

General Manuals

Getting Started

- How to Setup Cron
- How to enable Magento Profiler
- How to Translate Extension Texts and Emails
- How to Recover Store After Installation With Enabled Compilation

How to Setup Cron for Magento

Path to Magento root folder: Path to php: What access to your server do you have? CPanel Plesk SSH Set up on Windows Other How often Cron job must be executed? every minute every five minutes (**preferable**) every ten minutes every hour Get additional period patterns at crontab generator

Setting up the cron job using a CPanel

1. Open settings of cron jobs in your control panel.
2. Before setting up a new cron job, check that it has not been set up yet.
3. Add a new cron job with the following parameters:
 - Minute: *
 - Hour: *
 - Day: *
 - Month: *
 - Weekday: *
 - We suggest to set cron job execution time to every 5 minutes: */5 * * * *
 - Command: {{php_path}} -f {{magento_path}}/cron.php >> {{magento_path}}/var/log/cron.log 2>&1

Add New Cron Job

Common Settings:	Every 5 minutes (* / 5 * * * *)	
Minute:	*/5	Every 5 minutes (* / 5) ✓
Hour:	*	Every hour (*) ✓
Day:	*	Every day (*) ✓
Month:	*	Every month (*) ✓
Weekday:	*	Every weekday (*) ✓
Command:	<input type="text" value="<php_path> -f <magento_path>/cron.php >> <magento_path>/var/log/cron.log 2>&1"/> ✓	
<input type="button" value="Add New Cron Job"/>		

Problems with cron

If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

Setting up the cron job using a Plesk

1. Open settings of cron jobs in your control panel.
2. Before setting up a new cron job, check that it has not been set up yet.
3. Add a new cron job with the following parameters:
 - o Minute: *
 - o Hour: *
 - o Day: *
 - o Month: *
 - o Weekday: *
 - o We suggest to set cron job execution time to every 5 minutes: */5 * * * *
 - o Command: {{php_path}} -f {{magento_path}}/cron.php >> {{magento_path}}/var/log/cron.log 2>&1

Schedule a Task

Task

Switched on

Minute *

Hour *

Day of the month *

Month *

Enter the value

in UNIX crontab format, for example 1, 5-7, */4

Select the month

Day of the week *

Enter the value

in UNIX crontab format, for example 1, 4-6

Select the day of the week

Command *

* Required fields

OK

Cancel

Problems with cron

If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

Setting up the cron job via SSH

1. Login to your server
2. Before setting up a new cron job, check that it has not been set up yet. Use the following command to get a list of cron jobs: `crontab -l`
3. To add a new cron task, open crontab for editing: `crontab -e`
4. Add the following record to the end of the list of jobs: `{{time_type}} {{php_path}} -f {{magento_path}}/cron.php >> {{magento_path}}/var/log/cron.log 2>&1`

Problems with cron

If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

Setting up the cronjob on Windows

Windows has a **Scheduled Tasks** service which is accessible from the Control Panel.

Go to the scheduled task manager and add a task to run the following command: `{{php_path}}`

`C:{{magento_path}}\cron.php` To read more information how to create **Scheduled Tasks** on Windows, follow this link

Problems with cron

If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

Setting up the cronjob

1. Open settings of cron jobs in your control panel.
2. Before setting up a new cron job, check that it has not been set up yet.
3. Create a new cron job and add the following record to the end of the list of jobs: `{{time_type}} {{php_path}} -f {{magento_path}}/cron.php >> {{magento_path}}/var/log/cron.log 2>&1`

Problems with cron

If you do not have access to crontab on your service, you can set up the page that needs to be run periodically as a Home Page in your personal computer browser: `<your_domain>/cron.php`. Every time you open this page, it will execute the scheduled task(s) on your server. You could also set up a Scheduled Task on a computer to which you have access and which is usually running. It could then access a web accessible page that will run a cron job. In UNIX/BSD/Linux it can be done with **wget** or **curl** utilities. Lastly, if either of these won't work for you, there are a number of online cron services (complete list: <http://onlinecronservices.com>, <http://onlinecronjobs.com>) that may be useful. Many are free, but have restrictions on execution frequency or job count. If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

How to enable Magento Profiler

Magento comes with a very useful tool called the **Profiler**. If you've ever wanted to know how long a

piece of code takes to run, or which parts of your website are running slow, then the profiler is possibly the best tool for the job. Firstly, you have to turn it on. In your admin area, go to **System > Configuration > Advanced > Developer > Debug**, and switch **Profiler** to **Yes** for the appropriate store view:

The screenshot shows the Magento Admin configuration interface. On the left, the 'Configuration' sidebar is expanded to 'Developer'. The main content area is titled 'Developer' and contains a 'Debug' section. Within the 'Debug' section, the 'Profiler' setting is set to 'Yes'. Below this, there are several other settings sections: 'Template Settings', 'Translate Inline', 'Log Settings', 'JavaScript Settings', and 'CSS Settings'. At the top left, the 'Current Configuration Scope' is set to 'Default Config'.

Then load any page on your front end, and scroll to the bottom.

You'll see a link to **[profiler]**. Click it and you'll see this:

```

Memory usage: real: 29884416, emalloc: 29392232
Code Profiler|Time|Cnt|Emalloc|RealMem

```

Not particularly useful in it's current format. That's because the Profiler is switched off by default. To enable it, open **index.php** file located at the <MAGENTO_ROOT> folder. Find the line with `#Varien_Profiler::enable();` and un-comment it (remove the hash sign #): `Varien_Profiler::enable();` Reload your front end page and the profiler will be spitting out all sorts of information. The information is hierarchical, but it isn't displayed as such so it can be difficult to understand:

Code Profiler

```

mage
mage::app::init::system_config
CORE::create_object_of::Mage_Core_Model_Cache
mage::app::init::config::load_cache
mage::app::init::stores
CORE::create_object_of::Mage_Core_Model_Resource_Website
CORE::create_object_of::Mage_Core_Model_Resource_Website_Collecti
DISPATCH EVENT:resource_get_tablename
init_config_section:stores_admin
init_config_section:stores_default
init_config_section:stores_french
init_config_section:stores_german
init_config_section:stores_hwd
mage::app::init_front_controller
mage::app::init_front_controller::collect_routers
init_config_section:admin
DISPATCH EVENT:controller_front_init_routers
OBSERVER: cms
mage::dispatch::db_url_rewrite
mage::dispatch::routers_match
mage::dispatch::controller::action::predispatch
CORE::create_object_of::Mage_Core_Model_Layout
CORE::create_object_of::Mage_Core_Model_Session
Mage_Core_Model_Session_Abstract_Varien::start/start

```

How to enable SQL Profiling

To display in the **Profiler** information about SQL queries, such as **Number of queries executed and the time taken for it, Average query length, Queries per second, Longest query length, and Longest query** you need to enable **SQL Profiling**. 1) To enable **SQL Profiling**, you need to enable **Profiler** from admin as described above. 2) Open the local.xml config file located in <magento_root>/app/etc/ folder and **add** the node <profiler>true</profiler> into the **global/resources/default_setup/connection/** <default_setup> <connection> <host>localhost</host> <username>root</username> <password>astral</password> <dbname>feedexport1810</dbname> <initStatements><![CDATA[SET NAMES utf8]]></initStatements> <model>mysql4</model> <type>pdo_mysql</type> <pdoType></pdoType> <active>1</active> <profiler>true</profiler> </connection> 3) Save the file and clear the cache.

Now if you refresh the page, you will see the **SQL Profiling** output below the code profiling results:

__EAV_LOAD_MODEL_ATTRIBUTES__	0.0012	1	85,688	0
__EAV_LOAD_MODEL_AFTER_LOAD__	0.0088	1	342,040	262,144
frontend/base/default/template/page/html/footer.phtml	0.0028	1	128,880	0
frontend/base/default/template/page/html/cookie_notice.phtml	0.0001	1	16,480	0

Executed 41
Average qu
Quer
Longest

SHOW TAB

The first thing to note is the memory usage at the top. The numbers are in bytes, so divide the number by 1048576 to get the number of megabytes of RAM used to load the page. High memory usage can affect loading speeds, especially when lots of users are loading lots of different pages.

In the table the following headers **Code Profiler**, **Cnt**, **Emalloc**, **RealMem** show every step the system takes in the loading of a page, where:

- **Code Profiler** - the name of the block of code being executed
- **Time** - The time it took to complete in seconds
- **Cnt** - Short for Count, represents the number of times this individual block ran to generate the output
- **Emalloc** - The amount of memory PHP assigned to this single operation, Its a mix of core PHP `memory_get_usage()` function without true parameter passed to it, minus timer values
- **RealMem** - The actual amount of memory used to perform the operation. Its final value is also obtained through `memory_get_usage(true)` function minus timer values but this time with true parameter passed in.

How to use information from Profiler

Let us bring some light on all this information. The execution time param is the most interesting for us. However, it's a bit tricky. If you try to make the sum of all values of the column, you won't get the actual execution time. Very often the profiler's method calls are placed one inside the other. On the screenshot above you can see that `mage::dispatch::routers_match` has the biggest execution time (**2.6948 seconds**). However, it does not mean that there's something wrong with the routers match operation in Magento. In that case, other profiler's records are placed 'inside' the routers match record. That should be kept in mind. If you go down on the list you will find the most nested process. Let's check the real example. Look at the following image:

```

frontend/base/default/template/page/html/header.phtml
frontend/base/default/template/catalogsearch/form.mini.phtml
frontend/base/default/template/page/template/links.phtml
frontend/base/default/template/page/switch/languages.phtml
frontend/base/default/template/custom_extension/cart.phtml
frontend/base/default/template/page/html/topmenu.phtml

```

Note that `frontend/base/default/template/page/html/header.phtml` execution time is quite long – **1.8251 seconds**. We can see the set of header's child templates in the profiler's list at the bottom of `frontend/base/default/template/page/html/header.phtml`. The slowest child is `frontend/base/default/template/custom_extension/cart.phtml`. Ok then, it's a custom extension in our

store which displays the shopping cart contents as a pull down menu in the header. If we disable this extension we will get the page load performance increased by more than one second.

There are two major reasons of the bottlenecks like this one:

1) The first one – it's a bad code. An extension might have a big number of unnecessary database queries or the logic is simply bad organised and, as a result, resource consuming. In that case, you can ask your developer to check the extension or try to fix it yourself.

2) The second reason is the need to get some additional information from the database or make some additional operations to have the necessary information. That is exactly what happened in our case: the header's shopping cart pull down menu extension makes the additional database requests to get the shopping cart items information. By default (when there's no additional shopping cart information on each page), Magento does not fetch this information. So, in cases like this one you should think twice: worth the header's shopping cart one second of the page loading time or not. Not always **.phtml** templates execution time reflects the bottlenecks. It's useful to pay attention to **OBSERVER** records in the profiler's report. Some extensions might execute an additional logic on the different system events. Just check what observer is the slowest and make sure that it's not Magento core observer. Usually, the developers give names to their observers by adding a vendor name, something like this: `developer_extmenu_add_topmenu_items`. So, if some custom observer slows down your system you should either disable the extension or investigate it for a reason of the bottleneck.

```
DISPATCH EVENT:controller_action_predispatch
```

```
OBSERVER: atwix_monitor_controller_action_predispatch
```

```
OBSERVER: log
```

```
CORE::create_object_of::Mage_Log_Model_Visitor
```

How to add Profiling to custom module code

To add profiling to custom module code add next lines:

```
Varien_Profiler::start('LabelForCodeFragmentToProfile'); //... here some code we like to get statistics for  
Varien_Profiler::stop('LabelForCodeFragmentToProfile');
```

If cumulative time of execution for this code is more than .0001 second it will be shown in profiler table with cumulative time, number of executions and cumulative memory consumption.

How to Translate Extension Texts and Emails

Path to Magento root folder: Select Mirasvit Extension to Translate Select Extension ... Advanced Newsletter Advanced Product Feeds Advanced Reports Advanced SEO Suite Advanced Sphinx Search Pro Asynchronous Cache Automatic Internal Links Banner Management System Fast Asynchronous Re-indexing Follow Up Email Full Page Cache Gift Registry Help Desk MX Knowledge Base Menu Manager Pro Product Labels Product Questions Promotional Offers Reward Points + Referral program RMA Search AutoComplete & Suggest Pro Search Spell-Correction Sphinx Search Ultimate Store Credit & Refund Video For Products Pro Yandex Market Export Select Desired for Translation Language Select Language and Country ... Afrikaans (Suid-Afrika) / Afrikaans (South Africa) Azrbaycan (Azrbaycan) / Azerbaijani (Azerbaijan) Bahasa Indonesia (Indonesia) / Indonesian (Indonesia) Bahasa Melayu (Malaysia) / Malay (Malaysia) Bosanski (Bosna i Hercegovina) / Bosnian (Bosnia and Herzegovina) Català (Espanya) / Catalan (Spain) Cymraeg (Y Deyrnas Unedig) / Welsh (United Kingdom) Dansk (Danmark) / Danish (Denmark) Deutsch (Deutschland) / German (Germany)

Deutsch (Schweiz) / German (Switzerland) Deutsch (Österreich) / German (Austria) Eesti (Eesti) / Estonian (Estonia) English (Australia) / English (Australia) English (Canada) / English (Canada) English (Ireland) / English (Ireland) English (New Zealand) / English (New Zealand) English (United Kingdom) / English (United Kingdom) English (United States) / English (United States) Español (Argentina) / Spanish (Argentina) Español (Chile) / Spanish (Chile) Español (Colombia) / Spanish (Colombia) Español (Costa Rica) / Spanish (Costa Rica) Español (España) / Spanish (Spain) Español (México) / Spanish (Mexico) Español (Panamá) / Spanish (Panama) Español (Perú) / Spanish (Peru) Español (Venezuela) / Spanish (Venezuela) Filipino (Pilipinas) / Filipino (Philippines) Français (Canada) / French (Canada) Français (France) / French (France) Galego (España) / Galician (Spain) Hrvatski (Hrvatska) / Croatian (Croatia) Italiano (Italia) / Italian (Italy) Italiano (Svizzera) / Italian (Switzerland) Kiswahili (Kenya) / Swahili (Kenya) Latviešu (Latvija) / Latvian (Latvia) Lietuvi (Lietuva) / Lithuanian (Lithuania) Magyar (Magyarország) / Hungarian (Hungary) Nederlands (Nederland) / Dutch (Netherlands) Norsk Bokmål (Norge) / Norwegian Bokmål (Norway) Nynorsk (Noreg) / Norwegian Nynorsk (Norway) Polski (Polska) / Polish (Poland) Português (Brasil) / Portuguese (Brazil) Português (Portugal) / Portuguese (Portugal) Român (România) / Romanian (Romania) Shqip (Shqipëri) / Albanian (Albania) Slovenina (Slovensko) / Slovak (Slovakia) Slovenšina (Slovenija) / Slovenian (Slovenia) Srpski (Srbija) / Serbian (Serbia) Suomi (Suomi) / Finnish (Finland) Svenska (Sverige) / Swedish (Sweden) Ting Vit (Vit Nam) / Vietnamese (Vietnam) Türkçe (Türkiye) / Turkish (Turkey) íslenska (Ísland) / Icelandic (Iceland) eština (eská republika) / Czech (Czech Republic) () / Greek (Greece) () / Belarusian (Belarus) () / Bulgarian (Bulgaria) () / Macedonian (Macedonia) () / Mongolian (Mongolia) () / Russian (Russia) () / Ukrainian (Ukraine) () / Hebrew (Israel) () / Arabic (Algeria) () / Arabic (Kuwait) () / Arabic (Morocco) () / Arabic (Saudi Arabia) () / Arabic (Egypt) () / Persian (Iran) () / Hindi (India) () / Bengali (Bangladesh) () / Gujarati (India) () / Thai (Thailand) () / Lao (Laos) () / Georgian (Georgia) () / Khmer (Cambodia) () / Chinese (China) () / Chinese (Hong Kong SAR China) () / Chinese (Taiwan) () / Japanese (Japan) () / Korean (South Korea)

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied `Mirasvit_{{extension_name}}.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All extension templates are named according to next pattern: `{{extension_name}} - [template_name]`.
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to **System > Configuration**. Click on the Extension at the **Mirasvit Extensions** tab.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Email Notifications** list, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied `Mirasvit_{{extension_name}}.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.

You can add all desired for translation texts of the extension into the file according next pattern:

Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "

2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All newsletter templates are named according to next pattern: Newsletter - [template_name].
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to **System > Configuration > Newsletter**.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Subscription Options**, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied `Mirasvit_{{extension_name}}.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"
"Add New Field", "Add New Field"
"Add New Gateway", "Add New Gateway"
"Add New Pattern", "Ajouter nouveau modèle"
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa
"Add New Field", "Ajouter nouveau champ"
"Add New Gateway", "Ajouter nouvelle passer
"Add New Pattern", "Ajouter nouveau modèle"
"Add New Priority", "Ajouter nouvelle prior
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All extension templates are named according to next pattern: `Gift Registry - [template_name]`.
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to **Gift registry > Settings**.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Notifications** list, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at

{{magento_path}}/app/locale/en_US/ to the folder {{magento_path}}/app/locale/{{locale_code}}/, where {{locale_code}} - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied Mirasvit_{{extension_name}}.csv file at the desired for translation language code folder: {{magento_path}}/app/locale/{{locale_code}}/ and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to Translate Extension Email Templates

On your server, copy the files located at {{magento_path}}/app/locale/en_US/template/email/mst_{{extension_name | lowercase}}/ to the folder {{magento_path}}/app/locale/{{locale_code}}/template/email/mst_{{extension_name | lowercase}}/. Open the copied email templates at the desired for translation language code folder. Replace the words and phrases into the appropriate language.

How to Translate Texts of the Extension

On your server, copy the file Mirasvit_{{extension_name}}.csv located at {{magento_path}}/app/locale/en_US/ to the folder {{magento_path}}/app/locale/{{locale_code}}/, where {{locale_code}} - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied Mirasvit_{{extension_name}}.csv file at the desired for translation language code folder: {{magento_path}}/app/locale/{{locale_code}}/ and translate the text in the **right** column. Don't change the text in the **left** column.

You can add all desired for translation texts of the extension into the file according next pattern:

Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "

2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All extension templates are named according to next pattern: `{{extension_name}} - [template_name]`.
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to `{{extension_name}} > Settings`.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Email Notification Settings** list, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

Extension **Advanced Reports** unites such products as **Advanced Dashboard** and **Reports**.

Thus, to translate the extension, you need to go to the folder

`{{magento_path}}/app/locale/{{locale_code}}/` and create next files: `Mirasvit_Advr.csv` and `Mirasvit_Advd.csv`,

where `{{locale_code}}` - is a desired for translation language code folder.

1. To translate **Reports** texts, open the created `Mirasvit_Advr.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
2. To translate **Dashboard** texts, open the created `Mirasvit_Advd.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
3. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to Translate Extension Email Templates

To translate emails of the **Advanced Dashboard**, follow next steps:

On your server, copy the files located at `{{magento_path}}/app/locale/en_US/template/email/mst_advd/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/template/email/mst_advd/`.

Open the copied email templates at the desired for translation language code folder. Replace the words and phrases into the appropriate language.

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at

`{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`,

where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied `Mirasvit_{{extension_name}}.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right**

column. Don't change the text in the **left** column.

You can add all desired for translation texts of the extension into the file according next pattern:

Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "

2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior
```

How to Translate Texts of the Extension

Extension **Reward Points + Referral program** unites such products as **Reward Points** and **Social Rewards** Thus, to translate the extension, you need to copy on your server files: `Mirasvit_Rewards.csv` and `Mirasvit_RewardsSocial.csv`, which are located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. To translate **Reward Points** texts, open the copied `Mirasvit_Rewards.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
2. To translate **Social Networks** texts, open the copied `Mirasvit_RewardsSocial.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
3. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All extension templates are named according to next pattern: `{{extension_name}} - [template_name]`.
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to **Reward Points > Settings**.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Notification Settings** list, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

Sphinx Search Ultimate unites such products as **Search Spell-Correction**, **Search AutoComplete & Suggestions** and **Advanced Sphinx Search Pro**.

Thus, to translate the extension, you need to copy on your server files: `Mirasvit_Misspell.csv`, `Mirasvit_SearchAutocomplete.csv` and `Mirasvit_SearchIndex.csv`, which are located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. To translate **Search Spell-Correction** texts, open the copied `Mirasvit_Misspell.csv` file at the

desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.

2. To translate **Search AutoComplete & Suggestions** texts, open the copied `Mirasvit_SearchAutocomplete.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
3. To translate **Advanced Sphinx Search Pro** texts, open the copied `Mirasvit_SearchIndex.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.

You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "

4. Save the translated files and clear magento cache.

Example of the translated into the french language extension text:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to recover store after installation with enabled compilation

First of all, you need to disable compilation. Open the file `config.php` which is located at `<magento_path>/includes/`.

At around line 28, find next lines: `define('COMPILER_INCLUDE_PATH', dirname(__FILE__).DIRECTORY_SEPARATOR.'src');` `#define('COMPILER_COLLECT_PATH', dirname(__FILE__).DIRECTORY_SEPARATOR.'stat');` and comment them out:
`#define('COMPILER_INCLUDE_PATH', dirname(__FILE__).DIRECTORY_SEPARATOR.'src');`
`#define('COMPILER_COLLECT_PATH', dirname(__FILE__).DIRECTORY_SEPARATOR.'stat');`
Save the changes. After compilation is disabled, store must be recovered.

If you need to enable the Compilation mode, you need to run recompilation process. Go to **System > Tools > Compilation** and click on **Run Compilation Process**.

How to fix access issues after applying magento patch SUPEE-6285

Download the Patch and copy the file to the Magento Root folder on your server.
To apply this patch, use one of these methods:

1. Apply the patch via Browser

Open the patch file in your browser. For example, http://yoursite.com/m_controllers_patch.php

2. Apply the patch via Command Line

To run the script via command line (shell, SSH) you need to execute next command: `php {{magento_path}}/m_controllers_patch.php` If you receive an error message while executing the patch: **"Error updating the file. Please check file permissions, it should have write access"**, you need to set write access for the file and its path. Run following command in the command line to set full access: `sudo chmod 777 -R app/code/local/Mirasvit/ModuleName/controllers/Adminhtml/` where `ModuleName` is a name of the Mirasvit extension. If you don't have access to command line, you can set the appropriate write access in the **Filezilla** or ask your system administrator to do this.

How to fix access issues after applying magento patch SUPEE-6788

Download the Patch for SUPEE-6788 and copy the file to the `{{magento_path}}/` folder on your server. Before applying the patch, make backup of the folder `{{magento_path}}/app/` To apply this patch, use one of these methods:

1. Apply the patch via Command Line (Preferable)

To run the script via command line (shell, SSH) you need to execute next command: `php {{magento_path}}/fixsupee6788.php` where `{{magento_path}}` - path to Magento Root folder on your server.

2. Apply the patch via Browser

Execute the patch in your browser. For example, <http://yoursite.com/fixsupee6788.php> If you receive an error message while executing the patch: **"Unable to write new configuration to ..."**, you need to set write access for the file and its path.

Run following command in the command line to set full access: `sudo chmod 777 -R <file_path>/` After

script executing, patch will be applied to all **Mirasvit** extensions.
For security reasons, delete the executed patch from the server.

Getting Started

- How to Setup Cron
- How to enable Magento Profiler
- How to Translate Extension Texts and Emails
- How to Recover Store After Installation With Enabled Compilation

How to Setup Cron for Magento

Path to Magento root folder: Path to php: What access to your server do you have? CPanel Plesk SSH
Set up on Windows Other How often Cron job must be executed? every minute every five minutes (**preferable**) every ten minutes every hour Get additional period patterns at crontab generator

Setting up the cron job using a CPanel

1. Open settings of cron jobs in your control panel.
2. Before setting up a new cron job, check that it has not been set up yet.
3. Add a new cron job with the following parameters:
 - Minute: *
 - Hour: *
 - Day: *
 - Month: *
 - Weekday: *
 - We suggest to set cron job execution time to every 5 minutes: */5 * * * *
 - Command: {{php_path}} -f {{magento_path}}/cron.php >> {{magento_path}}/var/log/cron.log 2>&1

Add New Cron Job

Common Settings:	Every 5 minutes (* / 5 * * * *)	
Minute:	*/5	Every 5 minutes (* / 5) ✓
Hour:	*	Every hour (*) ✓
Day:	*	Every day (*) ✓
Month:	*	Every month (*) ✓
Weekday:	*	Every weekday (*) ✓
Command:	<input type="text" value="<php_path> -f <magento_path>/cron.php >> <magento_path>/var/log/cron.log 2>&1"/> ✓	
<input type="button" value="Add New Cron Job"/>		

Problems with cron

If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

Setting up the cron job using a Plesk

1. Open settings of cron jobs in your control panel.
2. Before setting up a new cron job, check that it has not been set up yet.
3. Add a new cron job with the following parameters:
 - o Minute: *
 - o Hour: *
 - o Day: *
 - o Month: *
 - o Weekday: *
 - o We suggest to set cron job execution time to every 5 minutes: */5 * * * *
 - o Command: {{php_path}} -f {{magento_path}}/cron.php >> {{magento_path}}/var/log/cron.log 2>&1

Schedule a Task

Task

Switched on

Minute *

Hour *

Day of the month *

Month *

Enter the value

in UNIX crontab format, for example 1, 5-7, */4

Select the month

Day of the week *

Enter the value

in UNIX crontab format, for example 1, 4-6

Select the day of the week

Command *

* Required fields

OK

Cancel

Problems with cron

If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

Setting up the cron job via SSH

1. Login to your server
2. Before setting up a new cron job, check that it has not been set up yet. Use the following command to get a list of cron jobs: `crontab -l`
3. To add a new cron task, open crontab for editing: `crontab -e`
4. Add the following record to the end of the list of jobs: `{{time_type}} {{php_path}} -f {{magento_path}}/cron.php >> {{magento_path}}/var/log/cron.log 2>&1`

Problems with cron

If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

Setting up the cronjob on Windows

Windows has a **Scheduled Tasks** service which is accessible from the Control Panel.

Go to the scheduled task manager and add a task to run the following command: `{{php_path}}`

`C:{{magento_path}}\cron.php` To read more information how to create **Scheduled Tasks** on Windows, follow this link

Problems with cron

If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

Setting up the cronjob

1. Open settings of cron jobs in your control panel.
2. Before setting up a new cron job, check that it has not been set up yet.
3. Create a new cron job and add the following record to the end of the list of jobs: `{{time_type}} {{php_path}} -f {{magento_path}}/cron.php >> {{magento_path}}/var/log/cron.log 2>&1`

Problems with cron

If you do not have access to crontab on your service, you can set up the page that needs to be run periodically as a Home Page in you personal computer browser: `<your_domain>/cron.php`. Every time you open this page, it will execute the scheduled task(s) on your server. You could also set up a Scheduled Task on a computer to which you have access and which is usually running. It could then access a web accessible page that will run a cron job. In UNIX/BSD/Linux it can be done with **wget** or **curl** utilities. Lastly, if either of these won't work for you, there are a number of online cron services (complete list: <http://onlinecronservices.com>, <http://onlinecronjobs.com>) that may be useful. Many are free, but have restrictions on execution frequency or job count. If you have installed cronjob, but still have problems with cron or not sure if cron is running, try to install a free extension **AOE Scheduler**. This extension can show you the status of cron and cron tasks including a timeline view.

How to enable Magento Profiler

Magento comes with a very useful tool called the **Profiler**. If you've ever wanted to know how long a

piece of code takes to run, or which parts of your website are running slow, then the profiler is possibly the best tool for the job. Firstly, you have to turn it on. In your admin area, go to **System > Configuration > Advanced > Developer > Debug**, and switch **Profiler** to **Yes** for the appropriate store view:

The screenshot shows the Magento Admin configuration interface. On the left, the 'Configuration' sidebar is expanded to 'Developer'. The main content area is titled 'Developer' and contains a 'Debug' section. Within the 'Debug' section, the 'Profiler' setting is set to 'Yes'. Below this, there are several other settings sections: 'Template Settings', 'Translate Inline', 'Log Settings', 'JavaScript Settings', and 'CSS Settings'. At the top left, the 'Current Configuration Scope' is set to 'Default Config'.

Then load any page on your front end, and scroll to the bottom.

You'll see a link to **[profiler]**. Click it and you'll see this:

```

Memory usage: real: 29884416, emalloc: 29392232
Code Profiler|Time|Cnt|Emalloc|RealMem

```

Not particularly useful in it's current format. That's because the Profiler is switched off by default. To enable it, open **index.php** file located at the <MAGENTO_ROOT> folder. Find the line with `#Varien_Profiler::enable();` and un-comment it (remove the hash sign #): `Varien_Profiler::enable();` Reload your front end page and the profiler will be spitting out all sorts of information. The information is hierarchical, but it isn't displayed as such so it can be difficult to understand:

Code Profiler

```

mage
mage::app::init::system_config
CORE::create_object_of::Mage_Core_Model_Cache
mage::app::init::config::load_cache
mage::app::init::stores
CORE::create_object_of::Mage_Core_Model_Resource_Website
CORE::create_object_of::Mage_Core_Model_Resource_Website_Collecti
DISPATCH EVENT:resource_get_tablename
init_config_section:stores_admin
init_config_section:stores_default
init_config_section:stores_french
init_config_section:stores_german
init_config_section:stores_hwd
mage::app::init_front_controller
mage::app::init_front_controller::collect_routers
init_config_section:admin
DISPATCH EVENT:controller_front_init_routers
OBSERVER: cms
mage::dispatch::db_url_rewrite
mage::dispatch::routers_match
mage::dispatch::controller::action::predispatch
CORE::create_object_of::Mage_Core_Model_Layout
CORE::create_object_of::Mage_Core_Model_Session
Mage_Core_Model_Session_Abstract_Varien::start/start

```

How to enable SQL Profiling

To display in the **Profiler** information about SQL queries, such as **Number of queries executed and the time taken for it, Average query length, Queries per second, Longest query length, and Longest query** you need to enable **SQL Profiling**. 1) To enable **SQL Profiling**, you need to enable **Profiler** from admin as described above. 2) Open the local.xml config file located in <magento_root>/app/etc/ folder and **add** the node <profiler>true</profiler> into the **global/resources/default_setup/connection/** <default_setup> <connection> <host>localhost</host> <username>root</username> <password>astral</password> <dbname>feedexport1810</dbname> <initStatements><![CDATA[SET NAMES utf8]]></initStatements> <model>mysql4</model> <type>pdo_mysql</type> <pdoType></pdoType> <active>1</active> <profiler>true</profiler> </connection> 3) Save the file and clear the cache.

Now if you refresh the page, you will see the **SQL Profiling** output below the code profiling results:

__EAV_LOAD_MODEL_ATTRIBUTES__	0.0012	1	85,688	0
__EAV_LOAD_MODEL_AFTER_LOAD__	0.0088	1	342,040	262,144
frontend/base/default/template/page/html/footer.phtml	0.0028	1	128,880	0
frontend/base/default/template/page/html/cookie_notice.phtml	0.0001	1	16,480	0

Executed 41
Average qu
Quer
Longest

SHOW TAB

The first thing to note is the memory usage at the top. The numbers are in bytes, so divide the number by 1048576 to get the number of megabytes of RAM used to load the page. High memory usage can affect loading speeds, especially when lots of users are loading lots of different pages.

In the table the following headers **Code Profiler**, **Cnt**, **Emalloc**, **RealMem** show every step the system takes in the loading of a page, where:

- **Code Profiler** - the name of the block of code being executed
- **Time** - The time it took to complete in seconds
- **Cnt** - Short for Count, represents the number of times this individual block ran to generate the output
- **Emalloc** - The amount of memory PHP assigned to this single operation, Its a mix of core PHP `memory_get_usage()` function without true parameter passed to it, minus timer values
- **RealMem** - The actual amount of memory used to perform the operation. Its final value is also obtained through `memory_get_usage(true)` function minus timer values but this time with true parameter passed in.

How to use information from Profiler

Let us bring some light on all this information. The execution time param is the most interesting for us. However, it's a bit tricky. If you try to make the sum of all values of the column, you won't get the actual execution time. Very often the profiler's method calls are placed one inside the other. On the screenshot above you can see that `mage::dispatch::routers_match` has the biggest execution time (**2.6948 seconds**). However, it does not mean that there's something wrong with the routers match operation in Magento. In that case, other profiler's records are placed 'inside' the routers match record. That should be kept in mind. If you go down on the list you will find the most nested process. Let's check the real example. Look at the following image:

```

frontend/base/default/template/page/html/header.phtml
frontend/base/default/template/catalogsearch/form.mini.phtml
frontend/base/default/template/page/template/links.phtml
frontend/base/default/template/page/switch/languages.phtml
frontend/base/default/template/custom_extension/cart.phtml
frontend/base/default/template/page/html/topmenu.phtml

```

Note that `frontend/base/default/template/page/html/header.phtml` execution time is quite long – **1.8251 seconds**. We can see the set of header's child templates in the profiler's list at the bottom of `frontend/base/default/template/page/html/header.phtml`. The slowest child is `frontend/base/default/template/custom_extension/cart.phtml`. Ok then, it's a custom extension in our

store which displays the shopping cart contents as a pull down menu in the header. If we disable this extension we will get the page load performance increased by more than one second.

There are two major reasons of the bottlenecks like this one:

1) The first one – it's a bad code. An extension might have a big number of unnecessary database queries or the logic is simply bad organised and, as a result, resource consuming. In that case, you can ask your developer to check the extension or try to fix it yourself.

2) The second reason is the need to get some additional information from the database or make some additional operations to have the necessary information. That is exactly what happened in our case: the header's shopping cart pull down menu extension makes the additional database requests to get the shopping cart items information. By default (when there's no additional shopping cart information on each page), Magento does not fetch this information. So, in cases like this one you should think twice: worth the header's shopping cart one second of the page loading time or not. Not always **.phtml** templates execution time reflects the bottlenecks. It's useful to pay attention to **OBSERVER** records in the profiler's report. Some extensions might execute an additional logic on the different system events. Just check what observer is the slowest and make sure that it's not Magento core observer. Usually, the developers give names to their observers by adding a vendor name, something like this: `developer_extmenu_add_topmenu_items`. So, if some custom observer slows down your system you should either disable the extension or investigate it for a reason of the bottleneck.

```
DISPATCH EVENT:controller_action_predispatch
```

```
OBSERVER: atwix_monitor_controller_action_predispatch
```

```
OBSERVER: log
```

```
CORE::create_object_of::Mage_Log_Model_Visitor
```

How to add Profiling to custom module code

To add profiling to custom module code add next lines:

```
Varien_Profiler::start('LabelForCodeFragmentToProfile'); //... here some code we like to get statistics for  
Varien_Profiler::stop('LabelForCodeFragmentToProfile');
```

If cumulative time of execution for this code is more than .0001 second it will be shown in profiler table with cumulative time, number of executions and cumulative memory consumption.

How to Translate Extension Texts and Emails

Path to Magento root folder: Select Mirasvit Extension to Translate Select Extension ... Advanced Newsletter Advanced Product Feeds Advanced Reports Advanced SEO Suite Advanced Sphinx Search Pro Asynchronous Cache Automatic Internal Links Banner Management System Fast Asynchronous Re-indexing Follow Up Email Full Page Cache Gift Registry Help Desk MX Knowledge Base Menu Manager Pro Product Labels Product Questions Promotional Offers Reward Points + Referral program RMA Search AutoComplete & Suggest Pro Search Spell-Correction Sphinx Search Ultimate Store Credit & Refund Video For Products Pro Yandex Market Export Select Desired for Translation Language Select Language and Country ... Afrikaans (Suid-Afrika) / Afrikaans (South Africa) Azrbaycan (Azrbaycan) / Azerbaijani (Azerbaijan) Bahasa Indonesia (Indonesia) / Indonesian (Indonesia) Bahasa Melayu (Malaysia) / Malay (Malaysia) Bosanski (Bosna i Hercegovina) / Bosnian (Bosnia and Herzegovina) Català (Espanya) / Catalan (Spain) Cymraeg (Y Deyrnas Unedig) / Welsh (United Kingdom) Dansk (Danmark) / Danish (Denmark) Deutsch (Deutschland) / German (Germany)

Deutsch (Schweiz) / German (Switzerland) Deutsch (Österreich) / German (Austria) Eesti (Eesti) / Estonian (Estonia) English (Australia) / English (Australia) English (Canada) / English (Canada) English (Ireland) / English (Ireland) English (New Zealand) / English (New Zealand) English (United Kingdom) / English (United Kingdom) English (United States) / English (United States) Español (Argentina) / Spanish (Argentina) Español (Chile) / Spanish (Chile) Español (Colombia) / Spanish (Colombia) Español (Costa Rica) / Spanish (Costa Rica) Español (España) / Spanish (Spain) Español (México) / Spanish (Mexico) Español (Panamá) / Spanish (Panama) Español (Perú) / Spanish (Peru) Español (Venezuela) / Spanish (Venezuela) Filipino (Pilipinas) / Filipino (Philippines) Français (Canada) / French (Canada) Français (France) / French (France) Galego (España) / Galician (Spain) Hrvatski (Hrvatska) / Croatian (Croatia) Italiano (Italia) / Italian (Italy) Italiano (Svizzera) / Italian (Switzerland) Kiswahili (Kenya) / Swahili (Kenya) Latviešu (Latvija) / Latvian (Latvia) Lietuvi (Lietuva) / Lithuanian (Lithuania) Magyar (Magyarország) / Hungarian (Hungary) Nederlands (Nederland) / Dutch (Netherlands) Norsk Bokmål (Norge) / Norwegian Bokmål (Norway) Nynorsk (Noreg) / Norwegian Nynorsk (Norway) Polski (Polska) / Polish (Poland) Português (Brasil) / Portuguese (Brazil) Português (Portugal) / Portuguese (Portugal) Român (România) / Romanian (Romania) Shqip (Shqipëri) / Albanian (Albania) Slovenina (Slovensko) / Slovak (Slovakia) Slovenščina (Slovenija) / Slovenian (Slovenia) Srpski (Srbija) / Serbian (Serbia) Suomi (Suomi) / Finnish (Finland) Svenska (Sverige) / Swedish (Sweden) Ting Vit (Vit Nam) / Vietnamese (Vietnam) Türkçe (Türkiye) / Turkish (Turkey) íslenska (Ísland) / Icelandic (Iceland) eština (eská republika) / Czech (Czech Republic) () / Greek (Greece) () / Belarusian (Belarus) () / Bulgarian (Bulgaria) () / Macedonian (Macedonia) () / Mongolian (Mongolia) () / Russian (Russia) () / Ukrainian (Ukraine) () / Hebrew (Israel) () / Arabic (Algeria) () / Arabic (Kuwait) () / Arabic (Morocco) () / Arabic (Saudi Arabia) () / Arabic (Egypt) () / Persian (Iran) () / Hindi (India) () / Bengali (Bangladesh) () / Gujarati (India) () / Thai (Thailand) () / Lao (Laos) () / Georgian (Georgia) () / Khmer (Cambodia) () / Chinese (China) () / Chinese (Hong Kong SAR China) () / Chinese (Taiwan) () / Japanese (Japan) () / Korean (South Korea)

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied `Mirasvit_{{extension_name}}.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All extension templates are named according to next pattern: `{{extension_name}} - [template_name]`.
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to **System > Configuration**. Click on the Extension at the **Mirasvit Extensions** tab.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Email Notifications** list, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied `Mirasvit_{{extension_name}}.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.

You can add all desired for translation texts of the extension into the file according next pattern:

Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "

2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All newsletter templates are named according to next pattern: Newsletter - [template_name].
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to **System > Configuration > Newsletter**.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Subscription Options**, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied `Mirasvit_{{extension_name}}.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"
"Add New Field", "Add New Field"
"Add New Gateway", "Add New Gateway"
"Add New Pattern", "Ajouter nouveau modèle"
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa
"Add New Field", "Ajouter nouveau champ"
"Add New Gateway", "Ajouter nouvelle passer
"Add New Pattern", "Ajouter nouveau modèle"
"Add New Priority", "Ajouter nouvelle prior
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All extension templates are named according to next pattern: `Gift Registry - [template_name]`.
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to **Gift registry > Settings**.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Notifications** list, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at

{{magento_path}}/app/locale/en_US/ to the folder {{magento_path}}/app/locale/{{locale_code}}/, where {{locale_code}} - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied Mirasvit_{{extension_name}}.csv file at the desired for translation language code folder: {{magento_path}}/app/locale/{{locale_code}}/ and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to Translate Extension Email Templates

On your server, copy the files located at {{magento_path}}/app/locale/en_US/template/email/mst_{{extension_name | lowercase}}/ to the folder {{magento_path}}/app/locale/{{locale_code}}/template/email/mst_{{extension_name | lowercase}}/. Open the copied email templates at the desired for translation language code folder. Replace the words and phrases into the appropriate language.

How to Translate Texts of the Extension

On your server, copy the file Mirasvit_{{extension_name}}.csv located at {{magento_path}}/app/locale/en_US/ to the folder {{magento_path}}/app/locale/{{locale_code}}/, where {{locale_code}} - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied Mirasvit_{{extension_name}}.csv file at the desired for translation language code folder: {{magento_path}}/app/locale/{{locale_code}}/ and translate the text in the **right** column. Don't change the text in the **left** column.

You can add all desired for translation texts of the extension into the file according next pattern:

Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "

2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All extension templates are named according to next pattern: `{{extension_name}} - [template_name]`.
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to `{{extension_name}} > Settings`.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Email Notification Settings** list, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

Extension **Advanced Reports** unites such products as **Advanced Dashboard** and **Reports**.

Thus, to translate the extension, you need to go to the folder

`{{magento_path}}/app/locale/{{locale_code}}/` and create next files: `Mirasvit_Advr.csv` and `Mirasvit_Advd.csv`,

where `{{locale_code}}` - is a desired for translation language code folder.

1. To translate **Reports** texts, open the created `Mirasvit_Advr.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
2. To translate **Dashboard** texts, open the created `Mirasvit_Advd.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
3. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa"  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior"
```

How to Translate Extension Email Templates

To translate emails of the **Advanced Dashboard**, follow next steps:

On your server, copy the files located at `{{magento_path}}/app/locale/en_US/template/email/mst_advd/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/template/email/mst_advd/`.

Open the copied email templates at the desired for translation language code folder. Replace the words and phrases into the appropriate language.

How to Translate Texts of the Extension

On your server, copy the file `Mirasvit_{{extension_name}}.csv` located at

`{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`,

where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. Open the copied `Mirasvit_{{extension_name}}.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right**

column. Don't change the text in the **left** column.

You can add all desired for translation texts of the extension into the file according next pattern:

Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "

2. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior
```

How to Translate Texts of the Extension

Extension **Reward Points + Referral program** unites such products as **Reward Points** and **Social Rewards** Thus, to translate the extension, you need to copy on your server files: `Mirasvit_Rewards.csv` and `Mirasvit_RewardsSocial.csv`, which are located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. To translate **Reward Points** texts, open the copied `Mirasvit_Rewards.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
2. To translate **Social Networks** texts, open the copied `Mirasvit_RewardsSocial.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
You can add all desired for translation texts of the extension into the file according next pattern:
Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "
3. Save the translated file and clear magento cache.

An example of the translated text into French:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior
```

How to Translate Extension Email Templates

In the Magento Admin Panel, go to **System > Transactional Emails**. Click on the button **Add New Template**.

In the opened window, follow next steps:

1. Select the extension email template from the list. All extension templates are named according to next pattern: `{{extension_name}} - [template_name]`.
2. In the **Locale** field, select the language.
3. Click on the **Load Template** button.
4. After loading template, will appear **Template Information** fields, where you can replace words and phrases into desired language.
5. After all changes made, specify **Template Name** and click on the **Save Template** button.

To activate this template, go to **Reward Points > Settings**.

At the top left, change your current **Configuration Scope** to the store you want to translate the email template.

At the **Notification Settings** list, select translated template for the appropriate email notification event.

How to Translate Texts of the Extension

Sphinx Search Ultimate unites such products as **Search Spell-Correction**, **Search AutoComplete & Suggestions** and **Advanced Sphinx Search Pro**.

Thus, to translate the extension, you need to copy on your server files: `Mirasvit_Misspell.csv`, `Mirasvit_SearchAutocomplete.csv` and `Mirasvit_SearchIndex.csv`, which are located at `{{magento_path}}/app/locale/en_US/` to the folder `{{magento_path}}/app/locale/{{locale_code}}/`, where `{{locale_code}}` - is a desired for translation language code folder. Follow next steps to translate the extension:

1. To translate **Search Spell-Correction** texts, open the copied `Mirasvit_Misspell.csv` file at the

desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.

2. To translate **Search AutoComplete & Suggestions** texts, open the copied `Mirasvit_SearchAutocomplete.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.
3. To translate **Advanced Sphinx Search Pro** texts, open the copied `Mirasvit_SearchIndex.csv` file at the desired for translation language code folder: `{{magento_path}}/app/locale/{{locale_code}}/` and translate the text in the **right** column. Don't change the text in the **left** column.

You can add all desired for translation texts of the extension into the file according next pattern: Both texts must be separated by **comma** and wrapped with quotation marks enclosure: "

4. Save the translated files and clear magento cache.

Example of the translated into the french language extension text:

```
"Add New Department", "Add New Department"  
"Add New Field", "Add New Field"  
"Add New Gateway", "Add New Gateway"  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Add New Priority"
```



```
"Add New Department", "Ajouter nouveau dépa  
"Add New Field", "Ajouter nouveau champ"  
"Add New Gateway", "Ajouter nouvelle passer  
"Add New Pattern", "Ajouter nouveau modèle"  
"Add New Priority", "Ajouter nouvelle prior
```

How to recover store after installation with enabled compilation

First of all, you need to disable compilation. Open the file `config.php` which is located at `<magento_path>/includes/`.

At around line 28, find next lines: `define('COMPILER_INCLUDE_PATH', dirname(__FILE__).DIRECTORY_SEPARATOR.'src');` `#define('COMPILER_COLLECT_PATH', dirname(__FILE__).DIRECTORY_SEPARATOR.'stat');` and comment them out: `#define('COMPILER_INCLUDE_PATH', dirname(__FILE__).DIRECTORY_SEPARATOR.'src');` `#define('COMPILER_COLLECT_PATH', dirname(__FILE__).DIRECTORY_SEPARATOR.'stat');` Save the changes. After compilation is disabled, store must be recovered.

If you need to enable the Compilation mode, you need to run recompilation process. Go to **System > Tools > Compilation** and click on **Run Compilation Process**.

How to fix access issues after applying magento patch SUPEE-6285

Download the Patch and copy the file to the Magento Root folder on your server.
To apply this patch, use one of these methods:

1. Apply the patch via Browser

Open the patch file in your browser. For example, http://yoursite.com/m_controllers_patch.php

2. Apply the patch via Command Line

To run the script via command line (shell, SSH) you need to execute next command: `php {{magento_path}}/m_controllers_patch.php` If you receive an error message while executing the patch: **"Error updating the file. Please check file permissions, it should have write access"**, you need to set write access for the file and its path. Run following command in the command line to set full access: `sudo chmod 777 -R app/code/local/Mirasvit/ModuleName/controllers/Adminhtml/` where `ModuleName` is a name of the Mirasvit extension. If you don't have access to command line, you can set the appropriate write access in the **Filezilla** or ask your system administrator to do this.

How to fix access issues after applying magento patch SUPEE-6788

Download the Patch for SUPEE-6788 and copy the file to the `{{magento_path}}/` folder on your server. Before applying the patch, make backup of the folder `{{magento_path}}/app/` To apply this patch, use one of these methods:

1. Apply the patch via Command Line (Preferable)

To run the script via command line (shell, SSH) you need to execute next command: `php {{magento_path}}/fixsupee6788.php` where `{{magento_path}}` - path to Magento Root folder on your server.

2. Apply the patch via Browser

Execute the patch in your browser. For example, <http://yoursite.com/fixsupee6788.php> If you receive an error message while executing the patch: **"Unable to write new configuration to ..."**, you need to set write access for the file and its path.

Run following command in the command line to set full access: `sudo chmod 777 -R <file_path>/` After

script executing, patch will be applied to all **Mirasvit** extensions.
For security reasons, delete the executed patch from the server.