

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.

CustomImage - Образ подписанный кастомным ключом далее - «**custom-key**» (цифровая подпись на заводе изготовителе не требуется). Генерация и использование «**custom-key**» необходима для реализации обновления ПО по HTTP и USB-(без bootstrap) на версии отличные от версий завода изготовителя. При этом остается возможность обновления ПО из меню бутлоадера с использованием multicast и USB-(с использованием bootstrap).

Порядок сборки

Выполнить подготовку необходимых файлов согласно пунктов **1,3,4** из "**Подготовка образов. Общие положения**"

1. Подготовить «**custom-key**». **Внимание!**«**custom-key**» необходимо поместить в корневую файловую систему **rootfs** в директорию **/usr/bin/ перед сборкой**;
2. Собрать «**промежуточный**» образ ПО с необходимыми кастомизациями **Внимание!**Промежуточный образ необходимо собирать без логотипа оператора;
3. Собрать «**финальный**» образ ПО со всеми необходимыми кастомизациями включая логотип оператора если необходимо;
4. Обновить STB в два этапа:
 1. На «**промежуточный**» образ ПО - устанавливается на версии 0.2.14-r8 и выше только через multicast или (USB&Bootstrap из меню бутлоадера); - На версии ПО ниже 0.2.14-r8 любым методом.
 2. На «**финальный**» образ ПО - **Внимание!** Обновление на образ подписанный «**Custom-key**» возможно только через **HTTP** или (USB без bootstrap) из меню портала.

1. Подготовка "custom-key"

- **1.1** Выполнить:

```
gpg --gen-key
```

Выбрать:

```
(5) RSA (sign only)
What keysize do you want? (2048) 2048
```

```
Key is valid for? (0) 0
Is this correct? (y/N) y
Real name: ID ключа
```

где: **ID ключа** - произвольное имя ключа, под этим ID ключ будет храниться в GPG

- **1.2** Экспорт ключа в файл:

```
gpg -o stb_custom.bin --export ID-ключа
```

- **1.3** Файл `stb_custom.bin` поместить в **rootfs** (**rootfs** - корневая файловая система приставки, берется из релиза) в директорию `/usr/bin/`.

Файл `stb_custom.bin` необходимо использовать при всех последующих (после 0.2.14-r8), сборках образов ПО.

2. Сборка промежуточного образа

- **2.1** Используется публичный ключ входящий в состав утилит. Выполнить:

```
gpg --import stb_secbin.key
export MAG200_OP_KEY=STB_PUBLIC
```

- **2.2** Подписать ядро публичным ключом. Выполнить:

```
./kernel_sign_324.sh
```

Внимание! «Промежуточный» образ собирается без логотипа оператора!

Необходимо закомментировать строку `export LOGO_PATH=./images/logo.bmp.gz` в профайле `./img_make.profile.mag324`

- **2.3** Выполнить сборку `imageupdate`:

```
./img_make.sh 220 "test_transitional" ../rootfs MAG324
./img_make.profile.mag324
```

- **2.4** Переименовать файл `imageupdate`. Например:

```
mv imageupdate imageupdate_trans
```

3. Сборка Финального образа

- **3.1** Используется **custom-key**, подготовленный ранее. Выполнить:

```
export MAG200_OP_KEY=ID кастомного ключа
```

- **3.2** Подписать ядро используя **custom-key**:

```
./kernel_sign_324_custom.sh
```

- **3.3** Указать в файле `img_make.profile.mag324`:

```
export MAG200_OP_KEY=ID кастомного ключа
```

- **3.4** Выполнить сборку imageupdate:

```
./img_make.sh 220 "test_final" ../rootfs MAG324 ./img_make.profile.mag324
```

- **3.5** Переименовать файл imageupdate. Например:

```
mv imageupdate imageupdate_final
```

4. Установка CustomImage в STB в два этапа

- **4.1** Установить в STB «Промежуточный» образ ПО - **Внимание!** Устанавливается на версии 0.2.14-r8 и выше **только** через **multicast** или **(USB&Bootstrap из меню бутлоадера)**; - На версии ПО ниже 0.2.14-r8 любым методом
- **4.2** Установить в STB «Финальный» образ ПО - **Внимание!** Обновление на образ подписанный «Custom-key» возможно **только** через **HTTP** или (USB без bootstrap) из меню портала.

В дальнейшем, для СТБ в которые были установлен промежуточный образ можно собирать и устанавливать только финальный образ ПО. Файл `stb_custom.bin`, в последствии, будет использоваться для всех собираемых образов и должен находиться в `rootfs`, в каталоге `/usr/bin`

Пример сборки CustomImage для MAG-254

Пример сборки CustomImage для MAG-254

```
root@localhost:~/218-r7-test# wget -b
http://wiki.iptv.infomir.com.ua/pub/operators_utils_new_mag200_mag250_mag254
.tar.gz
Continuing in background, pid 28657.
Output will be written to 'wget-log'.

root@localhost:~/218-r7-test# wget -b
http://soft.infomir.com.ua/mag254/release/0.2.18-r8/rootfs-0.2.18r8.tar.gz
Continuing in background, pid 28659.
Output will be written to 'wget-log.1'.
root@localhost:~/218-r7-test# wget -b
http://soft.infomir.com.ua/mag254/release/0.2.18-r8/vmlinux.bin.mag254
Continuing in background, pid 28661.
Output will be written to 'wget-log.2'.

root@localhost:~/218-r7-test# tar -zxvf
operators_utils_new_mag200_mag250_mag254.tar.gz
```

```
root@localhost:~/218-r7-test# tar -zxf rootfs-0.2.18r8.tar.gz
root@localhost:~/218-r7-test# mv vmlinux.bin.mag254
operators_utils_new_mag200_mag250_mag254/images/
root@localhost:~/218-r7-test# rm
operators_utils_new_mag200_mag250_mag254.tar.gz
root@localhost:~/218-r7-test# rm rootfs-0.2.18r8.tar.gz
root@localhost:~/218-r7-test# gpg --gen-key
```

gpg (GnuPG) 1.4.16; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:

- (1) RSA and RSA (default)
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)

Your selection? 4

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048)

Requested keysize is 2048 bits

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0)

Key does not expire at all

Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID

from the Real Name, Comment and E-mail Address in this form:

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: testbuildmag254

E-mail address:

Comment:

You selected this USER-ID:

"testbuildmag254"

Change (N)ame, (C)omment, (E)-mail or (O)kay/(Q)uit? o

You need a Passphrase to protect your secret key.

gpg: gpg-agent is not available in this session

You don't want a passphrase - this is probably a *bad* idea!

I will do it anyway. You can change your passphrase at any time, using this program with the option "--edit-key".

We need to generate a lot of random bytes. It is a good idea to perform

some other action (type on the keyboard, move the mouse, use the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

Not enough random bytes available. Please do some other work to give the OS a chance to collect more entropy! (Need 187 more bytes)

To speed up process of random bytes generation try to download some file in duplicated console in /dev/null

```
wget -O /dev/null http://192.168.1.1/some_big_file.ts
```

```
gpg: key AAD87568 marked as ultimately trusted
public and secret key created and signed.
```

```
gpg: checking the trustdb
```

```
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
```

```
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 2048R/AAD87568 2014-11-17
```

```
Key fingerprint = E6CB 1AEC 14F6 ACDC 9B74 D10F 1767 2ABE AAD8 7568
uid testbuilddmag254
```

Note that this key cannot be used for encryption. You may want to use the command "--edit-key" to generate a subkey for this purpose.

```
root@localhost:~/218-r7-test# gpg -o stb_custom.bin --export
testbuilddmag254
```

```
root@localhost:~/218-r7-test# cp stb_custom.bin rootfs-0.2.18r8/usr/bin/
```

```
root@localhost:~/218-r7-test# cd operators_utils_new_mag200_mag250_mag254/
```

```
root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254# gpg
```

```
--import stb_secbin.key
```

```
gpg: key 6BEED1ED: already in secret keyring
```

```
gpg: Total number processed: 1
```

```
gpg: secret keys read: 1
```

```
gpg: secret keys unchanged: 1
```

```
root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254#
export MAG200_OP_KEY=STB_PUBLIC
```

```
root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254#
./kernel_sign_254.sh
```

```
File vmlinux.sign create - successfully!!!
```

```
Image Name: MAG254 SH4 Kernel Linux 2.6.17
```

```
Created: Mon Nov 17 13:48:38 2014
```

```
Image Type: SuperH Linux Kernel Image (gzip compressed)
```

```
Data Size: 3244259 Bytes = 3168.22 kB = 3.09 MB
```

```
Load Address: 0x80800000
```

```
Entry Point: 0x80801000
```

```
File uImzlib_mag254.img create - successfully!!!
```

```
root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254# cat
img_make.profile.mag254
#   Kernel's file system
export KERNEL_PATH=./uImzlib_mag254.img
#   File name for enviroment variable
export ENV_VARIABLE_PATH=./images/env_mag254.txt

#   Userfs
export USERFS_VERSION=1
export USERFS_PATH=./images/userfs.img
#   File name for SecondBoot
export SECONDBOOT_PATH=./images/SbootIm_mag254

#   File name for Logotype
#export LOGO_PATH=./images/logo.bmp.gz
export MAG200_OP_KEY=STB_PUBLIC

root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254#
./img_make.sh 218 "test_transitional" ../rootfs-0.2.18r8/ MAG254
./img_make.profile.mag254
Make rootfs image ../rootfs-0.2.18r8/
Append digital signature MAG200_OP_KEY=STB_PUBLIC
File ./sumsubfsnone.img.sign create - successfully!!!
gpg (GnuPG) 1.4.16
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: ~/.gnupg
Supported algorithms:
Pubkey: RSA, RSA-E, RSA-S, ELG-E, DSA
Cypher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
File result:./imageupdate
Create section "Kernel size:" and append file ./uImzlib_mag254.img.
Create section "Image size:" and append file ./sumsubfsnone.img.sign.
Create section "Env size:" and append file ./images/env_mag254.txt.
Create section "Userfs size:" and append file ./images/userfs.img.
File ./imageupdate create - successfully!!!

root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254# mv
imageupdate imageupdate_tr
```

Transitional image was created. Now we proced to making final image.

```
root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254#
export MAG200_OP_KEY=testbuildmag254
```

```
root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254#
./kernel_sign_254_custom.sh
File vmlinux.sign create - successfully!!!
Image Name:   MAG254 SH4 Kernel Linux 2.6.17
Created:      Mon Nov 17 13:50:34 2014
Image Type:   SuperH Linux Kernel Image (gzip compressed)
Data Size:   3244259 Bytes = 3168.22 kB = 3.09 MB
Load Address: 0x80800000
Entry Point: 0x80801000
File uImzlib_mag254.img create - successfully!!!

root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254# vi
img_make.profile.mag254
root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254# cat
img_make.profile.mag254
#   Kernel's file system
export KERNEL_PATH=./uImzlib_mag254.img
#   File name for enviroment variable
export ENV_VARIABLE_PATH=./images/env_mag254.txt

#   Userfs
export USERFS_VERSION=1
export USERFS_PATH=./images/userfs.img

#   File name for SecondBoot
export SECONDBOOT_PATH=./images/SbootIm_mag254

#   File name for Logotype
export LOGOTYPE_PATH=./images/logo.bmp.gz
export MAG200_OP_KEY=testbuilddmag254

root@localhost:~/218-r7-test/operators_utils_new_mag200_mag250_mag254#
./img_make.sh 218 "test_final" ../rootfs-0.2.18r8/ MAG254
./img_make.profile.mag254
Make rootfs image ../rootfs-0.2.18r8/
Append digital signature MAG200_OP_KEY=testbuilddmag254
File ./sumsubfsnone.img.sign create - successfully!!!
gpg (GnuPG) 1.4.16
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: ~/.gnupg
Supported algorithms:
Pubkey: RSA, RSA-E, RSA-S, ELG-E, DSA
Cypher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

```
File result:./imageupdate
Create section "Kernel size:" and append file ./uImzlib_mag254.img.
Create section "Image size:" and append file ./sumsubfsnone.img.sign.
Create section "Env size:" and append file ./images/env_mag254.txt.
Create section "Userfs size:" and append file ./images/userfs.img.
Create section "Logotype size:" and append file ./images/logo.bmp.gz.
File ./imageupdate create - successfully!!!
```

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

Permanent link:
https://docs.infomir.com.ua/doku.php?id=stb_webkit:faq:customimage

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

