

**NAME**

mkimage – Generate image for U-Boot

**SYNOPSIS**

**mkimage** -l [**uimage file name**]

**mkimage** [*options*] -f [image tree source file] [**uimage file name**]

**mkimage** [*options*] -F [uimage file name]

**mkimage** [*options*] (**legacy mode**)

**DESCRIPTION**

The **mkimage** command is used to create images for use with the U-Boot boot loader. These images can contain the linux kernel, device tree blob, root file system image, firmware images etc., either separate or combined.

**mkimage** supports two different formats:

The old *legacy image* format concatenates the individual parts (for example, kernel image, device tree blob and ramdisk image) and adds a 64 bytes header containing information about target architecture, operating system, image type, compression method, entry points, time stamp, checksums, etc.

The new *FIT (Flattened Image Tree) format* allows for more flexibility in handling images of various types and also enhances integrity protection of images with stronger checksums. It also supports verified boot.

**OPTIONS****List image information:**

-l [*uimage file name*]

mkimage lists the information contained in the header of an existing U-Boot image.

**Create old legacy image:**

-A [*architecture*]

Set architecture. Pass -h as the architecture to see the list of supported architectures.

-O [*os*]

Set operating system. bootm command of u-boot changes boot method by os type. Pass -h as the OS to see the list of supported OS.

-T [*image type*]

Set image type. Pass -h as the image to see the list of supported image type.

-C [*compression type*]

Set compression type. Pass -h as the compression to see the list of supported compression type.

-a [*load address*]

Set load address with a hex number.

-e [*entry point*]

Set entry point with a hex number.

-l List the contents of an image.

-n [*image name*]

Set image name to 'image name'.

- d** [*image data file*]  
Use image data from 'image data file'.
- x** Set XIP (execute in place) flag.
- Create FIT image:**
- b** [*device tree file*]  
Appends the device tree binary file (.dtb) to the FIT.
- c** [*comment*]  
Specifies a comment to be added when signing. This is typically a useful message which describes how the image was signed or some other useful information.
- D** [*dtc options*]  
Provide special options to the device tree compiler that is used to create the image.
- E** After processing, move the image data outside the FIT and store a data offset in the FIT. Images will be placed one after the other immediately after the FIT, with each one aligned to a 4-byte boundary. The existing 'data' property in each image will be replaced with 'data-offset' and 'data-size' properties. A 'data-offset' of 0 indicates that it starts in the first (4-byte aligned) byte after the FIT.
- f** [*image tree source file* | *auto*]  
Image tree source file that describes the structure and contents of the FIT image.  
  
This can be automatically generated for some simple cases. Use "-f auto" for this. In that case the arguments -d, -A, -O, -T, -C, -a and -e are used to specify the image to include in the FIT and its attributes. No .its file is required.
- F** Indicates that an existing FIT image should be modified. No dtc compilation is performed and the -f flag should not be given. This can be used to sign images with additional keys after initial image creation.
- i** [*ramdisk\_file*]  
Appends the ramdisk file to the FIT.
- k** [*key\_directory*]  
Specifies the directory containing keys to use for signing. This directory should contain a private key file <name>.key for use with signing and a certificate <name>.crt (containing the public key) for use with verification.
- K** [*key\_destination*]  
Specifies a compiled device tree binary file (typically .dtb) to write public key information into. When a private key is used to sign an image, the corresponding public key is written into this file for for run-time verification. Typically the file here is the device tree binary used by CONFIG\_OF\_CONTROL in U-Boot.
- p** [*external position*]  
Place external data at a static external position. See -E. Instead of writing a 'data-offset' property defining the offset from the end of the FIT, -p will use 'data-position' as the absolute position from the base of the FIT.
- r** Specifies that keys used to sign the FIT are required. This means that they must be verified for the image to boot. Without this option, the verification will be optional (useful for testing but not for release).

## EXAMPLES

List image information:

```
mkimage -l uImage
```

Create legacy image with compressed PowerPC Linux kernel:

```
mkimage -A powerpc -O linux -T kernel -C gzip \  
-a 0 -e 0 -n Linux -d vmlinux.gz uImage
```

Create FIT image with compressed PowerPC Linux kernel:

```
mkimage -f kernel.its kernel.itb
```

Create FIT image with compressed kernel and sign it with keys in the /public/signing-keys directory. Add corresponding public keys into u-boot.dtb, skipping those for which keys cannot be found. Also add a comment.

```
mkimage -f kernel.its -k /public/signing-keys -K u-boot.dtb \  
-c "Kernel 3.8 image for production devices" kernel.itb
```

Update an existing FIT image, signing it with additional keys. Add corresponding public keys into u-boot.dtb. This will resign all images with keys that are available in the new directory. Images that request signing with unavailable keys are skipped.

```
mkimage -F -k /secret/signing-keys -K u-boot.dtb \  
-c "Kernel 3.8 image for production devices" kernel.itb
```

Create a FIT image containing a kernel, using automatic mode. No .its file is required.

```
mkimage -f auto -A arm -O linux -T kernel -C none -a 43e00000 -e 0 \  
-c "Kernel 4.4 image for production devices" -d vmlinuz kernel.itb
```

Create a FIT image containing a kernel and some device tree files, using automatic mode. No .its file is required.

```
mkimage -f auto -A arm -O linux -T kernel -C none -a 43e00000 -e 0 \  
-c "Kernel 4.4 image for production devices" -d vmlinuz \  
-b /path/to/rk3288-firefly.dtb -b /path/to/rk3288-jerry.dtb kernel.itb
```

## HOME PAGE

<http://www.denx.de/wiki/U-Boot/WebHome>

## AUTHOR

This manual page was written by Nobuhiro Iwamatsu <iwamatsu@nigauri.org> and Wolfgang Denk <wd@denx.de>. It was updated for image signing by Simon Glass <sjg@chromium.org>.