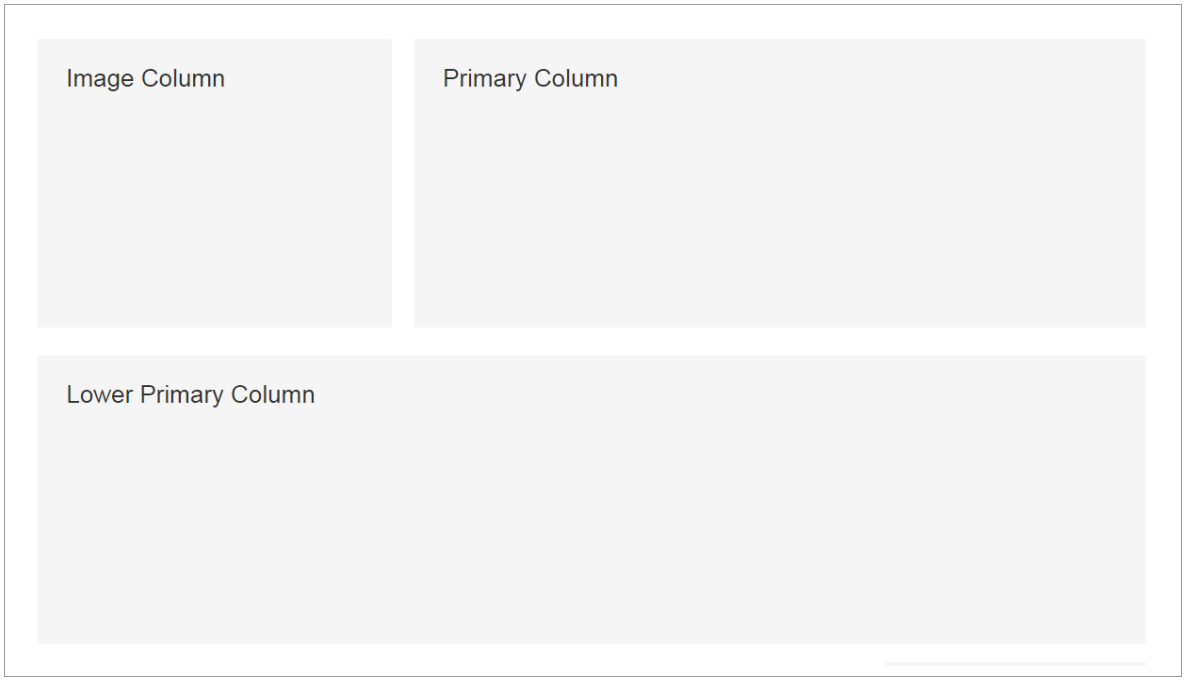
The Image Column and the Primary Column will take the entire row and the Secondary Column will disappear:



Example 3:

If you want to get rid of both: the Secondary Column and the Lower Secondary Column, leave the **Secondary Column Width** and the **Lower Secondary Column Width** fields empty and make the Primary Column and the Lower Primary Column wider. For example:

```
Image Column = 4, Primary Column = 8, Secondary Column = leave empty
4 + 8 + ... = 12

Lower Primary Column = 12, Lower Secondary Column = leave empty
12 + ... = 12
```

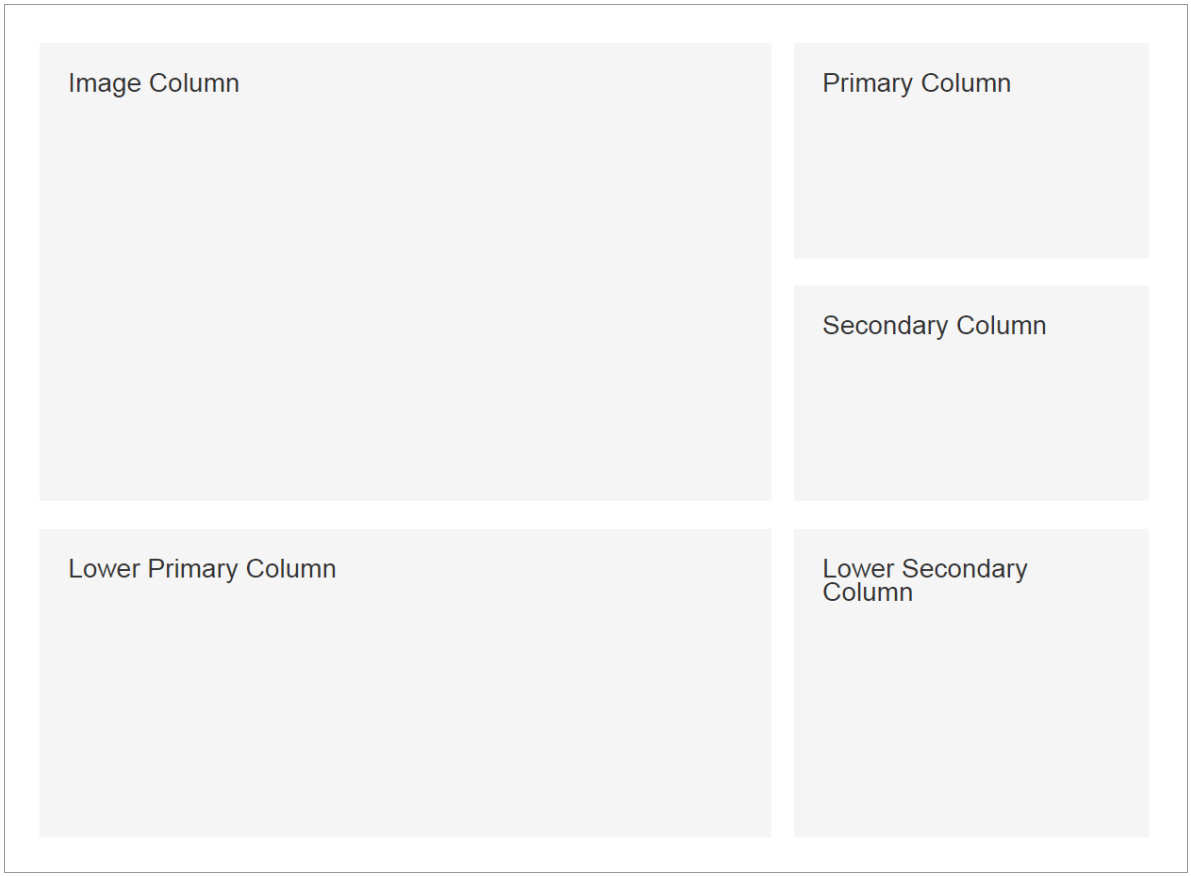


Example 4:

If sum of the Image Column and the Primary Column equals 12, and the Primary Column and the Secondary Column both have the same width, sections will have the following layout:

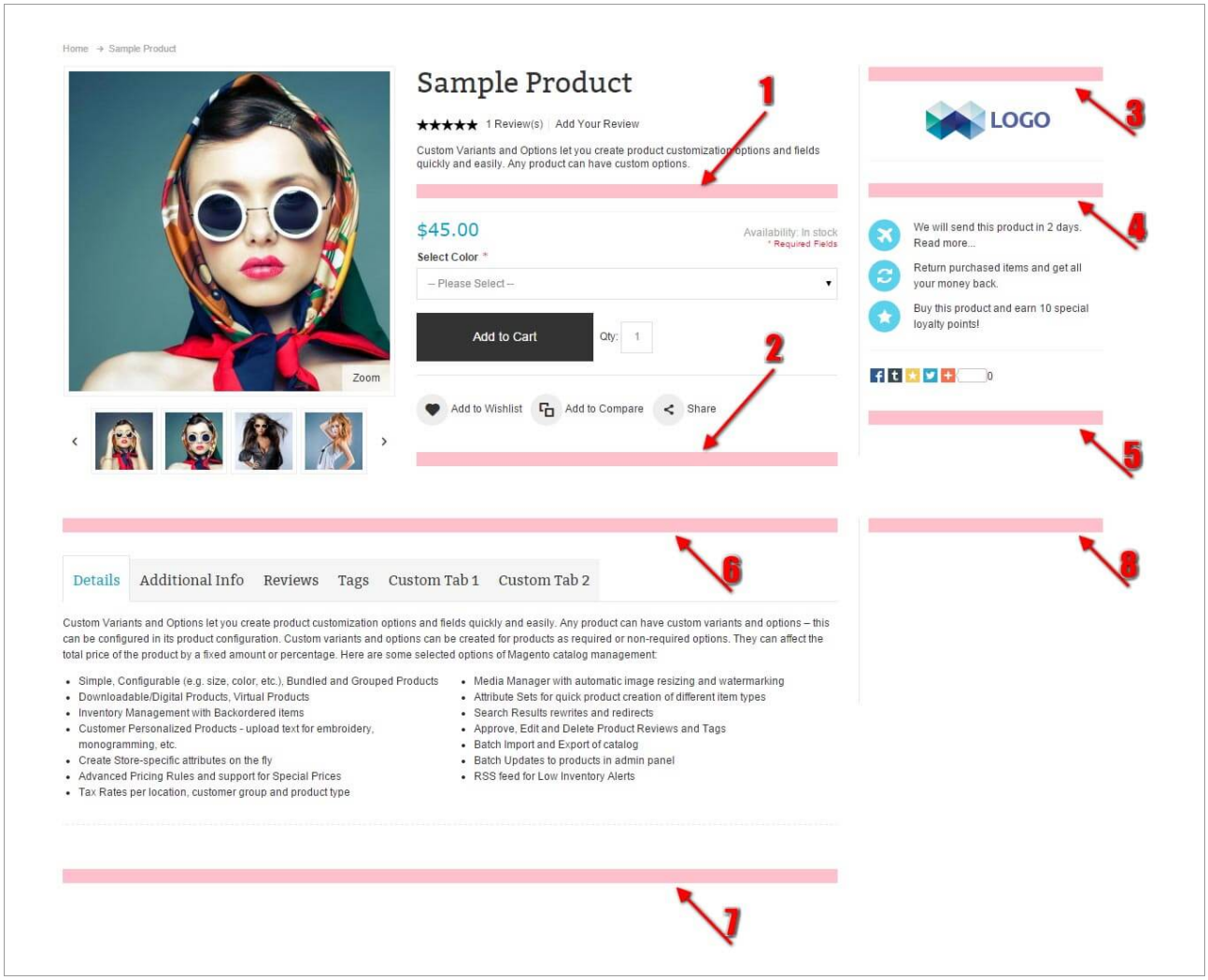
```
Image Column = 8, Primary Column = 4, Secondary Column = 4
8 + 4 + 4 = 16

Lower Primary Column = 9, Lower Secondary Column = 3
8 + 4 = 12
```

This allows you to present large product images. In this example, both the Primary Column and the Secondary Column will be displayed as a simple sidebar:



7.6.2 Product page block positions

In section `Stores > Configuration > Theme Settings > Product Page` of admin panel you can select where to display some of the product page blocks. Below you can see a list of all available positions:



1. **Primary Column, Position 1**
2. **Primary Column, Position 2**
3. **Secondary Column, Position 1**

4. Secondary Column, Position 2
5. Secondary Column, Position 3
6. Lower Primary Column, Position 1
7. Lower Primary Column, Position 2
8. Lower Secondary Column, Position 1

7.6.3 Product page configuration options

In this chapter you can find a list of options available in the following section of the theme admin panel: `Stores > Configuration > Theme Settings > Product Page`.

Width of Columns

- Image Column Width – width of the section with the product image. Specify value in grid units. By default it is equal to 4.
- Primary Column Width – width of the section with the product name, price, options etc. Specify value in grid units. By default it is equal to 5.
- Secondary Column Width – width of the section with additional product data and custom content. Specify value in grid units. By default it is equal to 3.

> **Important:**
>
> leave empty to remove this column.

- Lower Primary Column Width – width of the section with tabs, reviews etc. Specify value in grid units. By default it is equal to 9.
- Lower Secondary Column Width – width of the section with additional custom content. Specify value in grid units. By default it is equal to 3.

> **Important:**
>
> leave empty to remove this column.

Collateral Data

Container for product collateral data. It can be displayed as tabs, accordion or stacked blocks.

- **Position** – position of Collateral Data container. Screenshot with all available positions can be found in this chapter: **7.6.2 Product page block positions**.
- **Display Collateral Data as Tabs** – display Collateral Data container as simple tabs. If set to "No", Collateral Data will be displayed as vertically stacked blocks.
- **Style of Tabs** – visual style of the tabs.

Related Products

- **Position** – position of the related products block on product page. Screenshot with all available positions can be found in this chapter: **7.6.2 Product page block positions**.
- **Layout** – layout of the related products block on product page:
  - `Simple Grid` – Magento's default grid view
  - `Slider` – simple horizontal slider
- **Display As Thumbnails** – display products as thumbnails: small image with name and price displayed at the side of the image.
- **Number of Columns (Breakpoints)** – products are displayed in columns – with this parameter you can specify the number of columns (products) in a row visible with a particular browser viewport width. Leave this field empty to use the default value. Use the format presented in the examples below.

> **Note:**

> this option is available only if the *`Layout`* field was set to `Slider`.

If you display related products **as thumbnails** and place the block **inside a narrow container**, it's recommended to set this parameter to: `[0, 1], [480, 2], [768, 1]`

Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers `[A, B]` number `A` indicates the web browser viewport width, number `B` indicates the number of columns visible if browser viewport width is greater than the width specified in `A`.
In other words: display `B` columns if viewport width is above `A` pixels.

- **Example 1 – the default value:**

```
[0, 1], [480, 2], [768, 1]
```

  The default value is useful if the slider is placed inside narrow containers, for example, inside the Secondary Column or the Lower Secondary Column.

  This value can be translated as: display 1 column if viewport width is between 0px and 480px, display 2 columns if viewport width is above 480px, display 1 column if viewport width is above 768px.

  This means that **if the screen is wider than 768px** only 1 column will be visible. Thanks to that, the slider can be displayed inside narrow containers.

- **Example 2:**

```
[0, 1], [320, 2], [480, 3], [992, 4], [1200, 5]
```

  This example is useful if the slider is displayed inside wide section, for example: inside the Lower Primary Column.

  This example can be translated as: display 1 column if viewport width is between 0px and 320px, display 2 columns if viewport width is above 320px, display 3 columns if viewport width is above 480px, display 4 columns if viewport width is above 992px, and display 5 columns if viewport width is above 1200px.

- **Example 3:**

```
[0, 3]
```

  which can be translated as: always display 3 columns.

- **Number of Rows** – specify number of rows of products displayed in the slider. Enter value between 1 and 10.
- **Timeout** – to enable automatic scrolling of the slider, specify time (in milliseconds) between transitions. Leave empty to disable automatic scrolling. Note that 1000 miliseconds = 1 second.
- **Related Products Checkbox** – disable Magento's default checkbox displayed next to each related product (it's only displayed next to simple products which have no options).

Up-sell Products

- **Position** – position of up-sell products block on product page. Screenshot with all available positions can be found in this chapter: **7.6.2 Product page block positions**.
- **Layout** – layout of the up-sell products block on product page:
  - `Simple Grid` – Magento's default grid view
  - `Slider` – simple horizontal slider
- **Display As Thumbnails** – display products as thumbnails: small image with name and price displayed at the side of the image.
- **Number of Columns (Breakpoints)** – products are displayed in columns – with this parameter you can specify the number of products in a row visible with a particular browser viewport width. Leave this field empty to use the default value. Use the format presented in the examples below.

> **Note:**
>
> this option is available only if the *`Layout`* field was set to `Slider`.

If you display up-sell products **as thumbnails** and place the block **inside a narrow container**, it's recommended to set this parameter to: `[0, 1], [480, 2], [768, 1]`

Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers `[A, B]` number `A` indicates the web browser viewport width, number `B` indicates the number of columns visible if browser viewport width is greater than the width specified in `A`.
In other words: display `B` columns if viewport width is above `A` pixels.

- **Example 1:**

```
[0, 1], [480, 2], [768, 1]
```

This value is useful if the slider is placed inside narrow containers, for example, inside the Secondary Column or the Lower Secondary Column.

This value can be translated as: display 1 column if viewport width is between 0px and 480px, display 2 columns if viewport width is above 480px, display 1 column if viewport width is above 768px.

This means that **if the screen is wider than 768px** only 1 column will be visible. Thanks to that, the slider can be displayed inside narrow containers.

- **Example 2 – the default value:**

```
[0, 1], [320, 2], [480, 3], [992, 4], [1200, 5]
```

The default value is useful if the slider is displayed inside wide section, for example: inside the Lower Primary Column.

This value can be translated as: display 1 column if viewport width is between 0px and 320px, display 2 columns if viewport width is above 320px, display 3 columns if viewport width is above 480px, display 4 columns if viewport width is above 992px, and display 5 columns if viewport width is above 1200px.

- **Example 3:**

```
[0, 3]
```

This value can be translated as: always display 3 columns.

- **Number of Rows** – specify number of rows of products displayed in the slider. Enter value between 1 and 10.

- **Timeout** – to enable automatic scrolling of the slider, specify time (in milliseconds) between transitions. Leave empty to disable automatic scrolling. Note that 1000 miliseconds = 1 second.

Brand Logo

- **Position** – select position of the brand logo block.

7.7 Product Page Gallery

In this section of admin panel you can configure the options related to image gallery on product page.

- **Magnifier** – if enabled, magnifier block will be displayed at the right side of the main image on product page to zoom the currently selected image. Magnifier block will be displayed on mouse hover over the image.
- **Magnifier Width** – width of the magnifier block. Enter value in pixels (do not enter "px" suffix).
- **Magnifier Height** – height of the magnifier block. Enter value in pixels (do not enter "px" suffix).

7.8 Product Labels

In this section of admin panel you can configure the options related to product labels.

> **Important:**
>
> in order to display product labels, you may need to flush the cache (**4.1 Caching system**) and reindex your Magento (**4.2 Index management**) after marking products as new and after setting a special price.

"New" Label

- Show "New" Label – if a product is marked as new, the "New" label will be displayed over the product.

> **Note:**
>
> There are two ways to promote a product as new. Go to *Catalog > Manage Products* and click selected product to go to the product edit page. Next, do one of the following:
>
> 1. In the *Set Product as New From/To* fields, enter the date range in which the product will be promoted as a new product.
>
> 2. Or set the *New* attribute to `Yes`. Out of the box this attribute doesn't exist in Magento so you would need to create it and add to products. In this chapter you can see how to create such an attribute: **4.7.1.A.3 Attribute: new**.

"Sale" Label

- Show "Sale" Label – if a product is on sale, the "Sale" label will be displayed over the product.

> **Note:**
>
> There are two ways to add the "Sale" label to a product. Go to *Catalog > Manage Products* and click selected product to go to the product edit page. Next, do one of the following:
>
> 1. Click the *Advanced Pricing* link which can be found below the *Price* field. Then, enter a special price in the *Special Price* field.
>
> In the *Special Price From/To* fields you can also enter (it's optional) the date range in which the product will be automatically promoted with the special price. If you leave these fields empty, Magento will fill in the *Special Price From* field with the current date and the special price will apply until you remove it from the product.
>
> 2. Or set the *Sale* attribute to `Yes`. Out of the box this attribute doesn't exist in Magento so you would need to create it and add to products. In this chapter you can see how to create such an attribute: **4.7.1.A.2 Attribute: sale**.

- **Display Saved Amount** – if this option is enabled and if a product has a special price, the label will be displayed over the product to show the percentage which reflects the saved amount of money.

  For example, if the regular price is $100 and the special price is $90, the label will show `-10%`.

  This label has priority over the standard "Sale" label. This means that if the *Show "Sale" Label* field and the *Display Saved Amount* field are enabled and the product has a special price, only the label with saved amount will be displayed.

- **Precision of Saved Amount** – the saved amount of money will be rounded. Specify number of digits after the decimal point. Allowed values: between `0` and `4`.

Custom Label

- **Show Custom Label** – if a product is marked with a custom attribute, a label will be displayed over the product. The values of the attribute will be displayed as a text on the label.

- **Custom Label Attribute Code** – specify code of an attribute which will be used to create custom label. It should be a **dropdown attribute**. In this chapter you can see how to create such an attribute: **4.7.1.A.4 Attribute: custom_label**.

Style of Labels

- **Round Labels** – enable to change the shape of labels. If disabled, labels are rectangular.

7.9 Footer

In this tab, you can configure the options related to the footer section.

- **Automatic Column Width Calculation** – if enabled, width of the columns in the footer (a row of built-in static blocks) will be calculated automatically and columns will always have maximum available width – grid units will be applied to columns. Refer to **6.4 Grid system** for more details about grid units and grid system.

  It will take effect on built-in static blocks with the following identifiers:

  - block_footer_column1
  - block_footer_column2
  - …
  - block_footer_column6

  For example, if only 3 static blocks are enabled in the first row, those 3 blocks will be displayed as 3 equal columns.

  If this option is disabled, then the width of each column will be equal to 2 grid units.

> **Note:** this option will not be applied if exactly 5 blocks are enabled. This is due to limitations of the 12-column grid system – it isn't possible to display 5 equal columns, because 12 can't be divided by 5.

- **Footer Links** – links in the footer section: Search Terms, Contact Us etc. This block can be substituted by the static block `block_footer_links`.

- **Newsletter Subscribe Form** – block with the newsletter subscription form located in the footer section. All visitors have the option to sign up by entering their email address. This block can be substituted by the content of static block `block_footer_primary_bottom_right`.

- **Store Switcher** – store switcher displayed at the bottom of the page. It is only displayed if there is more than one active store in your Magento.

  Note that this is something else than the store view switcher which is displayed in

the header. In Magento "store" is not the same as "store view". In this case, a "store" is a group of "store views". Refer to Magento documentation for more details.

7.10 Product sliders

In this tab, you can configure the options related to product sliders.

Settings from this tab will be applied to all existing product sliders, also to Related Products and Ups-Sell Products on product page.

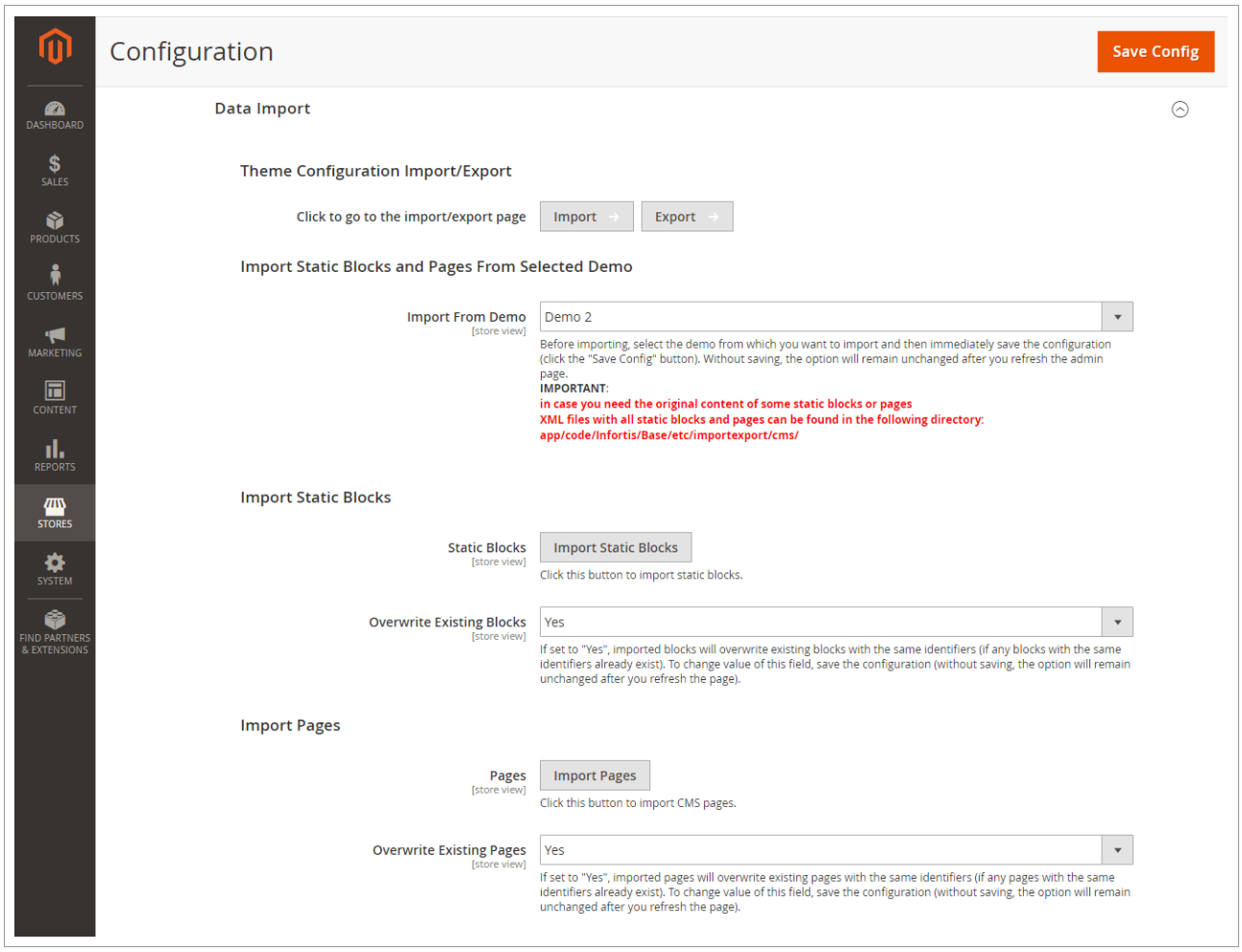
> **Note:**
>
> ***Timeout*** for *Related Products* and *Ups-Sell Products* can be specified individually, refer to **7.6 Product Page** for more details.

More info about product sliders can be found in **6.1.4 Featured Products Slider**.

- **Timeout** – to enable automatic scrolling of the product sliders, you need to specify the time (in milliseconds) between slide transitions. Leave it empty to disable automatic scrolling. Note that 1000 milliseconds = 1 second.
- **Speed** – duration (in milliseconds) of scrolling animation. Applied to scrolling with arrows (and touch swipe).
- **Automatic Scrolling Speed** – duration (in milliseconds) of scrolling animation. Applied to automatic scrolling (if **Timeout** was specified) and to scrolling with pagination.
- **Pause on Hover** – if enabled, when the slider is mouseovered then the automatic scrolling will pause.
- **Loop** – if enabled, scrolling will loop.
- **Lazy Loading** – delays loading of images. If enabled, images outside of viewport will not be loaded before user scrolls to them.

7.11 Data Import

This tab is responsible for the data import procedures after installation.

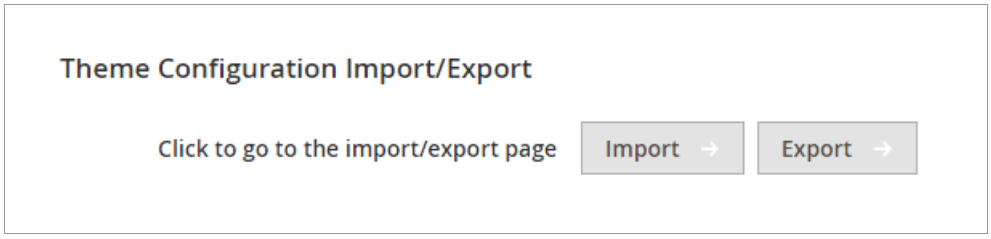


The following functionalities are available:

- import and export theme configuration
- import sample content: static blocks and CMS pages

7.11.1 Theme Configuration Import/Export

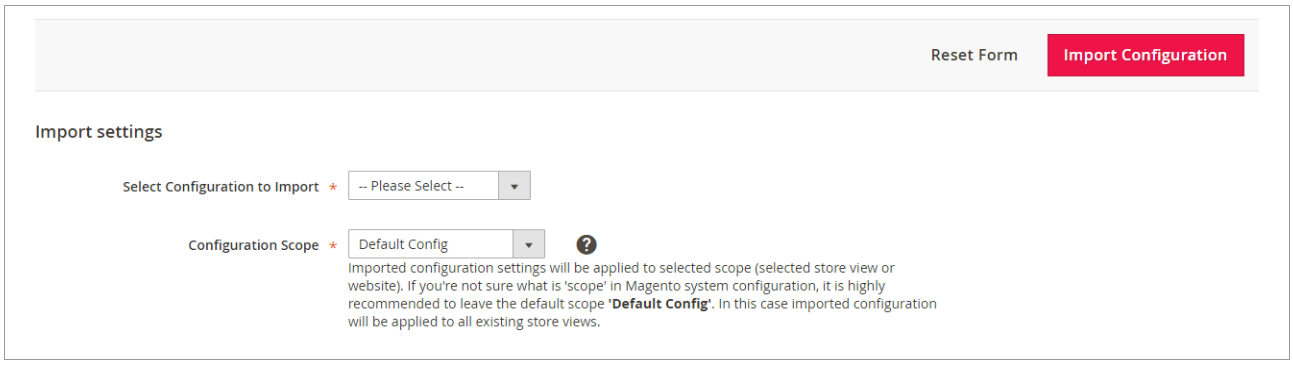
To import/export theme configuration, go to the **Data Import** tab under *Stores > Configuration > Theme Settings* and click the appropriate button to display forms with additional settings:



Both options are described below.

Import

Click the *Import* button – you will be redirected to the configuration import form:



Using that form you can quickly import theme configuration from selected demo store. It will import theme settings to these sections in admin panel:

- *Stores > Configuration > Infortis Themes - Configuration*
- *Stores > Configuration > Infortis Extensions*

After importing you can go to these sections and adjust the configuration to your needs. You can import as many times as you want, so the import functionality can be used to reset the configuration to default values.

To import, follow the instructions below:

1. On the form, choose one of the available demos from the *Select Configuration to Import* list.

2. From the *Configuration Scope* list select configuration scope – imported configuration settings will be saved in the selected scope.

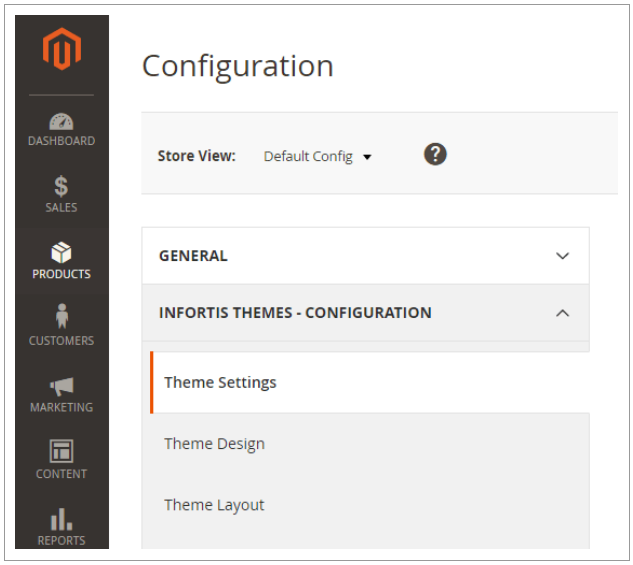
> **Note:**
>
> it is recommended to import to the "Default Config" scope. Especially if you have only one store view in your Magento or if want all your store views to look similar. Settings from the "Default Config" scope can be later overridden on store view level, if needed.

> **Important:**
>
> note that the scope of Magento configuration has three levels:
> ***1.** "Default Config", **2.** Website, **3.** Store view*
>
> A lower level can override a higher level:
> - settings of a store view override settings of a website
> - and settings of a website override settings of "Default Config"

3. If you import to the `Default Config` scope, imported settings will be applied to all existing store views in your Magento. If you're not sure what is "scope" in Magento system configuration, please refer to the following page: *http://docs.magento.com/m2/ce/user_guide/configuration/scope.html*

4. Finally, click the *Import Configuration* button. Then, flush Magento cache (**4.1 Caching system**) in order to see changes.

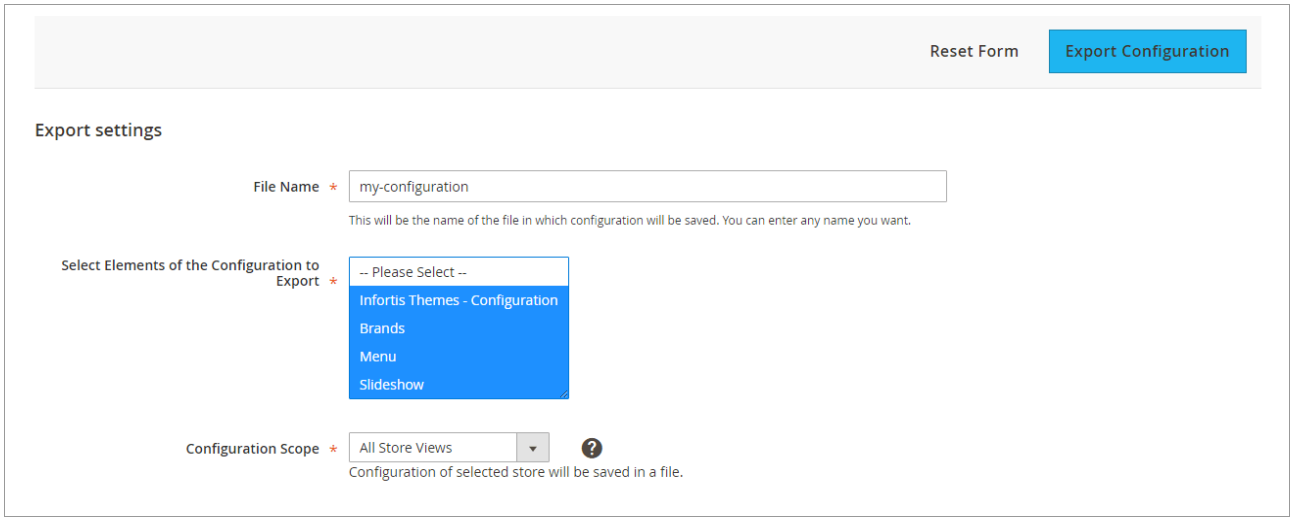
5. After import, if you want to change the imported configuration, all the settings can be found in section *Stores > Configuration*. Before making changes in the configuration, select the scope using the *Store View* switcher which can be found in the top left corner – see the screenshot below:



For example, if you imported to the "Default Config", you need to select that scope using the `Store View` switcher. **Otherwise your changes will not take any effect.**

Export

Click the `Export` button – you will be redirected to the configuration export form:



Using that form you can quickly export theme configuration to a file. It will export theme settings which can be found in the following sections of admin panel:

- *Stores > Configuration > Infortis Themes - Configuration*
- *Stores > Configuration > Infortis Extensions*

On the *Select Elements of the Configuration to Export* list you can choose which sections of the configuration will be exported.

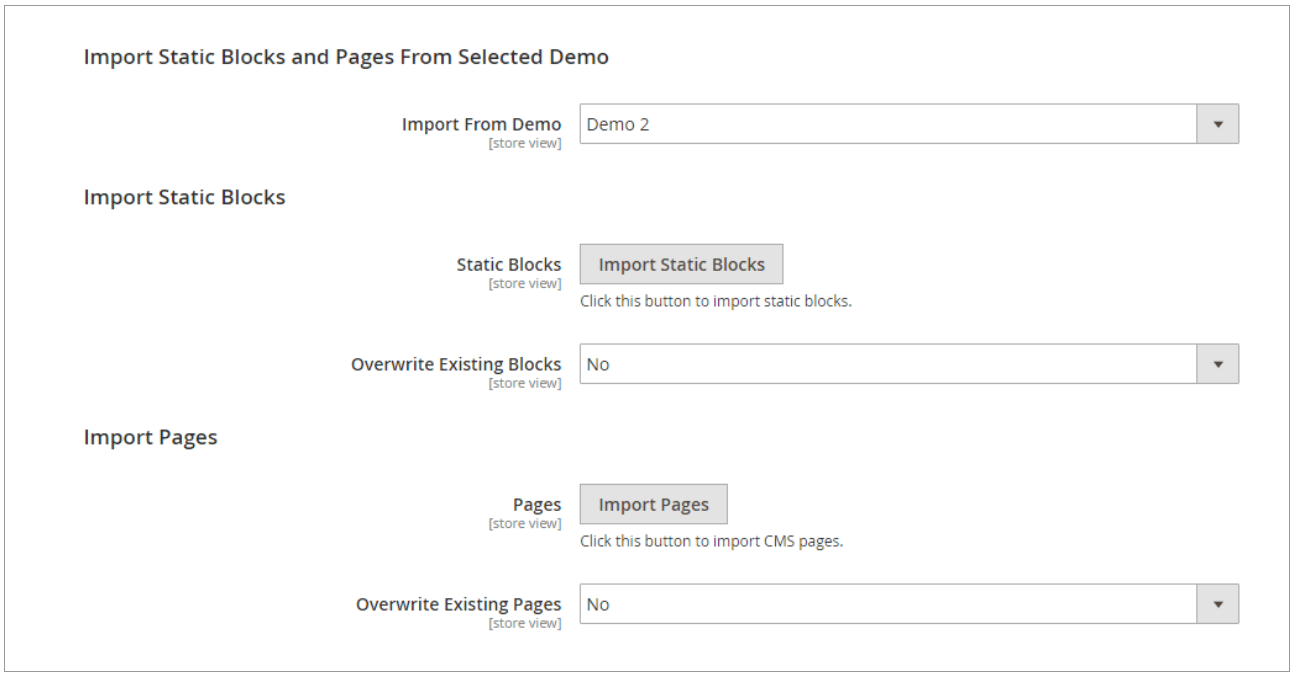
Specify the name of the file in which configuration will be saved. This name will appear on the configuration import form (which was described earlier in this chapter) so you will be able to import your configuration any time you want.

Exported configuration will be saved in a file in the following directory:
`app/code/Infortis/Base/etc/importexport/config/`

7.11.2 Import Static Blocks and Pages From Selected Demo

Ultimo theme ships with a number of useful CMS pages and static blocks. To import these items into your system, navigate to the **Data Import** tab under `Stores > Configuration > Theme Settings`.

Import Static Blocks and Pages From Selected Demo



Before importing, choose the demo from which you want to import blocks and pages.

- **Import From Demo** – from this dropdown list choose the demo from which you want to import blocks and pages, and immediately save the configuration by clicking the `Save Config` button. Save it before you import. Without saving, the `Import From Demo` option will remain unchanged after you refresh the admin page.

Import Static Blocks

- **Import Static Blocks** – click this button to import all predefined static blocks provided with this theme. Refer to **6.2 Built-in blocks and containers** for more details about the static blocks. Imported blocks will appear under *Content > Blocks*.

  To reset (re-import) selected static blocks, delete blocks which you want to reset, and then import blocks again. This way you will only import blocks which are missing.

> **Important:**
>
> in case you need the original content of some static blocks, XML file with all static blocks can be found in the following directory:
> `app/code/Infortis/Base/etc/importexport/cms/`

- **Overwrite Existing Blocks** – if set to **Yes**, imported blocks will overwrite existing blocks with the same identifiers (if any blocks with the same identifiers already exist). You can use this option when you want to restore the original content of all blocks.

> **Note:**
>
> to change this option select **Yes** or **No** and then **save the configuration**. Without saving, the option will remain unchanged after you refresh the page.

Import Pages

- **Import Pages** – click this button to import sample CMS pages provided with this theme. Imported pages will appear under *Content > Pages*.

  Sample home pages have identifiers such as `ultimo-home-page-01`. Home page configuration is described in this chapter: **6.1 Home page and CMS pages**

  To reset (re-import) selected pages, delete pages which you want to reset, and then import pages again. This way you will only import pages which are missing (which were deleted).

> **Important:**
>
> in case you need the original content of CMS pages, XML file with all pages can be found in the following directory:
> `app/code/Infortis/Base/etc/importexport/cms/`

- **Overwrite Existing Pages** – If set to **Yes**, created pages will overwrite existing pages with the same identifiers (if you have any). You can use this option when you want to restore the original content of all pages.

> **Note:**
>
> to change this option select **Yes** or **No** and then **save the configuration**. Without saving, the option will remain unchanged after you refresh the page.

8. Theme Design

This chapter refers to the following section of the theme admin panel:
Stores > Configuration > Infortis Themes - Configuration > Theme Design

Ultimo is equipped with administrative module which gives you the ability to configure the visual appearance of your store. You can change the colors of many elements by using color pickers or by specifying the color code manually. You can also apply textures (patterns) for the main sections of the page.

Names of the most of the options are pretty self-explanatory. Some of them are described in this chapter.

> **Important:**
>
> after creating new store view (or after changing the store view code), re-save theme settings (clicking **Save Config** button) in *Stores > Configuration > Ultimo > Theme Design* and in *Stores > Configuration > Ultimo > Theme Layout*. Otherwise design/layout settings will not be applied to that new store view.

8.1 Colors

In this tab, you can configure basic colors for the entire page. These colors will be inherited by all sections of the page.

Basic Colors

Text and links.

Buttons

Buttons such as “Add to cart”, “Log In” etc.

Interface Icons

Small buttons with icons, e.g.: arrows in product sliders, “sort order” in category view etc.

Custom Icons

Customizable icons (e.g. social icons) which can be used to build custom content.

Horizontal Links

Inline links, e.g. Magento's default Top Links in the header (set of links: Account, Wishlist, Log In), Footer Links etc.

Product Labels

Product labels “New” and “Sale”.

Other

Other elements not classified elsewhere.

8.2 Font

In this tab, you can configure the options related to fonts.

8.3 Effects

In this tab, you can set additional effects for some of the page elements.

- Content Area Shadow – If set to **Yes**, all sections with nontransparent **Inner Background Color**, will drop soft shadow.

  For example, if you set any color in **Inner Background Color** filed in **Footer – Primary** tab, shadow will be displayed in the main area of the footer section.

- Main Menu Shadow – select which container of the menu will drop shadow. This field will be visible if Content Area Shadow was enabled.

> **Note:**
>
> unlike other sections, menu doesn't need to have **Inner Background Color** to drop shadow.

- Structural Elements Transition – if enabled, main structural elements of the page layout will be animated during web browser window resizing. It is recommended to leave this option disabled for browser compatibility.

8.4 Page

In this tab, you can configure the options related to the background of the entire page.

- Background Color – background color of the entire page.

- Background Image – here you can upload an image which will be displayed as a background of the entire page. Click the **Browse...** button and locate the image on your computer.

> **Note:**
>
> sample backgrounds can be found in `pub/media/wysiwyg/infortis/ultimo/custom/background` directory.

Once configuration has been saved, a small icon of the existing image will appear to the left of the button. You can then configure the way the image is displayed with the following options:

- **Background Repeat** – defines how the background image is repeated. It can be repeated along the vertical axis, the horizontal axis, both, or not repeated at all. By default it is repeated both vertically and horizontally.
- **Background Attachment** – determines whether the background image is fixed or scrolls with the rest of the page. By default it scrolls.
- **Background Position (x-axis)** – defines the initial position of the background image along the horizontal axis. By default it is aligned to the center of the page.
- **Background Position (y-axis)** – defines the initial position of the background image along the vertical axis. By default it is aligned to the top of the page.

- **Pattern** – background pattern of the entire page. Pattern covers the background image.

8.5 Header

In this tab, you can configure the options related to the header section of the page. Background image can be set in exactly the same way as for entire page (which is described in previous chapter).

8.6 Header – Top

In this tab, you can configure the area at the top of the header. Settings from this tab will override settings from the `Header` tab.

8.7 Header – Primary

In this tab, you can configure elements of the primary area of the header.

8.8 Main Menu

In this tab, you can configure the options related to the main menu bar.

8.9 Main

In this tab, you can configure the options related to the main section of the page (the section located between the header and the footer).

8.10 Category View

In this tab, you can configure selected elements of the category view.

8.11 Product Page

In this tab, you can configure the product page (in Magento also called "product view").

8.12 Slideshow

In this tab, you can configure elements related to slideshows.

8.13 Footer

In this tab, you can configure the entire footer section of the page. Settings from this tab are inherited by all the other sub-sections of the footer (`Footer-Top`, `Footer-Primary` and `Footer-Bottom`). Selected settings can be overridden in sub-sections.

8.14 Footer – Top

In this tab, you can configure the options related to the area at the top of the footer. Settings from this tab will override settings from the `Footer` tab.

8.15 Footer – Primary

In this tab, you can configure the options related to the primary area of the footer. Settings from this tab will override settings from the `Footer` tab.

8.16 Footer – Secondary

In this tab, you can configure the options related to the secondary area of the footer (just below the primary area). Settings from this tab will override settings from the `Footer` tab.

8.17 Footer – Bottom

In this tab, you can configure the options related to the bottom area of the footer. Settings from this tab will override settings from the `Footer` tab.

9. Theme Layout

This chapter refers to the following section of the theme admin panel:
Stores > Configuration > Infortis Themes - Configuration > Theme Layout

> **Important:**
>
> after creating new store view (or after changing the store view code), re-save theme settings (clicking **Save Config** button) in *Stores > Configuration > Ultimo > Theme Design* and in *Stores > Configuration > Ultimo > Theme Layout*. Otherwise design/layout settings will not be applied to that new store view.

9.1 Responsive Layout

In this tab, you can configure the options related to the responsive layout.

- **Responsive Layout** – enable or disable responsive layout

- **Maximum breakpoint** – specify maximum breakpoint of the responsive layout. Maximum breakpoint also defines the **maximum width of the content area** of the page.

  Content area

  The content area is the region at the center of the viewport where the content is displayed. It is usually narrower than the viewport itself (unless you set the *Maximum Breakpoint* to `Full width` – then the content area will be stretched to cover the entire width of the viewport).

  Note that each of the predefined breakpoints in the *Maximum Breakpoint* field corresponds with a value of the maximum width of the content area. But **the actual width of the content area is always a little bit narrower** than the selected value – the design just needs some additional space at the sides so that the content doesn't touch the edges of the viewport.

  Maximum width of the content area

  Here's the table which presents the maximum width of the content area depending on the maximum breakpoint:

Maximum Breakpoint	Maximum width of the content area
1920	1740 px
1680	1500 px
1440	1380 px
1200	1170 px
992	970 px
Full width	width of the viewport

  Actual width of the content area

  The actual width of the content area also depends on the value of the **side padding**

which can be configured by the admin in the *Side Padding* field under *Stores > Configuration > Theme Design > Page > Inner Container* (the side padding can also be configured individually for each section of the page, e.g. under *Stores > Configuration > Theme Design > Header-Top > Inner Container*).

The side padding increases the empty space at the sides of the content area. The default value of the side padding is `15 px` (at each side).

Here's the table which presents the actual width of the content area depending on the maximum breakpoint and the default value of the side padding:

Maximum Breakpoint	Actual width of the content area
1920	1710 px
1680	1470 px
1440	1350 px
1200	1140 px
992	940 px
Full width	width of the viewport minus 30 px

You can also specify a custom width by selecting the `Custom width...` from the dropdown list (a new field will appear below).

- **Custom Maximum Breakpoint** – this field defines custom maximum breakpoint and custom maximum width of the content area. To display this field you need to select `Custom width...` in the *Maximum Breakpoint* field.

  Specify value in pixels, do not add "px" suffix.

- **Fluid Width** – if this option is enabled, when the web browser window is resized, the width of the content area is fluid.

  If this option is disabled, when the web browser window is resized, the width of the content area is static and changes only on layout breakpoints.

> **Note:**
>
> note that this option does not stretch the content area to full width (to cover the entire width of the viewport). To do that you need to set the *Maximum Breakpoint* field to `Full width`.

```
Admin panel: Stores > Configuration > ...
```

[placeholder for additional chapter]

10. Slideshow

This chapter refers to the following section of the theme admin panel:
`Stores > Configuration > Slideshow`

Description of all the settings of the Slideshow available in the admin panel can be found later in this chapter: **10.7 Slideshow configuration**.

10.1 Slideshow on home page

The slideshow can be added **to the home page** in two ways:

1. **Through the module admin panel** – in this field: `Stores > Configuration > Slideshow > General > Main Slideshow on the Home Page > Slides`.

2. **Through the shortcode** – the shortcode which adds the slideshow is described in the next chapter.

Both ways are described below.

10.1.1 Add home page slideshow through the module admin panel

In the `Slides` field in the admin panel under `Stores > Configuration > Slideshow > General > Main Slideshow on Home Page`, you need to enter an **identifier of a static block** which will contain your slides. You have two options:

- You can leave the identifier of the default static block `sample_slideshow_1` which contains a few sample slides.

- Or enter an identifier of your own static block with slides.

Whichever option you choose, go to `Content > Blocks`, open your static block and edit its content to change the content of your slides.

For more information about creating content of the slides, refer to chapter **10.3 Create**

slides. You can also find there information about sample static blocks (with sample slides) which can be **imported** after theme installation.

10.1.2 Add home page slideshow through the shortcode

Second method to add a slideshow to the home page is to use the shortcode which is described with more details in chapter **10.2 Slideshow on any page**.

Go to `Content > Pages`, open your home page and enter the shortcode in the main content field. For example, add it at the very top of the page content – this way your slideshow will be displayed just below the header on the home page. Magento will **automatically replace the shortcode with a slideshow** during page rendering.

10.2 Slideshow on any page

Slideshow can be displayed on any CMS page (and inside any static block). To add a slideshow to a page, use the following shortcode:

```
{{block class="Infortis\UltraSlideshow\Block\Slideshow" template="slideshow.phtml"
slides="sample_slideshow_1"}}
```

Enter the shortcode in the content field of your page, Magento will **automatically replace the shortcode with a slideshow** during page rendering.

The `slides` parameter must contain an identifier of a single static block which contains your slides. The slides from that static block will be displayed in the slideshow. You can also leave the default static block `sample_slideshow_1`, which contains a few sample slides for quick start.

To create the content of the slideshow (the slides), refer to chapter **10.3 Create slides** for more details. You can also find there information about how to **import sample static blocks** (with sample slides).

You can also import a sample CMS page which contains a few examples of slideshow shortcodes. Refer to chapter **10.5.2 Import page with examples of shortcodes** for

details about the import.

10.2.1 List of parameters

The shortcode which creates a slideshow can take a few parameters. Only the `slides` parameter is required. The other parameters are optional, they can be used to override the global settings of the Slideshow module. If you create multiple slideshows on a single page (by adding multiple shortcode), then in each shortcode you can have different values of parameters.

```
{{block class="Infortis\UltraSlideshow\Block\Slideshow" template="slideshow.phtml"
slides="sample_slideshow_1" timeout="15000"}}
```

All available parameters are presented below.

Required parameters:

- **slides** – an identifier of a single **static block** which contains your slides, for example: `sample_slideshow_1`. To create slides, the content of that static block needs to have proper structure, refer to **10.3.1 Basic structure of the static block with slides** for more details.

  Alternatively, you can also enter comma separated list of static block identifiers (but this option **is deprecated and will be removed in the future versions)**. For example: `block_slide1, block_slide2, block_slide3`. In this case, the content of **each block** will be displayed **as a separate slide**.

Optional parameters:

- **timeout** – to animate the slideshow automatically, specify the time (in milliseconds) between slide transitions. Enter `0` to disable automatic transition. Note that 1000 milliseconds = 1 second.

- **slide_transition** – the effect to be used for transition between two slides. Available values:

  - `fade`

  - `backSlide`

  - `goDown`
  - `fadeUp`

> **Note:**
>
> Enter `0` to use simple horizontal slide animation: `slide_transition="0"`

- **pagination** – display or remove pagination. Allowed values:
  - `0` – remove pagination.
  - `1` – display pagination, visual style 1.
  - `2` – display pagination, visual style 2.
- **pagination_position** – select position of the pagination. Allowed values:
  - `bottom-centered` – bottom, centered, below the slides.
  - `bottom-left` – bottom, left, below the slides.
  - `bottom-right` – bottom, right, below the slides.
  - `over-bottom-centered` – bottom, centered, over the slides.
  - `over-bottom-left` – bottom, left, over the slides.
  - `over-bottom-right` – bottom, right, over the slides.
- **loop** – enter `1` if the slideshow should loop.
- **banner** – an identifier of a single **static block** which contains additional banners (or any other content) displayed at the side of the slideshow. For example: `sample_slideshow_additional_banners_1`.

  To create additional side banners, the content of that static block needs to have proper structure, refer to **10.4 Create additional side banners** for more details.

> **Note:**

> Do not use this parameter if the slideshow will be displayed in a full width container.

- **banner_position** – position of the additional banners displayed at the side of the slideshow. Available values:
  - `right` – the additional banners will be displayed at the right side of the slideshow. This is the default value, if you want the additional banners to be displayed at the right side of the slideshow, you can omit this parameter.
  - `left` – the additional banners will be displayed at the left side of the slideshow.
- **gutter** – enter `0` to remove the space between the slides and the additional banners.

10.3 Create slides

Content of the slideshow (the slides) needs to be placed inside a **static block**. Follow these simple steps to create a block with slides:

1. Create a static block

So, to create the slides, first you need to create a static block. To do that, go to `Content > Blocks`. Refer to **4.8.2 Static blocks** for more information about creating static blocks in Magento.

> **Important:**
>
> when you're creating a static block, make sure to set it as **Enabled** and assign it to a proper **store view**. Otherwise the static block will not be used in the slideshow even if the slideshow was configured correctly.

2. Or import static blocks

Alternatively, you can also import sample static blocks (with sample slides) which contains dozens of ready-to-use code examples. Refer to chapter **10.5.1 Import static blocks with sample slides** for more details about the import.

For example, the main slideshow from the home page of **demo 1** have the following identifier: `sample_slideshow_1`.

3. Add content

When the static block is ready, add some **content**.

The content of the static block with slides needs to have a **proper structure** – refer to chapter **10.3.1 Basic structure of the static block with slides** for more details.

4. Enter the identifier of that static block

Finally, to "tell" Magento where your slides are stored, the **identifier** of your static block needs to be entered in:

- the `slides` parameter, if you decided to create a slideshow through a shortcode (which is described with more details in chapter **10.2 Slideshow on any page**).
- or in the *Slides* field in admin panel under *Stores > Configuration > Slideshow > General > Main Slideshow on Home Page*, if you decided to create the main slideshow on the home page.

10.3.1 Basic structure of the static block with slides

The content of the static block with slides **needs to have the following structure**:

```
<div class="item">
     This is slide 1...
</div>

<div class="item">
     This is slide 2...
```

```
</div>

<div class="item">
     This is slide 3...
</div>
```

As you can see, the content of the static block is **divided into items** (slides). Each item is a simple `<div>` element with the `item` class. Each item will be displayed as a separate slide in the slideshow. You can add as many items as you want.

> **Note:**
>
> make sure to close each item with the closing `</div>` tag. Otherwise the slideshow on frontend will be broken (e.g. two slides will be merged into one slide, the slideshow may be misplaced on the page, content of the page below the slideshow will be broken etc.).

To add custom content inside the first slide, replace the sample text `This is slide 1...` with any other content. For example, replace it with a simple image banner:

```
<div class="item">
     <div class="ban">
             <img class="image" src="{{media url="wysiwyg/infortis/banners/wide01.jpg"}}" />
     </div>
</div>

<div class="item">
     This is slide 2...
</div>

<div class="item">
     This is slide 3...
</div>
```

Then, add another banner – replace the sample text `This is slide 2...` with an image banner:

```
<div class="item">
     <div class="ban">
             <img class="image" src="{{media url="wysiwyg/infortis/banners/wide01.jpg"}}" />
     </div>
</div>
```

```
<div class="item">
     <div class="ban">
             <img class="image" src="{{media url="wysiwyg/infortis/banners/wide02.jpg"}}" />
     </div>
</div>

<div class="item">
     This is slide 3...
</div>
```

And so on. You can create as many slides as you want (to read more about creating content of slides, refer to the next chapter: **10.3.2 Content of a slide**).

You need to use this structure if you keep your slides in a **single static block**, i.e. if you entered an identifier of a single static block in the `slides` parameter of the slideshow shortcode (or in the *`Slides`* field in admin panel under *`Stores > Configuration > Slideshow > General > Main Slideshow on Home Page`*).

But if you decided to keep **each of your slides in a separate static block**, i.e. if you entered a comma separated **list of static block identifiers** in the `slides` parameter of the slideshow shortcode (or in the *`Slides`* field in admin panel under *`Stores > Configuration > Slideshow > General > Main Slideshow on Home Page`*), then **you don't use this structure**. However, the option with comma separated list of static block identifiers is deprecated – since version 2.5.0 it's recommended to keep all slides in a single static block (for clarity and for better performance).

10.3.2 Content of a slide

Inside each slide you can use any HTML: text, images, icons etc. In most cases slides contain image banners with additional captions, so to create slides you can use the built-in system of banners which offers lots of useful features. Full description of banners with many examples can be found in chapter **6.5.5 Banners**.

For example, to create a clickable image banner inside a slide, use the following markup:

```
<a href="{{store url="about-us"}}">
     <div class="ban">
             <img class="image" src="{{media url="wysiwyg/infortis/banners/wide01.jpg"}}" />
     </div>
</a>
```

Important:

a slideshow is always only as big as the container in which it was placed. Even if you put very large images inside your slides, the **visual size** of your slideshow will be always limited by the size of the containing block.

10.3.3 Animated content of a slide

Inside each slide you can use any HTML: text, images, icons etc. In most cases slides contain image banners with additional captions, so to create slides you can use the built-in system of banners which offers lots of useful features. Full description of banners with many examples can be found in chapter **6.5.5 Banners**.

When a banner is used in a slide, it can be easily extended with eye-catching **animations**. In this chapter you can find a few simple examples.

Note:

this chapter describes the animations which can be added **to the content of a slide** (e.g. to the text inside a slide). Do not confuse this with the animations which are used for transitions between slides – the slide transition animation can be configured globally in the `Transition Effect` field under `Stores > Configuration > Ultimo > Slideshow > General`.

Here's an example of a banner with caption (a heading and two lines of text). This is our basic code which will be extend with animations (it doesn't have any animations yet):

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/wide01.jpg"}}" />
     <div class="cap cap-top-left cap-no-bg cap-text-bg cap-text-bg-dark-1">
             <h3 class="text" data-animate-in="fadeInRight">Caption Example</h3>
             <p class="text" data-animate-in="fadeInRight delay-1">This is a line of text</p>
             <p class="text" data-animate-in="fadeInRight delay-2">Second line of text</p>
     </div>
</div>
```

To add an animation effect, add the `data-animate-in` attribute to selected line of text and specify the class name of the animation, for example: `fadeInRight`. In the example below, the animation was added to the heading element:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/wide01.jpg"}}" />
     <div class="cap cap-top-left cap-no-bg cap-text-bg cap-text-bg-dark-1">
             <h3 class="text" data-animate-in="fadeInRight">Caption Example</h3>
             <p class="text">This is a line of text</p>
             <p class="text">Second line of text</p>
     </div>
</div>
```

The `fadeInRight` animation class will make the text fade in and move horizontally from right to left when the slide is first shown on the screen. The list of all available animation classes can be found in chapter **6.5.6.B List of animations**.

> **Note:**
>
> note that we add the animation name to the `data-animate-in` attribute of the selected element, **not to the `class` attribute**. The slideshow script will automatically add the animation name to the `class` attribute when the slide is shown on the screen (so you don't need to worry about how and where the animation classes will be actually used).

We can now add the same animation effect to the rest of the text elements:

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/wide01.jpg"}}" />
     <div class="cap cap-top-left cap-no-bg cap-text-bg cap-text-bg-dark-1">
             <h3 class="text" data-animate-in="fadeInRight">Caption Example</h3>
             <p class="text" data-animate-in="fadeInRight">This is a line of text</p>
             <p class="text" data-animate-in="fadeInRight">Second line of text</p>
     </div>
</div>
```

To make the animation more eye-catching we can use the delay classes (such as `delay-1` and `delay-2`). These classes delay the animation start. The number at the end of each class name indicates the value (in seconds) by which the animation will be delayed.

Let's try these useful classes. Add `delay-1` to the first line of text (this will add 1 second of

delay) and `delay-2` to the second line of text (this will add 2 seconds of delay):

```
<div class="ban">
     <img class="image" src="{{media url="wysiwyg/infortis/banners/wide01.jpg"}}" />
     <div class="cap cap-push-up-10 cap-push-right-10 cap-no-bg cap-text-bg cap-text-bg-dark-
1">
              <h3 class="text" data-animate-in="fadeInRight">Caption Example</h3>
              <p class="text" data-animate-in="fadeInRight delay-1">This is a line of text</p>
              <p class="text" data-animate-in="fadeInRight delay-2">Second line of text</p>
     </div>
</div>
```

More utility classes for animations can be found in chapter **6.5.6.C Additional options**.

10.3.4 Size of slides and banners

The slideshow is responsive so the **width of the slideshow will expand to fit the width of the content area of the page** (however, the height of the slideshow is always the same as the height of the images inside the slides).

> **Note:**
>
> the width of the content area of the page depends on the value of the *Maximum Breakpoint* field (under *Stores > Configuration > Theme Layout*). More details can be found in chapter **9. Theme Layout**. So when you create images for your slides, you need to adjust the image size to the maximum width of the content area.

Inside your slides you can use images of any size. If an image is larger than the width of the slide, it will be automatically scaled down to fit the available space.

But to achieve the best visual effect it's recommended to use image sizes presented in the table below. It shows the width of slides (in pixels) for each available value of the *Maximum Breakpoint* field.

For example, for the default value of the *Maximum Breakpoint* (which is `1200 px`) and the default value of the *Side Padding* (which is `15 px` and can be found under *Stores > Configuration > Theme Design > Page > Inner Container*), the recommended

width of an image inside a slide is 1140 px:

Maximum Breakpoint	Slide width
992	940
1200	1140
1440	1350
1680	1470
1920	1710

Height of the slideshow is not limited so you can use images of any height. However, to get consistent look, it's recommended to use images of the same height in all slides.

10.3.4.A Slideshow with additional side banners

If the slideshow has additional side banners, it's recommended to use image sizes presented in the following table:

Maximum Breakpoint	Slide size	Additional side banners	
		Banner size	Banner size (no gutter)
992	690 x 333	220 x 101	235 x 111
1200	855 x 390	270 x 120	285 x 130
1440	997 x 390	322 x 120	337 x 130
1680	1102 x 420	352 x 130	367 x 140
1920	side banners not recommended		

In this table we assume that you use three equal banners which are displayed one below the other.

If the main slides and side banners are **not separated with a gutter**, for the side banners use the sizes from the column *"Banner size (no gutter)"*. But if the main slides and side banners **are separated with a gutter**, the side banners needs to be a bit smaller (because they also have a bottom gutter) so use the sizes from the column *"Banner size"*.

10.4 Create additional side banners

For every slideshow you can create additional small banners displayed at the side of the slideshow. In this example we will create three equal banners which are displayed one below the other.

The content of the static block with the additional side banners **needs to have the following structure**:

```
<div class="row row-bottom-gutter-half">

     <div class="col-md-12 small-banner">
             This is unit 1...
     </div>

     <div class="col-md-12 small-banner">
             This is unit 2...
     </div>

     <div class="col-md-12 small-banner">
             This is unit 3...
     </div>

</div>
```

To build the basic structure, we use the grid system. Each unit is a simple `<div>` element with the `small-banner` class and with the grid class (in this case we use the `col-md-12` class).

> **Note:**
>
> make sure to close each element with the closing `</div>` tag. Otherwise the slideshow on frontend will be broken (e.g. banners will be misplaced, content of the page below the slideshow will be broken etc.).

To add small bottom gutter between units, add the `row-bottom-gutter-half` class to the row element. Remove this class if you don't want to have any space between units.

To add a banner inside the first unit, replace the sample text `This is unit 1...` with the code of a simple image banner. For example:

```
<div class="row row-bottom-gutter-half">

     <div class="col-md-12 small-banner">
             <div class="ban ban-effect-1">
                     <img class="image" src="{{media
url="wysiwyg/infortis/slideshow/banners/a01.jpg"}}" alt="Banner 1" />
             </div>
     </div>

     <div class="col-md-12 small-banner">
             This is unit 2...
     </div>

     <div class="col-md-12 small-banner">
             This is unit 3...
     </div>

</div>
```

Then, add another banner:

```
<div class="row row-bottom-gutter-half">

     <div class="col-md-12 small-banner">
             <div class="ban ban-effect-1">
                     <img class="image" src="{{media
url="wysiwyg/infortis/slideshow/banners/a01.jpg"}}" alt="Banner 1" />
             </div>
     </div>

     <div class="col-md-12 small-banner">
             <div class="ban ban-effect-1">
                     <img class="image" src="{{media
url="wysiwyg/infortis/slideshow/banners/a02.jpg"}}" alt="Banner 2" />
             </div>
     </div>

     <div class="col-md-12 small-banner">
             This is unit 3...
     </div>

</div>
```

Finally, add the third banner:

```
<div class="row row-bottom-gutter-half">

     <div class="col-md-12 small-banner">
             <div class="ban ban-effect-1">
                     <img class="image" src="{{media
url="wysiwyg/infortis/slideshow/banners/a01.jpg"}}" alt="Banner 1" />
             </div>
     </div>

     <div class="col-md-12 small-banner">
```

```
            <div class="ban ban-effect-1">
                    <img class="image" src="{{media
url="wysiwyg/infortis/slideshow/banners/a02.jpg"}}" alt="Banner 2" />
            </div>
      </div>

      <div class="col-md-12 small-banner">
            <div class="ban ban-effect-1">
                    <img class="image" src="{{media
url="wysiwyg/infortis/slideshow/banners/a03.jpg"}}" alt="Banner 3" />
            </div>
      </div>

</div>
```

10.5 Import sample slides and shortcodes

10.5.1 Import static blocks with sample slides

For quick start, you can import sample static blocks which contain ready-to-use examples of slides. Refer to chapter **7.11 Data Import** for details about the import procedure.

After import, you can find all imported static blocks under `Content > Blocks`. Identifiers of the static blocks with sample slides start with the `sample_slideshow_` prefix. For example, the main slideshow from the home page of **demo 1** have the following identifier: `sample_slideshow_1`.

You can modify the imported static blocks to create your own slides for slideshows.

10.5.2 Import page with examples of shortcodes

You can import a sample CMS page which contains a few examples of slideshow shortcodes. Refer to chapter **7.11 Data Import** for more details about the import procedure.

The identifier of the page is: `magento-banner-slideshow`. After import, the page can be found under `Content > Pages`.

Page contains examples of shortcodes which create slideshows of different sizes. More

information about size of slides and banners can be found in **10.3.4 Size of slides and banners**.

> **Important:**
>
> To use these shortcodes first you need to import sample static blocks with sldies.

10.6 Full width slideshow on home page

To add a full width slideshow to the home page, follow the instructions in chapter **10.1.1 Add home page slideshow through the module admin panel**.

When the slideshow is ready, to make it full width, follow these steps:

1. Go to *Stores > Configuration > Slideshow > Main Slideshow on the Home Page*.

2. Enable the *Position: Above Content Area (Full Width)* option.

3. **Disable** the *Position: Above Content Area (Boxed)* option. It has to be disabled because the slideshow can't be added to both positions at the same time.

4. Finally, flush the cache.

The main slideshow on the home page will be displayed at the top of the main content area, just below the header. The slideshow will take full available width.

List and description of all the options available in the admin panel can be found in chapter **10.7 Slideshow configuration**.

10.7 Slideshow configuration

Below you can find a list of all slideshow settings available in the admin panel in `Stores > Configuration > Ultimo > Slideshow`.

10.7.1 General

Main Slideshow on the Home Page

In this tab you can quickly configure and enable the main slideshow on the home page. Settings from this section will be applied **only to the main slideshow on the home page**. To display slideshows in other places and on other pages, use the shortcode described earlier in this chapter.

- **Slides** – enter an identifier of a static block which contains your slides. The slides from that static block will be displayed in the main slideshow on the home page.

  You can use the default static block `sample_slideshow_1` which contains a few sample slides for quick start.

  > **Important:**
  >
  > Leave this field empty to remove the main slideshow from the home page.

  Alternatively you can also enter a comma separated list of static block identifiers. Content of each block will be displayed as a separate slide. Note that **this option is deprecated and will be removed in the future versions**, we recommend to enter only a single static block which contains all slides.

- **Position: Above Content Area (Boxed)** – if enabled, the main slideshow will be displayed at the top of the main content area of the home page (just below the header). Slideshow will be boxed – it will have the same width as the content area and sidebars.

  > **Note:**
  >
  > Do not enable slideshow in both positions at the same time.

- **Position: Above Content Area (Full Width)** – if enabled, the main slideshow will be displayed at the top of the main content area of the home page (just below the header). **Slideshow will take the full width of the viewport**.

> **Note:**
>
> Do not enable slideshow in both positions at the same time.

- **Hide Slides On Mobile Devices** – if set to `Yes`, slideshow will be hidden on mobile devices (if browser viewport width is less than 768 pixels). Important: the additional banners displayed at the side of the slideshow can be hidden separately, see options described later in this chapter.

- **Top Margin** – additional top margin for the slideshow. You can set margin to add extra space above the slideshow. Enter negative value to pull the slideshow up.

- **Bottom Margin** – additional bottom margin for the slideshow. You can set margin to add extra space below the slideshow.

General Settings

In this tab you can configure basic settings of a slideshow, such as speed and timeout. These settings will be applied to all slideshows on frontend.

- **Transition Effect** – the effect which will be used for transition between two slides.

> **Note:**
>
> Leave empty to use simple horizontal slide animation.

- **Timeout** – to animate the slideshow automatically, specify the time (in milliseconds) between transitions. Enter `0` to disable automatic transition. Note that 1000 milliseconds is equal to 1 second.

- **Scrolling Speed** – duration (in milliseconds) of horizontal scrolling animation. Defines how fast the current slide is replaced with the next slide. Applied to horizontal scrolling with arrows and touch swipe.

- **Automatic Scrolling Speed** – duration (in milliseconds) of horizontal scrolling

animation. Applied to automatic horizontal scrolling (if `Timeout` was specified) and to horizontal scrolling with pagination.

- **Smooth Height** – enable, if slides do not have the same height. It will allow height of the slideshow to animate smoothly.
- **Pause on Hover** – if enabled, when the mouse is over the slideshow the automatic transition functionality will pause.
- **Loop** – if enabled, slideshow will automatically loop.

10.7.2 Additional Banners

In this tab you can configure additional banners displayed at the side of the main slideshow on the home page. These settings are applied **only to the main slideshow on the home page**.

- **Banners** – enter an identifier of a single static block which contains additional side banners for the slideshow. These banners will be displayed at the side of the slideshow.

  Note that these banners will be applied **only to the main slideshow on the home page**. To add side banners to other slideshows, use proper parameter in the shortcode of each slideshow (the shortcode is described earlier in this chapter).

  You can use the default static block `sample_slideshow_additional_banners_1`, which contains sample banners for quick start.

> **Important:**
>
> Leave this field **empty** if you don't want to add side banners to the main slideshow on the home page.

- **Position of Banners** – select position of the banners.

- **Show Gutter Between Slides and Banners** – if set to `Yes`, there will be space between the slides and the additional banners.

- **Hide Banners On Mobile Devices** – if set to `Yes`, banners will be hidden on mobile devices (if browser viewport width is less than 768 pixels).

10.7.3 Navigation

In this tab you can configure slideshow navigation elements. These settings will be applied to all slideshows on frontend.

- **Navigation Buttons** – enable/disable "next" and "previous" buttons.
- **Pagination** – enable/disable and select style of pagination.
- **Pagination Position** – select position of the pagination.

11. Brands

This chapter refers to the following section of the theme admin panel:
`Stores > Configuration > Brands`

List of all settings available in the admin panel can be found later in this chapter:
11.2 Brands configuration.

11.1 Basic information

Below you can find some basic information about brand configuration.

> **Important:**
>
> after you finish configuration of all settings, remember to flush the cache (**4.1 Caching system**).

11.1.1 Brand attribute

First of all, to show brands, you will need an attribute which will store the information about brands. By default there is an attribute `manufacturer` in Magento which you can use for this (but you can of course create your own attribute, if needed).

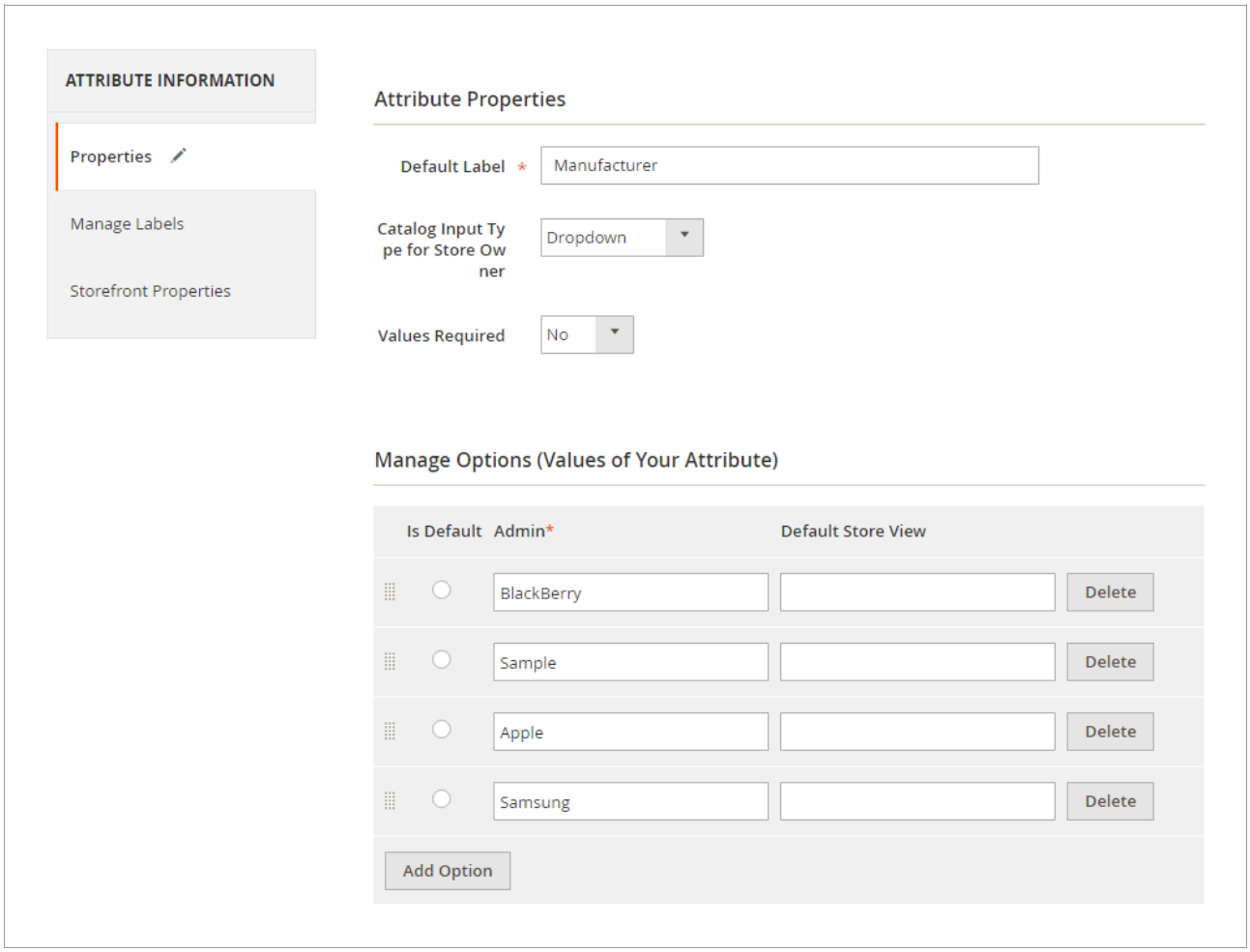
If you decide to create your own brand attribute, the most important thing is to select proper type. There are many types of attributes in Magento, but for brands you need to select `dropdown` in the *`Catalog Input Type for Store Owner`* field. This type of attribute will let you define a list of values (list of brands) available for products. For more information about attribute creation and management refer to **4.7 Attributes**.

11.1.2 Brand images

When your attribute is ready and was added to products, using your FTP client upload your brand logo images to the following directory: `pub/media/wysiwyg/infortis/brands/` Optimal size of the image is 210 x 50 pixels. All images should have the same file extension (the extension can be specified in brand settings described later in this chapter).

11.1.3 Brand names and image file names

All images should be named as per the entries in the brand attribute, but the names of the files has to be simplified a little bit.



Below you can find the list of rules which you need to follow in order to convert a brand name to a correct file name. The module will follow the same rules to generate the path of the image file for each brand. If files are not named correctly, then the module will not be

able to find and display brand images in your store.

The same rules will be used to generate links to custom pages if you configure brand logos to be links to custom pages.

> **Note:**
>
> in exactly the same way Magento simplifies category names and product names to automatically create **URL Key** for category pages and product pages. Full symbol conversion table can be found in Magento file: `app/code/core/Mage/Catalog/Helper/Product/Url.php`

Brand name conversion rules:

1. Replace all uppercase letters with lowercase letters, e.g.:

   for brand `BlackBerry` create an image named like this → `blackberry.png`

2. **Define the word separator** (for simplicity we will call it **the separator**) in the *`Image URL Key Separator`* field in *`Stores > Configuration > Brands > General`* section of admin panel. The separator will be used to replace all whitespace characters and other non-letter and non-digit characters in file names.

   It is recommended to use the default separator: a single hyphen (`-`). More info about separator can be found later in this chapter: **11.2 Brands configuration**.

3. Replace single space with the separator, e.g.:

   `Louis Vuitton` → `louis-vuitton.png`

4. Replace single special characters (non-letter and non-digit characters such as apostrophe, dot, ampersand, percent sign, exclamation mark etc.) with the separator, e.g.:

   `McDonald's` → `mcdonald-s.png`
   `AT&T` → `at-t.png`
   `Amazon.Com` → `amazon-com.png`
   `Coca-Cola` → `coca-cola.png`

5. Replace series of whitespace and special characters with a single separator, e.g.:

   `Marks & Spencer` → `marks-spencer.png`
   `J.P. Morgan` → `j-p-morgan.png`

6. Replace language specific symbols and characters (e.g. letters with diacritical marks) with letters from latin alphabet, e.g.:

   `Huáwéi` → `huawei.png`
   `Citroën` → `citroen.png`
   `Crédit Agricole` → `credit-agricole.png`

11.2 Brands configuration

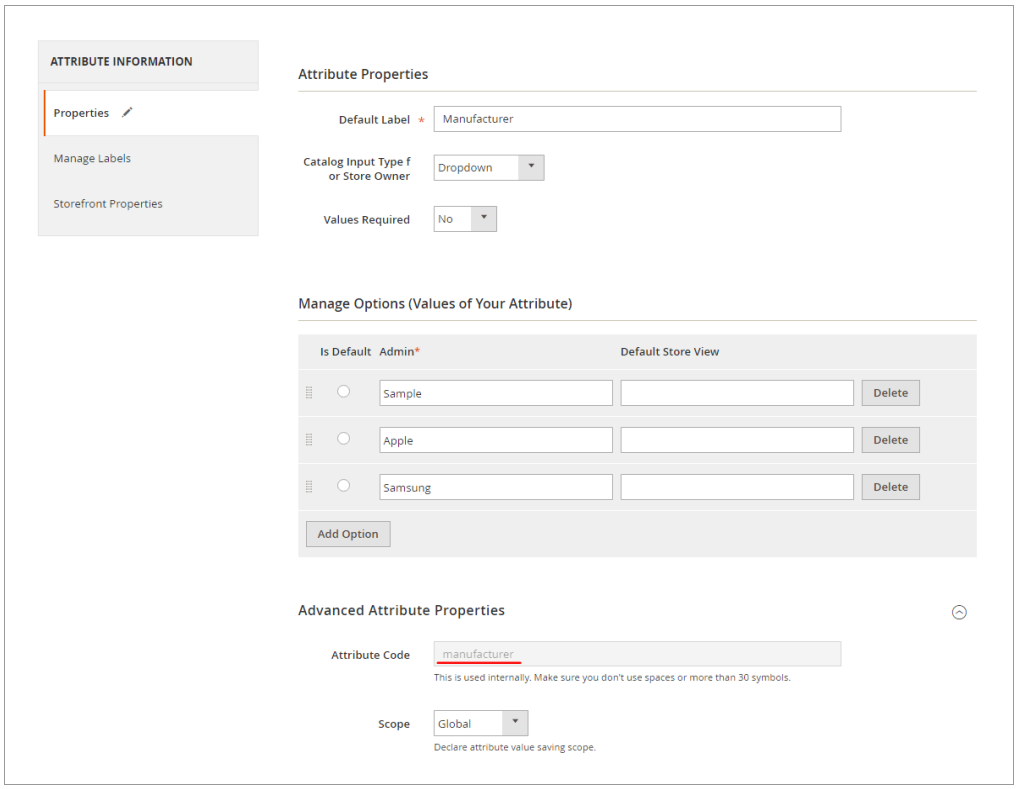
Below you can find a list of settings available in the admin panel `Stores > Configuration > Brands`.

11.2.1 General

- **Brand Attribute Code** – specify the **attribute code** of the attribute which stores the information about brands in your Magento (you can use Magento's default attribute `manufacturer` or create your own attribute).

  If such attribute doesn't exist in your store, you will need to create it and add it to the attribute sets which you use for your products.

  To find the **attribute code** of existing attribute go to `Stores > Attributes > Products`, click on the attribute and search the `Attribute Code` field:



- **Brand Image File Extension** – specify the file extension (jpg, png or gif) of your brand logo images.
- **Show Brand Image** – if set to `No`, brand image will be replaced with brand name (simple text). This option is useful when you don't want to show brand images, only the brand names.
- **Show Text If No Brand Image** –
  - if set to `Yes`: if brand image doesn't exist, it will be replaced with brand name (simple text).
  - if set to `No`: if brand image doesn't exist, it will be not displayed and there will be no other information about brand.

Link

- **Brand Logo is a Link to...** – with this option, you can decide whether brand logo is a link (e.g. to search results page).
  - **No Link** – select this option if you don't want brand logo to be a link.
  - **Quick Search Results** – brand logo is a link to quick search results page. For example: logo of brand *Apple* on product page and in brand sliders will be a link to the following page:

    `www.example.com/catalogsearch/result/?q=apple`

    which displays a list of products returned by Magento quick search for term “Apple”.
  - **Advanced Search Results** – brand logo is a link to advanced search results page which displays a list of products returned by Magento advanced search for brand attribute value “Apple”.
  - **Custom Page** – brand logo can also be a link to custom pages (e.g. category view pages, CMS pages, pages created by 3rd-party extensions etc.). In that case, each brand logo on product page and in brand sliders will be a link to a custom page.

> **Note:**
>
> this option doesn't create any custom pages for brands. It only makes it

> possible to link your brand logos to pages such as: category view pages, CMS pages or other pages created by 3rd-party extensions.

After selecting this option, more settings will appear below this field.

- **Base Path of Brand Pages** – path specified in this field will be used to build a link to a custom brand page (for each brand on product page and in brand sliders). For example, brand logo can be a link to a category which contains products from that brand. It can also be a link to any CMS page you want. To use this option select `Custom Page` in **Brand Logo is a Link to...** field.

  Path will be appended to the store base URL before the brand name.
  For example: `www.example.com/custom/base/path/apple`

`www.example.com/`	`custom/base/path`	`/apple`
Store base URL	*Base Path of Brand Pages*	*Brand name*

  Here you can see some examples.

  - **Example 1:**

    You sell smartphones and you want *Apple* logo to be a link to a category with other Apple products.

    - You created top-level categories for all your brands. If the category page URL is *www.example.com/apple*, then you should leave the **Base Path of Brand Pages** field empty.

    - You created sub-categories for all your brands. They are sub-categories of the main category named “All Brands” and the URLs of category pages are the following:
      - *www.example.com/all-brands/apple*,
      - *www.example.com/all-brands/samsung* etc.
      In that case you should enter `all-brands` in the **Base Path of Brand Pages** field. Thanks to that the module will be able to build correct links to category pages for all your brands.

  - **Example 2:**

You sell smartphones and you want *Apple* logo to be a link to a custom CMS page with some additional info about the brand. In Magento custom pages can be easily created in `Content > Pages`. For each page you need to specify value of **URL Key** field which is a unique identifier of a page. E.g., custom page with URL Key `apple` will have the following URL: *www.example.com/apple*

- You created custom pages for all your brands. If the URLs of your custom pages are the following:
  - *www.example.com/apple*,
  - *www.example.com/samsung* etc.,
  then you should leave the **Base Path of Brand Pages** field empty.

- URL Key of a custom page can be more complex than just a single word. All your brand pages can have some common part, e.g., `my-brands/apple`, `my-brands/samsung` etc. You created custom pages for all your brands. The URLs of your custom pages are the following:
  - *www.example.com/my-brands/apple*,
  - *www.example.com/my-brands/samsung* etc.
  In that case you should enter `my-brands` in the **Base Path of Brand Pages** field. Thanks to that the module will be able to build correct links to custom pages for all your brands.

- **Append Category URL Suffix** – if **Category URL Suffix** exists (in `Stores > Configuration > Catalog`), it can be appended at the end of the brand link. For example: if `.html` was set as the **Category URL Suffix**, logo of brand *Apple* on product page and in brand sliders will be a link to the following page:

  `www.example.com/custom/base/path/apple.html`

SEO Settings

- **URL Key Separator** – specify character which will be used as word separator in URLs of brand pages. Hyphen (`-`) is the recommended character. Other separators, such as underscores (`_`) or plus-signs (`+`) should be avoided. More information: *https://www.youtube.com/watch?v=AQcSFsQyct8*

  Module will strip all whitespace characters and other non-letter and non-digit characters and replace them with the separator. For example:

  if brand name is `Marks & Spencer` and separator is hyphen (`-`), the brand name in URL will be the following: `marks-spencer`

  More information about how brand names are converted (to generate name of the brand image file or brand page URL) can be found in **11.1.3 Brand names and image file names**.

- **Image URL Key Separator** – specify character which will be used as word separator in the names of brand image files. For example:

  if brand name is `Louis Vuitton`, separator is hyphen (`-`) and file extension is `png`, the name of the brand image file will be the following: `louis-vuitton.png`

> **Important:**
>
> the **Image URL Key Separator** field was added for backward compatibility. In next versions this field will be removed and the character specified in the **URL Key Separator** field will also be used in the names of brand image files.
>
> To prepare your store for this, we recommend to use hyphen (`-`) as the separator for images. In previous versions underscore (`_`) was the default separator for images. So what you need to do is to enter hyphen in the Image URL Key Separator field and edit names of all your brand images: replace underscores with hyphens.
>
> For example: change file name `louis_vuitton.png` to `louis-vuitton.png`

11.2.2 Brand on Product Page

- **Show Brand on Product Page** – if enabled, will show a brand logo on the product page (if product has an attribute which specifies the brand).

11.2.3 Brand List

Select Brands

In this tab you can select list of brands which will be displayed in brand sliders. If you have multiple sliders (e.g., on different pages of the store), then each slider will display the same list of brands.

> **Note:**
>
> you can also override the list of selected brands by using `brands` parameter in the slider block code – parameter is described later in this chapter. With parameter it is possible to display different set of brands in each slider.

- **All Brands** – if set to `Yes`, all existing brands will be displayed. If set to `No`, you will be able to select brands in the field below.
- **Selected Brands** – select brands which you want to display.

Filter Brands

- **Brands Assigned To Products** – if set to `Yes`, display only those brands, which are currently assigned to products.
- **Brands Assigned To Products In Stock** – if set to `Yes`, display only those brands, which are currently assigned to products which are in stock.

11.2.4 Brand Slider

Brand slider can be displayed on any CMS page and in any static block. For more datails see the block code which is described later in this chapter.

Below you can find a list of global settings for all brand sliders, some of the settings can be overridden per slider by using parameters which are described later in this chapter.

- **Show All Brands** – if set to **Yes**, slider will display all existing brands. Otherwise slider will display only those brands, which are currently assigned to products.
- **Timeout** – Time (in milliseconds) between slide transitions. Enter `0` to disable automatic transition. Note that 1000 milliseconds is equal to 1 second.
- **Speed** – duration (in milliseconds) of scrolling animation. Applied to scrolling with arrows (and touch swipe).
- **Automatic Scrolling Speed** – duration (in milliseconds) of scrolling animation. Applied to automatic scrolling (if **Timeout** was specified) and to scrolling with pagination.
- **Pause on Hover** – if enabled, when the slider is mouseovered then the automatic scrolling will pause.
- **Loop** – if enabled, scrolling will loop.
- **Lazy Loading** – delays loading of images. If enabled, images outside of viewport will not be loaded before user scrolls to them.

Brand slider block code and parameters

To display brand slider, enter the following code in the content field of any page or static block, Magento will automatically replace the code with a slider during page rendering:

```
{{block class="Infortis\Brands\Block\Brands" template="brand_slider.phtml" breakpoints="[0, 1],
[320, 2], [480, 2], [768, 3], [992, 4], [1200, 5]" block_name="Our Brands"}}
```

Here is the list of available parameters. Some of them can override global settings:

Required parameters:

- **block_name** – the name of the block displayed as a heading.

Optional parameters:

- **brands** – with this parameter you can specify the list of brands (separated by comma) which will be displayed in the slider. You can have multiple sliders and each can display completely different set of brands – just provide different list of brands in each slider code using this parameter, for example:

```
brands="Apple, Samsung, Louis Vuitton, Coca-Cola"
```

  This list will override the list of brands which was selected directly in the admin panel in `Stores > Configuration > Brands > Brand List` section. But if this parameter is not specified, slider will display brands selected in the admin panel.

- **breakpoints** – with this parameter you can specify the number of items in a row visible with a particular browser viewport width. Use the following format (this is the default value, it will be used if this parameter is not specified in the block code):

```
breakpoints="[0, 1], [320, 2], [480, 2], [768, 3], [992, 4], [1200, 5]"
```

  Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers `[A, B]` number `A` indicates the web browser viewport width, number `B` indicates the number of items visible if browser viewport width is greater than the width specified in `A`.
  In other words: display `B` items if viewport width is above `A` pixels. Below you can find some examples:

  - **Example 1:**

```
breakpoints="[0, 2], [480, 3], [992, 5]"
```

    which can be translated as: display 2 items if viewport width is between 0px and 480px, display 3 items if viewport width is above 480px, display 5 items if viewport width is above 992px.

  - **Example 2:**

```
breakpoints="[0, 4]"
```

which can be translated as: always display 4 items.

- **Example 3:**

```
breakpoints="[0, 3], [992, 6]"
```

which can be translated as: display 3 items on narrow screens (if viewport width is below 992px), and display 6 items on wider screens (above 992px).

- `is_responsive` – enter `0` if the slider **should not be responsive**. If this parameter is not specified, it is always assumed that the slider is responsive.

> **Important:**
>
> If the slider **is not** responsive, the number of visible items should be specified with the additional **show_items** parameter (see below).

  - *`show_items`* – the number of products in a row in the non-responsive slider. Use this parameter only if responsive behavior was disabled with parameter `is_responsive` (see above)

- `timeout` – to animate the slider automatically, specify the time (in milliseconds) between transitions. Note that 1000 milliseconds = 1 second.

- `move` – number of items that should move on animation. Allowed values:

  - enter `1` to move one item.

  - enter `0` to move all visible items. This is also the default value which is used if the parameter is not specified.

- `loop` – enter `1` if the slider should loop.

- `pagination` – enter `1` to display pagination.

12. Menu

This chapter refers to the following sections of the admin panel:

- General menu settings:
  `Stores > Configuration > Menu`

  Description of general settings can be found in this chapter:
  12.1 Menu configuration

- Menu settings which can be applied individually for each category:
  `Catalog > Categories`, in the `Menu` tab.

  Description of individual category settings can be found in this chapter:
  12.2 Drop-down boxes

More information about category management in Magento can be found in chapter **4.5 Categories**.

> **Note:**
>
> in Magento you can limit number of category levels displayed in the menu. Refer to **4.5.4 Depth of the menu** for more details.

12.1 Menu configuration

Below you can find a list of the menu settings available in the admin panel `Stores > Configuration > Menu`.

12.1.1 Top Menu

- **Top Menu** – enable/disable the horizontal drop-down menu in the header section of the page.
- **Mode** – select menu mode.
  - **Drop-down** – standard drop-down menu.
  - **Mobile** – collapsible accordion dedicated for mobile websites.
  - **Drop-down/Mobile** – combination of both modes, dedicated for responsive websites. You can specify the width of the browser viewport below which the drop-down menu turns into the mobile menu (see the **Threshold** field under the **Mobile Menu** tab).
- **Display Categories** – display categories in the top menu.

Vertical Layout

- **Vertical Drop-down Menu** – enable vertical layout of the menu.

  In this layout, categories are stacked vertically inside a drop-down box which can be shown/hidden when user clicks the button inside the menu bar. Subcategories are displayed in drop-downs at the right side of the main drop-down.

Home Link

- **Home Link: Text** – if enabled, will display menu item with text "Home" as a link to the home page. This link will be displayed with the same colors as other menu items.
- **Home Link: Icon** – if enabled, will display menu item with icon as a link to the

home page. This link will be displayed with the same colors as other menu items.

- **Single Icon as Home Link** – enable to show single icon as a link to the home page. Icon will be displayed without any additional background color. If it's enabled, options *Home Link: Text* and *Home Link: Icon* will be ignored.

Additional Settings

- **Align Drop-down With...** – the drop-down box (of top-level category) will be aligned with the right edge of the selected container, if the right edge of the drop-down box sticks out further to the right than the selected container.
  - **Viewport** – drop-down box will be aligned with the user's visible area of the page.
  - **Menu bar** – drop-down box will be aligned with the right edge of the menu bar.
  - **Primary header, inner container** – drop-down box will be aligned with the inner container of the primary header section.
- **Full-width Drop-down As Wide As...** – if a drop-down box of a top-level category was set as full-width (the *Submenu Type* field was set to `Mega drop-down` and the *Drop-down Width* field was left empty), it will be as wide as the selected container.

> **Note:**
>
> width of a drop-down box can be configured individually for each category in category settings.

- **Custom Static Block - Align Right** – align custom static block to the right side of the menu bar.

  To add custom static block to the menu you need to create a static block with ID `block_nav_dropdown` under *Content > Blocks*. Content of that static block will be displayed as a drop-down box.

> **Note:**
>
> there is another method to add custom drop-down boxes to the menu, for more details refer to **12.4 Custom drop-down boxes in the menu**.

12.1.2 Mobile Menu

- **Threshold** – width of the browser viewport (in pixels), below which the drop-down menu turns into a mobile menu (an accordion) dedicated for mobile devices. The threshold is needed only if the top menu is in the default `Drop-down/Mobile` mode.

> **Important:**
>
> The default value of the threshold:
>
> **- Horizontal menu:**
> for horizontal menu (when Vertical Drop-down Menu is not enabled) it's recommended to leave the default value of the threshold: 960. Values below 960 are not recommended - horizontal drop-down menus usually don't look good on narrow screens.
>
> **- Vertical menu:**
> if the Vertical Drop-down Menu is enabled, it's recommended to set the threshold to 770.

- **Show Category Blocks From X Levels of Categories** – specify number of levels of categories (counting from the top level) for which category blocks will be displayed in mobile menu.

  E.g. enter `1` to display blocks of 1st-level categories, enter `2` to display blocks of 1st-level and 2nd-level categories, enter `3` to display blocks of 1st-level, 2nd-level and 3rd-level categories, and so on.

  The allowed range is from `0` to `5`.

> **Important:**
>
> enter `0` to hide blocks of all categories on all levels in the mobile menu.

- **Hide Category Blocks Below 320px** – category blocks will be hidden in the mobile menu on small mobile devices (if width of the browser viewport is below 320px).

12.1.3 Sidebar Menu

In this section you can enable and configure the sidebar menu: vertical list of categories which can be displayed at the top of the sidebars on selected pages.

> **Note:**
>
> you can also display the sidebar menu inside any static block and on any CMS page by using the sidebar menu block code. The code is described later in this chapter.

Enable Sidebar Menu

Here you can enable/disable sidebar menu in different locations of the store.

You can choose in which sidebar the menu will be displayed. Depending on the layout (one, two or three columns), each page might have one or two sidebars (or no sidebar at all). Layout can be configured individually for each page (e.g. in category's settings, in CMS page's settings etc.). To change the layout of a page, change the value of its `Layout` field.

Note that if the `Layout` of a page was set to `3 columns`, the Main Sidebar will be displayed at the left side of the main content column and the Additional Sidebar will be displayed at the right side of the main content column.

If the `Layout` was set to `2 columns with left bar`, the Main Sidebar and the Additional Sidebar will be displayed at the left side of the main content column. In two column layout the Main Sidebar is always displayed above the Additional Sidebar.

If the `Layout` was set to `2 columns with right bar`, the Main Sidebar and the Additional Sidebar will be displayed at the right side of the main content column. In two column layout the Main Sidebar is always displayed above the Additional Sidebar.

Refer to Magento documentation for more details about page layouts:
– http://docs.magento.com/m2/ce/user_guide/design/page-layout-standard.html
– http://docs.magento.com/m2/ce/user_guide/design/page-layout-examples.html

Here is the list of options to enable the sidebar menu:

- **Category View (Main Sidebar)** – enable/disable menu in category view at the top of the Main Sidebar. It will be displayed above Magento's Layered Navigation – for

more details about Layered Navigation refer to **4.5.5 Layered Navigation ("Shop By" filters)**.

- **Category View (Additional Sidebar)** – enable/disable menu in category view at the top of the Additional Sidebar.
- **Home Page (Main Sidebar)** – enable/disable menu on home page at the top of the Main Sidebar.
- **Home Page (Additional Sidebar)** – enable/disable menu on home page at the top of the Additional Sidebar.
- **CMS Pages (Main Sidebar)** – enable/disable menu on CMS pages at the top of the Main Sidebar.
- **CMS Pages (Additional Sidebar)** – enable/disable menu on CMS pages at the top of the Additional Sidebar.
- **Product Page (Main Sidebar)** – enable/disable menu on product pages at the top of the Main Sidebar. Note that by default a product page have no sidebars, so you need to change the layout of a product page to display the sidebar menu. You can change the layout in product's settings.
- **Product Page (Additional Sidebar)** – enable/disable menu on product pages at the top of the Additional Sidebar. Note that by default a product page have no sidebars, so you need to change the layout of a product page to display the sidebar menu. You can change the layout in product's settings.

Sidebar Mega Menu

- **Sidebar Mega Menu** – enable sidebar mega menu. Sidebar mega menu can display drop-downs in a similar way to the top menu. Categories are stacked vertically and sub-categories are displayed in drop-downs at the right side of the menu.

  If this option is not enabled, the sidebar menu will be displayed as a simple accordion. Additional settings for accordion can be found below.

- **Use Sidebar Mega Menu As Mobile Menu** – when the sidebar menu exists on the page, it will be used as mobile menu (when mobile menu is displayed).

Block Settings

- **Block Title** – the name of the block displayed at the top of the block.

Additionally you can use variable `[current_category]` inside the title to create dynamic name of the block. Variable will be replaced with the name of the current category. On pages other than category view or product view this variable will be empty.

Accordion Settings

- **Parent Category** – the menu will only contain children (subcategories) of the selected parent category. Available options:
  - **Root** – show all top-level categories in the store.
  - **Parent of current category** – show the current category and its siblings (other categories from the same level as the current category).
  - **Parent of current category (no siblings)** – show the current category. Other categories from the same level as the current category will be ignored.
  - **Current category** – show subcategories of the current category.
- **Maximum Depth** – this parameter controls how many levels in the hierarchy of categories (starting from the parent category) are to be included in the menu. Menu with a Maximum Depth of `1` will just be an un-nested list of categories from a single level (with none of the subcategories displayed). The default value is `0`, it will display all subcategories.
- **Fallback To Root Category** – if current category doesn't exist, go back to root category. This option is useful if sidebar menu should be displayed on pages with no current category (on pages other than category view). If set to “Yes” and if current category doesn't exist, the menu will behave like **Parent Category** field was set to “Root” (it will display top-level categories). If set to "No", menu will not be displayed at all if there's no current category.

  For example, if menu is displayed on the home page (**Parent Category** field is set to “Current category”) and **Fallback To Root Category** field is set to “Yes”, then the menu will display top-level categories.

  If in the same example **Fallback To Root Category** field is set to “No”, then the menu will not show any categories and will not be displayed at all.
- **Fallback Block Title** – this block title will be used instead of the standard block title, if **Fallback To Root Category** field is set to “Yes” and if current category doesn't exist.
- **Show Number of Products** – show number of products next to each category.

Additional Settings

- **Hide Sidebar Menu on Mobile Devices** – list of categories in the sidebar can be hidden on mobile devices (if browser viewport is narrower than 768 pixels). All categories are displayed in the top main menu by default so with this option you can save some space on mobile devices.

- **Hide Magento's Category Filter** – if you enabled sidebar menu, you may want to disable category filter in Magento's Layered Navigation block ("Shop By") in category view.

Sidebar menu block code

You can also display the sidebar menu inside the content of any CMS page or custom static block (and custom static block can be later added to any page through Magento widgets system – see this chapter for more details about adding custom content to selected pages: **4.8.1 Add blocks to selected pages**).

To display the menu, type in the menu code inside the static block content (Magento will automatically replace the code with the menu in the frontend), save the static block and flush Magento cache. Here's an example of the sidebar menu block code:

```
{{block class="Infortis\UltraMegamenu\Block\Navigation" template="categories.phtml" parent="X"
depth="X" block_name="X"}}
```

There are several parameters that can help you configure the menu to suit your needs. Replace **X** with proper values of the parameters:

- **parent** – the menu will only contain children (subcategories) of the selected parent category. This parameter is similar to field **Parent Category** which is described earlier in this chapter. Available options:

  - `root` – show all top-level categories in the store.

  - `parent` – show the current category and its siblings (other categories from the same level as the current category).

  - `parent_no_siblings` – show the current category. Other categories from the same level as the current category will be ignored.
  - `current` – show subcategories of the current category.
  - Or specify the ID (an integer number) of any existing category.

- **depth** – specify how many levels in the hierarchy of categories (starting from the parent category) are to be included in the menu. This parameter is similar to Maximum Depth in the admin panel.
- **block_name** – the name of the block displayed as a heading.

Examples

Here you can see some examples of the block code:

```
{{block class="Infortis\UltraMegamenu\Block\Navigation" template="categories.phtml"
template="categories.phtml" parent="root" depth="2" block_name="All Categories in our store"}}
```

```
{{block class="Infortis\UltraMegamenu\Block\Navigation" template="categories.phtml"
parent="parent" depth="1" block_name="Current level of categories"}}
```

```
{{block class="Infortis\UltraMegamenu\Block\Navigation" template="categories.phtml"
parent="parent_no_siblings" depth="3" block_name="Current category with subcategories"}}
```

```
{{block class="Infortis\UltraMegamenu\Block\Navigation" template="categories.phtml"
parent="current" depth="4" block_name="Subcategories of current category"}}
```

12.1.4 Category Labels

- **Label 1** – define the text of the category label 1. Labels can be assigned to every category under `Products > Categories`.
- **Label 2** – define the text of the category label 2. Labels can be assigned to every category under `Products > Categories`.

12.2 Drop-down boxes

12.2.1 Drop-down box configuration options

Each category in the menu can have a drop-down box containing subcategories or any other custom content. Below you can find a list of the drop-down box settings available in the admin panel. To configure the drop-down, go to `Products > Categories`, select a category and open the `Menu` tab:

- **Submenu Type** – if category has subcategories, choose how subcategories should be displayed. You can choose the type individually for each category. For more details refer to **12.2.2 Submenu types**.

- **Drop-down Width** – in this field you can override the default width of the drop-down box. Width can be specified:

  - in pixels, for example: `150px`,

  - or as a percentage, for example: `200%` (percentage of the containing block's width).

  In top-level categories (and only in top-level categories) if the *Submenu Type* is set to `Mega drop-down`, then the width of the drop-down box can also be specified in grid units (number between 1 and 12). For example: enter `4` and the drop-down box will have a width of 4/12 of the entire menu bar.

  Default width of drop-downs:

  - The mega drop-down of a top-level category is by default as wide as the menu bar.

  - The mega drop-down of a category on lower levels is by default as wide as the menu item of that category. So it is necessary to override the default width. For example, if you enter `300%`, the drop-down will be three times as wide as the category item.

  - The classic drop-down is by default around 240 pixels wide.

- **Drop-down Content Proportions** – proportions between three sections of the mega drop-down box:

  a) content of the *Left Block* field

  b) block of subcategories

  c) content of the *Right Block* field

  You need to enter three numbers. Each number defines width of one section. Width is expressed in grid units (numbers between 0 and 12). For each section enter a number between 0 and 12. Sum of the numbers entered for all three sections has to be equal to 12.

> **Note 1:**

category blocks are displayed only in mega drop-down boxes so this option is applicable only for categories with `Mega drop-down` selected in the *`Submenu Type`* field.

Note 2:

Since Magento 2.1.0 proportions between these three sections have to be specified in the following format: `3;6;3;`

Numbers are separated with semicolon, each number defines width of one section.

You can also leave this field empty to use the **default value**, which is: `4;4;4;`

Below you can see a few examples of values which can be entered in the *`Drop-down Content Proportions`* field.

Example 1:

If you want the block of subcategories to be as wide as the Left Block and the Right Block, enter `4;4;4;` This is also the default value of this field.

Remember that the sum of the numbers entered for all three sections has to be equal to 12. In this example we have 4+4+4=12, so the sum is correct.

Example 2:

If you don't want to display the Right Block but you want to have wide block of subcategories, enter `4;8;0;`

Note:

If you don't want to display the Right Block, you also need to leave the *`Right Block`* field **empty**.

As you can see, we entered `0` for the Right Block, and we increased the value for subcategories. In this example we have 4+8+0=12, so again the sum is correct.

Example 3:

If you don't want to display the Right Block but you want the block of subcategories to be as wide as the Left Block, enter `6;6;0;` Remember to leave the *Right Block* field empty if you don't want to display it.

Example 4:

If you want to have wide block of subcategories, very narrow Left Block and very narrow Right Block, enter `2;8;2;`

- **Number of Columns With Subcategories** – in this field you can select the number of columns in which the subcategories are displayed in the drop-down box. For example, select `3` to display subcategories in three columns. Leave this field empty to use the default value. Default value is `4`.

> **Note:**
>
> subcategories are displayed in columns only in mega drop-down boxes so this option is applicable only for categories with `Mega drop-down` selected in the *Submenu Type* field (in the classic drop-downs subcategories are displayed one below the other).

- **Top Block** – category block displayed in the drop-down box above the block of subcategories. It's one of the four blocks with custom content.

  If `Mega drop-down` was selected in the *Submenu Type* field, you can add custom content to the drop-down box. Content can be inserted into four fields: *Top Block*, *Left Block*, *Right Block*, *Bottom Block*. Content of these fields will be display in the drop-down box.

  To distinguish these blocks from other blocks in Magento, we will call them **category blocks**. For more details refer to **12.2.3 Category blocks**.

  If you don't want to display any additional content in the drop-down box, simply **leave these four fields empty**.

> **Note:**
>
> if you need to split your content inside category blocks into columns, you can use the built-in grid system – refer to **6.4 Grid system** for more details about the grid.

- **Left Block** – category block displayed at the left side of the block of subcategories.
- **Right Block** – category block displayed at the right side of the block of subcategories.
- **Bottom Block** – category block displayed below the block of subcategories.
- **Category Label** – eye-catching labels for categories. Labels needs to be defined in *Stores > Configuration > Menu > Category Labels*. Refer to **12.1.4 Category Labels** for more details.
- **Custom URL** – enter a single hash character (#) to make the category **not clickable** in the main menu. It is especially useful when you want to create a drop-down box in the menu (to display some custom content) but don't want to redirect customers to any specific category.

  Leave this field empty if no changes are needed and the category in the menu has to be a link to the category page.

  You can also use this field **to replace the default link of the category** with any custom link. To do this, enter custom URL path into this field. The path will be appended to the base URL of the store to create a new link. For example, enter `contact/` to create a link to Magento's default “Contact Us” page, or `customer/account/create/` to create a link to “Create an Account” page.

  More info about adding custom links to the menu can be found in **12.3 Custom links in the menu**.

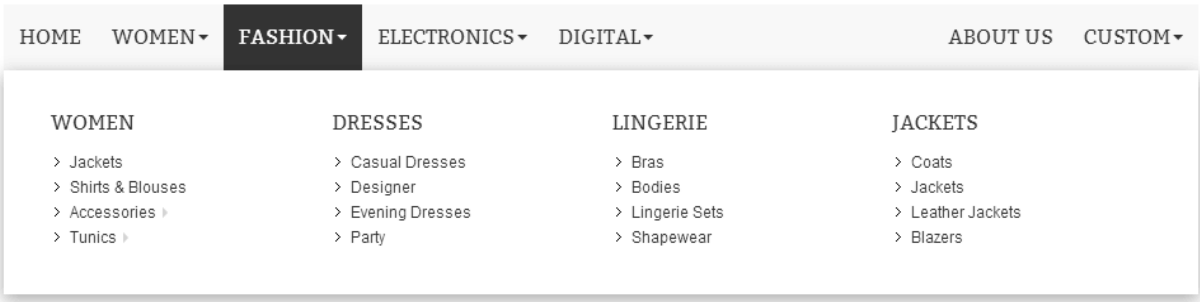
12.2.2 Submenu types

Available types

If the category has subcategories (i.e. it has a submenu), you can choose how the subcategories will be displayed. Go to *Products > Categories*, select a category and open the *Menu* tab. There are three available options in *Submenu Type* field:

1. **Mega drop-down**

   Wide drop-down box which can display subcategories in multiple columns. It can also display additional content from the category blocks (which are described later in this chapter).



2. **Classic drop-down**

   Simple drop-down box which displays subcategories one below the other. It can display any number of levels of categories. Classic drop-down is the default type and it will be used if no type was specified for the category.

3. Simple submenu (no drop-down)

Additional type which can be used to display subcategories not in a drop-down box, but just below the category as a simple bulleted list.

> **Note:**
>
> this option can't be applied to top-level categories.

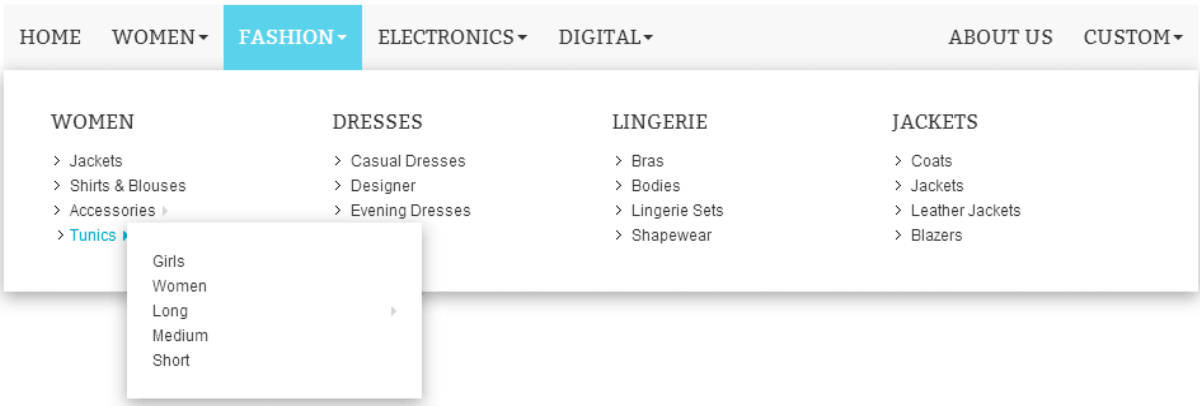
If in the previous example we will change the type of the “Dresses” category to `Simple submenu (no drop-down)`, the drop-down box will look like this:

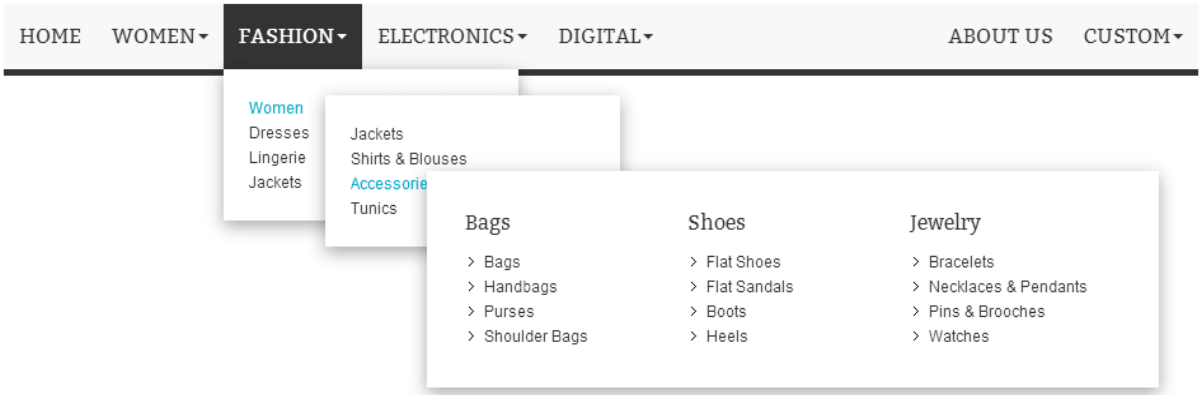
So the subcategories of the “Dresses” category will be displayed as a simple bulleted list.

Multilevel menus

Multilevel menus are fully supported, categories inside a drop-down box can have their own subcategories which will also be displayed in drop-down boxes.

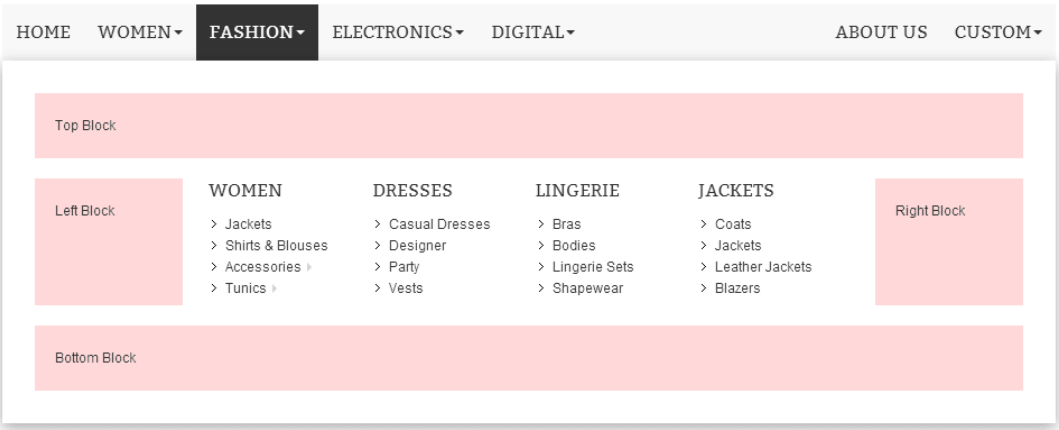
Additionally, all of the submenu types can be mixed together. For example, a classic drop-down can be nested inside a mega drop-down, and a mega drop-down can have classic drop-downs inside. See examples below:


HOME
WOMEN
FASHION
ELECTRONICS
DIGITAL
ABOUT US
CUSTOM
WOMEN
Jackets
Shirts & Blouses
Accessories
Tunics
DRESSES
Casual Dresses
Designer
Evening Dresses
LINGERIE
Bras
Bodies
Lingerie Sets
Shapewear
JACKETS
Coats
Jackets
Leather Jackets
Blazers
Girls
Women
Long
Medium
Short

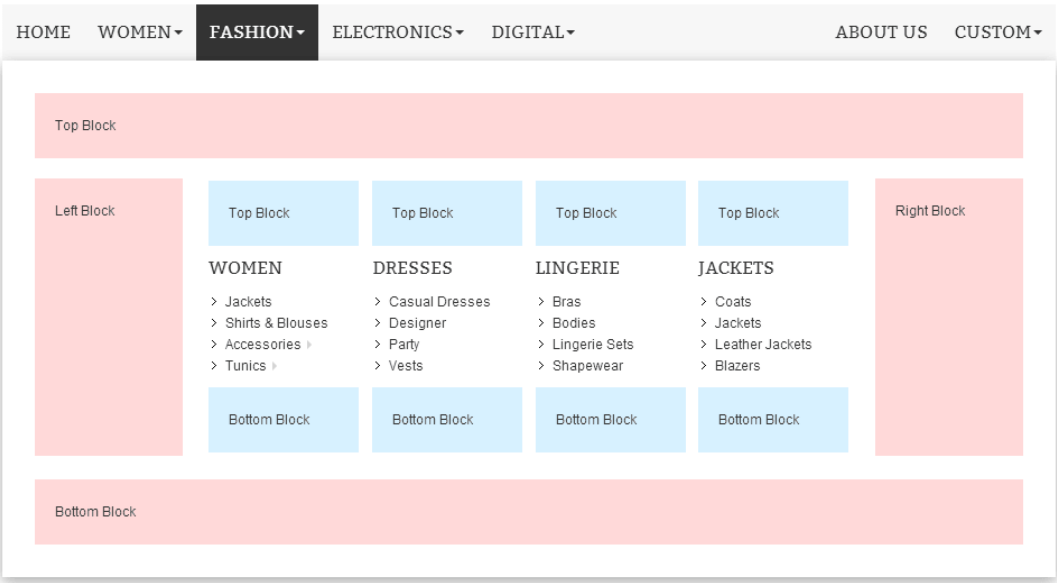

HOME
WOMEN
FASHION
ELECTRONICS
DIGITAL
ABOUT US
CUSTOM
Women
Dresses
Lingerie
Jackets
Jackets
Shirts & Blouses
Accessorie
Tunics
Bags
Bags
Handbags
Purses
Shoulder Bags
Shoes
Flat Shoes
Flat Sandals
Boots
Heels
Jewelry
Bracelets
Necklaces & Pendants
Pins & Brooches
Watches

12.2.3 Category blocks

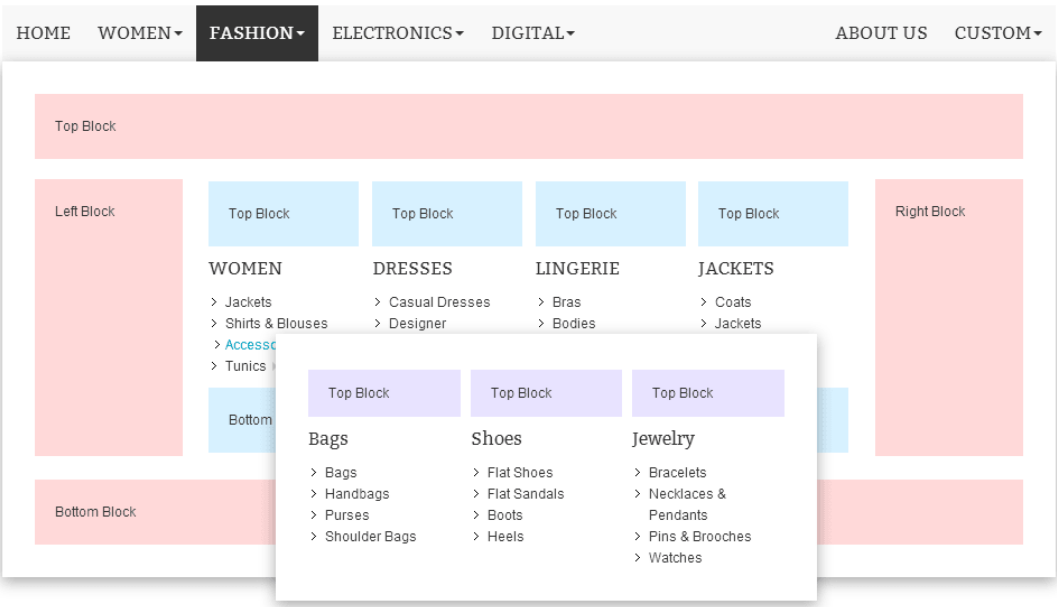
If `Mega drop-down` was selected as the *`Submenu Type`* field, you can display additional content inside the drop-down box. Custom content can be inserted into four **category blocks**. On the example below, blocks of the “Fashion” category are marked with light red:



Mega drop-downs can also display category blocks of lower categories. For example, if the subcategories of the “Fashion” category (“Women”, “Dresses”, “Lingerie” and “Jackets”) have any content inside Top Block and Bottom Block, these blocks will be displayed in the drop-down. On the example below, these blocks are marked with light blue color:



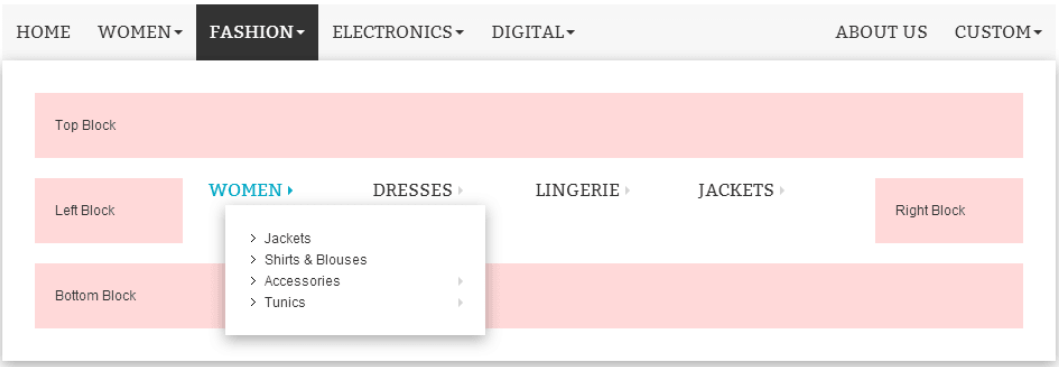
Moreover, subcategories of the “Fashion” category can have their own drop-down boxes. For example, if the “Accessories” category is configured as a mega drop-down (i.e. `Mega drop-down` was selected in the *`Submenu Type`* field), then a drop-down box will be displayed on mouse hover over the “Accessories” category. That drop-down can also contain its own category blocks with custom content (on the example below, these blocks are marked with light purple color):



Note that in this example, categories “Women”, “Dresses”, “Lingerie” and “Jackets” can't

have `Mega drop-down` selected in the *`Submenu Type`* field (if it was selected, it will be ignored) because their subcategories are already displayed in the mega drop-down.

But categories “Women”, “Dresses”, “Lingerie” and “Jackets” **can have** `Classic drop-down` selected in the *`Submenu Type`* field. If they are configured this way, menu will look like this:

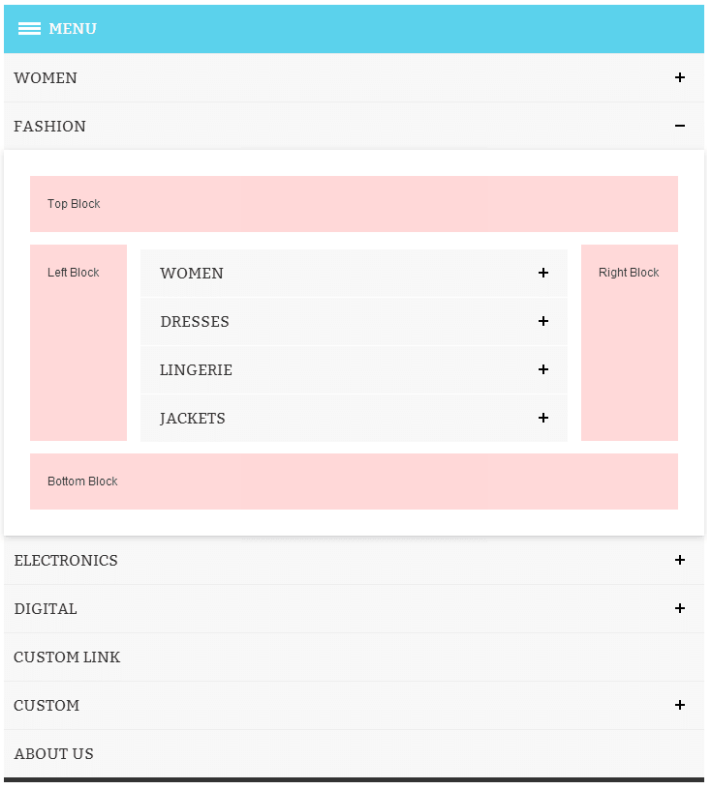


12.2.3.A Category blocks in mobile menu

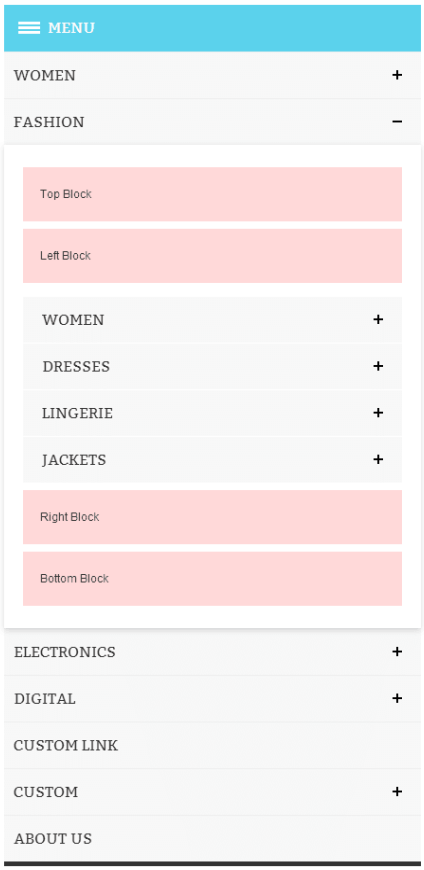
If the mobile menu is enabled, category blocks will also be displayed in the mobile menu (picture 1.). Mobile mode is activated when browser viewport is narrower than the threshold specified in the menu settings.

If screen is narrower than 768 pixels, category blocks will be displayed one below another (picture 2.), even the **Left Block** and the **Right Block**.

Picture 1.



Picture 2.



To hide category blocks in the mobile menu, you can use the **Show Category Blocks From X Levels of Categories** option which can be found in `Stores > Configuration > Menu > Mobile menu`. Refer to **12.1.2 Mobile Menu** for details.

You can also hide selected portions of the content inside the category blocks. There are two special classes:

- `hide-in-mobile-menu` – this class will hide the element when the menu is in the mobile mode
- `hide-in-desktop-menu` – this class will hide the element when the menu is in the standard mode.

Example below shows how to use these classes. Add this code inside one of the category blocks, e.g. **Top Block**:

```
<div class="hide-in-mobile-menu">
   This text will not be visible in mobile menu
</div>

<div class="hide-in-desktop-menu">
   This text will not be visible in desktop menu
</div>
```

12.2.4 Sample content for category blocks

Here you can find some samples of the content for category blocks. They are based on the samples from the demo.

Category "Women"

Content of the `Right Block`:

```
<a href="{{store direct_url='dress7.html'}}">
     <div class="ban ban-effect-3 ban-caption-fade-out">
             <img class="image" src="{{media url='wysiwyg/infortis/other/menu/01.jpg'}}"
alt="Sample custom content of the category block" />
             <div class="cap" style="background-color: rgba(0,0,0, 0.2);">
                     <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                             <span class="ic ic-gift ic-6x animated infinite pulse"></span>
                     </div>
             </div>
     </div>
</a>
```

Category "Fashion"

Content of the `Right Block`:

```
{{block class="Infortis\Base\Block\Product\ProductList\Featured"
template="product/list_featured_slider.phtml"
category_id="3" product_count="4" breakpoints="[0, 1], [480, 2], [768, 1], [960, 2]"
timeout="4000" move="1" centered="1" size="size-s" hide_button="1" img_width="110"
block_name="Recommended Bags"}}
```

Category "Electronics"

Content of the `Top Block`:

```
<div class="links-wrapper-separators">
     <ul class="links">
             <li class="label hidden-xs hidden-sm">Recommended:</li>
             <li>
                     <a href="{{store
```

```
direct_url='electronics/smartphones/phone3.html'}}">Configurable</a>
            </li>
            <li>
                    <a href="#">Galaxy S3</a>
            </li>
            <li>
                    <a href="#">Galaxy S4</a>
            </li>
            <li>
                    <a href="#">iPhone 5</a>
            </li>
     </ul>
</div>
<div class="links-wrapper-separators pull-right hide-in-mobile-menu hidden-xs hidden-sm">
     <ul class="links">
            <li>
                    <a href="#">Add</a>
            </li>
            <li>
                    <a href="#">Custom</a>
            </li>
            <li>
                    <a href="#">Links</a>
            </li>
     </ul>
</div>
<span class="section-line"></span>
```

Content of the `Left Block`:

```
<a class="hidden-xs" href="{{store direct_url='electronics/phone-accessories.html'}}">
     <div class="ban ban-caption-hinge-left">
            <img class="image" src="{{media url='wysiwyg/infortis/other/menu/02.jpg'}}"
alt="Sample custom content of the category block" />
            <div class="cap" style="background-color: #ffc60f;">
                    <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                            <span class="ic ic-star ic-4x animated infinite pulse margin-
bottom"></span>
                            <h3 class="no-margin">Go to category</h3>
                    </div>
            </div>
     </div>
</a>
<br/>
<h3>Featured Category</h3>
<p>This is a custom block ready to display any content. You can add blocks to any category in
the catalog...</p>
<a class="button" href="#">View Category  <span class="ic ic-arrow-right"></span></a>
```

Category "Phones & Tablets"

Content of the `Top Block`:

```
<a href="{{store direct_url='electronics/smartphones.html'}}">
     <div class="ban ban-caption-hinge-up">
             <span class="image ic ic-3x ic-smartphones ib ib-hover ib-square ib-size-xxl"
style="background-color: #333;"></span>
             <div class="cap">
                     <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                             <h4 class="no-margin">Phones & Tablets</h4>
                     </div>
             </div>
     </div>
</a>
```

Category "Accessories"

Content of the `Top Block`:

```
<a href="{{store direct_url='electronics/phone-accessories.html'}}">
     <div class="ban ban-caption-hinge-up">
             <span class="image ic ic-3x ic-callcenter ib ib-hover ib-square ib-size-xxl"
style="background-color: #333;"></span>
             <div class="cap">
                     <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                             <h4 class="no-margin">Accessories</h4>
                     </div>
             </div>
     </div>
</a>
```

Category "Cameras"

Content of the `Top Block`:

```
<a href="{{store direct_url='electronics/cameras.html'}}">
     <div class="ban ban-caption-hinge-up">
             <span class="image ic ic-3x ic-gift ib ib-hover ib-square ib-size-xxl"
style="background-color: #333;"></span>
             <div class="cap">
                     <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                             <h4 class="no-margin">Cameras</h4>
                     </div>
             </div>
     </div>
</a>
```

12.3 Custom links in the menu

The main menu typically includes the selection of categories from the catalog. But you can also add links to content pages and external websites. In order to add a link, follow the steps described in the example below.

Let's say you want to create a link to Magento's “Create an Account” page.

1. Go to *Products > Categories* and create a new category in your catalog by clicking the **Add Subcategory** button.

2. Fill in all the required fields. Especially enable fields **Enable Category** and **Include in Menu**, and then save the category.

3. The name of the category will be the name of the link so you can use any name you like.

4. Open the *Menu* tab. In the **Custom URL** field, enter URL path of the “Create an Account” page which is `customer/account/create/`. This path will be appended to the base URL of the store to create a link to that page.

   To create a link to some other page you just need to enter proper URL path. For example, to create a link to a CMS page, in the **Custom URL** field, enter the **URL Key** of that page. To check what is the **URL Key** of a page, go to *Content > Pages*, select a page and find the **URL Key** column in the table.

5. Finally, go to your store and make sure the new link appears in the main menu. If it's not, you may need to flush the cache (**4.1 Caching system**) and reindex your Magento (**4.2 Index management**).

12.3.1 Custom links inside a static block

Another method to add custom links to the main menu, is to create a static block (under *Content > Blocks*) with identifier `block_nav_links`. Content of this block will be displayed in the menu bar so you can quickly add multiple links to the menu.

To display links properly, content of the block should have the following structure:

*Admin panel: Stores > Configuration > **Menu***

```
<li class="nav-item level0 level-top right">
     <a class="level-top" href="{{store direct_url='about-magento-demo-store'}}">
             <span>Sample Internal Link</span>
     </a>
</li>
<li class="nav-item level0 level-top right">
     <a class="level-top" href="http://infortis-themes.com/">
             <span>Sample External Link</span>
     </a>
</li>
```

Important:

note that there are no `<ul>` tags at the beginning and at the end of this code snippet. This code will be inserted inside the menu bar HTML which already contains `<ul>` tags and additional tags would break the structure of the menu.

You can remove class `right` if you don't want the links to float to the right side of the menu bar.

Also, you may need to disable Magento's WYSIWYG editor, otherwise it will add `<ul>` tags automatically. It is recommended to disable the WYSIWYG completely so that it doesn't process the code when you don't need it. Turn WYSIWYG on only when it is needed. Go to *`Stores > Configuration > Content Management`* and set it as *`Disabled by Default`*.

For more information about CMS **markup tags**, such as `{{store direct_url='about-magento-demo-store'}}`, refer to **4.8.3 CMS markup tags**.

12.4 Custom drop-down boxes in the menu

Drop-down boxes with custom content can be easily added to any top-level category in the main menu. In order to add a drop-down, follow the steps described in the example below.

1. Go to `Products > Categories` and create a new category in your catalog by clicking the **Add Subcategory** button.

2. Fill in all the required fields. Especially enable fields **Enable Category** and **Include in Menu**, and then save the category.

3. The name of the category will be the name of the item in the menu so you can use any name you like. Drop-down box will be displayed on mouse hover over that item.

4. Open the `Menu` tab. In the **Submenu Type** field, select “Mega drop-down”.

5. In the **Top Block** field, enter the content which you want to display in the drop-down box. It can be any HTML or just a simple text – whatever you need:

6. In the **Custom URL** field, enter a single hash character (`#`) to make this category not clickable (or leave this field empty if no changes are needed and the category in the menu has to be a link to a category page).

7. Finally, go to your store and make sure the new item appears in the menu. If it's not, you may need to flush the cache (**4.1 Caching system**) and reindex your Magento (**4.2 Index management**).

Static block as a drop-down box

Another method to add custom drop-down box to the main menu, is to create a static block (under `Content > Blocks`) with identifier `block_nav_dropdown`. Content of this block will be displayed in the menu as a drop-down box. Title of that block will be used as the name of the menu item.

To align this menu item to the right side of the menu bar, use option **Custom Static Block - Align Right** which can be found in `Stores > Configuration > Menu > Top Menu`.

13. Troubleshooting

13.1 General issues after installation

13.1.1 Some features don't work or give errors

Here's the list of possible sources of problems:

a) Magento cache (or external cache) was not flushed after installation of the theme.

b) CSS files which are responsible for the visual appearance of the theme were not generated automatically. Go to `Stores > Configuration > Ultimo > Theme Design` and click **Save Config** button. Then go to `Stores > Configuration > Ultimo > Theme Layout` and click **Save Config** button again. Then, flush Magento cache and refresh your browser's cache.

c) Some of the theme files were not uploaded or are broken. Re-upload original theme files to your server (overwrite existing files) and flush the cache. Note that files can get broken if you send them in large numbers via FTP without zipping. So it is highly recommended to upload zipped files.

d) File permissions issue – refer to **13.1.4 File permissions** for more details.

e) Theme wasn't enabled properly.

f) You have some third-party extensions which change/break theme's default behavior or cause some conflicts. Disable all extensions and flush the cache, then enable extensions one by one to check which one is causing problems. Refer to **2.3 Third-party extensions** for more details about third-party extensions.

g) You edited theme files and accidentally removed or changed something in the files. Re-upload original theme files to your server (overwrite existing files) and flush the cache.

h) You have customized the theme by creating custom sub-theme and your modifications override or break the default behavior of the theme.

13.1.2 Blank page or 404 not found – when trying to open theme admin pages

Log out from Magento admin panel, delete all files in `var/cache/` directory (to clear Magento cache), and log in again.

13.1.3 There is no theme admin tab in the admin panel, or I get a blank page, or Access Denied error

Log out from Magento admin panel, delete all files in `var/cache/` directory, clear web browser cache/cookies for your store domain and log in again.

13.1.4 File permissions

Incorrect file permissions on your server may cause many issues. After uploading any files to your server in order to install themes or extensions you need to set correct file permissions for all the uploaded files – the same permissions as for the default Magento files.

File permissions may differ depending on server – usually 755 for folders and 644 for files. In case of any doubts about permissions, please contact your hosting provider. Refer to this article for more details:
http://www.magentocommerce.com/wiki/1_-_installation_and_configuration/magento_filesystem_permissions

In case of problems related to images refer to: **13.1.9 Images are not displayed**

13.1.5 Theme features don't work after installation or upgrade

If some theme features don't work after installation or some resources (e.g. images) are not displayed in the frontend, it often indicates problems with file permissions – your Magento doesn't have access to the theme files because of incorrect file permissions on your server. For more details refer to **13.1.4 File permissions**

After theme upgrade

If you upgraded the theme in your store and you're using custom sub-theme, you will also need to **upgrade your sub-theme**. This is extremely important step: review all the files in your sub-theme to make sure they don't override new features added to the theme.

For example: if the product page template file was changed in the update (e.g. a new option was added: ability to display product SKU), you will need to apply that change in your sub-theme (in the product page template file). Otherwise SKU will not be displayed in your store because Magento will use the template file in your sub-theme to render the product page and your template file doesn't have the new feature. So you will need to add the new feature manually (copy the new code from the original template file to the file in your sub-theme).

13.1.6 "Package with this name does not exist" message is displayed when trying to enable the theme

This message means that:

a) theme files were not uploaded,

b) or they are not in the correct place,

c) or there is a problem with file permissions on your server – refer to chapter **13.1.4 File permissions** for more details.

13.1.7 After theme installation the store gives blank page or an error

Log out from Magento admin panel. Change the owner of the uploaded files to the web server user and set correct file permissions for them. Delete all files in `var/cache/` directory (do not delete the directory) and log in again.

13.1.8 Tabs, sliders or other interactive elements don't work

This may be caused by jQuery conflict: one of your custom extensions is loading another version of jQuery. You need to remove that version of jQuery or remove that extension.

13.1.9 Images are not displayed

Sometimes after installation some images on custom CMS pages or inside static blocks (e.g. images in the slideshow) are not displayed. First, check if images were uploaded and if correct paths to those files can be found in the source code of your page (click `CTRL+U` in most browsers).

> **Note:**
>
> in fresh Magento installation, the default sample home page tries to load sample images which doesn't exist in this theme (so you will see blank rectangles instead of images). Go to *Content > Pages > Home page*, remove the default content or replace it with your own content.

If all files were uploaded and paths are correct, it means that your Magento doesn't have access to those images because of incorrect file permissions on your server. You need to set correct file permissions for all images and folders containing images (in most cases images are stored in Magento's `media` directory). Refer to chapter **13.1.4 File permissions** for more details.

13.1.10 Sliders: previous/next buttons don't work in some browsers

Please see this thread on Flexslider forum (Flexslider is a jQuery plugin used for sliders), some users reported that the issue can be caused by a Wacom Driver, they were able to resolve the issue by unplugging their tablets:
https://github.com/woothemes/FlexSlider/issues/589#issuecomment-14804424

You may also want to reinstall your web browser and remove all plugins to check if some of them can cause conflicts.

13.1.11 Slideshow is not displayed on the home page

Make sure the configuration of the slideshow is correct. Enable all the static blocks which are used as slides in the slideshow. Finally, flush the cache.

If this doesn't help:

- Non-standard value in *Stores > Configuration > Web > Default Pages > Default Web URL*. Set the default value: `cms`

- Some custom extensions change the default behavior of Magento's home page, so the slideshow module is not able to load on the home page anymore. How to recognize that case: open the source code of the home page in your web browser (`CTRL+U` in most browsers). The `<body>` tag on the home page should have CSS class `cms-index-index` (this is Magento's default class). If you don't see this class, then it is probably that case, and you will need to disable the extension which is changing the default behavior of Magento's home page.

13.2 Configuration

13.2.1 Theme configuration changes do not appear on the frontend

Flush Magento cache in `System > Cache Management`, clear web browser cache/cookies for your store domain and refresh the page in your web browser.

> **Note:**
>
> if you created a sub-theme to customize the theme, do not copy folder `skin/frontend/<ThemeName>/default/css/_config/` (and files stored inside) to your sub-theme. If you do this, files inside this folder will override your theme settings.

13.3 Third-party extensions

13.3.1 One of my extensions doesn't work with this theme

This theme may not be compatible with some third-party extensions. It’s impossible for theme author to make the theme compatible out of the box with all existing extensions because there are thousands of available Magento extensions.

Generally all extensions can work with all themes but sometimes they have to be integrated manually. That’s because sometimes the extension and the theme override the same Magento template file so there can be a conflict. In such cases you need to customize the theme and merge two conflicting template files into one template file.

This isn't something specific to our themes, this is general rule for all Magento themes and extensions.

Refer to chapter **2.3 Third-party extensions** for more details.

13.3.2 One of the features stopped working after third-party extension was installed

There can be a conflict, if the extension and the theme override the same Magento template file. You should contact author of the extension for more information. Also, check your log files in `var/log/` for more information about the error. Refer to chapter **2.3 Third-party extensions** for more details about third-party extensions.