

ultimo theme

Update Guide

Copyright © 2012-2015 Infortis

All rights reserved

Table of Contents

1. Important changes.....	3
1.1 General guidelines.....	3
1.2 Ultimo version 1.15.4.....	3
1.3 Ultimo version 1.15.0.....	5
1.4 Ultimo version 1.14.0.....	9
1.5 Ultimo version 1.13.0.....	9
1.6 Ultimo version 1.12.0.....	10
1.7 Ultimo version 1.11.1.....	12
1.8 Ultimo version 1.11.0.....	13
1.9 Ultimo version 1.10.4.....	13
1.10 Ultimo version 1.10.3.....	13
1.11 Ultimo version 1.10.2.....	14
1.12 Ultimo version 1.10.0.....	15
1.13 Ultimo versions 1.9.0 - 1.9.4.....	19
1.14 Ultimo version 1.8.0.....	24
1.15 Ultimo version 1.7.1.....	24
1.16 Ultimo version 1.7.0.....	24
1.17 Ultimo version 1.6.0.....	26
1.18 Ultimo version 1.5.10.....	29
1.19 Ultimo version 1.5.6.....	30
1.20 Ultimo version 1.5.3.....	31
1.21 Ultimo version 1.5.0.....	32
1.22 Ultimo version 1.4.0.....	34
1.23 Ultimo version 1.3.1.....	34
1.24 Ultimo version 1.3.0.....	35
2. Upgrade.....	36
2.1 Prepare your Magento.....	36
2.2 Upload files.....	38
2.3 Import sample blocks and pages.....	40
2.4 Problems after upgrade.....	41

1. Important changes

Here you can find description of the most important changes in selected versions. List of all changes in all versions can be found in **Changelog.txt** in the download package. You can upgrade directly to the latest version (and omit all versions between) but you need to follow all the instructions provided for each version which you want to skip.

1.1 General guidelines

a) Re-save theme settings after upgrade

To get rid of the old outdated CSS generated automatically by the theme (the old CSS may break the layout after you upload new version of the theme), re-save settings after upgrade. Go to *System > Configuration > Ultimo > Theme Design* and click **Save Config** button. Then go to *System > Configuration > Ultimo > Theme Layout* and click **Save Config** button again.

b) Review and update your sub-theme

This is extremely important step. If you're using sub-theme, review your modified files to make sure they don't override new features added to the theme.

c) Review and update your CSS

Review your CSS modifications in `custom.css` to check if they don't break any new styles added to the theme.

1.2 Ultimo version 1.15.4

This is a compatibility update for **Magento 1.9.2.2** and the patch bundle **SUPEE-6788**. Magento 1.9.2.2 and patch SUPEE-6788 address several security issues, all details can be found here:

- <http://magento.com/security/patches/supee-6788-technical-details>
- <http://magento.com/security/patches/supee-6788>
- <http://magento.stackexchange.com/a/87262/764>

You can simply upgrade your installation to Magento 1.9.2.2 version which includes all the latest security patches (SUPEE-5344, SUPEE-5994, SUPEE-6285, SUPEE-6482, SUPEE-6788).

If Magento upgrade is not possible due to some reasons, you still can apply the patch via FTP upload as shown in this article (useful if you have no SSH access):

<https://magentary.com/kb/install-supee-6788-without-ssh/>

Before applying patch SUPEE-6788, make sure to apply all previous patches.

White list of allowed blocks

Magento 1.9.2.2 introduces a white list of allowed block types. If a block was added to a CMS page or to a static block through a CMS shortcode (for example: `{{block type="cms/block"}}`), and the block is not on the white list, it will not be displayed on frontend. In that case you will see a notice in your system log file under `/var/log/`, similar to the following:

```
Notice: Undefined variable: block in D:/xampp/htdocs/_STAGE/m1922-00/app/code/core/Mage/Core/Model/Email/Template/Filter.php on line 187
```

Note:

log file will only be created if logging is enabled in your Magento. Logging can be enabled under *System > Configuration > Developer*.

The white list of allowed blocks can be found under *System > Permissions > Blocks* (only in Magento 1.9.2.2 and later).

The permission system normally only allows a reduced number of blocks:

```
core/template
catalog/product_new
```

If you are using Magento 1.9.2.2, Ultimo update 1.15.4 will add theme's blocks to the white list:

```
cms/block
catalog/product_list
ultimo/product_list_featured
ultraslideshow/slideshow
brands/brands
ultramegamenu/navigation
```

If installation of the theme update goes correctly, you will see all the added blocks on the white list. You will also find a log file `Infortis_Ultimo.log` under `/var/log/` with a confirmation: *"Successfully added allowed blocks to Magento's white list"*.

If some of the blocks are already on the white list (e.g. if you manually added blocks to the list), you can see a notice in the system log file similar to the following:

```
SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'cms/block' for key
'UNQ_PERMISSION_BLOCK_BLOCK_NAME', query was: INSERT INTO `permission_block` (`block_name`,
`is_allowed`) VALUES (?, ?)
```

1.3 Ultimo version 1.15.0

Below you can find the list of changes and improvements in this version.

Full width page

- You can select `1 column, full width` in the **Layout** field of the CMS page configuration and the content of the page will be stretched to full width of the browser viewport. See chapter **6.1.2.C Full width page** in the [user guide](#) for more

details

- Selected page sections can be stretched to full width (*System > Configuration > Theme Design*).

Header

- Mini Cart drop-down and Mini Compare drop-down can be displayed inside the menu bar. Position of these blocks can be changed under *System > Configuration > Theme Settings > Header*
- You can change the height of the top header and height of the following blocks in the primary header: Mini Cart drop-down, Mini Compare drop-down and Top Links. These options can be found under *System > Configuration > Theme Design*.

Footer

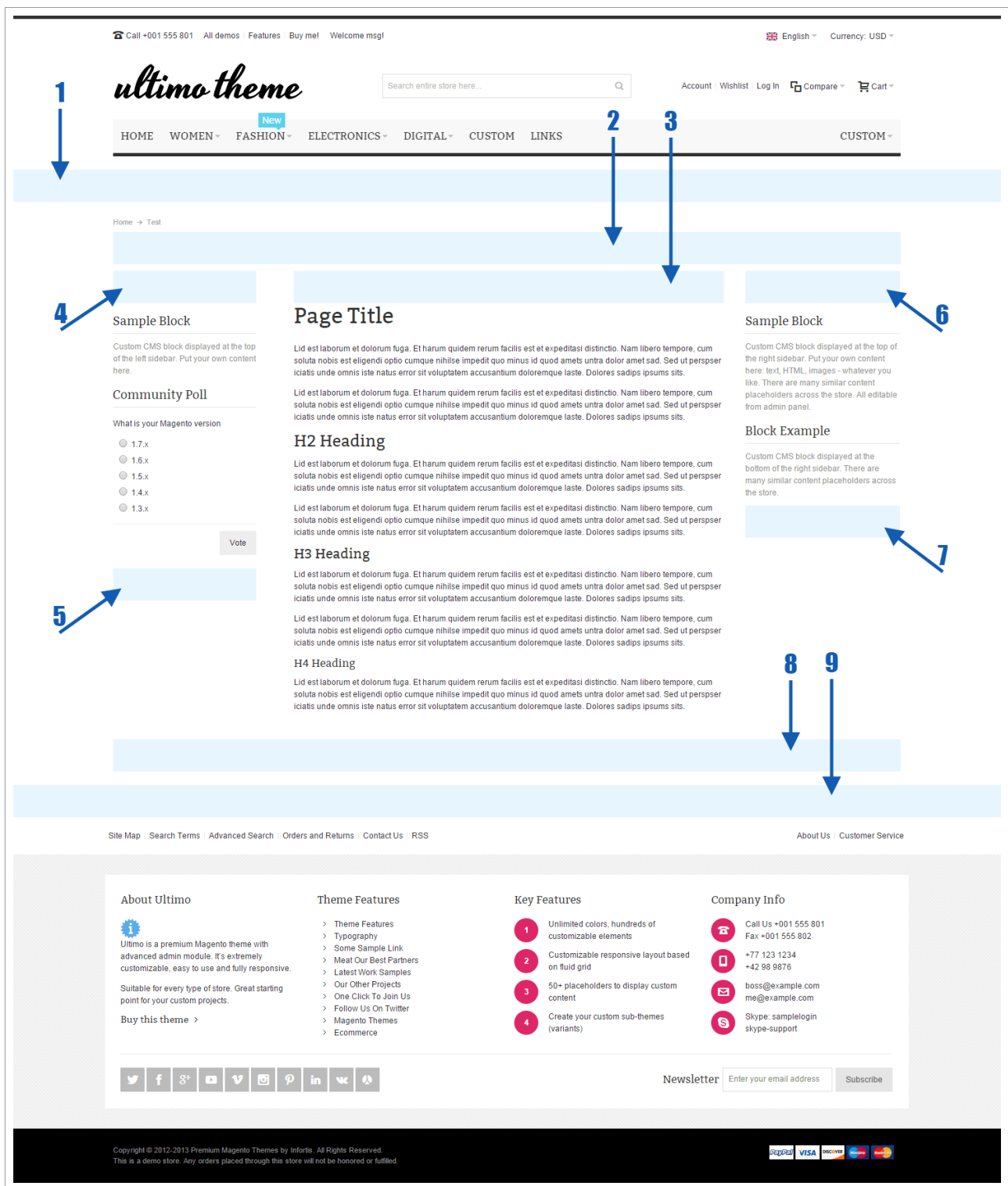
You can add footer background image.

Changes in static blocks

Below you can see identifiers of the static blocks which are deprecated and will be removed in next versions.

Instead of these static blocks **you can now use containers** which are much more flexible. Containers give you ability to **assign custom static blocks to selected areas** of the page (e.g. to the left sidebar), but additionally you can also choose the type of pages on which your blocks will be displayed. See chapter **4.8.1 Add blocks to selected pages** in the [user guide](#) for more details.

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



List and description of all containers can be found in chapter **6.2.1 Page** in the **user guide**.

Deprecated static blocks:

The following static blocks will be removed in next versions. Instead of these predefined static blocks you can now assign your custom static blocks to containers.

- `block_left_top` – block at the top of the left sidebar.
- `block_left_bottom` – block at the bottom of the left sidebar.
- `block_right_top` – block at the top of the right sidebar.
- `block_right_bottom` – block at the bottom of the right sidebar.
- `block_home_left1` – block at the top of the left sidebar on the home page.
- `block_home_left2` – the same as block above.
- `block_home_left3` – the same as block above.
- `block_home_right1` – block at the top of the right sidebar on the home page.
- `block_home_right2` – the same as block above.
- `block_home_right3` – the same as block above.

Removed static blocks:

The following static blocks were removed in this version. Instead of these predefined static blocks you can now assign your custom static blocks to containers.

- `block cms_page_top` – block at the top of any CMS page.
- `block_category_above_collection` – block above the products listing in the category view.
- `block_opc_below_progress` – block below the checkout progress block.
- `block_home_preface` – block at the top of the home page, above sidebars and main content area.
- `block_home_postscript` – block at the bottom of the home page, below sidebars and main content area.

1.4 Ultimo version 1.14.0

In this version we added more options for site header configuration. For example:

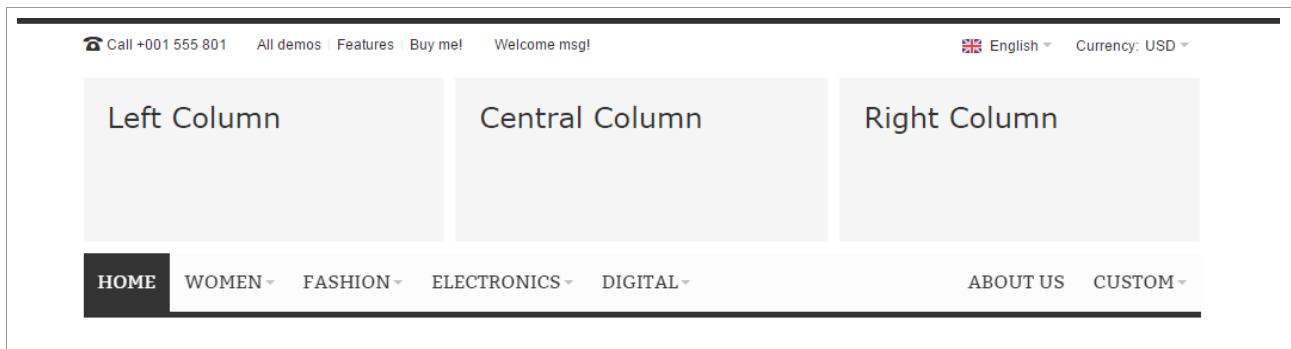
- You can move elements such as search box or Mini Cart to the top of the header. See chapter **7.1 Header** in the [user guide](#) for more details.
- Main menu items can be centered within the menu bar
- Mini Cart and Mini Compare blocks are displayed in the sticky header
- There are also new containers for custom blocks in the header. You can easily assign any number of custom static blocks to each container:



See chapter **6.2.1 Header** in the [user guide](#) for more details about new header containers.

1.5 Ultimo version 1.13.0

In this version we added new options for site header configuration. You have more control over the width of header sections (left column, central column, right column). You can also assign header elements (such as search box, logo, links) to selected sections.



All new options are described in the [user guide](#), chapter **7.1 Header**.

Important:

if you notice any problems after the upgrade, for example, some elements of the header (such as search box) are not displayed as expected (or are not displayed at all), please flush Magento cache and clear your web browser's cache.

You may also need to go to *System > Configuration > Theme Settings > Header* and manually assign the search box to proper section. After that, again flush Magento cache in order to see changes on front-end.

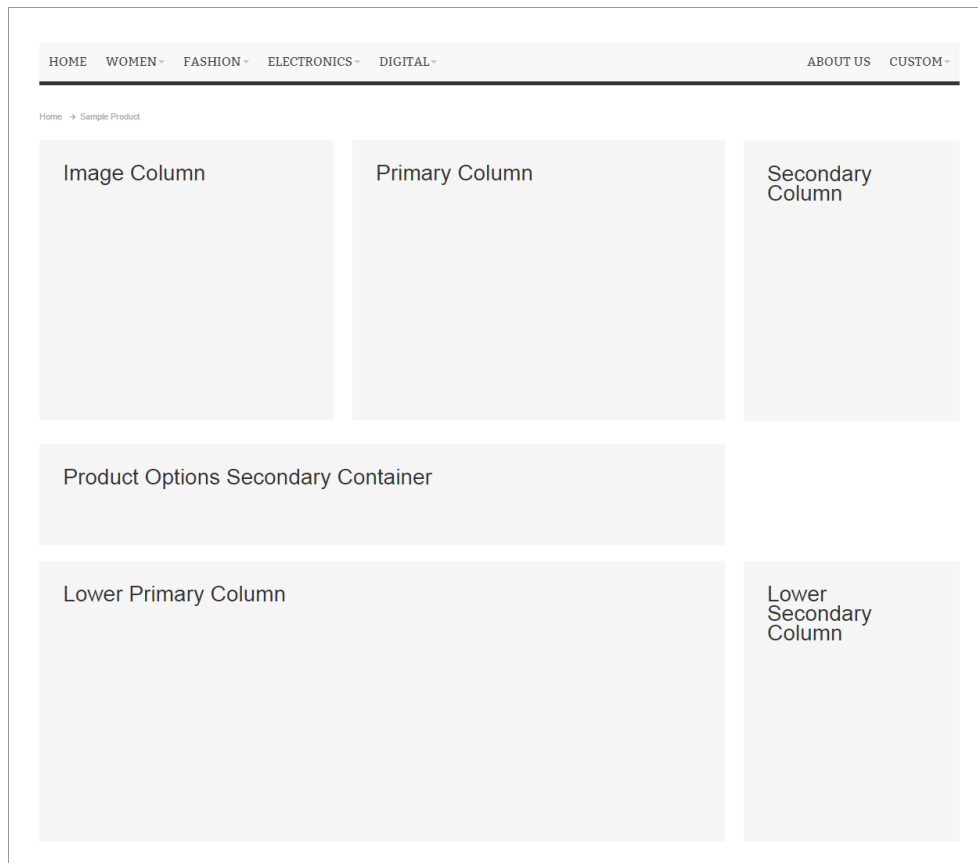
1.6 Ultimo version 1.12.0

In this version we rebuilt and much improved the product page. There are also many other improvements, especially in the layout of the category view.

Product page

There are many new options and the store admin has now much more control over many different elements of the product page. For example:

- position of tabs, related products and up-sell products
- display: product reviews, product tabs, related products, up-sell products in the tabs or outside the tabs
- configuration of tabs/accordion
- width of the product page sections



All new options are described with details in the [user guide](#), chapter **7.5 Product Page**.

There are also new containers for custom blocks on product page. You can easily assign any number of custom static blocks to each container. See chapter **6.2.4 Product page** for more details. Details and example of assigning static blocks to containers can be also found in chapter **4.8.1 Add blocks to selected pages**.

Category view

- Layered navigation is now displayed above the products in mobile view and the filters are collapsed.

- Sidebar blocks are collapsed in mobile view.

Collapsible blocks

- Collapsible blocks are now opened/collapsed by clicking the title bar of the block. Opener icons (small squares with `[+]` sign) were removed.
- Don't use both classes `collapsible` and `mobile-collapsible` in one block.
- The `active` class (if you want the block to be collapsed by default) has to be added to the title bar, not to the main wrapper (like it was in previous versions). Here is an example of the new structure:

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

See more details in the user guide, chapter **6.5.2 Collapsible blocks**.

Other changes

With 2 new parameters, you can now control order and sorting direction of products in the Featured Products slider. For details, refer to chapter **6.1.4 Featured Products Slider** in the user guide.

1.7 Ultimo version 1.11.1

In this version we fixed some minor bugs (e.g. for Magento 1.9.1.0 bug with "Only X Left" table on configurable product page). List of all changes can be found in `Changelog.txt`.

1.8 Ultimo version 1.11.0

Since this version Ultimo is compatible with Magento 1.9.1.0. If you are using older version of Magento, you can ignore this update.

1.9 Ultimo version 1.10.4

In this version we improved rich snippets on product page. You can choose (in *System > Configuration > Theme Settings > Rich Snippets*) if rich snippets will use product prices including or excluding tax. We also added support for bundle products and different price types: price range, "As low as", fixed price. All these price types are now included in rich snippets.

Important:

this file (which was added in version 1.10.2) needs to be removed from your Magento:

```
/app/design/frontend/ultimo/default/template/catalog/product/price.phtml
```

Otherwise rich snippets will not work correctly.

jQuery will be upgraded in next major update of Ultimo. If you would like to upgrade it now, please follow the instructions in the previous chapter: **1.11 Ultimo version 1.10.2**.

1.10 Ultimo version 1.10.3

In this version we added a workaround for the bug in Safari on iPads and iPhones with iOS8 (description of the bug: https://bugs.webkit.org/show_bug.cgi?id=137032). Because of this bug, the product grid in the category view is sometimes displayed incorrectly in Safari – each row (except the last row) should have the same number of products, but sometimes products drop to another row. The workaround for this bug is to replace CSS pseudo-class *:nth-child* with *:nth-of-type* in CSS styles of the product grid.

After you install the new version, re-save theme settings (clicking **Save Config** button) in *System > Configuration > Ultimo > Theme Layout*, and then flush the cache.

If you're using Magento Compilation mode, you will also need to re-compile.

1.11 Ultimo version 1.10.2

In version 1.10.2 we added support for rich snippets (the few lines of text that appear under every search result in Google search). Several template files had to be slightly modified.

Breadcrumbs

There were also some minor changes in CSS of the breadcrumbs (to prepare that block for rich snippets). So if you will notice (after upgrade) that the breadcrumbs on your site have double arrows, it means that the CSS of the breadcrumbs was not upgraded properly. What can cause this:

1. Not all new CSS files were uploaded or old files were not overwritten:
 - `/skin/frontend/ultimo/default/css/override-theme.css`
 - `/skin/frontend/ultimo/default/css/styles.css`
2. Or you have some custom CSS styles (e.g. inside `custom.css`) which overrides the new CSS of the breadcrumbs.

Upgrade of jQuery in next version

We also want to announce that in next release of Ultimo we will upgrade jQuery library to version 1.11.1.

It was already tested and works correctly with Ultimo, but we just want to give more time to users which have third-party extensions installed in their Magento – many extensions still relies on older versions of jQuery so to avoid problems with compatibility we want to ask you to check and test your extension with newer jQuery before the next release of Ultimo.

In case some extensions are not compatible with new jQuery, you will be able to leave the current version of jQuery in Ultimo, it will still work without problems.

If you would like to upgrade it (or just test it) now, here's the location of jQuery in Ultimo:
`js/infortis/jquery/jquery-1.7.2.min.js`

To upgrade jQuery:

- remove that file

- then download new version from <http://code.jquery.com/jquery-1.11.1.min.js>
- upload it to the folder `js/infortis/jquery/`
- remove the existing file `jquery-1.7.2.min.js`
- and finally rename uploaded file `jquery-1.11.1.min.js` to `jquery-1.7.2.min.js`

Thanks to renaming the file, you will not need to make any additional changes in Magento layout XML files which load the jQuery to the frontend.

1.12 Ultimo version 1.10.0

In version 1.10.0 there are improvements of the page header displayed on mobile devices with narrow screens. We also added font icons and made some changes in CSS of different elements to make them work correctly with new icons. All the details about new features and available options can be found in the **user guide**.

Header

Mobile header

If mobile header is enabled, the standard header is transformed into a mobile-friendly header on small mobile devices (if browser viewport is narrower than 770 pixels). Thanks to that the header takes less space than in previous versions and the layout is cleaner.

New option can be found in *System > Configuration > Theme Settings > Header*.

If mobile header is enabled, option *System > Configuration > Menu > Main Menu Mode* has to be set to **Drop-down/Mobile** or **Mobile**.

Sticky menu

If sticky menu is enabled, the main menu does not disappear when the user scrolls down the page. This option can be found in *System > Configuration > Theme Settings > Header*.

Font icons

Old icons were replaced with much better and more powerful font icons. Font icons give you more control over the visual appearance of the icons. Now changing color or size of the icon is as easy as adding CSS class or applying inline styles.

You will need to rebuild the markup of your icons to make them look correctly and work correctly with all the new options. Below you can find some examples of the old markup and the changes that have to be made.

Note:

all the details about the new markup can be found in the user guide.

"Image sprite" icons

Many new icons were added in this version, old icons created with image sprites were replaced with font icons. Full list of available icons can be found in the user guide. Replace old classes with new classes in your icon containers to display new font icons correctly. Just like in the example below:

Old:

```
<span class="icon i-letter"></span>
```

New:

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

There are also many new options to control the look of the icons (e.g. **size**, **color**, **background color**, **position** etc.). For example, to increase the letter icon from the previous example, all you need to do is to add a class to the icon element, for example `ic-lg`, `ic-2x`, `ic-3x` etc.

```
<span class="ib ic ic-letter ic-lg"></span>
```

No need to edit image files anymore to increase the size or change the color of the icon. All can be done with predefined classes or with simple CSS.

Social icons

To add icons of social networking services you need to use exactly the same markup as for other font icons.

Old:

```
<a href="http://twitter.com/infortis">
  <span class="icon icon-hover i-twitter"></span>
</a>
```

New:

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

Icons with custom images

Old:

```
<span class="icon">
  
</span>
```

New:

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

If you have used class `large` to increase the size of the icon, this class has been changed too. Now there are more classes to control the size - refer to the user guide for more details, chapter **6.5.3.B Icons with background color (iconbox)**. Here's an example:

Old:

```
<span class="icon large">
  
</span>
```

New:

```
<span class="ic ic-img ib ib-size-l">
  
</span>
```

Text with icon

A block of text (paragraphs, links, headings etc.) with additional icon.

Here's an example with custom images as icons:

Old:

```
<div class="feature feature-icon-hover indent large">
  <span class="icon large">
    
  </span>
  <h3>Block Heading Example</h3>
  <p>This is a paragraph of sample text.</p>
  <a href="#">Read more...</a>
</div>
```

New:

```
<div class="feature feature-icon-hover indent indent-size-l">
  <span class="ic ic-img ib ib-size-l">
    
  </span>
  <h6 class="above-heading">The Secondary Heading</h6>
  <h3>Block Heading Example</h3>
  <p>This is a paragraph of sample text.</p>
  <a href="#">Read more...</a>
</div>
```

Note that in the main container of the block, class `large` was replaced with class `indent-size-l`. And in the icon, class `large` was replaced with class `ib-size-l`.

Here's an example with predefined icons:

Old:

```
<div class="feature feature-icon-hover">
  <span class="icon i-telephone-w"></span>
  <h3>Block Heading Example</h3>
  <p>This is a paragraph of sample text.</p>
  <a href="#">Read more...</a>
</div>
```

New:

```
<div class="feature feature-icon-hover">
  <span class="left ic ic-phone ib"></span>
  <h3>Block Heading Example</h3>
  <p>This is a paragraph of sample text.</p>
  <a href="#">Read more...</a>
</div>
```

Note that if the div with class `feature` doesn't have class `indent`, you will need to add class `left` to the icon in order to place the icon along the left side of its container.

Characters as icon

Letters, numbers and characters as icons.

Old:

```
<span class="icon i-char">2</span>
```

New:

```
<span class="ib ic ic-char">2</span>
```

1.13 Ultimo versions 1.9.0 - 1.9.4

In version 1.9.x there are many improvements and changes in the main menu. The most important changes are described below. All the details about new features and available options can be found in the **user guide**.

Make a **backup** of category blocks before upgrade

Important:

before you upgrade Ultimo to version **1.9.0** (or to later version), you need to

make a copy of the custom content of all the category blocks in your categories.

You can copy it to a simple text file. It will be needed to recreate your drop-down boxes in the new version of the menu. Go to *Catalog > Category Management*, select a category, open the *Menu* tab and make a copy of the content which you have in these fields:

Block Right, Block Top and Block Bottom:

The screenshot shows the Magento Category Management interface. On the left, there's a sidebar with 'Categories' and 'Add Root Category' and 'Add Subcategory' buttons. Below that, a 'Choose Store View' dropdown is set to 'All Store Views'. A list of categories is shown: Default Category (0), Women (12), Fashion (28), Electronics (17), and Digital (5). The main content area is titled 'Fashion (ID: 3)' and has buttons for 'Reset', 'Delete Category', and 'Save Category'. The 'Menu' tab is selected, showing a 'Menu' section with four blocks: 'Block Right', 'Proportions: Subcategories / Block Right', 'Block Top', and 'Block Bottom'. Each block has a 'WYSIWYG Editor' and a '[STORE VIEW]' button. Red arrows point to the 'Block Right', 'Block Top', and 'Block Bottom' sections, indicating where to copy the custom content.

Block Right content: `{{block type="ultimo/product_list_featured" template="catalog/productlist_featured_slider.phtml" category_id="85" product_count="4" breakpoints="[0, 2]" timeout="4000" move="1" centered="1" size="size-s" hide_button="1" img_width="110" block_name="Recommended Bags"}}`

Block Top content: `Evening Tops`

Block Bottom content: `<div class="show-separators clearer" style="background-color: #eee; padding: 10px 15px;>
<ul class="links">
<li class="label">Featured Categories:
<li class="first">
Evening Tops

</div>`

To avoid issues with menu upgrade

To avoid issues with the upgrade of the menu module, strictly follow all the instructions provided in chapter **2. Upgrade**. Especially remember to do the following:

- Before you upload theme files, make sure your store is disabled: put Magento in the **maintenance mode**.

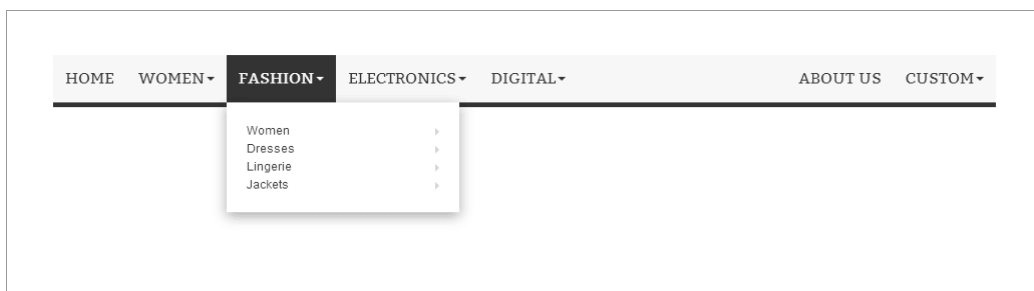
- Files can sometimes get broken if you sent loose files (without zipping) via FTP. So it is highly recommended to upload zipped files: extract the `Theme.zip` archive directly on your server. This will eliminate potential issues with missing/broken files.
- You can see all sorts of errors if Magento doesn't have access to all uploaded files (e.g. because of incorrect file permissions on your server). You need to set correct file permissions for all the theme files (the same as for other Magento files). Permissions may differ depending on server. In case of any doubts, please contact your hosting provider. Refer to this article for more details:
http://www.magentocommerce.com/wiki/1_-_installation_and_configuration/magento_filesystem_permissions
- Files inside this directory are crucial to make a correct upgrade of the menu module:
`app/code/local/Infortis/UltraMegamenu/data/ultramegamenu_setup/`
They have to be accessible for Magento during the upgrade process.
- Reindex Magento and flush the cache after upgrade.

Menu re-configuration

Set drop-down type

In this version, options “*Enable Classic Style*” and “*Enable Wide Style*” were removed. Now the style (type) of the drop-down box can be selected individually for each category.

After upgrading, you will notice that all the drop-downs were reset to “classic” type, and they look like on this screenshot:



You will need to re-configure your menu to set the same look as you had in previous version. To do this, go to *Catalog > Category Management*, select a category, open the *Menu* tab and in the **Submenu Type** field select the type of the drop-down:

- If in the previous version you have used “*Wide style*”, select **Mega drop-down**

- If in the previous version you have used “*Classic style*”, select **Classic drop-down** (or leave it empty because the classic type is the default type)

You need to do this for each top-level category which has a drop-down box (meaning: for each top-level category which has any active subcategories).

Important:

after making these changes you may need to reindex Magento and flush the cache.

Restore content of your category blocks

In the same **Menu** tab, you can restore the content of your category blocks. If you made a copy of the category blocks before the upgrade, you can now insert it back to the category blocks of the new menu.

You will notice that there is an additional block named **Left Block**. Content inserted into that block will be displayed at the left side of the subcategories in the mega drop-down box.

The screenshot shows the Magento Admin interface for editing a category named "Fashion (ID: 3)". The "Menu" tab is selected, showing the following settings:

- Submenu Type:** Mega drop-down
- Drop-down Width:** 2
- Drop-down Content Proportions:** Left Block: 2, Subcategories: 8, Right Block: 2
- Number of Columns With Subcategories:** 4
- Category Blocks:**
 - Top Block:** Sample content of the block. [WYSIWYG Editor]
 - Left Block:** Sample content of the block. [WYSIWYG Editor]
 - Right Block:** Sample content of the block. [WYSIWYG Editor]
 - Bottom Block:** Sample content of the block. [WYSIWYG Editor]
- Category Label:** [Text field]
- Custom URL:** [Text field]

On the left sidebar, the "Categories" section is expanded, showing a tree structure of categories including "Fashion (28)", "Women (12)", "Jackets (8)", "Shirts & Blouses (0)", "Accessories (0)", "Tunics (0)", "Dresses (0)", "Lingerie (0)", "Jackets (0)", "Swimwear (0)", "Electronics (17)", and "Digital (5)".

Drop-down box configuration

Here's a short summary of new options related to drop-down boxes:

- **Drop-down style**

Options "Enable Classic Style" and "Enable Wide Style" were removed. Now the style (type) of the drop-down can be selected individually for each category.

- **Multiple styles in one menu**

Additionally, all of the drop-down types can be mixed together. For example, a simple classic drop-down can be nested inside a mega drop-down.

- **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.

- **Drop-down width**

Width of the drop-down box can be changed individually for each category.

- **Custom blocks in the mobile menu**

Category blocks with custom content can be now displayed also in the mobile version of the menu.

Custom drop-down boxes and custom links in the menu

In the new version you can easily add multiple custom drop-down boxes to the menu. For more details refer to the user guide, chapter **13.4 Custom drop-down boxes in the menu**.

There's also a new better method to add custom links to the menu. For more details refer to the user guide, chapter **13.3 Custom links in the menu**.

Old method for custom links in the menu

If you're using static block `block_nav_links` to display custom links in the menu, the content of that block needs to be slightly modified – a new class was added to the `` tags. To display links properly, that class is required. Details can be found in the user guide (**13.3 Custom links in the menu**).

1.14 Ultimo version 1.8.0

In this version we added ability to import color presets (and other theme configuration settings) from the demo. All the details about the new feature can be found in the user guide.

1.15 Ultimo version 1.7.1

Since this version Ultimo is compatible with Magento 1.9.0.0 and 1.9.0.1. If you are using older version of Magento, you can ignore this update.

1.16 Ultimo version 1.7.0

In this version there were many improvements in the Brands module. The most important changes are described below.

Important changes in Brands module

- **Set custom separators for brand names.** In *System > Configuration > Brands > General > SEO Settings* section you can select characters which will be used:
 - as a word separator in the names of brand image files,
 - and in links to custom brand pages.

There are two fields for this: **URL Key Separator** and **Image URL Key Separator**. By default both fields define the same character for separators (hyphen: -). It is recommended to use the same separator for both cases.

So why there are two fields for this?

The **Image URL Key Separator** field was added only for backward compatibility. In previous versions underscore (_) was the default separator for images. So if this is not a fresh installation of the theme, but an upgrade, you can use this field to set the old separator. Thanks to that you will not need to make any changes in names of your brand images.

In next versions the **Image URL Key Separator** 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. 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`

Note:

hyphen (-) is the recommended separator, it can give better results in SEO. Other separators, such as underscores (_) or plus-signs (+) should be avoided. Source: <https://www.youtube.com/watch?v=AQcSFsQyct8>

- **Added support for language specific symbols and characters** (e.g. letters with diacritical marks) in brand names. These symbols will be properly replaced with letters from latin alphabet to create correct path to a brand image file.

If your names of your brands contain such characters, you may need to fix the names of your brand image files. The list of rules which you need to follow in order to convert a brand name to a correct file name is provided in the user guide, chapter **12. Brands**

- Since this version, **module will check if brand image file exists**. If file doesn't exist, there will be no info about brand displayed on frontend.

Optionally, if brand image doesn't exist, it can be replaced with brand name (simple text). This option is useful for stores which can't display logos of all brands (e.g, because of copyright restrictions).

- More info about all available options related to brands can be found in the user guide.

New parameter in brand slider

There is a new optional parameter in brand slider:

- **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 *System > Configuration > Brands > Brand List* section.

Here's an example of the slider block code with new parameter:

```
{{block type="brands/brands" template="infortis/brands/brand_slider.phtml"
breakpoints="[0, 1], [320, 2], [480, 2], [768, 3], [960, 4], [1280, 5]" block_name="Our Brands"
brands="Apple, Samsung, Louis Vuitton, Coca-Cola"}}
```

1.17 Ultimo version 1.6.0

Many important changes were made in this version, description can be found below.

Flexslider is no longer supported

jQuery plugin Flexslider was replaced with Owl Carousel plugin. Resources (JavaScript and CSS files) related to Flexslider are not loaded by the theme anymore but all files are still on their place. If any of your custom code relies on Flexslider, you can still use it.

Note:

it is highly recommended to use the new slider plugin instead of Flexslider to avoid possible conflicts in the future. New plugin is more reliable, lighter, compatible with all browsers, it has better support for touch-screen devices and it lets us create sliders which slide like native apps. It can do everything that Flexslider can do and much more.

To use Flexslider you would need some small customization:

in `app/design/frontend/ultimo/default/layout/local.xml` uncomment all lines which are marked with tag `obsolete: flexslider`. For example, replace:

```
<!-- obsolete: flexslider -->
<!--<action
method="addJs"><script>infortis/jquery/plugins/jquery.flexslider.min.js</script></action-->
```

with:

```
<!-- obsolete: flexslider -->
<action
method="addJs"><script>infortis/jquery/plugins/jquery.flexslider.min.js</script></action>
```

Files with Flexslider

Flexslider plugin was used in all sliders in the theme. In order to replace it with new plugin we had to modify the following files:

```
template/catalog/product/list_featured_slider.phtml
template/catalog/product/new.phtml
template/catalog/product/list/related_multi.phtml
template/catalog/product/list/related_tabbed.phtml
template/catalog/product/list/upsell.phtml
template/infortis/brands/brand_slider.phtml
template/infortis/cloudzoom/product/view/media.phtml
template/infortis/ultraslideshow/slideshow.phtml
skin/frontend/ultimo/default/css/infortis/brands/brands.css
skin/frontend/ultimo/default/css/infortis/cloud-zoom/cloud-zoom.css
skin/frontend/ultimo/default/css/infortis/ultra-slideshow/ultra-slideshow.css
skin/frontend/ultimo/default/css/infortis/_shared/generic-nav.css
skin/frontend/ultimo/default/css/infortis/_shared/itemslider.css
```

If you copied any of these files to your custom sub-theme (for customization), you will need to delete it from your sub-theme and use the new version instead.

Product sliders

Block code

Block code of **Featured Products Slider**, **New Products Slider** and **Brand Slider** was simplified. To use default settings related to the number of visible items (products/brands), you don't need to use additional parameters anymore. Below you can see examples of all sliders:

Featured Products Slider:

```
{{block type="ultimo/product_list_featured"
template="catalog/product/list_featured_slider.phtml" category_id="3" product_count="12"
hide_button="1" block_name="My Products"}}
```

New Products Slider:

```
{{block type="catalog/product_new" template="catalog/product/new.phtml" products_count="10"
hide_button="1" block_name="My New Products"}}
```

Brand Slider:

```
{{block type="brands/brands" template="infortis/brands/brand_slider.phtml" block_name="Our
Brands"}}
```

Below you can find more details about block parameters.

Parameters

There were some changes in the parameters of product sliders:

- **is_responsive** parameter is only needed if you want to disable responsive behavior of the slider:

```
is_responsive="0"
```

So if you want your slider to be responsive, you can omit this parameter.

- **show_items** parameter is only needed if responsive behavior of the slider was disabled. In that case the number of visible items should be specified with **show_items** parameter.
- **breakpoints** parameter is now the only way to control the number of products in a row visible with a particular browser viewport width. This is the default value of this parameter:

```
breakpoints="[0, 1], [320, 2], [480, 3], [768, 4], [960, 5], [1280, 6]"
```

If you want to use the default value, you can simply omit this parameter in your block code.

- **init_delay** parameter is no longer supported.
- many new parameter were added so that now each slider can be configured

individually. Global settings are still available, but now they can be overridden per slider. Full list of all available parameters can be found in the **User Guide** (chapter 6. *Theme features and elements*).

Image slideshow

Slideshow module was improved, now you can display custom slideshow on any CMS page and in any static block. To display a slideshow in a selected place you just need to use the slideshow block code (the same way as you do to display product sliders):

```
{{block type="ultraslideshow/slideshow" template="infortis/ultraslideshow/slideshow.phtml"
slides="block_slide1, block_slide2"}}
```

Slideshow on home page

To display a slideshow on home page you still can use the same admin settings as it was in previous versions. You don't need to use the block code for this (block code is only an option, a way to display slideshows in other places). Just go to *System > Configuration > Slideshow* and configure options in *Main Slideshow on Home Page* sections. These sections are related only to the main slideshow on home page and settings from these sections will not be applied to any other slideshow which was added via block code. More details can be found in the **User Guide** (chapter 11. *Slideshow*).

Full width slideshow

The main home page slideshow can be configured to take full available width of the page. List of all available settings can be found in the **User Guide**.

1.18 Ultimo version 1.5.10

Since version 1.5.10 Ultimo is compatible with Magento 1.8.1.0. Follow the entire upgrade procedure described in **2. Upgrade**. If you are not using Magento 1.8.1.0, you can ignore this update.

Important:

If you want to upgrade your Magento to 1.8.1.0, you need to use Ultimo 1.5.10 (or newer). Previous versions of Ultimo are not fully compatible with

Magento 1.8.1.0. Magento team made many changes in version 1.8.1.0, so you need to install proper version of the theme (**don't forget to install theme patch for Magento 1.8.1.0 which adds files required in Magento 1.8.1.0**). Otherwise you may encounter many issues, e.g. your customers will not be able to log in.

Directories and files modified in version 1.5.10 to make the theme compatible with Magento 1.8.1.0:

```
template/bundle/catalog/product/view/type/bundle.phtml
template/checkout/cart/item/default.phtml
template/checkout/onepage/review/item.phtml
template/checkout/cart.phtml
template/customer/form/login.phtml
template/persistent/checkout/onepage/login.phtml
template/persistent/customer/form/login.phtml
template/page/1column.phtml
template/page/2columns-left.phtml
template/page/2columns-right.phtml
template/page/3columns.phtml
app/code/local/Infortis/Ultimo/
```

1.19 Ultimo version 1.5.6

Since version 1.5.6 Ultimo is compatible with **Magento 1.8.0.0**. If you are not using Magento 1.8.0.0, you can ignore this update.

If you're planning to upgrade your Magento to version 1.8.0.0 (or you are already using it):

- if you are using Ultimo **older** than 1.5.5:
follow the entire upgrade procedure described in **2. Upgrade**
- if you are using Ultimo 1.5.5:
follow the entire upgrade procedure described in **2. Upgrade** but you can omit steps **10.** and **11.** The only files that you need to install are stored in Theme (patch for Magento 1.8.0.0).

List of changes

A new directory is now available inside the package: `Theme (patch for Magento 1.8.0.0)`. If you are using Magento 1.8.0.0, you need to upload files from this directory to Magento's root directory. But do it **after** installation of the files from the main zip package `Theme.zip` (which contains theme files for all supported versions of Magento) – simply follow the instructions in **2. Upgrade**.

Here is the list of changes that were made in order to make the theme compatible with Magento 1.8.0.0:

- Line `echo $this->getBlockHtml('formkey')` was added in:
 - `template/catalog/product/view.phtml`
- Conditional clause: `if ($this->displayProductStockStatus())` was added in:
 - `template/bundle/catalog/product/view/type/bundle.phtml`
 - `template/catalog/product/view/type/default.phtml`
 - `template/catalog/product/view/type/grouped.phtml`

Listed files can be found in `Theme (patch for Magento 1.8.0.0)` directory.

1.20 Ultimo version 1.5.3

Slideshow on home page

Layout (HTML markup) of the image slideshow on home page has been improved and the default size of slides has changed a little bit. You will need to change your images if you are displaying slideshow with additional banners at the side. See the image size table in the user guide for details.

Sidebar menu

There are new settings available for the sidebar menu. Refer to the user guide to check the list of new options. Sidebar menu block code has also changed a little bit:

```
{{block type="ultramegamenu/navigation" parent="root" depth="2" block_name="All Categories in our store" template="infortis/ultramegamenu/categories.phtml"}}
```

Update the code if you're displaying the menu with the code, e.g. inside some static blocks.

1.21 Ultimo version 1.5.0

Before you upgrade to 1.5.0 (or higher)

In order to keep the theme directories clean, remove the files of the previous versions. Remove files from the following directories:

- `app/design/frontend/ultimo/default/layout/`
- `app/design/frontend/ultimo/default/template/`
- `skin/frontend/ultimo/default/`
- `app/code/local/Infortis/Brands`
- `app/code/local/Infortis/CloudZoom`
- `app/code/local/Infortis/Infortis`
- `app/code/local/Infortis/Ultimo`
- `app/code/local/Infortis/UltraMegamenu`
- `app/code/local/Infortis/UltraSlideshow`

Before you start deleting, make sure you don't have any custom files inside those folders.

Simplified backend

In the previous versions you had to click “Refresh CSS...” button every time the configuration was saved in *System > Configuration > Ultimo > Theme Design* and *System > Configuration > Ultimo > Theme Layout* sections. **It is not needed anymore and the button was removed.** All the settings are now refreshed automatically after you click **Save Config** button.

After upgrading to version 1.5.0 you need to refresh the CSS one last time to get rid of old settings. Go to *System > Configuration > Ultimo > Theme Design* and click **Save Config** button. Then, do exactly the same in section *System > Configuration > Ultimo > Theme Layout*.

Product sliders

There were major changes in the sliders. The slider block code (the code in double curly braces: `{{block...}}`) has also changed. For example, there are two new parameters: **show_items** and **is_responsive**. Make sure you update your sliders' block code on your pages. You can read more about sliders in the user guide.

Changes in static blocks

Blocks which ID has changed:

- `block_header_top_help` change ID to `block_header_top_left`
- `block_header_links_upper` change ID to `block_header_top_left2`

Blocks which position has changed:

- `block_header_top_right`: now is displayed **after** language switcher and currency switcher in the header

New blocks:

- `block_header_top_right2` it replaced `block_header_top_right`
- `block_nav_links`: custom links in the main menu

Changes in CMS content

There were changes in the icons markup so you will need to update static blocks and pages which contain icons. Refer to **2.3 Import sample blocks and pages** for more details about updating static blocks. By default icons are used in the following static blocks:

- `block_footer_primary_bottom_left`
- `block_footer_column` (from 1 to 6)
- `block_product_secondary_bottom`
- `block_product_tab1`

New markup of icons is described in the user guide.

Modified CSS

The structure of the CSS files of the theme was reorganized and improved, there were also many changes in the file names. Make sure you review all your CSS modifications, for example, in your custom sub-themes or in `custom.css` file.

1.22 Ultimo version 1.4.0

All menu settings were moved to *System > Configuration > Menu*.

1.23 Ultimo version 1.3.1

- **Bug in the checkout**

Error: when trying to make a purchase you are sent back to basket when clicking from step 1. to step. 2 (if logged in).

This code (which is responsible for adding CSS class to BODY tag, using default Magento functionality) in `app/design/frontend/ultimo/default/layout/local.xml` was causing this error:

```
<customer_logged_in>
  <reference name="root">
    <action method="addBodyClass">
      <classname>is-logged-in</classname>
    </action>
  </reference>
</customer_logged_in>
```

Code was removed.

1.24 Ultimo version 1.3.0

In the previous versions brand images were stored in `media/infortis/attributes/` directory. Please move all your brand images to the new directory:
`media/wysiwyg/infortis/brands/`.

2. Upgrade

Important:

before you upgrade Ultimo to version **1.9.0** (or to later version), you need to make a copy of the custom content of all the category blocks in your categories. Refer to chapter **1.13 Ultimo versions 1.9.0 - 1.9.4** for details.

Important:

before you start upgrading, refer to chapter **1. Important changes** to check if there are any additional instructions for the version which you want to install and for the versions which you have skipped. Do not omit this step!

Note:

in case you have made any changes directly in the theme files, backup all your changes. You can restore those changes after upgrade.

Note that the correct way to customize the theme in Magento is to create a custom sub-theme, you should never modify the original files.

To upgrade the theme you need to follow almost the same steps as when you installed it. To avoid unexpected problems after installation, you should first prepare your Magento. Please strictly follow the steps described in this guide.

2.1 Prepare your Magento

1. **Always test the new version on a test Magento installation** before you upgrade your live store. When the tests are finished and you want to upgrade your live store, always do it in the **maintenance mode** (when your Magento is offline) because no page can be refreshed during the upgrade process. Maintenance mode is described later in this chapter.

2. **Backup Magento** files and the store database.

Important:

creating backup before installation of any themes or extensions in Magento is extremely important, especially when you are working on a live store.

3. **Disable compilation** mode in `System > Tools > Compilation`.

4. **Disable all cache systems** which you have in your Magento:

4.a. Magento cache

4.b. full page cache and caching modules for Magento (such as Speedster)

4.c. any additional cache on your server, PHP cache engines, APC, etc.

Important:

when developing your Magento store, in order to be able to see changes immediately, you should completely disable the cache. You can enable it after you finish configuring your store.

5. Disable **Merge JavaScript Files** and **Merge CSS Files** in `System > Configuration > Developer`. You can enable this feature after update.

6. **Remove all possible custom modifications** of the Magento's "base" theme. Ultimo (same as any Magento theme) relies on "base" theme, so any modifications of the "base" theme can change the default behavior of Ultimo and break some functionality. **You should never edit "base" theme's files.**

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

Important:

do not just close the browser window: you need to click the **Log Out** link to refresh the access control system.

8. Before you install the theme files, make sure your store is disabled during the installation process. Put Magento in the **maintenance mode**. To do this you can create a simple empty file named `maintenance.flag` in your Magento's root directory. Magento will be offline after that (and you will not be able to log in to the admin panel). You can remove the file after you finish.

2.2 Upload files

Before you install/upgrade the theme **you need to prepare Magento**, so please do not omit the previous step: **2.1 Prepare your Magento**.

9. Extract the theme package on your computer and navigate inside `Theme Files` directory.
10. Using your FTP client upload `Theme.zip` file to your Magento's root directory (Magento's root directory is where you can find Magento's basic files, such as: `index.php` and `get.php`).
11. Unzip `Theme.zip` – it contains four directories: `app`, `js`, `media`, `skin`. Magento already contains the same directories so unzipping in Magento's root directory will merge theme's and Magento's directories.

Unzip `Theme.zip` file – it contains four directories: `app`, `js`, `media`, `skin`. 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.

Important:

extracted files have to **overwrite** files from the previous version which already exist on the server. So if you'll see a system message, confirm that you want to overwrite existing files.

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

Alternatively, you can unzip `Theme.zip` file on your computer, and upload its content (four directories: `app`, `js`, `media`, `skin`) to your Magento's root directory. Note that files can be broken if you send them in large numbers via FTP without zipping. So it is highly recommended to upload zipped files.

12. **Install theme patches for specific versions of Magento**. Note that in this case patch doesn't fix anything, it only adds theme files which are required in specific versions of Magento. Please note that no patch is needed for *Magento 1.7.x*.

12.a. **NEW:** *Omit this step if you are not using Magento 1.9.2.0 or newer*

This step is required only if you are using Magento **1.9.2.0** (or newer). In this step you will install files which are required only in Magento 1.9.2.0 (or newer). Navigate inside

Theme (patch for Magento 1.9.2.x) directory. Using your FTP client upload content of Theme (patch for Magento 1.9.2.x) directory to Magento's root directory. Overwrite existing files.

12.b. Omit this step if you are not using Magento 1.9.1.0 or 1.9.1.1

This step is required only if you are using Magento **1.9.1.0** or **1.9.1.1**. In this step you will install files which are required only in Magento 1.9.1.0. Navigate inside Theme (patch for Magento 1.9.1.0, 1.9.1.1) directory. Using your FTP client upload content of Theme (patch for Magento 1.9.1.0, 1.9.1.1) directory to Magento's root directory. Overwrite existing files.

12.c. Omit this step if you are not using Magento 1.8.1.0, 1.9.0.0 or 1.9.0.1

This step is required only if you are using Magento **1.8.1.0**, **1.9.0.0** or **1.9.0.1**. In this step you will install files which are required only in those versions of Magento. Navigate inside Theme (patch for Magento 1.8.1.0, 1.9.0.0, 1.9.0.1) directory. Using your FTP client upload content of Theme (patch for Magento 1.8.1.0, 1.9.0.0, 1.9.0.1) directory to Magento's root directory. Overwrite existing files.

12.d. Omit this step if you are not using Magento 1.8.0.0

This step is required only if you are using Magento **1.8.0.0**. In this step you will install files which are required only in Magento 1.8.0.0. Navigate inside Theme (patch for Magento 1.8.0.0) directory. Using your FTP client upload content of Theme (patch for Magento 1.8.0.0) directory to Magento's root directory. Overwrite existing files.

12.e. Omit this step if you are not using Magento 1.6.x

This step is required only if you are using Magento **1.6.x**. In this step you will install files which are required only in Magento 1.6.x. Navigate inside Theme (patch for Magento 1.6.x) directory. Using your FTP client upload content of Theme (patch for Magento 1.6.x) directory to Magento's root directory. Overwrite existing files.

12.f. Omit this step if you are not using Magento 1.5.x

This step is required only if you are using Magento **1.5.x**. In this step you will install files which are required only in Magento 1.5.x. Navigate inside Theme (patch for Magento 1.5.x) directory. Using your FTP client upload content of Theme (patch for Magento 1.5.x) directory to your Magento root directory. Overwrite existing files.

- 13. Set correct file permissions** for all the theme files which you uploaded to your server.

Note:

theme files should have the same permissions as other Magento files. Otherwise Magento will not have access to the theme files, images etc. File permissions may differ depending on server. In case of any doubts about the permissions, please contact your hosting provider. Refer to this article for more details: [**Magento Filesystem Permissions**](#)

14. 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 the root directory of Magento. After that, you will be able to log in to the admin panel again.

Note:

if you get **404 not found** message when trying to open theme configuration pages in the admin panel after installation, you need to **log out** and log in again to refresh the access control system.

15. **Reindex Magento** and **flush Magento cache** (and any other additional cache that you have in your Magento installation).
16. **Now you need to refresh the CSS files** which are responsible for the visual appearance of the theme (the old outdated CSS generated automatically by the theme may break the layout after you upload new version of the theme). Go to `System > Configuration > Ultimo > Theme Design` and click **Save Config** button. Then go to `System > Configuration > Ultimo > Theme Layout` and click **Save Config** button again.

2.3 Import sample blocks and pages

After the theme was updated, you can import sample static blocks and pages provided with the theme (in some updates we add new blocks and pages or we change their default content).

17. Go to `System > Configuration > Ultimo > Theme Settings > Data Import` to import static blocks and pages. Click the **Import Static Blocks** and **Import Pages** buttons – this will only import new blocks and pages which were added in the new version of the theme. Options **Overwrite Existing Blocks** and **Overwrite Existing Pages** should be set to **No**.

Set **Overwrite Existing Blocks** and **Overwrite Existing Pages** to **Yes** if you want

to overwrite existing blocks and pages with new blocks and pages.

Note:

in case you don't want to overwrite your current content but you want to check what changes were made in the selected blocks or pages, the XML file with the content of all blocks and pages can be found in the following directory: `app/code/local/Infortis/Ultimo/etc/import/`

Now you can open up your web browser. Clear your web browser's cache and cookies for your store domain (this will eliminate potential problems with unrefreshed cache).

Important:

it is a good idea to **disable Magento cache** before you start to configure your store. Otherwise you will need to flush the cache in order to see any changes you have made in the config.

2.4 Problems after upgrade

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

- a) Magento cache (or external cache) was not flushed after installation of the theme.
- b) You can see all sorts of error messages if the **Compilation** mode was not disabled before installation.
- c) CSS files which are responsible for the visual appearance of the theme were not generated automatically. Go to `System > Configuration > Ultimo > Theme Design` and click **Save Config** button. Then go to `System > Configuration > Ultimo > Theme Layout` and click **Save Config** button again. Then, flush Magento cache and refresh your browser's cache.
- d) 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.

- e) File permissions issue – refer to the user guide for more details, chapter **“Troubleshooting”**.
- f) Theme wasn't enabled properly. Theme should only be enabled in *System > Configuration > Design* (the same as it was described in the installation steps). Refer to the user guide for more details.
- g) 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 the user guide for more details.
- h) 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.
- i) You have customized the theme by creating custom sub-theme and your modifications override or break the default behavior of the theme.
- j) You have some elements that left from your previous theme (e.g. in Magento's "base" theme in `app/design/frontend/base/`) and that breaks the new theme.

Also check your log files in `var/log/` (log settings should be enabled in *System > Config > Developer*) for more information about possible errors.

Please refer to chapter **“Troubleshooting”** in the user guide for more information about the most common issues.