

Sysadmin HOWTOs

From Matchi Wiki

Contents

- 1 HOWTO Reboot a Server
 - 1.1 When do we need to do this?
 - 1.2 Overview
 - 1.3 Reboot via a SSH console session
 - 1.4 Reboot via the Virtuozzo Power Panel
 - 1.5 Reboot via the Plesk Control Panel
 - 1.6 Reboot via the Serial Console
 - 1.7 Reboot via the Hosting Administration Panel
- 2 HOWTO Upgrade a server's kernel
 - 2.1 When do we need to do this?
 - 2.2 Overview
- 3 HOWTO Refresh a technical environment
 - 3.1 When do we need to do this?
 - 3.2 Overview
 - 3.3 Refresh an environment
 - 3.3.1 Backup the existing target environment's application
 - 3.3.2 Backup the existing target environment's database
 - 3.3.3 Copy the Source environment's application to the Target environment
 - 3.3.3.1 Configure the new environment
 - 3.3.4 Set up the database for the new environment
 - 3.3.4.1 Dump the production database to a file
 - 3.3.4.2 Create a new database instance
 - 3.3.4.3 Load the database dump into the new instance
 - 3.3.5 Database Changes for non-production environments
 - 3.3.5.1 Anonymize all external users' email addresses
 - 3.3.5.2 Add [ENVIRONMENT] prefix to all notification email subjects
 - 3.3.5.3 Un-Hide hidden Innovations and Organizations
 - 3.3.5.4 Change System URL for the non-production environment
 - 3.3.5.5 Change Email Notifications for the non-production environment
 - 3.3.5.6 Set all Email Notifications Groups to the Test Group on the non-production environment
 - 3.3.6 Set the Apache configuration for the environment
 - 3.3.7 Set the Environment Logo in the Template
- 4 HOWTO Server Maintenance
 - 4.1 Keeping the server's patch level up to date
 - 4.1.1 Update application packages
 - 4.1.2 Updating the RKHUNTER file hashes
 - 4.2 Update the Linux Kernel
 - 4.2.1 What kernel am I running?
 - 4.2.2 What Linux Distribution am I running?
- 5 HOWTO Joomla Framework and Component Maintenance
 - 5.1 Installation of New Third-Party Joomla Components
 - 5.1.1 Pre-Production Testing
 - 5.2 Upgrading Third-Party Joomla Components
 - 5.2.1 Pre-Production Testing
 - 5.3 Upgrading Third-Party Joomla Components when upgrading the Joomla Core Framework
 - 5.3.1 Decision Criteria
 - 5.3.2 Pre-Production Testing

- 5.4 Upgrading Components and the Joomla Core Framework
 - 5.4.1 Decision Criteria
 - 5.4.2 Pre-Production Testing
- 5.5 Special Security Step for all Upgrades and new Installations

HOWTO Reboot a Server

When do we need to do this?

You would only ever need to reboot a server in the following situations:

- When a server has run out of memory due to memory leakage which is caused by poor code
- Too many open database sessions (attributable to poor code - you know who you are!)
- Upgrading the kernel due to security issues.

Overview

There are a number of ways to reboot a server. Depending on the state that the servers is in, e.g. it may suffer from a massive memory haemorrhage, not all methods to initiate a server reboot are possible:

- SSH session: Issue the `sudo reboot` command
- Virtuozzo Power Panel. This is only available on virtual servers.
- Plesk Control Panel. This is available on all types of servers that have Plesk is installed
- Serial Console. This option is only available for physical servers
- Hosting Administration Panel. This option is only available for physical servers

The duration of a server reboot and resumption of full service

- For a virtual server is about 60 seconds
- For a physical server is about 150 seconds

Note

If a server reboot takes much longer, you should inspect the server logs and determine why the reboot has taken so long. And obviously remedy the issue.

Reboot via a SSH console session

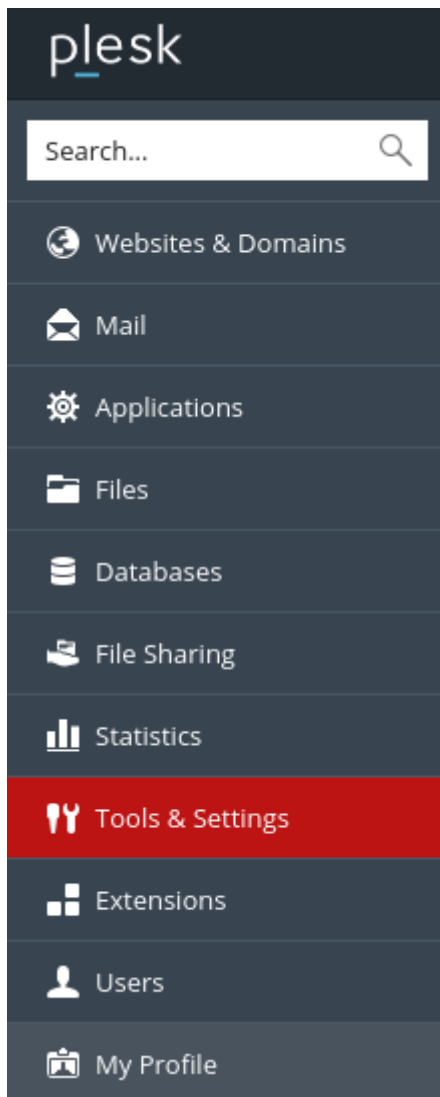
```
$
```

Reboot via the Virtuozzo Power Panel

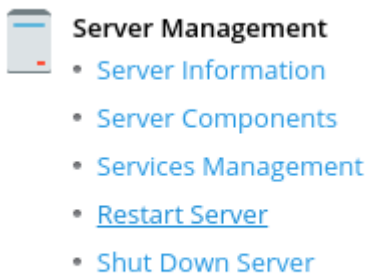
Go to the Virtuozzo Power Panel appliance through this URL: [https://\[server's IPv4 address\]/vz/cp](https://[server's IPv4 address]/vz/cp). Log in as user *root* and the Server Initial Password (see Infrastructure section for details).

Reboot via the Plesk Control Panel

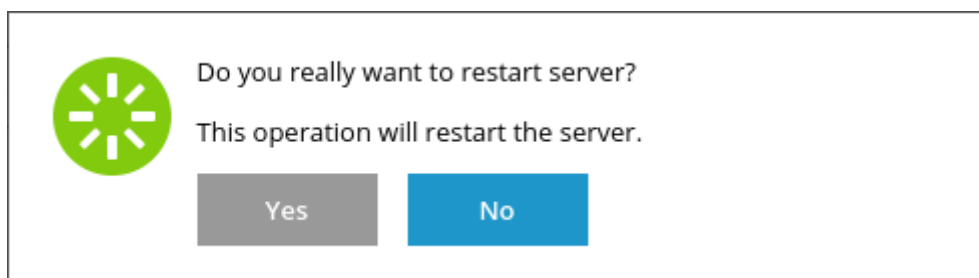
- Log into the Server's Plesk Control Panel and select the "Tools and Settings" option:



- In the "Server Management" box, select the "Restart Server" link:



- Confirm the operation by clicking on Yes:



- The browser window should immediately clear with an error message (depending on your browser). Be sure to check that server has come up again by trying to reestablish the Plesk Control Panel session.

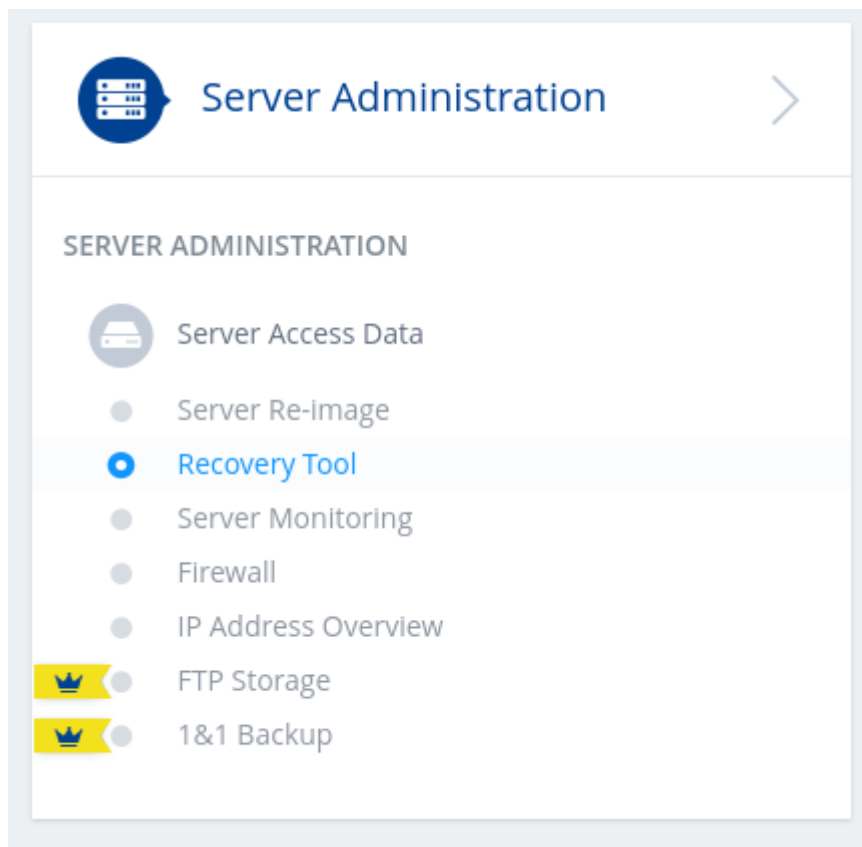
Reboot via the Serial Console

\$

Reboot via the Hosting Administration Panel

This method has some flaws, as it can not tell you whether the server was even running at all, or if the reboot was successful.

- Go to <https://admin.1and1.co.uk> and log in using the server's hosting account details.
- Select the contract that relates to the server in question. Find the "Server Administration" block and select "Recovery Tool":



- Make the selections as below:

Recovery Tool

Reboot your server from the hard drive or using an emergency system.

Please note: Improper usage can lead to loss of data or not being able to reach your server. No guarantees can be provided that your data or server will function as expected.



Warning!

- Please be aware that using this service is at your own risk. Improper use of this service can lead to loss of data or block access to your server.

Recovery Status

Boot mode : Normal system

Status : *Ready.*

Recovery Action

Reboot now : ☒

Boot in : ☒ Normal system
☐ Linux 64-bit Rescue System (debian/oldstable)
☐ Linux 64-bit Rescue System (debian/stable)
☐ Linux 64-bit Rescue System (debian/testing)

Cancel

Next

- Confirm your action:

Confirm Action

Please note: Improper usage can lead to loss of data or not being able to reach your server. No guarantees can be provided that your data or server will function as expected.

Boot in : *Normal system*

Reboot now : *Yes*

Cancel

OK

- Wait for a confirmation:

Process Successful

The process was successful.

Your server will be available again in about 10 minutes.

Please note that in the event of a reset, your server will not be available until the process is finished.

[Go To Overview](#)

HOWTO Upgrade a server's kernel

When do we need to do this?

As a guide, security improvements are main reason for upgrading a kernel. It is unlikely that you would need to upgrade the kernel due to new features in the kernel or promises of performance improvements.

You can only upgrade the kernel of a physical server. You can not upgrade the kernel of a virtual server yourself. If the kernel version of the virtual server is not satisfactory, then move to a physical server or wait for the hosting provider to eventually upgrade the virtualizer's operating system.

Overview

In some cases after a kernel upgrade, it may be necessary to upgrade packages that are affected by a kernel upgrade.

HOWTO Refresh a technical environment

When do we need to do this?

This is only necessary when a non-production environment needs to be aligned with the production environment in order to create a baseline for a new tranche of development or testing.

You should not ever need to refresh the production environment from a non-production environment, as you will lose recent data changes and potentially cause a production outage while doing this.

Overview

The following environments are accessible from the public internet - albeit that all but the PRODUCTION environment are password protected:

- **PROD** - Production. Visible to the public. <https://matchi.biz>
- **DEV** - Development. Password protected. <http://dev.matchi.biz>
- **TEST** - Test environment. Password protected. <http://test.matchi.biz>
- **MERGE** - Code merge environment. Password protected. <http://merge.matchi.biz>
- **STARTUP** - Code merge environment. Password protected. <http://startup.matchi.biz>
- **REPORT** - Report Development environment. Password protected. <http://report.matchi.biz>
- **PERF** - Performance Testing environment. Password protected and on a dedicated server. <http://perf.matchifranchizing.biz>

Things you should know before proceeding:

- Most of these environments are completely backed up every night.

- The subdomains have previously been created on the hosting provider's administrative interface
- The non-production environments are hardly used.
- Most environments are on the same server instance (for the time being), excepts for the performance testing environment, which necessarily needs to be on a dedicated server.
- The environments are based off the directory `/var/www/vhosts/s16972616.onlinehome-server.info` on server `mweb01`, and

`/var/www/vhosts/s17502230.onlinehome-server.info` on server `mfran01`.

- The performance environment is on a dedicated server and on a separate domain, so that it can benefit from Cloudflare's Content Delivery Network (CDN) features.
- The refresh date of the environment is always obvious from the name as it appears on a browser, and is in the form `[ENVIRONMENT [YYYYMMDD]`, e.g. **TEST 20151114**.

Refresh an environment

It is unlikely that you would ever need to refresh the production environment from another environment. It you ever feel that this is required, think again!

The steps to refresh a target environment from a source environment are done on the web server (`mweb01`), followed by a number of operations that are performed on the database server (`mdb01`).

In this example the source environment is the PRODUCTION environment, and the target environment is one called TEST.

Backup the existing target environment's application

On the web server (`mwebXX`):

Take a current snapshot of the target environment's application in case you need to revert back to it it. In the example here, the backup file is held in the HOME directory of the currently logged-in user and is a BZip-version-2 compressed tape archive, i.e. a `.tar.bz2` file. It is useful to include a datestamp in the name of the backup file, in the form `YYYYMMDD`.

There are the available application directories:

```
$ cd /var/www/vhosts/s16972616.onlinehome-server.info
$ ls
demo.matchi.biz
dev.matchi.biz
launch.matchi.biz
matchi.org
merge.matchi.biz
report.matchi.biz
startup.matchi.biz
test.matchi.biz
...
```

Create a snapshot of the TEST-environment's application:

```
$ tar -cjf ~/test.matchi.biz.YYYYMMDD.tar.bz2 test.matchi.biz
```

Backup the existing target environment's database

On the database server (called `mdbXX`):

Take a current snapshot of the target environment's database in case you need to revert back to it. In the example here, the backup file is held in the HOME directory of the currently logged-in user and is a BZip-version-2 compressed SQL dump, i.e. a `.sql.bz2` file. A date stamp is included in the name of the backup file in the ISO date format of YYYYMMDD.

List all the databases:

```
$ mysql -e 'show databases'
```

Database
information_schema
build_20140119
build_20140325
build_20140507
build_20140725
build_20150129
build_20150731
build_20151105
demo_20150929
dev_20150129
dev_20150227
dev_20150316
dev_20150414
dev_20150623
dev_20150804
launch_20130906
merge_20140702
merge_20151029
multi_20141012
mysql
perf_20150216
perf_20151110
startup_20140512
startup_20150129
startup_20150825
test_20150312
test_20150316
test_20150320
etc

Create a backup of the existing target environment's database, say, `test_20150316`, to `test_20150320_YYYYMMDD.sql.bz2`, where YYYYMMDD is today's date:

```
$ mysqldump --routines test_20150320 | bzip2 > test_20150320_YYYYMMDD.sql.bz2
```

Copy the Source environment's application to the Target environment

*Remember that it is safer to first go to the target directory and to **pull** the content of the source directory into it.*

First, clear all but the configuration files out of the target directory:

```
$ cd /var/www/vhosts/s16972616.onlinehome-server.info/test.matchi.biz
```

Confirm that you are indeed in the target directory before proceeding:

```
$ pwd
/var/www/vhosts/s16972616.onlinehome-server.info/test.matchi.biz
```

So you are sure that you are in the correct directory? If so, remove all the directories from `/var/www/vhosts/s16972616.onlinehome-server.info/test.matchi.biz`


```
$ find . -type d -exec sudo rm -fr {} 2>/dev/null \;
```

Copy the source content from the PRODUCTION environment to 'here'. You should remain in the `var/www/vhosts/s16972616.onlinehome-server.info/test.matchi.biz` directory and *pull* the content from the source directory into the current target directory:

```
$ sudo cp -r ../matchi.biz/* .  
[sudo] password for madman:
```

Configure the new environment

There should be a pre-built configuration file for this environment ready, what needs to be altered slightly and copied to the proper name. This file is called `configuration.php.ENVIRONMENT`, where `ENVIRONMENT` is the environment name, e.g. `test`, `dev`, etc... In the case, we have the file `configuration.php.test` and edit the following changes in it:

- Change the name of the database - by convention, this holds today's date
- Change the site name - by convention, this holds today's date in ISO form `YYYYMMDD`

```
$ sudo nano configuration.php.test  
...  
public $sitename = 'TEST 20151129';  
...  
public $db = 'test_20151129';  
...
```

Set the originator of emails sent from the website as follows:

```
public $mailfrom = 'test.support@matchi.biz';  
public $fromname = 'Test Matchi.Biz Support';  
public $sendmail = '/usr/sbin/sendmail';  
public $smtpauth = '1';  
public $smtpuser = 'support@matchi.biz';
```

A non-production environment is likely to not have SSL (The exception to this may be for a pre-production environment). Set the SSL flag to 0:

```
...  
public $force_ssl = 0;  
...
```

Also confirm that you have the correct *tmp* and *logs* directories set up:

```
...  
public $log_path = '/var/www/vhosts/[SERVERNAME].onlinehome-server.info/[ENVIRONMENT]/logs';  
public $tmp_path = '/var/www/vhosts/[SERVERNAME].onlinehome-server.info/[ENVIRONMENT]/tmp';  
...
```

To recap:

- `[SERVERNAME]` is the server Id that allocated by the hosting provider
- `[ENVIRONMENT]` is the name of the environment that you are busy creating

Some file manipulations are required to complete the system-level configuration:

- Advise search engines to not search in this environment. `robots.txt` should just contain the following:

```
User-agent: *  
Disallow: /
```

- Set the access control for all the files

```
$ sudo chown -R plesk:psaserv *
```

Save the changes to configuration.php: `Ctrl O`, `Ctrl X`

Set up the database for the new environment

This is performed on the database server (mdb01).

Dump the production database to a file

The production database is named build_YYYYMMDD, where YYYYMMDD is the date the database was instantiated for the first time, build_20151105. Do a complete database dump, including all stored procedures, to a dump file, build_20151105_20151109.sql, where the second date is today's date.

```
$ mysqldump --routines build_20151105 > build_20151105_20151129.sql
```

Create a new database instance

The target database is named ENVIRONMENT_YYYYMMDD, where YYYYMMDD is today's date.

```
mysql -e 'create database test_20151109'
```

Load the database dump into the new instance

Note the `-c` option when importing the data as this prevents the comments in the stored procedures from getting stripped out.

```
$ mysql -c test_20151109 < build_20151105_20151109.sql
```

Database Changes for non-production environments

The following changes need to be made to prevent both operations staff and external user from receiving unwanted notifications, and for the end-to-end system behaviour to be consistent with the environment that it is in.

Anonymize all external users' email addresses

This prevents accidental emails getting sent to real users. System users mostly have a user_id < 900. Members who belong to the testing group (Group Id 22) should not have their email addresses mangled.

```
set @ENVIRONMENT='test';  
update mtchi_users  
  set email = concat(substr(md5(id),1,10), '@[ENVIRONMENT].matchi.biz')  
  where id >= 900  
     and id not in (select user_id  
                   from mtchi_user_usergroup_map
```

```
) where group_id = 22
```

Replace '[ENVIRONMENT]' with the environment name so that non-production emails can be traced to their origin.

The result is that you should have email addresses like this:

```
mysql> select email from mtchi_users where id > 900 limit 10;
d045c59a90@matchi.biz
043c3d7e48@matchi.biz
4daa3db355@matchi.biz
e820a45f1d@matchi.biz
90794e3b05@matchi.biz
b7892fb3c2@matchi.biz
74bba22728@matchi.biz
5c936263f3@matchi.biz
1ce927f875@matchi.biz
8065d07da4@matchi.biz
```

Add [ENVIRONMENT] prefix to all notification email subjects

To prevent any confusion when emails arrive from a non-production environment in a developer's and tester's mail client (and the emails should not arrive on anyone else's mail client, BTW), the email's subject line should be prefixed with the environment's name.

For example, when preparing a test environment, do this:

```
set @ENVIRONMENT='test';
update mtchi_inno_system_picklist_event_types
set subject = concat('[',upper(left(@ENVIRONMENT,1)),lower(substring(@ENVIRONMENT,2)),'] ',subject);
```

All system-emitted emails will now be prefixed with "[Test] ".

Warning

Remember that these records can't be worked back into the production environment again, now that they permanently are tainted.

Un-Hide hidden Innovations and Organizations

Some test innovations and test organization exist on the production system but are marked as 'hidden' so that they do not appear in any real searches to actual external users. These 'hidden' innovations and organizations are useful for a test environment as it can save a tester from creating a Test Innovation or Test Organizations, so can be 'un-hidden'. The 'is_hidden' flag needs to be reset on all innovations and organisations in the test environment:

```
mysql> update mtchi_inno set is_hidden = 0 where is_hidden = 1;
mysql> update mtchi_inno_organizations set is_hidden = 0 where is_hidden = 1;
```

Change System URL for the non-production environment

Set the URLs that are contained in system-generated notifications so that they point to the non-production environments and exhibit the correct behaviour when tested:

```
mysql> update mtchi_inno_config set value = 'http://test.matchi.biz' where param = 'SystemURL';
```

Change Email Notifications for the non-production environment

In any of the non-production environments, copies of system-generated notifications need to be sent to the test email account instead of Zoho.

```
mysql> update mtchi_inno_config set value = 'test@matchi.biz' where param = 'NotificationDispatchOpsRecipients';
```

Set all Email Notifications Groups to the Test Group on the non-production environment

In any of the non-production environments, system-generated notifications that are sent to user groups need to be sent to only members of the Test user group, Id=22.

```
mysql> update mtchi_inno_system_picklist_event_types set notificant_groups = 22 where notificant_groups is not null;
```

Set the Apache configuration for the environment

You should not need to make any changes, but if you have to, such as setting different password criteria, make the changes here:

```
$ sudo nano /etc/httpd/conf/httpd.conf
```

Get Apache to re-read the configuration without having to restart the service:

```
$ sudo /etc/init.d/httpd graceful
```

Set the Environment Logo in the Template

To avoid any further ambiguity, it helps to also indicate in the site what environment it is. This is done:

- The site name is set in the configuration.php file - see previous steps.
- A modified Matchi logo is selected from the template configuration, as follows:

In the administrator panel in [http://\[ENVIRONMENT\].matchi.biz/administrator](http://[ENVIRONMENT].matchi.biz/administrator), go to Extensions -> Templates -> Matchi Template -> Features, and select the logo image `images/MatchiBizLogo_TEST.png` in the case of the TEST environment.

HOWTO Server Maintenance

Keeping the server's patch level up to date

Update application packages

You can update all the packages on a server to the latest release:

```
$ sudo yum update
```

This lists all the packages that are upgraded or to be installed as a result of new dependencies.

```
=====
Package                Arch      Version                Repository    Size
=====
Updating:
ORBit2                 x86_64    2.14.17-5.el6          base         168 k
audit-libs             x86_64    2.3.7-5.el6            base          71 k
authconfig             x86_64    6.1.12-19.el6          base        377 k
...
Transaction Summary
=====
Install      3 Package(s)
Upgrade    220 Package(s)

Total download size: 285 M
Is this ok [y/N]:
```

Warning

An impact analysis should be performed before confirming the execution of the update.

- Not all updates are good - even though they have been verified and tested by the trusted repositories such as REDHAT, REMI, EPEL and others.
- Some updates include new features or behaviours that may impact the stability of your existing application.

The update can be processor intensive, and may even knock some existing processes out. Keep a watch on log files for process that have been knocked out and be ready to restart them. It is best to perform any kind of update during low server usage.

Updating the RKHUNTER file hashes

Update the file hashes of the new files on RKHUNTER, or you will get false alert emails that the server may been compromised, similar to the following:

```
Subject: [rkhunter] Warnings found for s16972617
Please inspect this machine, because it may be infected.
```

Run this command to update the file hashes:

```
$ sudo rkhunter --propupd
[ Rootkit Hunter version 1.4.0 ]
File updated: searched for 167 files, found 138
```

Update the Linux Kernel

This is currently managed by the providers of the server virtual hosting environment. In this type of environment, it is not possible to update a Linux kernel. It is however possible to update the kernel on dedicated servers (a.k.a. 'tin').

What kernel am I running?

Use this command to check the kernel version:

```
$ uname -r
2.6.32-042stab108.2
```

You can also see all the details of the operating system in one go:

```
$ uname -a
Linux s16972616.onlinehome-server.info 2.6.32-042stab108.2 #1 SMP Tue May 12 18:07:50 MSK 2015 x86_64 x86_64 x86_64 GNU/Linux
```

What Linux Distribution am I running?

This is the vendor of the Linux distribution:

```
$ cat /etc/issue | head -1
CentOS release 6.6 (Final)
```

HOWTO Joomla Framework and Component Maintenance

Installation of New Third-Party Joomla Components

Pre-Production Testing

The installation of all third-party component should first be regression tested in a non-production environment. Apart from a successful regression test outcome, the new component should also comply to the following criteria:

- Flawless installation, with no error messages or warnings
- Flawless un-installation, with no error messages or warnings
- No impact on performance
- The components is fit for purpose and performs the expected function adequately

Upgrading Third-Party Joomla Components

Pre-Production Testing

The upgrade of existing third-party components should be regression-tested on a pre-production environment before it is deployed on the Production environment. Since many of the components in use are from third parties, there is no guarantee that the behaviour and integration between one to the next version will be consistent.

Upgrading Third-Party Joomla Components when upgrading the Joomla Core Framework

Decision Criteria

It is useful to cross-reference component-provider's latest release notes (on their respective websites usually) to verify if a component needs to be upgraded in line with a new Joomla Core Framework release. Some components also announce on the Joomla Administrator panel if they are due for an upgrade.

Pre-Production Testing

If so, the third-party component should also be upgraded at the same time and included in the regression test.

Upgrading Components and the Joomla Core Framework

Decision Criteria

The same is true when the Joomla core needs to be updated, since this can cause third-party components to fail. This is a frequently documented phenomenon where third-party components fail to stay in lock-step with the Joomla core. If the third-party components fail to work with after an upgrade of the Joomla Core Framework and no component upgrades are available, then the upgrade of the Joomla Core Framework should be abandoned.

Pre-Production Testing

It is possible to reduce some of the regression testing effort by doing a code comparison between the AS-IS and TO-BE versions. In some cases, a security patch for the Joomla Core consist of a few lines of code only. A useful tool for comparing code between directories on a wholesale basis is the open-source Meld tool from <http://meldmerge.org> and works on all OSs.

Special Security Step for all Upgrades and new Installations

All script files in the Joomla directory tree are set to read-only and owned by the Apache daemon for security purposes. There is an hourly process, on the hour, that sets all script files in the Joomla directory tree to read-only again if they were accidentally left writable by an installation procedure.

Therefore, before you can install a new component or upgrade an existing one, the directory needs to momentarily be set to writable:

```
$ cd /var/www/vhosts/[servername]/matchi.biz
$ sudo find administrator bin cli components includes language layouts libraries media modules plugins templates \
-not -path "*/cache*" -not -path "*/.svn/*" -type f \( -name "*.php" -or -name "*.ini" \) -exec chmod u+w {} \; -print |
```

When the upgrade/installation is complete, remove the write-privileges again:

```
$ sudo find administrator bin cli components includes language layouts libraries media modules plugins templates \
-not -path "*/cache*" -not -path "*/.svn/*" -type f \( -name "*.php" -or -name "*.ini" \) -exec chmod u-w {} \; -print |
```

Since there are no process semaphores to mark that an installation is in progress, upgrades and installations need to be done in the period between the hours, in case you are faced with an inexplicable upgrade/installation failure.

Retrieved from "http://wiki.matchi.info/index.php?title=Sysadmin_HOWTOs&oldid=1500"

Category: Pages with syntax highlighting errors

-
- This page was last modified on 15 October 2016, at 21:43.
 - Content is available under Creative Commons Attribution unless otherwise noted.