

Operators Utilities and Manuals for 0.2.14-r8 version and higher

Requirements to OS

*nix system **i386**. All examples in this manual are given for *Ubuntu Server 12.04 i386*.



For 64-bit Systems they must additionally install the appropriate packages to supporting the 32-bit libraries. For example, for Ubuntu 14.04 - 16.04, must be installed `lib32z1` package (32-bit shared libraries for AMD64)

Required packets

Example: For Ubuntu Server 12.04 i386 it is enough to install next package `mtd-utils`

Utilities

Operator utilities allow to make three different variants of firmware image:

- **PublicImage** - image which is signed with standard “public key” - STB_PUBLIC.
Updating variants: Starting from 0.2.14-r8 **updates via HTTP or USB from portal on manufacturer firmware version only**, («manufacturer firmware» - firmware versions that were assembled directly by the manufacturer and provided for automatic and manual updates by manufacturer's URL).
From Bootloader menu can be updated on PublicImage or CustomImage (transitional version) via Multicast/USB with bootstrap/TFTP
- **CustomImage** - image which is signed with “custom-key”. This key is created by operator **without manufacturer**.
Updating variants: Updates via **HTTP or USB** from portal on firmware versions that are signed by the same key (custom-key). **It is used if there is a need in STB update from portal (HTTP or USB update method)**.
From Bootloader menu can be updated on PublicImage or CustomImage (transitional version) via Multicast/USB with bootstrap/TFTP
- **OperatorImage** - image which is signed with “operator key”. “Operator key” should be **signed by manufacturer**.
Updating variants: **Updates on firmware versions which are signed by “operators key only”**.





Utilities: Operator Utilities



Before building the image of the firmware, we recommend to read the following documentation:
[JavaScript API](#), [Operator's Guide](#)

It is recommended: Run all commands with **root** permission. Use **tar** console archiver.

Warning! The **Command Shell** which is in the scripts and system shell can be different from each other!

Key points

1. Preparing of ulmage, ulmzlib_null.img, ulmzlib.img

MAG-250/254/270 using files:

- `vmlinux.bin.mag<model_number>`,

MAG-256/324/351 using files:

- `uImage_mag<model_number>.clean`

where is `<model_number>` - STB model number

`vmlinux.bin.mag<model_number>` or `uImage_mag<model_number>.clean` take from release <http://soft.infomir.com.ua> in according to STB model and place to `./images` directory.

2. Kernel sign

- For example: MAG-250 using the next script: `./kernel_sign_250.sh`

3. Profile preparing

* For MAG-250 using the profile - `'./img_make.profile.mag250'`

[Example of ./img_make.profile.mag250](#)

```
# Kernel's file system
export KERNEL_PATH=./uImzlib_mag250.img
# File name for enviroment variable
```

```
export ENV_VARIABLE_PATH=./images/env_mag250.txt
# Userfs
export USERFS_VERSION=1
export USERFS_PATH=./images/userfs.img
# File name for SecondBoot
export SECONDBOOT_PATH=./images/SbootIm_mag250
# File name for Logotype
#export LOGOTYPE_PATH=./images/logo.bmp.gz
export MAG200_OP_KEY=STB_PUBLIC
```

For correct work of KERNEL_PATH operator utility should be ./uImzlib_mag250.img for MAG250 and ./uImzlib.img for MAG200. Variables ENV_VARIABLE_PATH, USERFS_VERSION, USERFS_PATH, SECONDBOOT_PATH, LOGOTYPE_PATH can be commented. In this case there will not be such section. Variable MAG200_OP_KEY should contain key identifier (ID), with which image will be signed.

- Example:

```
export MAG200_OP_KEY=ID_key
```

where ID_KEY should be:

- **STB_PUBLIC** - for publicimage making. Utilities contain public key;
- **Custom-key ID** - for customimage making;
- **Operators key ID** - for operatorimage making.

Profile variables description

Name	Description
KERNEL_PATH	The location of the image file system that contains the core. If the variable is not assignment, then use the ./UImzlib.img .
ENV_VARIABLE_PATH	This variable should contain the path to the file, which contains the bootloader variables and their meaning. An example of such a file, see env.txt , He shows how to set variables of bootloader, in particular, font color and background color. "\$" Symbol should be screened.
USERFS_VERSION	Number of image version, which should be present in NAND partition named "Userfs" . If number of current image and new one are the same, update will not work.
USERFS_PATH	The location of the image user file system (image to be recorded in the section "NAND" with the title "Userfs"). The image is prepared using the userfs_img.sh . Previously the operator must put the necessary files to it in a subdirectory named /userfs .
SECONDBOOT_PATH	The path to the second image bootloader .
LOGOTYPE_PATH	The path to the file logo, prepared by the operator.

Example of env.txt

```
bg_color=0x00006498
fg_color=0x00FFFFFF
portal1=http://10.1.0.1/stalker_portal/c/index.html
language=en
```

```
update_url=http://10.1.0.1/imageupdate
ntpurl=10.1.0.1
timezone_conf=Europe/Kiev
```

The most used variables

4. "Imageupdate" making

Syntax:

```
./img_make.sh <version_number> "<description>" <path_to_rootfs> <modelname>
<path_to_profile>
```

where:

version_number	version number of the image, must be a number (3 digits) . After successfully upgrading variable of bootloader "Image_Version" takes this value.
"description"	description of firmware image Attention! "Spaces" are not allowed. description should be escaped! After successfully upgrading variable of bootloader "Image_Desc" takes this value
path_to_rootfs	the location/path to the Root File System STB . Root File System rootfs-...tar.gz you can get from release http://soft.infomir.com.ua
modelname	Model of STB for which the assembled image. Can be MAG200, MAG250, MAG254, MAG270
path_to_ptofile	Profile/path to profile , in which additive sections can be configured. Can be ./img_make.profile.mag200 or ./img_make.profile.mag<model_number>

Example for MAG-250:

```
./img_make.sh 218 "Test_my_version" ../250/rootfs-0.2.18r14 MAG250
./img_make.profile.mag250
```

Example for MAG-256:

```
./img_make.sh 220 "Test_my_version" ../256/i256-splash-7.7 MAG256
./img_make.profile.mag256
```

Update, Automatic software update, Variables cheking system

By default, update and automatic software update is performed on the factory version.

"Software update" in "System settings" - By default, updates on manufacturer version. It is recommended to change URL of your firmware image version. Variable: `update_url`

"Autoupdate module" from "Settings" in main menu - By default, autoupdate is turned on and

initialize update to manufacturer firmware versions. Update also available in manual mode. it is recommended to turn off or change URL to yours. Variables: **autoupdate_cond**, **betaupdate_cond**. Read more - [Software autoupdate module](#)

Variables cheking system - With purpose to increase security and control variables changing it is recommended to check necessary/critical variables while firmware version loades (for example: portal1, portal2, update_url, autoupdate_cond etc.). Read more - [Checking of environment variables while loading](#)

Notes on using GPG

The program **gpg** is used for working with keys and creating the digital signature of images. gnupg.org, [GNU Privacy Guard - Wikipedia](#)

To transfer the key from one device to another you may use the following commands:

- for preserving the information of the key in the file and:

```
gpg -o opsecbin.KEY --export-secret-keys ID_Key
```

- for adding this key to gpg and:

```
gpg --import opsecbin.KEY
```

- for viewing currently available keys:

```
gpg --list-keys
```

PublicImage - prepearing, making

[Publicimage - prepearing, making - example for MAG-250](#)

CustomImage - prepearing, making

[CustomImage - prepearing, making - example for MAG-250](#)

From:
<https://docs.infomir.com.ua/> -

Permanent link:
https://docs.infomir.com.ua/doku.php?id=en:stb_webkit:faq:make_image_mag-200_250_new

Last update: **2019/05/17 11:23**

