

Script Authors

How to Add Your Own Scripts to Fantastico F3 Locally

Adding your own scripts (or a new flavor of an existing script) to **Fantastico F3** is very easy. It requires only two steps.

Step 1: Prepare your script file. As mentioned above, you can add your own script or a new flavor of an existing script.

You can use the following as references. They show fully annotated source codes of 10 existing scripts.

- <http://174.120.165.106/Chive.html>
- <http://174.120.165.106/Dolibarr.html>
- <http://174.120.165.106/Drupal7.html>
- <http://174.120.165.106/iWare.html>
- <http://174.120.165.106/limny.html>
- <http://174.120.165.106/OmegaBB.html>
- <http://174.120.165.106/PivotX.html>
- <http://174.120.165.106/REDAXOCMS.html>
- <http://174.120.165.106/Redaxscript.html>
- <http://174.120.165.106/WordPress.html>

Step 2: Copy your script file to **`/var/netenberg/fantastico_f3/sources/vendors/Netenberg/Script.`**

That's it.

You can now fire up your browser and open **Fantastico F3** and see your new script, which is fully ready to be installed (no additional steps will be required).

Highlights:

- Need an older version? Simply copy the existing script and change the download URL.
- Need to set a non-default language during installation? Simply copy the existing script and change the default installation parameters (if available).
- Need to override requirements? Simply copy the existing script and override `getRequirements()`.

The possibilities are endless because we offer the full source code. No part of the script files are encoded or encrypted or obfuscated in anyway. Our scripts API is 100% transparent.

Script Authors

Walkthrough

Let us assume that we have a script named **Rambo** which can be downloaded using the following URL: <http://rambo.com/latest.zip>.

We will start with a skeleton.

```
// Rambo.php

<?php

class Netenberg_Script_Rambo extends Netenberg_Script
{
    public function install($parameters)
    {
    }

    public function getCategory()
    {
    }

    public function getDescription()
    {
    }

    public function getDetails($parameters)
    {
    }

    public function getForm()
    {
    }

    public function getImage()
    {
    }

    public function getName()
    {
    }

    public function getItems($parameters)
    {
    }

    public function getRequirements()
    {
    }
}
```

Script Authors

```
public function getSize()  
{  
}  
  
public function getSlug()  
{  
}  
  
public function getTimestamp()  
{  
}  
  
public function getUrls()  
{  
}  
  
public function getVersion()  
{  
}  
  
public function uninstall($parameters)  
{  
}  
}
```

These are the mandatory functions required within a script.

We will now formulate the basic functions one by one.

```
public function getCategory()  
{  
    return _('Blogs'); // our script is a blog, so we place it in the  
    Blogs category  
}
```

```
public function getDescription()  
{  
    return _('Rambo is the best blog script in the world.');
```

Script Authors

```
public function getDetails($parameters)
{
    $document_root_directory = array();
    if ($parameters['document_root']) {
        $document_root_directory[] = $parameters['document_root'];
    }
    if ($parameters['directory']) {
        $document_root_directory[] = $parameters['directory'];
    }
    $document_root_directory = implode('/', $document_root_directory);

    // We are extracting the version number from an installed copy of
    our script.
    $version_php = sprintf('%s/version.php', $document_root_directory)
;
    if (!is_file($version_php)) {
        return false;
    }
    $contents = file_get_contents($version_php);
    preg_match('#version\s*=\s*'([\d+\.]+'#', $contents, $version);

    return array(
        'version' => $version[1],
    );
}
```

```
// We will only require domain and directory from the end-user.
// We do not require any ther user-supplied information.
```

```
public function getForm()
{
    $control_panel = Zend_Registry::get('control_panel');
    $form = new Netenberg_Form();
    $form->addElement('select', 'domain', array(
        'label' => _('Domain'),
        'multiOptions' => $control_panel->getDomains(),
        'required' => true,
    ));
    $form->addElement('text', 'directory', array(
        'description' => _('Leave this field empty if you want to inst
all in the web root for the domain you've selected (i.e., http://domai
n.com/ ). If you'd like to install in a subdirectory, please enter the
path to the directory relative to the web root for your domain. The f
inal destination subdirectory should not exist, but all others can exi
st (e.g., http://domain.com/some/sub/directory - In this case, "direct
ory" should not already exist).'),
    ));
}
```

Script Authors

```
'filters' => array(new Netenberg_Filter_Directory()),
'label' => _('Directory'),
'validators' => array(new Netenberg_Validate_Directory()),
));
$form->addDisplayGroup(
    array('domain', 'directory'),
    'location_details',
    array(
        'decorators' => $form->getDefaultGroupDecorator(),
        'disableLoadDefaultDecorators' => true,
        'legend' => 'Location Details',
    )
);
$form->addDisplayGroup(
    array('submit', 'reset'),
    'buttons',
    array(
        'decorators' => $form->getButtonGroupDecorator(),
        'disableLoadDefaultDecorators' => true,
    )
);

return $form;
}
```

```
public function getImage()
{
    return 'https://rambo.com/logo.png';
}
```

```
public function getName()
{
    return 'Rambo';
}
```

```
public function getItems($parameters)
{
    $domain_directory = array();
    if ($parameters['domain']) {
        $domain_directory[] = $parameters['domain'];
    }
}
```

Script Authors

```
}
if ($parameters['directory']) {
    $domain_directory[] = $parameters['directory'];
}
$domain_directory = implode('/', $domain_directory);

// This is what the end-user will see after a successful install.
return array(
    _('Frontend') => array(
        sprintf(
            '<a href="http://%s" target="_blank">http://%s</a>',
            $domain_directory,
            $domain_directory
        ),
    ),
);
}
```

```
public function getRequirements()
{
    $control_panel = Zend_Registry::get('control_panel');
    $apache = $control_panel->getApache();
    $mysql = $control_panel->getMysql();
    $php = $control_panel->getPhp();

    // We will only have Disk Space requirements.
    return array(
        'Disk Space' => $control_panel->getSize() >= $this->getSize()?
true: false,
    );
}
```

```
public function getSize()
{
    return 1000000; // in bytes
}
```

```
public function getSlug()
{
    return 'word-press';
}
```

Script Authors

```
}

// This is the timestamp of the most recent release of our script.

public function getTimestamp()
{
    return '2016-01-01 01:00:00';
}

public function getUrls()
{
    return array(
        _('Home') => 'http://rambo.com',
        _('Documentation') => 'http://rambo.com/documentation',
        _('Support') => 'http://rambo.com/support',
    );
}

public function getVersion()
{
    return '1.0.0';
}
```

We will now formulate the install()and uninstall()functions.

```
public function install($parameters)
{
    $control_panel = Zend_Registry::get('control_panel');
    $operating_system = Zend_Registry::get('operating_system');

    $curl = new Netenberg_cURL;

    $step = 0;

    $document_root_directory = array();
    if ($parameters['document_root']) {
        $document_root_directory[] = $parameters['document_root'];
    }
}
```

Script Authors

```
}
if ($parameters['directory']) {
    $document_root_directory[] = $parameters['directory'];
}
$document_root_directory = implode('/', $document_root_directory);

$domain_directory = array();
if ($parameters['domain']) {
    $domain_directory[] = $parameters['domain'];
}
if ($parameters['directory']) {
    $domain_directory[] = $parameters['directory'];
}
$domain_directory = implode('/', $domain_directory);

// Step 1: We ask Fantastico F3 to create a new MySQL database and
user.
log_('DEBUG', sprintf(_('Step %d'), ++$step));
list(
    $parameters['mysql_hostname'],
    $parameters['mysql_username'],
    $parameters['mysql_password'],
    $parameters['mysql_database']
) = $control_panel->insertMysql();

// Step 2: We ask Fantastico F3 to download and extract our script
to the user-supplied directory.
log_('DEBUG', sprintf(_('Step %d'), ++$step));
$operating_system->transpose(
    'http://rambo.com/latest.zip',
    array(
        'rambo/*' => sprintf('%s', $document_root_directory),
    )
);

// Step 3: We write the MySQL-
specific information into the config.php file.
$config_php = sprintf('%s/wp-
config.php', $document_root_directory);
$content = file_get_contents($config_php);
$content = str_replace('{{ localhost }}', $parameters['mysql_host
name'], $content);
$content = str_replace('{{ database_name_here }}', $parameters['m
ysql_database'], $content);
$content = str_replace('{{ username_here }}', $parameters['mysql_
username'], $content);
$content = str_replace('{{ password_here }}', $parameters['mysql
_password'], $content);
file_put_contents($config_php, $content);
```


Script Authors

```
// Step 4: We invoke the install.php file.
log_('DEBUG', sprintf(_('Step %d'), ++$step));
list($output, $return_var) = $curl->request(
    sprintf('http://%s/install.php', $domain_directory),
    'GET',
    array(),
    array(),
    array()
);
if (strpos($output[1], 'Rambo has been installed successfully.') !
== false) {
    log_('DEBUG', 'Success');

    return parent::install($parameters);
}
log_('DEBUG', 'Failure');

return false;
}
```

```
public function uninstall($parameters)
{
    $control_panel = Zend_Registry::get('control_panel');
    $operating_system = Zend_Registry::get('operating_system');

    $document_root_directory = array();
    if ($parameters['document_root']) {
        $document_root_directory[] = $parameters['document_root'];
    }
    if ($parameters['directory']) {
        $document_root_directory[] = $parameters['directory'];
    }
    $document_root_directory = implode('/', $document_root_directory);

    // Step 1: We read the MySQL-
specific information from the config.php file.
    $config_php = sprintf('%s/wp-
config.php', $document_root_directory);
    if (!is_file($config_php)) {
        return false;
    }
    $contents = file_get_contents($config_php);
    preg_match('#DB_NAME', \s*'([\^']*)#', $contents, $database);
    preg_match('#DB_USER', \s*'([\^']*)#', $contents, $mysql_username);
}
```

Script Authors

```
// Step 2: We delete the MySQL database and user.
$control_panel->deleteMysql($mysql_username[1], $database[1]);

// Step 3: We ask Fantastico F3 to remove the directory and files.
$operating_system->dispose($document_root_directory);

return parent::uninstall($parameters);
}
```

Now that we have formulated all the functions, we can assemble the complete file.

```
<?php

class Netenberg_Script_Rambo extends Netenberg_Script
{
    public function install($parameters)
    {
        $control_panel = Zend_Registry::get('control_panel');
        $operating_system = Zend_Registry::get('operating_system');

        $curl = new Netenberg_cURL;

        $step = 0;

        $document_root_directory = array();
        if ($parameters['document_root']) {
            $document_root_directory[] = $parameters['document_root'];
        }
        if ($parameters['directory']) {
            $document_root_directory[] = $parameters['directory'];
        }
        $document_root_directory = implode('/', $document_root_directory);

        $domain_directory = array();
        if ($parameters['domain']) {
            $domain_directory[] = $parameters['domain'];
        }
        if ($parameters['directory']) {
            $domain_directory[] = $parameters['directory'];
        }
        $domain_directory = implode('/', $domain_directory);

        log_('DEBUG', sprintf(_('Step %d'), ++$step));
        list(
```

Script Authors

```
        $parameters['mysql_hostname'],
        $parameters['mysql_username'],
        $parameters['mysql_password'],
        $parameters['mysql_database']
    ) = $control_panel->insertMysql();

    log_('DEBUG', sprintf(_('Step %d'), ++$step));
    $operating_system->transpose(
        'http://rambo.com/latest.zip',
        array(
            'rambo/*' => sprintf('%s', $document_root_directory),
        )
    );

    $config_php = sprintf('%s/wp-
config.php', $document_root_directory);
    $contents = file_get_contents($config_php);
    $contents = str_replace('{{ localhost }}', $parameters['mysql_
hostname'], $contents);
    $contents = str_replace('{{ database_name_here }}', $parameter
s['mysql_database'], $contents);
    $contents = str_replace('{{ username_here }}', $parameters['my
sql_username'], $contents);
    $contents = str_replace('{{ password_here }}', $parameters['m
ysql_password'], $contents);
    file_put_contents($config_php, $contents);

    log_('DEBUG', sprintf(_('Step %d'), ++$step));
    list($output, $return_var) = $curl->request(
        sprintf('http://%s/install.php', $domain_directory),
        'GET',
        array(),
        array(),
        array()
    );
    if (strpos($output[1], 'Rambo has been installed successfully.
') !== false) {
        log_('DEBUG', 'Success');

        return parent::install($parameters);
    }
    log_('DEBUG', 'Failure');

    return false;
}

public function uninstall($parameters)
{
    $control_panel = Zend_Registry::get('control_panel');
```

Script Authors

```
$operating_system = Zend_Registry::get('operating_system');

$document_root_directory = array();
if ($parameters['document_root']) {
    $document_root_directory[] = $parameters['document_root'];
}
if ($parameters['directory']) {
    $document_root_directory[] = $parameters['directory'];
}
$document_root_directory = implode('/', $document_root_directory);

$config_php = sprintf('%s/wp-config.php', $document_root_directory);
if (!is_file($config_php)) {
    return false;
}
$content = file_get_contents($config_php);
preg_match('#DB_NAME', $content, $database);
preg_match('#DB_USER', $content, $mysql_username);

$control_panel->deleteMysql($mysql_username[1], $database[1]);

$operating_system->dispose($document_root_directory);

return parent::uninstall($parameters);
}

public function getCategory()
{
    return _('Blogs');
}

public function getDescription()
{
    return _('Rambo is the best blog script in the world.');
```

```
public function getDetails($parameters)
{
    $document_root_directory = array();
    if ($parameters['document_root']) {
        $document_root_directory[] = $parameters['document_root'];
    }
    if ($parameters['directory']) {
        $document_root_directory[] = $parameters['directory'];
    }
    $document_root_directory = implode('/', $document_root_directory);
}
```

Script Authors

```
ry);

    $version_php = sprintf('%s/version.php', $document_root_directory);
    if (!is_file($version_php)) {
        return false;
    }
    $contents = file_get_contents($version_php);
    preg_match('#version\s*=\s*'([\d+\.]+'#', $contents, $version);

    return array(
        'version' => $version[1],
    );
}

public function getForm()
{
    $control_panel = Zend_Registry::get('control_panel');
    $form = new Netenberg_Form();
    $form->addElement('select', 'domain', array(
        'label' => _('Domain'),
        'multiOptions' => $control_panel->getDomains(),
        'required' => true,
    ));
    $form->addElement('text', 'directory', array(
        'description' => _('Leave this field empty if you want to install in the web root for the domain you've selected (i.e., http://domain.com/ ). If you'd like to install in a subdirectory, please enter the path to the directory relative to the web root for your domain. The final destination subdirectory should not exist, but all others can exist (e.g., http://domain.com/some/sub/directory - In this case, "directory" should not already exist).'),
        'filters' => array(new Netenberg_Filter_Directory()),
        'label' => _('Directory'),
        'validators' => array(new Netenberg_Validate_Directory()),
    ));
    $form->addDisplayGroup(
        array('domain', 'directory'),
        'location_details',
        array(
            'decorators' => $form->getDefaultGroupDecorator(),
            'disableLoadDefaultDecorators' => true,
            'legend' => 'Location Details',
        )
    );
    $form->addDisplayGroup(
        array('submit', 'reset'),
        'buttons',
```

Script Authors

```
        array(
            'decorators' => $form->getButtonGroupDecorator(),
            'disableLoadDefaultDecorators' => true,
        )
    );

    return $form;
}

public function getImage()
{
    return 'https://rambo.com/logo.png';
}

public function getName()
{
    return 'Rambo';
}

public function getItems($parameters)
{
    $domain_directory = array();
    if ($parameters['domain']) {
        $domain_directory[] = $parameters['domain'];
    }
    if ($parameters['directory']) {
        $domain_directory[] = $parameters['directory'];
    }
    $domain_directory = implode('/', $domain_directory);

    return array(
        _('Frontend') => array(
            sprintf(
                '<a href="http://%s" target="_blank">http://%s</a>'
                ,
                $domain_directory,
                $domain_directory
            ),
        ),
    );
}

public function getRequirements()
{
    $control_panel = Zend_Registry::get('control_panel');
    $apache = $control_panel->getApache();
    $mysql = $control_panel->getMysql();
    $php = $control_panel->getPhp();
}
```

Script Authors

```
        return array(
            'Disk Space' => $control_panel->getSize() >= $this->getSiz
e()? true: false,
        );
    }

    public function getSize()
    {
        return 1000000;
    }

    public function getSlug()
    {
        return 'word-press';
    }

    public function getTimestamp()
    {
        return '2016-01-01 01:00:00';
    }

    public function getUrls()
    {
        return array(
            _('Home') => 'http://rambo.com',
            _('Documentation') => 'http://rambo.com/documentation',
            _('Support') => 'http://rambo.com/support',
        );
    }

    public function getVersion()
    {
        return '1.0.0';
    }
}
```

Unique solution ID: #1020
Author: Netenberg Staff
Last update: 2016-11-30 02:50