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). Is used if there is a need in STB update from portal (HTTP or USB update method).
From Booloader menu can be updated on PublicImage or CustomImage (transitional version) via Multicast/USB with bootstrap/TFTP

CustomImage - image which is signed with custom key, further - **“custom-key”** (digital signature by manufacturer doesn't need). Generating and using **custom-key** is for firmware updating via HTTP and (USB without bootstrap) on versions which are different of manufacturer version. There is opportunity to update firmware in bootloader menu via multicast and (USB with bootstrap using).

Build order

1. Prepare **“custom-key”**. You need to prepare **“custom-key”** and place it in **rootfs** in **`/usr/bin/`** directory before image making.
2. Make **“transitional”** image with necessary customizations Without operator logo!.
3. Make **“final”** image with all necessary customizations including operator logo (if it is needed).
4. **Update STB in two stages**:
    1. On the **“transitional”** image - update on firmware version 0.2.14-r8 and above **only via Multicast** or (USB&Bootstrap from Bootloader menu); - On firmware versions below 0.2.14-r8 using any update method.
    2. On the **“final”** image - Update to image which is signed by «Custom-key» possible **only via HTTP** or (USB without Bootstrap from portal system settings menu)

1. Prepeare custom-key

- **1.1** Perform:

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

Choose:

```
(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: Key ID
```

where: **Key ID** - random key name, which will be stored in GPG

- **1.2** Export key in the file:

```
gpg -o stb_custom.bin --export ID_Key
```

- **1.3** Place `stb_custom.bin` file in **rootfs** in `/usr/bin/` directory.

File `stb_custom.bin` should be used in all next image and firmware making (after 0.2.14-r8) .

2. "Transitional" image making

Make shure that stb_custom.bin file is in rootfs /usr/bin/ directory

- **2.1** “Public key” is used (the “Public key” is present in the Operators utilites). Perform:

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

- **2.2** kernel should be signed by “public key”. Perform:

```
./kernel_sign_250.sh
```

“Transitional” image making. Attention!Without operators logo!
Comment the string `export LOGOTYPE_PATH=./images/logo.bmp.gz` in profile `./img_make.profile.mag250`

- **2.3** Perform `imageupdate` making:

```
./img_make.sh 216 "test_transitional" ../rootfs MAG250
./img_make.profile.mag250
```

- **2.4** Rename `imageupdate`. For example:

```
mv imageupdate imageupdate_trans
```

3. "Final" image making

- **3.1 Custom-key** is used. Perform:

```
export MAG200_OP_KEY=ID of custom-key
```

- **3.2** Signed the kernel using **custom-key**:

```
./kernel_sign_250_custom.sh
```

- **3.3** Set in file `img_make.profile.mag250`:

```
export MAG200_OP_KEY=ID of custom-key
```

- **3.4** `imageupdate` making:

```
./img_make.sh 216 "test_final" ../rootfs MAG250 ./img_make.profile.mag250
```

- **3.5** Rename `imageupdate`. For example:

```
mv imageupdate imageupdate_final
```

4. Update STB on "CustomImage" in two stages:

- **4.1 “transitional”** firmware version - update on firmware version 0.2.14-r8 and above **only via Multicast** or (USB&Bootstrap from Bootloader menu); - On firmware versions below 0.2.14-r8 using any update method.
- **4.2 “final”** firmware version - Update to image which is signed by «Custom-key» possible **only via HTTP** or (USB without Bootstrap from portal system settings menu)

Transitional image should be installed in STB only once (to load on STB **stb_custom.bin**).

So there is no need to reinstall transitional on STB that has already your **stb_custom.bin** in rootfs. It will be enought to create only final version and update it on STB using HTTP or USB from portal.

File `stb_custom.bin` should be in `rootfs`, directory `/usr/bin/` in both transitional and final images

Example of making CustomImage for MAG-254

Example of making CustomImage for 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 -zxf
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
pub2048R/AAD87568 2014-11-17
Key fingerprint = E6CB 1AEC 14F6 ACDC 9B74  D10F 1767 2ABE AAD8 7568
uidtestbuildmag254

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 testbuildmag254
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 LOGOTYPE_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=testbuildmag254

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=testbuildmag254
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!!!
```