

BEST PRACTICES

CODING AND MAGENTO RULES

Ngu Nguyen - Tech Director.

BLUECOM SOLUTIONS

The E-Commerce Experts

www.bluecomgroup.com

COPYRIGHT © 2012-2014 BLUECOM GROUP. ALL RIGHTS RESERVED.



AGENDA

- 1. CODING RULES**
- 2. MAGENTO RULES**
- 3. HOW TO USE PHPSTORM AND SUBVERSION**
- 4. THE STEPS IN DEVELOPMENT PROCESS**
- 5. PRACTICES FOR PHP**
- 6. PRACTICES FOR MAGENTO**
- 7. DEBUG EXTENSIONS FOR MAGENTO**

REFERENCE URL

Very useful in best practices in Magento.

1. <http://inchoo.net/tools-frameworks/zend/zend-framework-coding-style-standard/>
2. <http://www.atwix.com/magento/best-practices/>
3. <http://scandiweb.com/files/Scandiweb-developmentguidelines.pdf>
4. <https://packagist.org/packages/magento-ecg/coding-standard>
5. <https://wiki.magento.com/display/MAGE2DOC/Magento+Coding+Standards>
6. <https://github.com/magento-ecg/coding-standard>
7. <https://github.com/cobhimself/phpcs-magento-rules>
8. <http://prattski.com/magento-dev-best-practices/>
9. <http://framework.zend.com/manual/1.12/en/coding-standard.coding-style.html>

1. CODING RULES

1. Don't hardcode anything (email, constant, values)
2. Commit all changes on SVN, don't work on dev server directly
3. Comment your code
4. More code does not mean better code. Keep it simple and reduce complexity.
5. Delete unused code
6. Use understandable and long names for variables

2. MAGENTO RULES

Don't touch the magento core

app/code/local/Mage

app/design/adminhtml/default/default

app/design/frontend/base

app/design/frontend/default/default

lib/Varien

lib/Zend

skin/frontend/base

skin/frontend/default/default

2. MAGENTO RULES

Always create a your package theme for your design

`app/design/frontend/<your_package>/<your_theme>`

`skin/frontend/<your_package>/<your_theme>`

2. MAGENTO RULES

Try to avoid overriding wherever it's possible.

Use event-listeners, helpers, or extend (not override) the core classes to form your own. It also saves your extension from conflicts with third party addons.

```
<block type="your_extension/checkout_onepage_billing"  
name="checkout.onepage.billing" as="billing" template="  
checkout/onepage/billing.phtml"/>
```

Do not remove generic blocks from *.phtml files or xml layouts.

Magento third party extensions usually use generic blocks to inject their own blocks there. If you have deleted it, you may spend a lot of time to find out why some extension doesn't work.

2. MAGENTO RULES

Use Magento generic CSS classes whenever possible.

There are thousands of CSS classes, that are defined in Magento default installation and if you are going to make a new store template, you should use those class names in your new template. It's a good practice, because there is a lot of third party extensions that use those class names for design integration.

2. MAGENTO RULES

Use Magento generic CSS classes whenever possible.

You should use those class names in your new template. It's a good practice, because there is a lot of third party extensions that use those class names for design integration.

2. MAGENTO RULES

Always document your own code.

As a good programmer, you should always use PHPDoc in your projects. If you were faced with problem and trying to find out the solution, it's always easier to explore documented code. Also you should think about other programmers who may be forced to work with your code :)

2. MAGENTO RULES

1. Use Magento generic CSS classes whenever possible.
2. Use cache for your own blocks.
3. Use Magento built in log system for logging the behavior of your extension.
4. Use Magento built in profiler to check the performance issues.
5. Use XML layouts to integrate features or extend functionality wherever it's possible.
6. Use understandable names for CMS pages, CMS blocks and pages URLs
7. For translation, always apply first a translation package and only translate in a customized CSVs project specific text
8. When defining customizable CMS pages, define a specific set of templates (as per needed) and have the admin to apply them to the page they are creating. The page design is related to the applied template
9. Always add custom modules in System -> Configuration -> Bluecom -> MY_EXTENSION
10. Always use clean dummy content for test products and content

4.THE STEPS IN DEVELOPMENT PROCESS

Before do task

1. Reading deep to task question. how many child task.
2. Feelback task if you are understand it, and require code, database to PM
3. Re-estimate time for task if you feel not enough time for this task.

Begin do task

1. Check and research to magento to support task exit.
2. Analytic and todo list to process steps, need Teach Lead to support if you don't clear about Magento.
3. Code text used formatter to read easier.
4. You must implement to Best Practices in Magento document of Bluecom company.
5. Test for this task and related case.
6. Upload(svn commit) code to dev server and comment to subversion with id task.

Finish do task

1. Test in dev server
2. Update task status.

HOW TO USE SUBVERSION

- Using subversion in PhpStorm.
- How to resolve conflict code.
- How to use reversion files.
- How to manager to code changed in local.

2.BASIC TEST CASES IN MAGENTO

- Task about price: you should enable all price types such as regular, tire, discount, etc
- Login and Logout
-

Popular practices of PHP applications debugging.

1. Error output
2. Variable values
3. Structured data
4. Code tracing
5. Objects analysis
6. Database query

Popular practices of PHP applications debugging.

Error output

- Structured data
- Code tracing
- Objects analysis
- Database query

Error output

Directly in the application:

```
ini_set('display_errors', 'On');  
error_reporting(E_ALL | E_STRICT);
```

In php.ini:

```
display_errors = On  
error_reporting = E_ALL | E_STRICT
```


Popular practices of PHP applications debugging.

Error output

Structured data

Code tracing

Objects analysis

Database query

Structured data output

```
echo '<pre>'.  
    print_r($myArray, true).  
'</pre>';
```

Array

```
(  
    [key1] => value 1  
    [key2] => value 2  
)
```

```
var_dump($myArray);
```

```
array(2) { ["key1"]=> string(7) "value 1" ["key2"]=>  
string(7) "value 2" }
```

Popular practices of PHP applications debugging.

Error output

Structured data

Code tracing

Objects analysis

Database query

Code tracing

`debug_backtrace()` and `print_debug_backtrace()`

Popular practices of PHP applications debugging.

Error output

Structured data

Code tracing

Objects analysis

Database query

What object do we
have?

`get_class()`

`get_class_vars()`

`get_declared_classes()`

`method_exists()`

Popular practices of PHP applications debugging.

Error output

Structured data

Code tracing

Objects analysis

Database query

Database query

a) `echo $sql = "SELECT `some_field1` FROM `table` WHERE `some_field2` = '$var'";`

b) `$res = mysql_query($sql) or die(mysql_error());`

BEST PRACTICES FOR MAGENTO

It's good enough for a majority of applications.

But what about Magento?



BEST PRACTICES FOR MAGENTO

Error output

```
error_reporting(E_ALL | E_STRICT);  
ini_set('display_errors', 1);
```

```
if (isset($_SERVER['MAGE_IS_DEVELOPER_MODE'])) {  
    Mage::setIsDeveloperMode(true);  
}
```

add
SetEnv MAGE_IS_DEVELOPER_MODE "true"
to .htaccess

BEST PRACTICES FOR MAGENTO

Structured data output

```
$customer = Mage::getModel('customer/customer')->load(1);  
  
print_r($customer);
```

BEST PRACTICES FOR MAGENTO

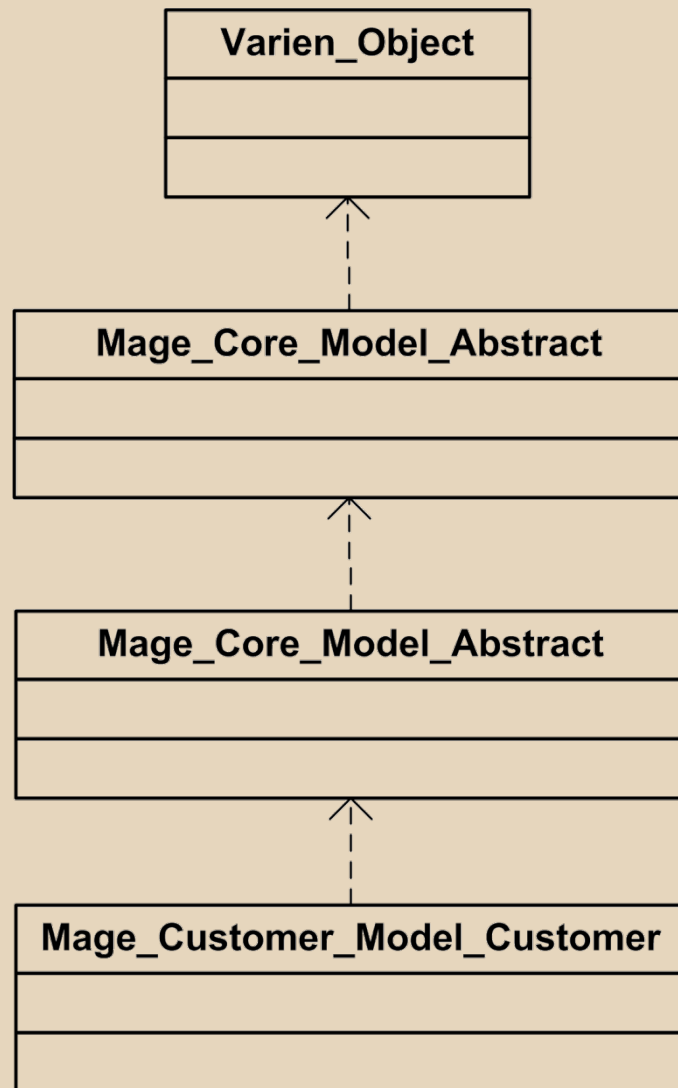
Structured data output

```
Mage_Customer_Model_Customer Object ( [_eventPrefix:protected] => customer [_eventObject:protected] => customer [_errors:protected] =>
Array ( ) [_attributes:protected] => [_addresses:protected] => [_addressesCollection:protected] => [_isDeleteable:protected] => 1 [
_isReadOnly:protected] => [_resourceName:protected] => customer/customer [_resource:protected] => [_resourceCollectionName:protected] =>
customer/customer_collection [_cacheTag:protected] => [_dataSaveAllowed:protected] => 1 [_isObjectNew:protected] => [_data:protected] =>
Array ( [entity_id] => 1 [entity_type_id] => 1 [attribute_set_id] => 0 [website_id] => 1 [email] => john.doe@example.com [group_id] => 1 [
increment_id] => 000000001 [store_id] => 1 [created_at] => 2007-08-30 23:23:13 [updated_at] => 2008-08-08 12:28:24 [is_active] => 1 [
firstname] => John [lastname] => Doe [password_hash] => 2049484a4020ed15d0e4238db22977d5:eg [prefix] => [middlename] => [suffix] =>
[taxvat] => [default_billing] => 274 [default_shipping] => 274 ) [_hasDataChanges:protected] => [_origData:protected] => Array ( [entity_id] =>
1 [entity_type_id] => 1 [attribute_set_id] => 0 [website_id] => 1 [email] => john.doe@example.com [group_id] => 1 [increment_id] =>
000000001 [store_id] => 1 [created_at] => 2007-08-30 23:23:13 [updated_at] => 2008-08-08 12:28:24 [is_active] => 1 [firstname] => John
[lastname] => Doe [password_hash] => 2049484a4020ed15d0e4238db22977d5:eg [prefix] => [middlename] => [suffix] => [taxvat] => [
default_billing] => 274 [default_shipping] => 274 ) [_idFieldName:protected] => entity_id [_isDeleted:protected] => [_oldFieldsMap:protected] =>
Array ( ) [_syncFieldsMap:protected] => Array ( ) )
```


BEST PRACTICES FOR MAGENTO

Useless again?

BEST PRACTICES FOR MAGENTO



Varien_Object::getData()

Varien_Object::debug()

BEST PRACTICES FOR MAGENTO

Structured data output

```
$customer = Mage::getModel('customer/customer')->load(1);
```

```
echo '<pre>'.print_r($customer->debug(), true);
```

BEST PRACTICES FOR MAGENTO

Structured data output

```
[entity_id] => 1
[entity_type_id] => 1
[attribute_set_id] => 0
[website_id] => 1
[email] => john.doe@example.com
[group_id] => 1
[increment_id] => 000000001
[store_id] => 1
[created_at] => 2007-08-30 23:23:13
[updated_at] => 2008-08-08 12:28:24
[is_active] => 1
[firstname] => John
[lastname] => Doe
[password_hash] => 2049484a4020ed15d0e4238db22977d5:eg
[prefix] =>
[middlename] =>
[suffix] =>
[taxvat] =>
[default_billing] => 274
[default_shipping] => 274
```

BEST PRACTICES FOR MAGENTO

Code tracing

<code>debug_backtrace()</code>		<code>Varien_Debug::backtrace()</code>
<code>print_debug_backtrace()</code>		

BEST PRACTICES FOR MAGENTO

Object analysis

`get_class()`

`get_class_vars()`

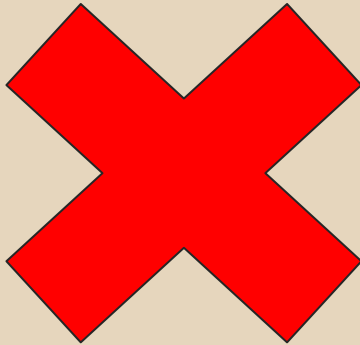
`get_declared_classes()`

`method_exists()`

Used everywhere in Magento

BEST PRACTICES FOR MAGENTO

Database queries



```
echo $sql = "SELECT `some_field1` FROM `table`";  
  
$res = mysql_query($sql) or die(mysql_error());
```

BEST PRACTICES FOR MAGENTO

Database queries

Display a query:

```
$myCollection->load(true);
```

Write a query to the system log:

```
$myCollection->load(false, true);
```


BEST PRACTICES FOR MAGENTO

Database queries

```
Mage::getModel('catalog/product')->getCollection()->load(true);
```



```
$model = Mage::getModel('catalog/product')->getCollection();  
$sql = $model->getSelect()->__toString();  
echo $sql;
```

```
SELECT `e`.* FROM `catalog_product_entity` AS `e`
```

BEST PRACTICES FOR MAGENTO

Database queries

Write all the queries to var/debug/pdo_mysql.log

lib/Varien/Db/Adapter/Pdo/Mysql.php:

```
protected $_debug    = true;  
protected $_logAllQueries= true;
```

Edit Block 'Right column trustmarks FR'

General Information

Block Title *

Right column trustmarks FR

Identifier *

footer_links

Store View *

All Store Views
Eiffel Tower eShop
Default Store
French
English



Status *

Activé ▼

How to write good commit messages?

1. Reference the Redmine and/or Mantis task

For Mantis, use one of:

```
Issue #123
```

```
Bug #123
```

```
Task #123
```

For Redmine, use one of:

```
Redmine #123
```

```
[123]
```

2. Explain briefly what this commit is about

Explain *why* the code was written:

- What was the problem to fix? (not just a reference to the big redmine task)
- What is the reasoning for your fix, and why do you think you fixed the bug?

3. If needed, include any backend configuration instruction

- Attribute to be added
- Procedure to configure
- ...

DEBUGGING and PROFILING

Installation

<http://xdebug.org/download.php>

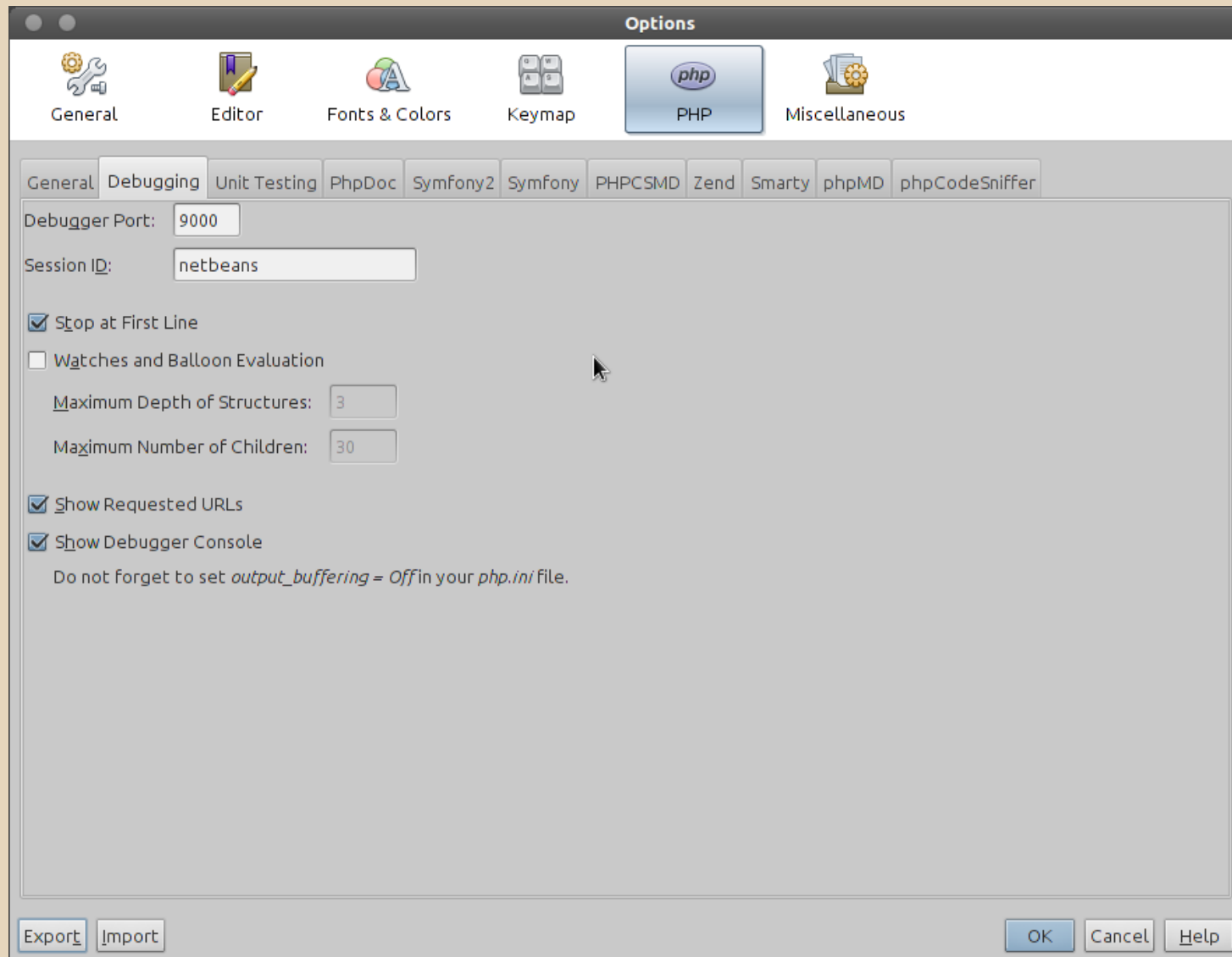
Setting up

```
xdebug.profiler_enable_trigger=on  
xdebug.remote_autostart=off  
xdebug.remote_enable=1  
xdebug.remote_host="127.0.0.1"  
xdebug.remote_port=9000  
xdebug.remote_handler="dbgp"  
xdebug.idekey="netbeans"  
xdebug.collect_vars=on  
xdebug.collect_params=4  
xdebug.show_local_vars=on  
xdebug.var_display_max_depth=5  
xdebug.show_exception_trace=on
```

```
zend_extension=/usr/lib/php5/20090626+libs/xdebug.so
```

```
; zend_extension_ts="c:\php\ext\php_xdebug-2.0.1-5.2.1.  
dll"
```

Setting up NetBeans IDE



Well, what good will that do?

http://example.com/index.php?XDEBUG_SESSION_START=netbeans

The screenshot shows the NetBeans IDE interface. The main editor displays a PHP file with a breakpoint set on line 846. The Variables window is open, showing the current state of variables during the debug session.

Code Snippet:

```
838 * @param array $modules
839 * @return array
840 */
841 protected function _sortModuleDepends($modules)
842 {
843     foreach ($modules as $moduleName => $moduleProps) {
844         $depends = $moduleProps['depends'];
845         foreach ($moduleProps['depends'] as $depend => $true) {
846             if ($moduleProps['active'] && (!isset($modules[$depend]) || empty($modules[$depend]['active']))) {
847                 Mage::throwException(
848                     Mage::helper('core')->__('Module "%1$s" requires module "%2$s".', $moduleName, $depend)
849                 );
850             }
851             $depends = array_merge($depends, $modules[$depend]['depends']);
852         }
853         $modules[$moduleName]['depends'] = $depends;
854     }
855     $modules = array_values($modules);
```

Variables Window:

Name	Type	Value
Superglobals		
depend	string	"Mage_Core"
depends	array[1]	
moduleName	string	"Mage_Page"
moduleProps	array[3]	
modules	array[68]	
true	boolean	1
this	Mage_Core_Model_Config	

The status bar at the bottom indicates the IDE is running, with the file name 'netbeans', the status 'running', and the line/column information '845 | 68 | INS'.

xDebug

http://example.com/index.php?XDEBUG_SESSION_START=netbeans



Xdebug helper <http://bit.ly/KuCo2c>



easy Xdebug <http://bit.ly/LKpvjC>

Profiling

```
xdebug.profiler_enable=1
```

Logs visualization

Webgrind <http://bit.ly/LXMGFJ>

Kcachegrind <http://bit.ly/KGzyAw>

WinCacheGrind <http://bit.ly/Nh4iPY>

Xdebugtoolkit <http://bit.ly/LmB4t9>

MacCallGrind <http://bit.ly/LlerGS>

CachegrindVisualizer <http://bit.ly/OD6dLy>

Magento®  connect

Magento Connect — developer tools



Commerce Bug <http://bit.ly/M8ggqh>

Debug

Welcome Controller/Request Models Collections Blocks **Layout** Class/URI Lookup System Tasks

Handles for this Request

```
<default />
<STORE_default />
<THEME_frontend_default_default />
<catalog_product_view />
<PRODUCT_TYPE_simple />
<PRODUCT_27 />
<customer_logged_out />
<MAP_popup />
<MAP_price_msrp_item />
<SHORTCUT_popup />
<SHORTCUT_uk_popup />
```

View Page Layout:: [XML](#) | [Text](#)
View Package Layout:: [XML](#) | [Text](#)

Path Information

Package	default
Default Theme	default
Custom Package	N/A
Custom Theme	N/A
Templates	app/design/frontend/default/default/template
Layout	app/design/frontend/default/default/layout
Translations	app/design/frontend/default/default/locale/en_US
Skin	skin/frontend/default/default

See Blocks tab for specific template locations

Magento Connect — developer tools



Developer Toolbar for Magento <http://bit.ly/LnSW8s>

This is a demo store. Any orders placed through this store will not be honored or fulfilled.

Magento®

Search entire store here...

[My Account](#) | [My Wishlist](#) | [My Cart](#)

Your Language

[Furniture](#) [Electronics](#) [Apparel](#)

Anashria
Get it while they last
Limited Offer 20% OFF

It's here - Magento end of summer SALE!!
Starts September 1st

20% OFF ELECTRONICS
15% OFF MEN'S SHOES
10% OFF APPLIANCES

Buy Canon REBEL
Starting at \$499

FREE SHIPPING on orders over \$50.
This offer is valid on all our store items.

Best Selling Products

[Sony VAIO VGN-TXN27N/B 11.1" Notebook PC](#)
See all [Laptops](#)

[Nine West Women's Lucero Pump](#)
See all [Shoes](#)

[Olympus Stylus 750 7.1MP Digital Camera](#)
See all [Digital Cameras](#)

[Acer Ferrari 3200 Notebook Computer PC](#)
See all [Laptops](#)

[ASICS® Men's GEL-Kayano® XII](#)

[Coalescer: Functioning On Impatience T-Shirt](#)

POPULAR TAGS
[Camera](#) [Hohoho](#) [SEXY](#)
[Test](#) [bones](#) [cool](#) [cool t-shirt](#)
[crap](#) [good](#) [green](#) [hip](#) [laptop](#)
[mobile](#) [nice](#) [notebook](#) [phone](#) [red](#)
[tight](#) [trendy](#) [young](#)
[View All Tags](#)

MY CART
You have no items in your shopping cart.

COMPARE
You have no items to compare.

COMMUNITY POLL
What is your favorite Magento feature?
☐ Layered Navigation
☐ Price Rules
☐ Category Management
☐ Compare Products
[Vote](#)

Layout
Keep your layout consistent with our standard layout handlers.
Blocks
36 USED BLOCKS

Performance
TIME: 1.16s MEM: 15 MB

Configuration
1680 CONFIGURATIONS

Controller
CONTROLLER AND REQUEST

Models
7 MODELS, 45 QUERIES

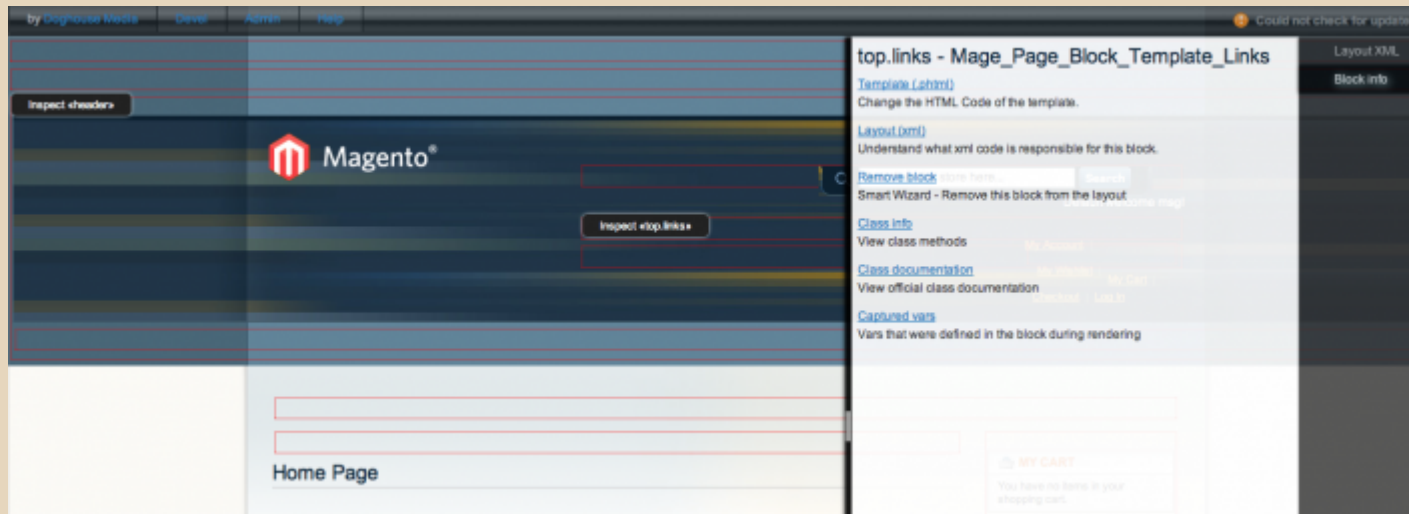
BACK TO
Keep your layout consistent with our standard layout handlers.
to S
are

NOW ACCEPTING PayPal

Magento Connect — developer tools

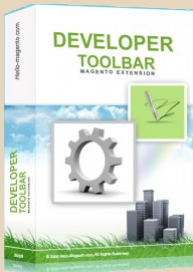


Advanced Developer Tools <http://bit.ly/Lo1Vqa>



Only for versions 1.6.1 and earlier

Magento Connect — developer tools



Developer Toolbar <http://bit.ly/LnD1Hk>

Brand	Sony	<input type="radio"/> Category Management
In Depth	Processor: The Ultra Low Voltage Intel Core Solo U1500 processor offers a 1.33 GHz speed paired with a fast 533 MHz front-side bus and large 2 MB L2 cache. (An L2, or	<input type="radio"/> Compare Products
		<input type="button" value="Vote"/>
Frontend Hints Backend Hints Logs Merge JS Store Code in Urls SEO Rewrites Translate Inline Clean & Disable All Cache		

Magento Connect — developer tools



Magento FirePHP <http://bit.ly/LnYGyX>

```
Mage::helper('firephp')->send('Lorem ipsum sit amet ..');  
Mage::helper('firephp')->debug(Mage::getModel('catalog/product')->load  
(54));
```

The screenshot shows the Magento Admin interface in a Mozilla Firefox browser. The browser's address bar displays the URL: `magento.loc/index.php/stores/adminhtml_index/index/key/eda95d425119e7f5b404c3668c9cda35/`. The page title is "Magento Admin". The main content area shows the "Store Management" section with a table of stores. The table has two columns: "Store #" and "Store Name". The table contains two rows: one with "15" and "test2", and another with "14" and "test".

Overlaid on the right side of the browser window is the "Variable Viewer" window, which displays the debug output of the FirePHP script. The output is an array of data for a product with ID 54:

```
array(  
  ['entity_id'] => 54  
  ['entity_type_id'] => 10  
  ['attribute_set_id'] => 42  
  ['type_id'] => 'grouped'  
  ['sku'] => 1114  
  ['created_at'] => '2007-08-28 16:38:54'  
  ['updated_at'] => '2008-08-08 15:00:22'  
  ['has_options'] => 0  
  ['required_options'] => 0  
  ['name'] => 'Magento Red Furniture Set'  
  ['meta_title'] => 'Magento Red Furniture Set'  
  ['meta_description'] => 'Magento Red Furniture Set'  
  ['image'] => '/m/a/magento-red-furniture-set.jpg'  
  ['small_image'] => '/m/a/magento-red-furniture-set.jpg'  
  ['url_key'] => 'magento-red-furniture-set'  
  ['thumbnail'] => '/m/a/magento-red-furniture-set.jpg'  
  ['gift_message_available'] =>  
  ['url_path'] => 'magento-red-furniture-set.html'  
  ['custom_design'] =>  

```

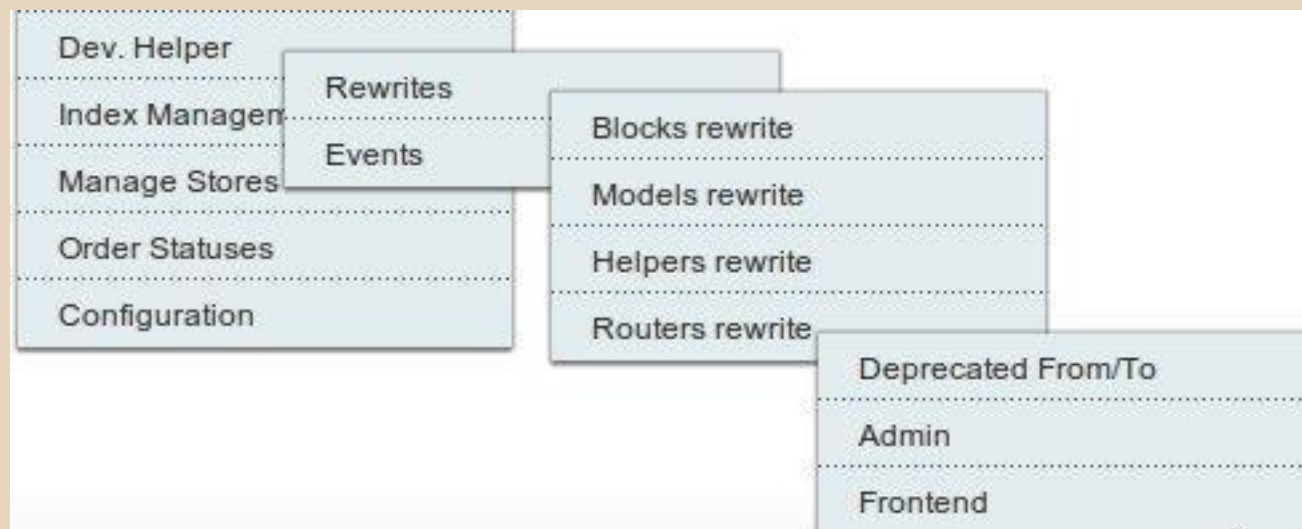
At the bottom of the browser window, the "Console" tab is active, showing the following log entry:

```
array('entity_id'=>'54', 'entity_type_id'=>'10', 'attribute_set_id'=> ... )
```

Magento Connect — developer tools



Developer Helper <http://bit.ly/OLauwz>



Dev. Helper / Rewrites / Models

From	To
Mage_Core_Model_Url_Rewrite	AW_Productquestions_Model_Urlrewrite
Mage_Eav_Model_Entity_Increment_Numeric	Aschroder_SetStartOrderNumber_Model_Backend_Increment_Numeric
Mage_Cron_Model_Schedule	Noovias_Cron_Model_Schedule

THANK YOU FOR YOUR
ATTENTION