Brand Manager 3

By Gilad Ventura

## Terminology

|  |  |
| --- | --- |
| Term | Description |
| Distribution | A single working casino package with all of its components and databases. |
| WP | Wordpress |
| BO | Back office |
| GS | Game Server |
| BOUI | Back office user interface |
| Wallet | Wallet components, based on scala |
| BM | Brand manager |

## Usage

bm3.sh -a <action> -b <brand>

### Available parameters

* -a <action>
  + create
  + delete
  + update

will svn update a specific distribution.

* + switch

will svn switch a specific distribution to a different svn path.

* -b <brand>
  + cosmikcasino
  + deuceclub
* -c <component> optional – default: all
  + wp
  + bo
  + gs
  + boui
  + wallet
* -t <type> optional – default: trunk
  + branch
  + tag
  + trunk
* -n <type name> optional (dependent on type) – default: trunk
  + Branch name
  + Tag name
* -d <distribution display name> optional
  + A custom name the distribution will be represented as
* -s optional
  + Simulate the action with debug data displayed
* -l optional
  + List all available installed distributions, should be used only as a single parameter
* -r <revision number> optional
  + Revision number, used for the update action to update to a specific revision. Wallet component will not be affected by revision number as its not versioned.
* -cleanup <bool> optional
  + Default: true, does the BM should cleanup on failure?

### Usage examples

* bm3.sh -a create -b cosmikcasino -t tag -n 116003 –d latestcosmik

will create the cosmikcasino brand, all components, based on tag 116003 and will be known as latestcosmik.devmachine.cosmikcasino.com

* bm3.sh -a create -b cosmikcasino –c wp –d temp

will create the cosmikcasino brand, wp component, based on trunk and will be known as temp.devmachine.cosmikcasino.com

* bm3.sh -a switch -b cosmikcasino –d temp –t tag –n 1.16.0\_03

will switch the cosmikcasino distribution custom-named 'temp' to be based on tag 1.16.0\_03

* bm3.sh -a update -b deuceclub –t tag –n 1.17.0 –r 1265421

will update the deuceclub tag 1.17.0 distribution to revision 1265421

## Routing

All distributions created should be accessible from every internal machine via Gamescale's DNS server. BM should register every new distribution created into the DNS server and vice versa.

### Distributions URL structure

[tag123 / branch123 / trunk / custom distribution display name].[machine name].www.[brand real URL].com

## Source structure

To make sure the BM will stay relevant, it should be modular and scalable. Thus the components actions scripts should be modular and a main brand configuration file should be available.

Generally, to add a new brand / component / action type to the BM, all that will be required is to add a new action script, brand configuration file or a new component folder with its relevant action scripts.

### Actions scripts

Actions scripts are files containing a set of actions required to create / delete / update (etc..) a specific component. That means that for every action type for every component a file should be available respectively.

e.g:

* wordpress
  + /create
  + /delete
  + /update
  + …
* Backoffice
  + /create
  + /delete
  + /update
  + …
* (other components)
* ..

### Brand configuration file

A configuration file representing a brand in the system, including every brand specific parameter needed as well as enabled components for the brand (some brands might require different components to work)

## General features

### Ubuntu based

BM3 should be based on Ubuntu server 10

### Backup before update / switch

When updating or switching there is a chance of a failure in the process, which may turn the distribution to be not functional. Thus when BM runs an update script, it should first backup the entire distribution (make a temp copy of everything including DB) so in case of an error restoration can use the copy.

### Cleanup on failure

When creating a distribution BM should know how to remove everything back in case of a failure in the process. It should act as a garbage collector.

### Database versioning

When using the "switch" or "update" actions, the distribution database should remain intact with all of it's contents. In order to upgrade the DB scheme to the new version without damaging the data a new upgrade – downgrade system should be integrated.

Each component which should support DB upgrading / downgrading should consists of a base.sql file which should contain the base scheme and content of the DB. From that point, each change of DB scheme should be done with an upgrade script named up%d.sql and a matching reverse script named down%d.sql . the %d digit represents the DB version, from 2 …. N (from 2 because the base.sql is 1).

To track a specific distribution DB version a new DB table will be introduced: DBVersionsLog

|  |  |
| --- | --- |
| Name | Type |
| Id | Int (auto increment) |
| Version | Int |
| Date | Datetime |

Each 'installed' DB upgrade by the BM should be appended by the BM and the last row is the current DB version. Consider the following example

|  |  |  |
| --- | --- | --- |
| Id | Version | Date |
| 1 | base | 2012-9-2 10:10:10 |
| 2 | 2 | 2012-9-2 10:10:11 |
| 3 | 3 | 2012-9-2 10:10:12 |
| 4 | 4 | 2012-9-2 10:10:13 |

#### Creating

When creating a distribution, BM should run the base.sql file followed by the up%d.sql files ordered while updating the DBVersionsLog table with the newly installed upgrade scripts accordingly.

#### Upgrading

When switching a distribution, BM should not delete the old DB and scripts folder, but instead inspect the DBVersionsLog table and compare the latest DB version installed with the available DB scripts. If there are additional DB scripts which are not installed according to the DBVersionsLog table, an upgrade is needed.

To upgrade, BM should run the new up%d.sql files ordered while updating the DBVersionsLog table with the newly installed upgrade scripts accordingly. At the end of the process BM should delete the old scripts folder.

#### Downgrading

When switching / updating a distribution, BM should not delete the old DB and scripts folder, but instead inspect the DBVersionsLog table and compare the latest DB version installed with the available DB scripts. If the latest DB version upgrade script according to the DBVersionsLog table is not available in the new version, a downgrade is needed.

To downgrade, BM should run the old scripts folder down%d.sql files from the latest to the new scripts folder latest (not included) while deleting the relevant DBVersionsLog rows accordingly. At the end of the process BM should delete the old scripts folder.

In case of a missing downgrade script needed, BM action should fail.

## Tags

Usage: ${TAG\_NAME} on script files or %TAG\_NAME@ on template files

|  |  |
| --- | --- |
| Tag | Description / Example |
| BRAND\_DOMAIN\_NAME | Cosmikcasino.com |
| READABLE\_BRAND\_NAME | Cosmic casino |
| BRAND­\_NAME | Cosmikcasino |
| SVN\_TYPE | empty(trunk) / tags / branches |
| SVN\_TYPE\_NAME | 1.16.0\_05 |
| SVN\_TYPE\_NAME\_CLEAN | 116005 |
| SVN\_TYPE\_NAME\_PREFIX\_DOT | 116005. |
| SVN\_TYPE\_NAME\_PREFIX\_US | 116005\_ |
| WWW\_ROOT | /var/www |
| SVN\_ROOT | <http://svn-master/svn/gs1> |
| APACHE\_CONF\_DIR | /etc/httpd/conf.d or /etc/apache2 |
| WWW\_DISTRIBUTION\_DIR | /var/www/[distribution display name / type + type name]  e.g  /var/www/latest\_cosmik  /var/www/cosmikcasino\_tag\_116005  /var/www/cosmikcasino\_branch\_2486  /var/www/cosmikcasino\_trunk |
| JETTY\_ DISTRIBUTION \_DIR | the distribution dir inside the jetty apps folder path  e.g  /[JETTY\_APPS\_PATH]/latest\_cosmik  /[JETTY\_APPS\_PATH]/cosmikcasino\_tag\_116005  /[JETTY\_APPS\_PATH]/cosmikcasino\_branch\_2486  /[JETTY\_APPS\_PATH]/cosmikcasino\_trunk |
| DISTRIBUTION\_NAME | The distribution custom display name or the svn type + the svn type name is missing  e.g  latest\_cosmik  cosmikcasino\_tag\_116005  cosmikcasino\_branch\_2486  cosmikcasino\_trunk |

## Actions breakdown

### Create

#### Wordpress

1. Install if required
   1. Apache
   2. Memcache
   3. Mysql
2. Create DB
   1. Name should be informative, consisting of distribution name and the relevant component type e.g wp\bo\gs\wallet
3. Get Wordpress
   1. Wp-config.php and .htaccess should be already inside the wordpress folder
4. Parse wp-config.php
   1. wp-config.php should be present in the wordpress framework including the relevant tokens to be parsed to real values by the BM
5. SVN checkout the Wordpress theme
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/themes/${BRAND\_NAME}
   2. To: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/themes/${BRAND\_NAME}
6. SVN checkout Wordpress plugin "GS\_core"
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/GS\_core/${BRAND\_NAME}
   2. To: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core
7. SVN checkout Wordpress plugin "GS\_block\_ie"
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/GS\_block\_ie
   2. To: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_block\_ie
8. SVN checkout Wordpress plugin "GS\_ajax"
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/GS\_ajax
   2. To: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_ajax
9. SVN checkout Wordpress plugin "GS\_DC\_flash"
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/GS\_DC\_flash
   2. To: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_DC\_flash
10. SVN checkout Wordpress plugin "wpml"
    1. From: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/sitepress-multilingual-cms
    2. To: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/sitepress-multilingual-cms
11. SVN checkout Wordpress plugin "w3 total cache"
    1. From: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/w3-total-cache
    2. To: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/w3-total-cache
    3. Copy all files from ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/w3-total-cache/installation\_files into ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content
12. Import reset.sql into DB
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/reset.sql
13. Import content.sql into DB
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/content.sql
14. Import CSV's into DB
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/\*.csv
15. Parse \_GS\_config.php
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/\_GS\_config.php
16. Create apache virtual host
    1. Should be according to routing requirements
    2. Should have a SYSTEM\_ENVIRONMENT variable set to dev/qa/stage/prod accordingly
17. Reset memcache

#### Back office

1. Install if required
   1. Apache
   2. Memcache
   3. Mysql
2. Create DB
   1. Name should be informative, consisting of distribution name and the relevant component type e.g wp\bo\gs\wallet
3. Get Yii
   1. To: ${WWW\_DISTRIBUTION\_DIR}/yii
4. SVN checkout the BO
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/backoffice/
   2. To: ${WWW\_DISTRIBUTION\_DIR}/bo
   3. Copy all files from ${WWW\_DISTRIBUTION\_DIR}/ bo/protected/config/${BRAND\_NAME}/files into ${WWW\_DISTRIBUTION\_DIR}/ bo/protected/config/
5. Parse main. ${env}.conf
   1. From: ${WWW\_DISTRIBUTION\_DIR}/ bo/protected/config/main. ${env}.conf
6. Import base and upgrade sql into DB
   1. As in "Database versioning" section
   2. Upgrade from: ${WWW\_DISTRIBUTION\_DIR}/bo/DB/${BRAND\_NAME}/upgrade/up%d.sql
   3. Base from: ${WWW\_DISTRIBUTION\_DIR}/bo/DB/${BRAND\_NAME}/base
7. Create + Parse cron from txt
   1. Should contain the relevant tokens to be parsed to real values by the BM
   2. From: ${WWW\_DISTRIBUTION\_DIR}/bo/cron/cron.txt
8. Create apache virtual host
   1. Should be according to routing requirements
   2. Should have a SYSTEM\_ENVIRONMENT variable set to dev/qa/stage/prod accordingly
9. Set CLI with SYSTEM\_ENVIRONMENT global variable
   1. variable set to dev/qa/stage/prod accordingly
10. Reset memcache

#### Back office UI

1. Install if required
   1. Apache
   2. Memcache
2. SVN checkout the BOUI
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/BOUI/
   2. To: ${WWW\_DISTRIBUTION\_DIR}/boui
3. Parse main.conf
   1. From: ${WWW\_DISTRIBUTION\_DIR}/boui/protected/config/main.conf
4. Create apache virtual host
   1. Should be according to routing requirements
   2. Should have a SYSTEM\_ENVIRONMENT variable set to dev/qa/stage/prod accordingly
5. Set CLI with SYSTEM\_ENVIRONMENT global variable
   1. variable set to dev/qa/stage/prod accordingly
6. Reset memcache

#### Game server

1. Install if required
   1. Apache
   2. Memcache
   3. Mysql
2. Create DB
   1. Name should be informative, consisting of distribution name and the relevant component type e.g wp\bo\gs\wallet
3. SVN checkout the games
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/games/
   2. To: ${WWW\_DISTRIBUTION\_DIR}/games
4. Parse site.conf.php
   1. From: ${WWW\_DISTRIBUTION\_DIR}/games/site.conf.php
5. Import base and upgrade sql into DB
   1. As in "Database versioning" section
   2. Upgrade from: ${WWW\_DISTRIBUTION\_DIR}/games/DB/${BRAND\_NAME}/upgrade/up%d.sql
   3. Base from: ${WWW\_DISTRIBUTION\_DIR}/games/DB/${BRAND\_NAME}/base
6. Create apache virtual host
   1. Should be according to routing requirements
   2. Should have a SYSTEM\_ENVIRONMENT variable set to dev/qa/stage/prod accordingly
7. Reset memcache

#### Area 51

1. Install if required
   1. Apache
   2. Memcache
2. SVN checkout the area51
   1. From: ${SVN\_ROOT}/${SVN\_TYPE}/games/
   2. To: ${WWW\_DISTRIBUTION\_DIR}/area51
3. Parse site.conf.php
   1. From: ${WWW\_DISTRIBUTION\_DIR}/area51/ site.conf.php
4. Create apache virtual host
   1. Should be according to routing requirements
   2. Should have a SYSTEM\_ENVIRONMENT variable set to dev/qa/stage/prod accordingly
5. Reset memcache

#### Wallet

1. Install if required
   1. Jetty
   2. Mysql
2. Create DB
   1. Name should be informative, consisting of distribution name and the relevant component type e.g wp\bo\gs\wallet
3. Get and extract .war file by required version
   1. Version from: ${SVN\_ROOT}/${SVN\_TYPE}/externals/wallet.txt
   2. Should recognize 'latest' as the newest code available
   3. BM should download the .war files from a Nexus server
   4. BM should know from which nexus url to get wallet .war file according to required version
   5. Should be saved and extracted at the Jetty web apps folder as
4. Import base and upgrade sql into DB
   1. As in "Database versioning" section
   2. Upgrade from: ${ JETTY \_BRAND\_DIR}/web\_inf/classes/sql/upgrade/up%d.sql
   3. Base from: ${ JETTY \_BRAND\_DIR}/web\_inf/classes/sql/base.sql
5. Parse props.txt
   1. From: ${JETTY\_DISTRIBUTION\_DIR}/web\_inf/classes/props/props.txt
6. Create Jetty virtual host
7. Restart Jetty

### Update

#### Wordpress

1. SVN update the Wordpress theme
2. SVN update Wordpress plugin "GS\_core"
3. SVN update Wordpress plugin "GS\_block\_ie"
4. SVN update Wordpress plugin "GS\_ajax"
5. SVN update Wordpress plugin "GS\_DC\_flash"
6. SVN update Wordpress plugin "wpml"
7. SVN update Wordpress plugin "w3 total cache"
   1. Copy & override all files from ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/w3-total-cache/installation\_files into ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content
8. Import reset.sql into DB
   1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/reset.sql
9. Import content.sql into DB
   1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/content.sql
10. Import CSV's into DB
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/\*.csv
11. Parse \_GS\_config.php
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/\_GS\_config.php
12. Reset memcache

#### Back office

1. SVN update bo folder
2. Import upgrade/downgrade sql into DB
   1. As in "Database versioning" section
   2. Upgrade from: ${WWW\_DISTRIBUTION\_DIR}/bo/DB/${BRAND\_NAME}/upgrade/up%d.sql
   3. Downgrade from: ${WWW\_DISTRIBUTION\_DIR}/bo/DB/${BRAND\_NAME}/downgrade/down%d.sql
3. Parse main. ${env}.conf
   1. From: ${WWW\_DISTRIBUTION\_DIR}/ bo/protected/config/main. ${env}.conf
4. Update + Parse cron from txt
   1. Should contain the relevant tokens to be parsed to real values by the BM
   2. From: ${WWW\_DISTRIBUTION\_DIR}/bo/cron/cron.txt
5. Reset memcache

#### Back office UI

1. SVN update boui folder
2. Parse main.conf
   1. From: ${WWW\_DISTRIBUTION\_DIR}/boui/protected/config/main.conf
3. Reset memcache

#### Game server

1. SVN update games folder
2. Parse site.conf.php
   1. From: ${WWW\_DISTRIBUTION\_DIR}/games/site.conf.php
3. Import upgrade/downgrade sql into DB
   1. Upgrade from: ${WWW\_DISTRIBUTION\_DIR}/games/db/${BRAND\_NAME}/upgrade/up%d.sql
   2. Downgrade from: ${WWW\_DISTRIBUTION\_DIR}/games/db/${BRAND\_NAME}/downgrade/down%d.sql
4. Reset memcache

#### Area 51

1. SVN update area51 folder
2. Parse site.conf.php
   1. From: ${WWW\_DISTRIBUTION\_DIR}/area51/ site.conf.php
3. Reset memcache

#### Wallet

1. Install if required
   1. Jetty
   2. Mysql
2. Get and extract .war file by required version
   1. Version from: ${SVN\_ROOT}/${SVN\_TYPE}/externals/wallet.txt
   2. Should recognize 'latest' as the newest code available
   3. BM should download the .war files from a Nexus server
   4. BM should know from which nexus url to get wallet .war file according to required version
   5. Should be saved and extracted at the Jetty web apps folder as
3. Import upgrade/downgrade sql into DB
   1. As in "Database versioning" section
   2. Upgrade from: ${ JETTY \_BRAND\_DIR}/web\_inf/classes/sql/upgrade/up%d.sql
   3. Downgrade from: ${ JETTY \_BRAND\_DIR}/ web\_inf/classes/sql/downgrade/down%d.sql
4. Parse props.txt
   1. From: ${ JETTY \_BRAND\_DIR}/web\_inf/classes/props/props.txt
5. Restart Jetty

### Switch

#### Wordpress

1. Rename DB
   1. Name should be informative, consisting of distribution name and the relevant component type e.g wp\bo\gs\wallet
2. Rename wordpress folder
3. Get new Wp-config.php
   1. Should be clean with all of the relevant tokens intact
4. Parse wp-config.php
   1. wp-config.php should be present in the wordpress framework including the relevant tokens to be parsed to real values by the BM
5. SVN switch the Wordpress theme
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/themes/${BRAND\_NAME}
6. SVN switch Wordpress plugin "GS\_core"
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/GS\_core/${BRAND\_NAME}
7. SVN switch Wordpress plugin "GS\_block\_ie"
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/GS\_block\_ie
8. SVN switch Wordpress plugin "GS\_ajax"
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/GS\_ajax
9. SVN switch Wordpress plugin "GS\_DC\_flash"
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/GS\_DC\_flash
10. SVN switch Wordpress plugin "wpml"
    1. To: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/sitepress-multilingual-cms
11. SVN switch Wordpress plugin "w3 total cache"
    1. To: ${SVN\_ROOT}/${SVN\_TYPE}/wordpress/wp-content/plugins/w3-total-cache
    2. Copy all files from ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/w3-total-cache/installation\_files into ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content
12. Import reset.sql into DB
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/reset.sql
13. Import content.sql into DB
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/content.sql
14. Import CSV's into DB
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/db/\*.csv
15. Parse \_GS\_config.php
    1. From: ${WWW\_DISTRIBUTION\_DIR}/wp/wp-content/plugins/GS\_core/\_GS\_config.php
16. Modify apache virtual host
    1. Should be according to routing requirements
    2. Should have a SYSTEM\_ENVIRONMENT variable set to dev/qa/stage/prod accordingly
17. Reset memcache

#### Back office

1. SVN switch bo folder
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/backoffice/
2. Import upgrade/downgrade sql into DB
   1. As in "Database versioning" section
   2. Upgrade from: ${WWW\_DISTRIBUTION\_DIR}/bo/DB/${BRAND\_NAME}/upgrade/up%d.sql
   3. Downgrade from: ${WWW\_DISTRIBUTION\_DIR}/bo/DB/${BRAND\_NAME}/downgrade/down%d.sql
3. Parse main. ${env}.conf
   1. From: ${WWW\_DISTRIBUTION\_DIR}/ bo/protected/config/main. ${env}.conf
4. Update + Parse cron from txt
   1. Should contain the relevant tokens to be parsed to real values by the BM
   2. From: ${WWW\_DISTRIBUTION\_DIR}/bo/cron/cron.txt
5. Reset memcache

#### Back office UI

1. SVN switch boui folder
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/BOUI/
2. Parse main.conf
   1. From: ${WWW\_DISTRIBUTION\_DIR}/boui/protected/config/main.conf
3. Reset memcache

#### Game server

1. SVN switch games folder
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/games/
2. Parse site.conf.php
   1. From: ${WWW\_DISTRIBUTION\_DIR}/games/site.conf.php
3. Import upgrade/downgrade sql into DB
   1. Upgrade from: ${WWW\_DISTRIBUTION\_DIR}/games/db/${BRAND\_NAME}/upgrade/up%d.sql
   2. Downgrade from: ${WWW\_DISTRIBUTION\_DIR}/games/db/${BRAND\_NAME}/downgrade/down%d.sql
4. Reset memcache

#### Area 51

1. SVN switch area51 folder
   1. To: ${SVN\_ROOT}/${SVN\_TYPE}/games/
2. Parse site.conf.php
   1. From: ${WWW\_DISTRIBUTION\_DIR}/area51/ site.conf.php
3. Reset memcache

#### Wallet

1. Install if required
   1. Jetty
   2. Mysql
2. Rename DB
3. Get and extract .war file by required version
   1. Version from: ${SVN\_ROOT}/${SVN\_TYPE}/externals/wallet.txt
   2. Should recognize 'latest' as the newest code available
   3. BM should download the .war files from a Nexus server
   4. BM should know from which nexus url to get wallet .war file according to required version
   5. Should be saved and extracted at the Jetty web apps folder as
4. Import upgrade/downgrade sql into DB
   1. As in "Database versioning" section
   2. Upgrade from: ${ JETTY \_BRAND\_DIR}/web\_inf/classes/sql/upgrade/up%d.sql
   3. Downgrade from: ${ JETTY \_BRAND\_DIR}/ web\_inf/classes/sql/downgrade/down%d.sql
5. Parse props.txt
   1. From: ${ JETTY \_BRAND\_DIR}/web\_inf/classes/props/props.txt
6. Modify Jetty virtual host
7. Restart Jetty

### Delete

#### Wordpress

1. Delete DB
2. Delete folder
3. Delete apache virtual host
4. Reset memcache

#### Back office

1. Delete DB
2. Delete Yii
3. Delete folder
4. Delete cron
5. Delete apache virtual host
6. Reset memcache

#### Back office UI

1. Delete folder
2. Reset memcache

#### Game server

1. Delete DB
2. Delete folder
3. Delete apache virtual host
4. Reset memcache

#### Area 51

1. Delete folder
2. Delete apache virtual host
3. Reset memcache

#### Wallet

1. Delete DB
2. Delete wallet from Jetty web apps folder
3. Delete Jetty virtual host
4. Restart Jetty