

Version-updater Script

If you have any questions about version-updater script please contact [Irina Ivanova](#).

Usage manual

▼ [One module update ./update-version.sh](#)

Parameters

Script requires 3 parameters: module name, version, JIRA username. Also right after running script asks JIRA password for provided username.

```
./update-version.sh admin 1.1.1.1 irina
```

How It Works

- Script automatically downloads module war file
- precimpiles it (only on WebLogic)
- renames it
- undeploys old version
- deploys new version
- checks whether new version is running (only on Tomcat, where successful deploy doesn't mean that that application is running)
- updates JIRA issue summary with new version

Lock

Right after running script creates lock file:

```
UPDATING_[PREDEMO]_user-module-version.loc
```

If this file exists it means that somebody is updating some module, so other user can't make updates at the same time. Script automatically removes lock file at the end of it's work.

NB! If script was terminated manually (for example Ctrl+C), then lock file should be also removed manually.

Logging

Script saves logs into following file:

```
version-updater/[update].log
```

If module was updates using this script, then you can see following rows in the log file:

```
-----one module update: ehealth-admin-3.9.1.24.war-----  
  
23.02.2015 13:20:41 irina OK: ehealth-admin-3.9.1.24.war is deployed  
23.02.2015 13:20:41 irina OK: ehealth-admin-3.9.1.24.war is running  
  
23.02.2015 13:20:42 irina OK: JIRA issue EHEALTHADM-426 summary is updated to  
ehealth-latest***3.9.1.24
```

Output Colors

red - error, that can't be ignored (for example, file download is failed)

yellow - warning, that can be ignored (for example, old version undeploy failed - may be it doesn't exist)

green - process is successfully completed (for example, file is downloaded)

cyan - info about performing some process (for example, script is downloading file right now)

▼ Update of several modules `./batch-update-versions.sh`

Parameters

Script requires 1 parameter: JIRA username. Also right after running script asks JIRA password for provided username.

```
./batch-update-versions.sh irina
```

Before running the script user has to describe modules and versions, that he want's to update, in the following file:

```
version-updater/batch-modules.txt
```

Content should be in the following format:

```
admin 1.1.1.1
billing 2.2.2.2
clinician-portal 3.3.3.3
tyk 4.4.4.4
```

NB! Order of modules is important - it defines in which order modules will be propagated. Right order can be found in version tracker: <http://ehealthtest.webmedia.int:7070/versiontracker/>.

There is a script that generates `batch-modules.txt` file with all last stable versions (assuming that there is an environment with stable versions). Script is located in HG and can be download from here: <http://ehealth.webmedia.ee/scripts/version-updater/>

By default it takes versions (ex lab-ordering) from KJL2-test environment and saves them into file on right order.

How It Works

- Script output starts with word *****START*****, so if needed script start could be easily found.
- automatically downloads first module (if failed, then continues with next one)
- precompile it (only on WebLogic, if failed, then continues with next one)
- renames it
- undeploys old version
- deploys new version (if failed, then continues with next one)
- checks whether new version is running (only in Tomcat)
- updates summary in JIRA issue (is failed, then continues with next module)
- repeats same steps with next module in the file

Statistics

At the end of all updates script shows statistics about which modules had errors and which modules were updates successfully.

```
*****
*****STATISTICS*****
DOWNLOAD ERRORS: 1
    billing-2.2.2.2
UNDEPLOY ERRORS: 0
DEPLOY ERRORS: 0
RUN ERRORS: 0
JIRA ERRORS: 0
DEPLOYED MODULES: 1
    admin-1.1.1.1
*****
```

Lock

Right after running script creates lock file:

```
UPDATING_[PREDEMO]_BATCH_MODULES_user.lock.
```

If this file exists it means that somebody is updating some module, so other user can't make updates at the same time.

Script automatically removes lock file at the end of it's work.

NB! If script was terminated manually (for example Ctrl+C), then lock file should be also removed manually.

Logging

Script saves logs into following file:

```
version-updater/[update].log
```

If module was updates using this script, then you can see following rows in the log file:

```
-----batch update: ehealth-reception-2.9.1.178.war-----  
23.02.2015 13:20:41 irina OK: ehealth-reception-2.9.1.178.war is deployed  
23.02.2015 13:20:41 irina OK: ehealth-reception-2.9.1.178.war is running  
23.02.2015 13:20:42 irina OK: JIRA issue EHEALTHREC-124 summary is updated to  
ehealth-latest***2.9.1.178
```

Output Colors

red - error, that can't be ignored (for example, file download is failed)

yellow - warning, that can be ignored (for example, old version undeploy failed - may be it doesn't exist)

green - process is successfully completed (for example, file is downloaded)

cyan - info about performing some process (for example, script is downloading file right now)

gray background - breaklines

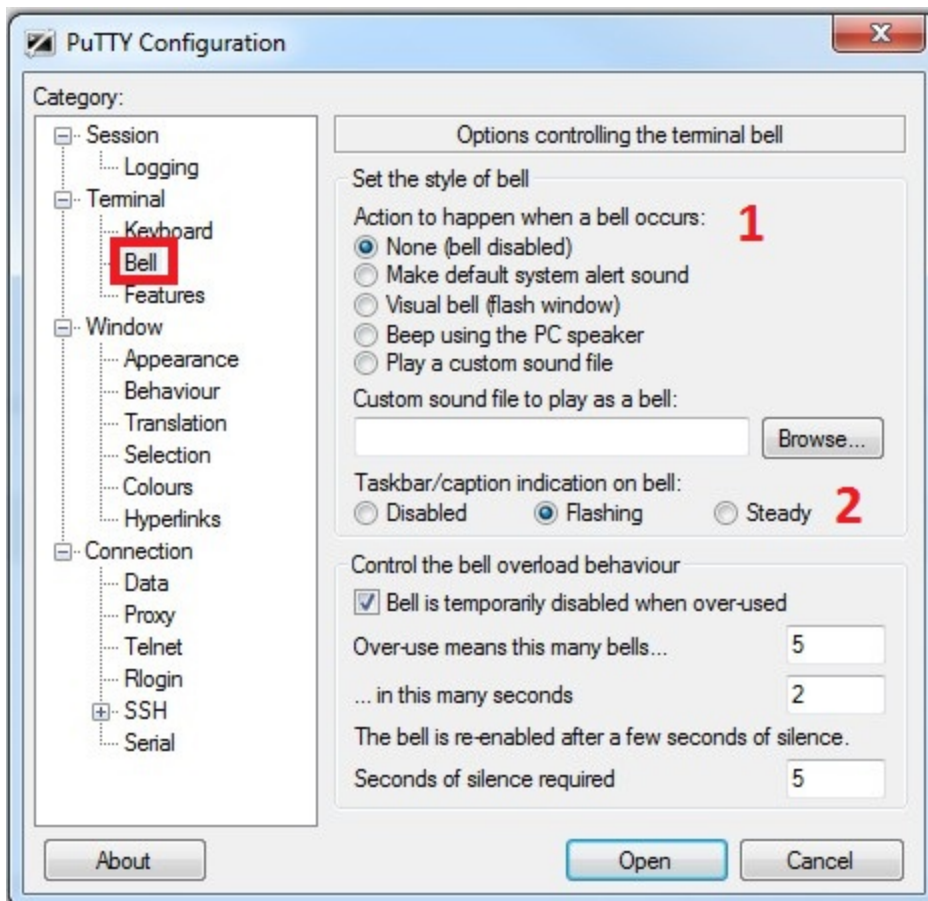
▼ Notifications

Script gives notifications in 2 cases:

- when it waits some response from user (JIRA password etc)
- when it finishes the work

The type of notification can be configured in terminal, that is used for updates.

Configurations in PuTTY:



In first block you can choose audio effects and on the second block - visual effects.

NB! These configurations applies to the session, so if you change them you change all notifications in the session, not only version-updater's ones.

▼ Problems

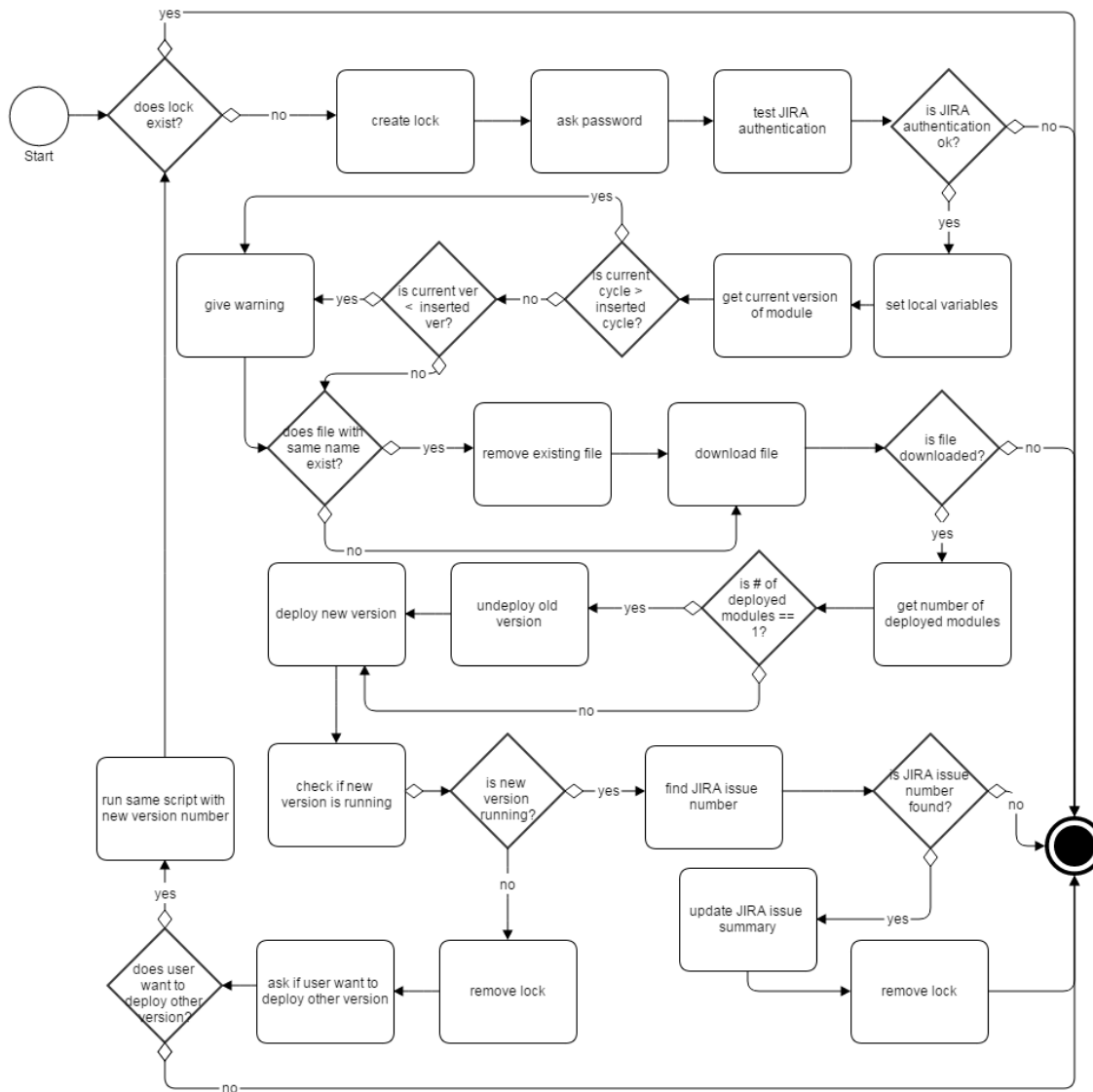
WebLogic: The action you performed timed out after 300 000 milliseconds

If deploy takes a lot of time and gives timeout error at the end, you should check memory usage in nodes - it's likely that some node is out of memory, so you need to restart it.

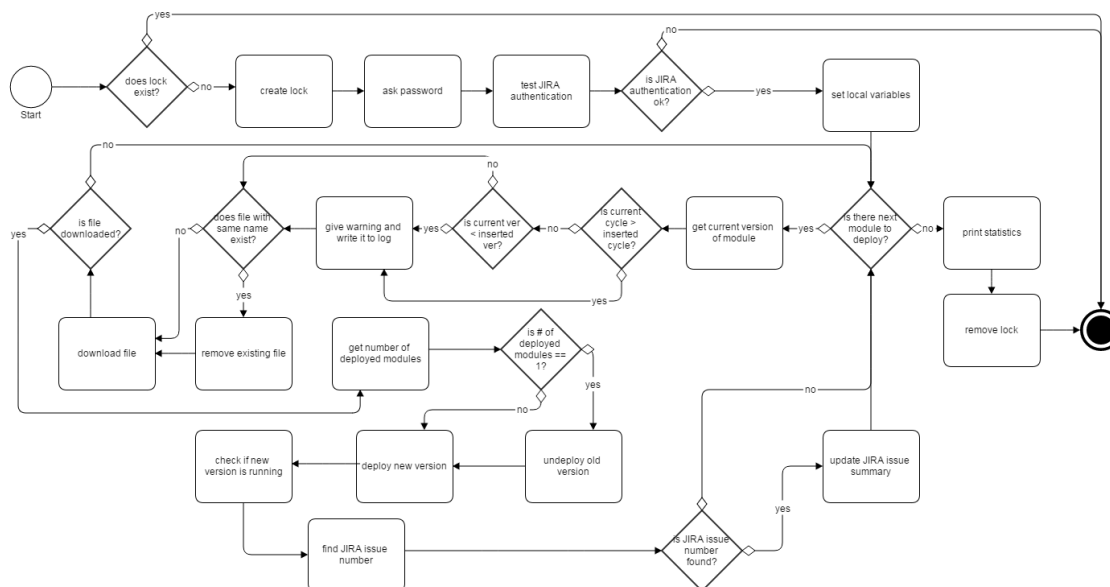
WebLogic: Precompiling gives an error

Try to precompile one more time - sometimes for unknown reason precompile script gives error on first precompiling.

▼ BPMN for one module in Tomcat



▼ BPMN for many modules in Tomcat



Installation and Configuration

▼ File structure

Files and their names may depend on different environments.

Centralized files (are available in <http://ehealth.webmedia.ee/scripts/version-updater>)

If you want to modify them you need to perform modifications in HG, not in application server.

Changes from HG will be synchronized every 10 minutes (see Centralization chapter for details).

- `version-updater/deploy-undeploy-script.py` - deploy/undeploy functions that are specific for WebLogic (you don't need to have it on Tomcat servers)
- `version-updater/functions-tomcat.sh` - deploy/undeploy functions that are specific for Tomcat (you don't need to have it on WebLogic). Works with both Tomcat 6 and 7
- `version-updater/functions-wl.sh` - functions that are specific for WebLogic (you don't need to have it on Tomcat servers)
- `version-updater/functions.sh` - general functions, that are used in all servers
- `version-updater/extended-modules.txt` - file with list of modules, that have `-lt` extension. NB! It is crucial that content of this file is up-to-date. If some `-lt` module is migrated and this file is not updated, then script deploys wrong estonian version.
- `version-updater/readme.txt` - brief description of the script
- `update-version-tomcat.sh` - one-module updater for test environments (where only 1 server in use), specific for any version of Tomcat (you don't need to have it on WebLogic)
- `update-version-live-tomcat.sh` - one-module updater for live environments (where 2 or more servers in use), specific for any version of Tomcat (you don't need to have it on WebLogic)
- `batch-update-versions-tomcat.sh` - batch modules updater. specific for any version of Tomcat (you don't need to have it on WebLogic)

Local configuration files

Configuration files depends on one environment.

- `version-updater/batch-modules.txt` - file, where user should specify modules and versions that he wants to update with batch script (see "Update of several modules" chapter for details)
- `version-updater/jira-issues.txt` - file, where are listed JIRA issues for all modules
- `version-updater/set-variables.sh` - file with general variables that depends on environment
- `version-updater/functions-local.sh` - file with functions that depend on users input (for example, to set module and version)

Local output files

These files should be modified only by script, user shouldn't edit them.

- `version-updater/update.log` - file where script saves logs about all updates that were made with script
- `version-updater/jira-rest.txt` - empty file, that script uses for performing REST request to JIRA

▼ How to install script into new environment

1. Create directory **version-updater** in the root
2. Configure cron jobs to automatically download centralized files - see chapter Centralization for details
 - a. In production servers jobs should be added under deploy user:

```
su - deploy
crontab -e
```

3. Create following files in the **version-updater** directory:
 - a. `batch-modules.txt`
 - b. `jira-issues.txt`
 - c. `jira-rest.txt`
 - d. `set-variables.sh`
 - e. `update.log`
 - f. `functions-local.sh`

```
touch batch-modules.txt jira-issues.txt jira-rest.txt set-variables.sh update.log
functions-local.sh
```

4. Modify file `version-updater/jira-issues.txt` and add module-issue connections for all modules in the following format:

```
admin:EHEALTHADM-NNN
billing:EHEALTHBIL-NNN
```

```
admin:
authentication:
authorization:
billing:
clinician-portal:
consumer-portal:
diagnostics:
diet:
docman:
integration:
integration-It:
lab-ordering:
person:
prevention:
reception:
register:
report-engine:
reports:
schedule:
system:
treatment:
ui:
```

5. Add variables into file `version-updater/set-variables.sh` - see chapter "Local variables in set-variables.sh" for details.
6. Add content into file `version-updater/functions-local.sh` - you can see in other environments where script is already installed which content should it contain
7. After crontab job downloads centralized files make script files `.sh` executable:

```
chmod +x *.sh version-updater/*.sh
```

8. In production environment you should also download script for checking running version on both servers: <http://ehealth.webmedia.ee/scripts/show-status-of-applications.sh>

NB! Before script first run be sure, that cron job downloads all files and all required files presents (especially `extended-modules.txt`).

▼ Local variables in set-variables.sh

File `version-updater/set-variables.sh` should contain certain variables. If some environment doesn't have value for the variable (for example, `$proxy`) then variable still need to be listed in the file with blank value.

```
# Local variables, that should be different in every environment

app="It-test"
extension="-It"
```

```
# tomcatManager variable in Tomcat 6 should not contain /text at the end - for example,
"http://username:password@localhost:7070/manager"
tomcatManager="http://username:password@localhost:7070/manager/text"
proxy=""
summaryTitle="LT_test***"
```

Local variables that are specific for live environments, where 2 or more servers in use

```
declare -A ehealthTomcatManagers
ehealthTomcatManagers["@prdapp01:8080"]="http://username:password@prdapp01:8080/manager/text"
ehealthTomcatManagers["@prdapp02:8080"]="http://username:password@prdapp02:8080/manager/text"
```

```
declare -A hisTomcatManagers
hisTomcatManagers["@prdapp01:8090"]="http://username:password@prdapp01:8090/manager/text"
hisTomcatManagers["@prdapp02:8090"]="http://username:password@prdapp02:8090/manager/text"
```

```
declare -A modules
modules["admin"]="ehealth"
modules["authentication"]="ehealth"
modules["authorization"]="ehealth"
modules["billing"]="ehealth"
modules["clinical-portal"]="ehealth"
modules["diet"]="ehealth"
modules["docman"]="ehealth"
modules["integration"]="ehealth"
modules["integration-It"]="ehealth"
modules["person"]="ehealth"
modules["prevention"]="ehealth"
modules["register"]="ehealth"
modules["report-engine"]="ehealth"
modules["schedule"]="ehealth"
modules["system"]="ehealth"
modules["treatment"]="ehealth"
modules["ui"]="ehealth"
modules["zk"]="ehealth"
modules["diagnostics"]="his"
modules["reception"]="his"
modules["treatment"]="his"
```

General variables, that may stay the same in different environment, but if needed may be changed

```
extendedModules="version-updater/extended-modules.txt"
lock="UPDATING_VERSION.loc"
batchLock="UPDATING_BATCH_VERSIONS.loc"
log="version-updater/update.log"
rest="version-updater/jira-rest.txt"
issues="version-updater/jira-issues.txt"
jira="https://jira.nortal.com/rest/api/2/issue"
jiraAuth="https://jira.nortal.com"
```



```
warLocation="http://ehealth.webmedia.ee/ivy/eu.webmedia.healthcare"
warLocationCom="http://ehealth.webmedia.ee/ivy/com.nortal.healthcare"
batch="version-updater/batch-modules.txt"
```

▼ Centralization

Some general files (that all environments are using) are centralized, so if you need to modify them you need to do it only in one place.

1. Centralized files are located in `hg/EHEALTH/UTILS/scripts/version-updater` directory
2. At `ehealth proxy` server is described job, that every 10 minutes synchronize `UTILS` directory content, so fresh files (with 10 min delay) are available in the web <http://ehealth.webmedia.ee/scripts/version-updater/>
3. At every environment (that is using `version-updater` script) job is described, which every 10 minutes downloads fresh files from the web (only if they were modified).

All changes is possible to synchronize manually.

How to download `HG/EHEALTH/UTILS` directory

```
>mkdir HG
>cd HG
\HG>hg init
\HG>hg pull https://repo.nortal.com/hg/EHEALTH/UTILS
Username not specified in .hg/hgrc. Keyring will not be used.
http authorization required
realm: SONIA :: SCM Manager
user: irina
password:
pulling from https://repo.nortal.com/hg/EHEALTH/UTILS
requesting all changes
adding changesets
adding manifests
adding file changes
added 42 changesets with 370 changes to 288 files
(run 'hg update' to get a working copy)

\HG2>hg update
270 files updated, 0 files merged, 0 files removed, 0 files unresolved
```

How to update `HG/EHEALTH/UTILS` content

```

\HG>hg status
? soapUI-projects\REST-price-in-amb-reservation.xml
\HG>hg add
\HG>hg commit -u irina -m [message, eg "add new-project.xml"]
\HG>hg push https://repo.nortal.com/hg/EHEALTH/UTILS
pushing to https://repo.nortal.com/hg/EHEALTH/UTILS
Username not specified in .hg/hgrc. Keyring will not be used.
http authorization required
realm: SONIA :: SCM Manager
user: irina
password:
searching for changes
remote: adding changesets
remote: adding manifests
remote: adding file changes
remote: added 1 changesets with 1 changes to 1 files

```

How to add or modify job for updating files

NB! There are different files with functions for Tomcat and WebLogic. See chapter "File structure" for details.

```
[server]$ crontab -e
```

File with jobs will be opened in vim editor:

```

*/10 * * * * cd /home/lt_test/version-updater/; wget -N --no-cache
http://ehealth.webmedia.ee/scripts/version-updater/functions-tomcat.sh
*/10 * * * * cd /home/lt_test/version-updater/; wget -N --no-cache
http://ehealth.webmedia.ee/scripts/version-updater/readme.txt
*/10 * * * * cd /home/lt_test/version-updater/; wget -N --no-cache
http://ehealth.webmedia.ee/scripts/version-updater/functions.sh
*/10 * * * * cd /home/lt_test/version-updater/; wget -N --no-cache
http://ehealth.webmedia.ee/scripts/version-updater/extended-modules.txt
*/10 * * * * cd /home/lt_test/; wget -N --no-cache
http://ehealth.webmedia.ee/scripts/version-updater/update-version-tomcat.sh
*/10 * * * * cd /home/lt_test/; wget -N --no-cache
http://ehealth.webmedia.ee/scripts/version-updater/batch-update-versions-tomcat.sh

```

▼ Environments, where Irina Ivanova manages scripts

- ehealth-latest
- TÜK predemo, demo, reports
- ITK test, edu, reports
- LT test, LT demo