How to create a new package / theme

Last modified by **farkie** on Tue, September 11, 2012 10:43

[Source](http://www.magentocommerce.com/wiki/4_-_themes_and_template_customization/0_-_theming_in_magento/package?do=edit&rev=)|[Old Revisions](http://www.magentocommerce.com/wiki/4_-_themes_and_template_customization/0_-_theming_in_magento/package?do=revisions)

**TABLE OF CONTENTS**

* [How to create a new package / theme](http://www.magentocommerce.com/wiki/4_-_themes_and_template_customization/0_-_theming_in_magento/package#how_to_create_a_new_package_theme)
  + [References](http://www.magentocommerce.com/wiki/4_-_themes_and_template_customization/0_-_theming_in_magento/package#references)

*This is a work-in-progress. Help making this article great is requested.*

The following was done with Magento v1.3.2.4.

I’ve been reading up on Magento’s packages and themes for a little while now. For the life of me, I can’t get a secondary theme to work. So I created a whole new package.

This procedure will create a minimal, 2-column (left), non-complete, bare-bones package/theme that you can then fully customize. I did it from the command-line in linux, but I suppose you could do it from FTP or SCP. Just make sure you keep the paths correct.

The reason for creating this package from scratch is so that I could get a better understanding of exactly how it works and what files do what. During this walk-through, you can reload your test Magento installation and either see a bare-bones theme, or an error message specifying the missing file. This will help you to see which files are necessary for which part of the system.

|  |
| --- |
| **The walk-through** |

**1.** In your installed magento directory, create a directory structure like this:

app

+ design

+ frontend

+ your\_new\_theme\_name (package)

+ default (theme in your new package)

+ layout

+ template

(Note that the directory structure app/design/frontend already exists. That’s where all themes/packages reside.)

**2.** In layout, copy these files from Magento’s default package:

catalog.xml

catalogsearch.xml

checkout.xml

cms.xml (homepage customization/content will not work without this file)

customer.xml

newsletter.xml

page.xml

wishlist.xml (if you want wishlist capability)

The page.xml file will be the default page layout .xml file used for every page in your site. Other layout .xml files will modify (or “update”) this base layout.

**Example copy command** in linux for this step:

cp ../../../default/default/layout/page.xml .

The above command will only work if your current directory isapp/design/frontend/your\_new\_theme\_name/default/layout.

**3.** In template, make these directories:

callouts

catalog

catalogsearch

checkout

customer

page

reports

review

**4.** In template/callouts, copy this file from Magento’s default package:

left\_col.phtml

**5.** In template/catalog, create these directories:

navigation

category (will need this to view products)

product (will need this to view products)

**6.** In template/catalog/navigation, copy these files from Magento’s default package:

top.phtml

left.phtml (will need this to view products)

**7.** In template/catalog/category, copy this file from Magento’s default package:

view.phtml (will need this to view products)

**8.** In template/catalog/product, copy these files from Magento’s default package:

list.phtml (will need this to view products)

price.phtml (will need this to view products)

new.phtml (will need this to view products on the homepage)

**9.** In template/catalog/product, create this directory:

list (will need this to view products)

**10.** In template/catalog/product/list, copy this file from Magento’s default package:

toolbar.phtml (will need this to view products)

**11.** In template/catalogsearch, copy this file from Magento’s default package:

form.mini.phtml

**12.** In template/checkout, create these directories:

cart

onepage

total

**13.** In template/checkout, copy this file from Magento’s default package:

cart.phtml

**14.** In template/checkout/cart, create this directory:

item

**15.** In template/checkout/cart, copy these files from Magento’s default package:

crosssell.phtml

coupon.phtml

shipping.phtml

totals.phtml

noItems.phtml

**16.** In template/checkout/cart/item, copy this file from Magento’s default package:

default.phtml

**17.** In template/checkout/onepage, copy this file from Magento’s default package:

link.phtml

**18.** In template/checkout/total, copy this file from Magento’s default package:

default.phtml

**19.** In template/customer, copy the entire directory structure from Magento’s default package:

cp -a ../../../../default/default/template/customer/\* .

This will copy the following files/directories from /template/customer:

.:

account/ address/ address.phtml balance.phtml dashboard.phtml form/

logout.phtml order/ orders.phtml widget/ wishlist.phtml

./account:

dashboard/ dashboard.phtml link/ navigation.phtml

./account/dashboard:

address.phtml hello.phtml info.phtml newsletter.phtml sidebar.phtml

./account/link:

back.phtml

./address:

book.phtml edit.phtml

./form:

address.phtml confirmation.phtml forgotpassword.phtml mini.login.phtml

newsletter.phtml changepassword.phtml edit.phtml

login.phtml mini.newsletter.phtml register.phtml

./order:

view.phtml

./widget:

dob.phtml name.phtml taxvat.phtml

**20.** In template/page, copy these files from Magento’s default package:

1column.phtml (used for the login screen and only a few other places)

2columns-left.phtml (used for virtually every page in your store)

**21.** In template/page, create these directories:

html

switch

template

**22.** In template/page/html, copy these files from Magento’s default package:

breadcrumbs.phtml

footer.phtml

header.phtml

head.phtml

notices.phtml

**23.** In template/page/switch, copy these files from Magento’s default package:

languages.phtml

stores.phtml

**24.** In template/page/template, copy this file from Magento’s default package:

links.phtml

**25.** In template/reports, copy this file from Magento’s default package:

home\_product\_viewed.phtml

**26.** In template/review, make this directory:

helper

**27.** In template/review/helper, copy this file from Magento’s default package:

summary\_short.phtml

**28.** In Magento’s admin, set your new package:

**a.** **System** → **Configuration**

**b.** In the upper-left area, set the configuration scope,

**c.** Click on **Design**, in the **Package** section enter your\_new\_theme\_name in **Current package name**.

Now go to your storefront and reload. You should be looking at the default Magento theme without the images. In this walk-through, none of the images were copied over (intentionally) so that you can start with a nearly-blank theme and build it up to what you want from there.

|  |
| --- |
| **Disabling Specific Features** |

For our site, we don’t have any coupons, so I turned off the coupons feature by editing layout/checkout.xml and commenting-out this line:

<block type="checkout/cart\_coupon" name="checkout.cart.coupon" as="coupon" template="checkout/cart/coupon.phtml"/>

It’s not necessary to comment-out the line in template/checkout/cart.phtml which calls ‘coupon’ (but you could if you wanted to):

<?php echo $this->getChildHtml('coupon') ?>

We also don’t need or want the “crosssell” section, so while I was in layout/checkout.xml I commented-out that line as well.

|  |
| --- |
| **How to turn on 'Template Path Hints'** |

When viewing your theme, it’s helpful to see which template file is being used for which part of the page. Magento has an excellent debugging tool called ‘Template Path Hints’.

**1.** **Admin** → **System** → **Configuration**.

**2.** Select your store from the drop-down in the upper-left corner and wait for the page to reload. Note that you have to be on the website level or lower. The ‘Template Path Hints’ option will not be visible if you are at a higher level.

**3.** **Advanced** → **Developer** (all the way at the bottom).

**4.** **Template Path Hints** → **Yes**.

**5.** Hit the orange **Save Config** button.

Go to your store and reload. You should see path specs in dashed boxes around all sections of the page. These paths tell you which template .phtml file is responsible for that section.

|  |
| --- |
| **How to turn off Magento's cache** |

When developing a theme/package, it’s helpful to not have to wait for your changes to expire the built-in cache. Here’s how to disable Magento’s cache so that you can see your changes right away:

**1.** **Admin** → **System** → **Cache Management**

**2.** In the **Cache Control** section, for the **All Cache** drop-down, select **Disable**.

**3.** Click the orange **Save Cache Settings** button.

|  |
| --- |
| **How to set the header details** |

If you open up/app/design/frontend/your\_new\_theme\_name/default/template/page/html/header.phtml, you’ll see a line like this:

1. **<h1 id="logo"><a href="<?php echo $this->getUrl('') ?>">**
2. **<img src="<?php echo $this->getLogoSrc() ?>" alt="<?php echo $this->getLogoAlt() ?>" />**
3. **</a></h1>**

Where does Magento get the value for getLogoSrc() and getLogoAlt() when building your logo image? This information is not in the template (though it could be). Instead, it’s in the admin section:

**1.** **Admin** → **System** → **Design** → section **Header**.

You could put the exact data directly into your header.phtml file, but then you’d have to edit the file by hand every time there were a change to the image. Having the data in the admin section gives you an easy-to-use web-interface for modifying the data.

Also in this header.phtml file is this code:

1. **<?php echo $this->getWelcome() ?>**

This “**Welcome Text**” is also defined in the **Header** section.

|  |  |
| --- | --- |
| **References** |  |

[Designer's Guide to Magento](http://www.magentocommerce.com/design_guide)

[Magento Design Terminologies](http://www.magentocommerce.com/design_guide/articles/magento-design-terminologies4)

[Working with Magento Themes](http://www.magentocommerce.com/design_guide/articles/working-with-magento-themes)

[Intro to Layouts](http://www.magentocommerce.com/design_guide/articles/intro-to-layouts)

*More to come.*