

# Операторские утилиты и инструкции для сборки образа ПО на основе 0.2.14-r8 и выше

## Требования к ОС

Все примеры используемые в данном руководстве приведены для *Ubuntu Server 12.04 i386*.



Для 64-разрядной системы, необходимо дополнительно установить поддержку 32-разрядных библиотек. Например, для Ubuntu 14.04 - 16.04 необходимо установить пакет `lib32z1` (32-bit shared libraries for AMD64).

## Необходимые пакеты

**Например:** Для **Ubuntu Server 12.04 i386**, достаточно установить пакет `mtd-utils`

## Утилиты

Утилиты оператора позволяют собирать образ ПО в трех вариантах:

- **PublicImage** - образ подписанный стандартным публичным ключом `STB_PUBLIC`.  
**Варианты обновления:** Начиная с версии ПО **0.2.14-r8** обновляется по HTTP и (USB без **Bootstrap** из меню портала) **только** на «версии ПО завода изготовителя», («версии ПО завода изготовителя» - версии ПО собранные непосредственно производителем и предоставляемые для автоматического и ручного обновления по URL завода изготовителя). На другие версии обновляется только из меню **Bootloader-a** (**multicast, USB&Bootstrap**). Заводской вариант образа ПО (`imageupdate`), подписанный стандартным публичным ключом `STB_PUBLIC`, выкладывается в каталоге с каждым текущим релизом, соответственно модели STB, по адресу: [soft.infomir.com.ua](https://soft.infomir.com.ua)
- **CustomImage** - образ подписанный кастомным ключом для обновления ПО. Ключ создается оператором самостоятельно без обращения к производителю.  
**Варианты обновления:** Обновляется по HTTP и «USB без **Bootstrap**» **только на образы ПО собранные на основе или после 0.2.14-r8 и подписанные этим же ключом**. На другие версии ПО (не подписанные «кастомным ключом»), обновляется только из меню **Bootloader-a** (**multicast, USB&Bootstrap**). Рекомендуется для операторов которые хотят реализовать безопасное обновление ПО через HTTP, но оставить вариант обновления на другие версии ПО из меню **Bootloader-a**.

- **OperatorImage** - образ подписанный ключом оператора. При этом ключ оператора в обязательном порядке подписан на заводе изготовителе.

**Варианты обновления:** Обновляется **только** на версии ПО подписанные этим же ключом оператора.



Утилиты для сборки образов ПО: [Operator Utilities](#)



Перед подготовкой образа основного ПО настоятельно рекомендуем ознакомиться со следующей документацией:

[JavaScript API](#), [Operator guide](#)

**Рекомендуется:** Выполнять все команды с правами «суперпользователя». При работе с архивами использовать консольный архиватор **tar**.

**Внимание!** Командная оболочка указанная в скриптах может отличаться от системной!

## Подготовка образов. Общие положения

### 1. Подготовка `ulmage`, `ulmzlib_null.img`, `ulmzlib.img`

**MAG-250/254/270** используются файлы:

- `vmlinux.bin.mag<model_number>`

**MAG-256/351/324** используются файлы:

- `uImage_mag<model_number>.clean`

где `<model_number>` - номер модели STB для которой собирается образ ПО.

`vmlinux.bin.mag<model_number>` или `uImage_mag<model_number>.clean` необходимо взять из релиза ПО в соответствии с моделью STB для которой собирается образ и поместить в каталог `./images`. Каталог `./images` находится в каталоге с утилитами.

Файлы:

- `uImage_mag<model_number>`
- `uImzlib_null_mag<model_number>.img`
- `uImzlib_mag<model_number>.img`

- будут созданы в каталоге с утилитами после окончания работы скриптов.

## 2. Подпись ядра

Пример для MAG-250 - используется скрипт `./kernel_sign_250.sh`

## 3. Подготовка профайла

Параметры сборки образа указываются в профайле:

```
* Пример для MAG-250 - используется профайл 'img_make.profile.mag250'
```

[Пример профайла 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
```

[Описание переменных профайла](#)

Название	Описание
KERNEL_PATH	Расположение образа файловой системы, содержащей ядро. Если переменная не заданна, то используется <code>./uImzlib.img</code> .
ENV_VARIABLE_PATH	Эта переменная должна содержать путь к файлу, в котором содержатся переменные бутлоадера и их значение. Пример, такого файла смотрите <b>Пример файла env.txt</b> , он показывает каким, образом можно установить переменные бутлоадера, в частности цвета шрифта и фона. Символ \$ должен экранироваться.
USERFS_VERSION	Номер версии образа, который должен располагаться в разделе «NAND» с названием «Userfs». Если номер текущего образа и обновляемого совпадает, то обновление не производится.
USERFS_PATH	Расположение образа файловой системы пользователя (образ который должен быть при обновлении записан в раздел «NAND» с названием «Userfs»). Образ подготавливается с помощью <code>userfs_img.sh</code> . Предварительно оператор должен положить необходимые ему файлы в подкаталог с именем <code>/userfs</code> .
SECONDBOOT_PATH	Путь ко второму образу бутлоадера.
LOGOTYPE_PATH	Путь к файлу логотипу, подготовленный оператором.

Для правильной работы утилит оператора, переменная `KERNEL_PATH` в профайлах, должна иметь значение `./uImzlib_mag250.img` для MAG250 и `./uImzlib.img` - для MAG200.

Переменные ENV\_VARIABLE\_PATH, USERFS\_VERSION, USERFS\_PATH, SECONDBOOT\_PATH, LOGO\_TYPE\_PATH могут быть закоментированы, в этом случае соответствующей секции не будет в образе. В переменной MAG200\_OP\_KEY должен быть установлен соответствующий идентификатор - (ID) ключа, которым будет подписан образ,

**Например:** export MAG200\_OP\_KEY=ID\_key, где ID\_KEY должен быть:

- STB\_PUBLIC - для сборки PublicImage. Публичный ключ входит в состав утилит;
- ID кастомного ключа - для сборки CustomImage;
- ID ключа оператора - для сборки OperatorImage.

## 4. Подготовка переменных Bootloader-а (Переменные окружения)

Необходимые переменные Bootloader-а указываются в файле ./images/env.txt

Наиболее часто используемые переменные

Пример файла env.txt

```
bg_color=0x00006498
fg_color=0x00FFFFFF
portall=http://10.1.0.1/stalker_portal/c/index.html
language=ru
update_url=http://10.1.0.1/imageupdate
ntpurl=10.1.0.1
timezone_conf=Europe/Kiev
```

## 5. Сборка образа (сборка imageupdate)

Синтаксис:

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

где:

version_number	<b>Номер версии образа</b> , должно быть трехзначным числом. После успешного обновления переменная начального загрузчика Image_Version принимает это значение.
description	<b>Краткое описание</b> !Внимание! «Пробелы» в описании не допускаются. Описание необходимо экранировать! После успешного обновления переменная начального загрузчика Image_Desc принимает это значение
path_to_rootfs	<b>путь</b> к каталогу с <b>корневой файловой системой</b> . Корневую файловую систему <b>rootfs-...tar.gz</b> можно взять из релиза <a href="http://soft.infomir.com.ua">http://soft.infomir.com.ua</a> в соответствии с моделью STB
modelName	<b>модель STB</b> для которой собирается версия, может быть <b>MAG200, MAG250, MAG254, MAG270</b> и т.д

path_to_profile	<b>Путь к профайлу</b> , в котором можно настроить подключение и отключение дополнительных секции. может быть <code>./img_make.profile.mag200</code> или <code>./img_make.profile.mag250</code> и т.д.
-----------------	--

Пример для MAG-250:

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

Пример для MAG-256:

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

«Обновление ПО» из меню «Системные настройки» - переменная **update\_url**. По умолчанию, обновление происходит по URL производителя на образ ПО подготовленный производителем. При сборке собственного образа рекомендуется изменить переменную на URL вашего образа.



"Модуль автоматического обновления" из пункта «Настройки» основного меню. Переменные: **autoupdate\_cond**, **betaupdate\_cond** По умолчанию, автоматическое обновление включено и инициирует обновление на версии ПО производителя. Обновление возможно так же и в ручном режиме. При сборке собственного образа рекомендуется отключить работу модуля автоматического обновления либо организовать собственную систему обновления, руководствуясь описанием и инструкциями.

**Проверка переменных окружения при старте**. С целью повышения безопасности и контроля несанкционированного изменения переменных, рекомендуется производить проверку необходимых/критичных переменных при загрузке ПО (например, таких как: `portal1`, `portal2`, `update_url`, `autoupdate_cond` и т.д.)

## Замечания по работе с программой gpg

Для работы с ключами и создания цифровой подписи образов используется программа **gpg** - [GnuPG.org](https://www.gnu.org/software/gnuPG/) [GnuPG Wikipedia](https://en.wikipedia.org/wiki/GnuPG)

Для переноса ключа, например, с одной машины на другую можно воспользоваться следующими командами:

- для сохранения информации о ключе в файле

```
gpg -o opsecbin.KEY --export-secret-keys ID-ключа
```

- для добавления этого ключа в gpg

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

- для просмотра текущих доступных ключей.

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

## PublicImage - подготовка, сборка

[PublicImage - подготовка, сборка - пример для MAG-250](#)

## CustomImage - подготовка, сборка

[CustomImage - подготовка, сборка - пример для MAG-250](#)

## OperatorImage

Инструкция по сборке и установке образа «**OperatorImage**» предоставляется после согласования процедуры подписи ключа оператора с [коммерческим отделом](#)

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

Permanent link:  
[https://docs.infomir.com.ua/doku.php?id=stb\\_webkit:faq:make\\_image\\_mag-200\\_250\\_new](https://docs.infomir.com.ua/doku.php?id=stb_webkit:faq:make_image_mag-200_250_new)

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

