Manufacturing Tool V2 (MFGTool2) Update Command List (UCL) User Guide

Contents

[1 Overview 2](#_Toc387744999)

[2 UTP mode operation 2](#_Toc387745000)

[2.1 Global Configuration 2](#_Toc387745001)

[2.2 Update Command List 2](#_Toc387745002)

[2.2.1 Host Specific Commands 3](#_Toc387745003)

[2.2.2 Firmware Specific Commands 4](#_Toc387745004)

[2.2.3 Condition execute 6](#_Toc387745005)

[2.2.4 Use variable 6](#_Toc387745006)

# Overview

Manufacturing (Mfg) tool provides a flexible way for users to set their own operations. An xml script file is used to configure the users operation.

An xml file typically consists of a number of tasks which can be executed by running the manufacturing tool just once.

# UTP mode operation

## Global Configuration

Mfg tool uses global configuration to recognize which device the user wants to flash among different USB devices connected to the PC.

Let’s explain it by an example.

<CFG>

<STATE name="BootStrap" dev="MX6Q" vid="15A2" pid="0054"/>

<STATE name="Updater" dev="MSC" vid="066F" pid="37FF"/>

</CFG>

Global configuration is contained between parameter <CFG> and </CFG>.

<STATE name="BootStrap" dev="MX6Q" vid="15A2" pid="0054"/> indicates the first phase of the burning process, the phase name is “BootStrap”, and a device named “MX6Q” should be connected with the USB pid “0054” and vid “15A2”. For i.MX 6 serial, in the phase “BootStrap”, the valid strings for dev are: “MX6Q”, “MX6D”, “MX6SL”; in the phase “Updater”, the valid string for dev is: “MSC”.

<STATE name="Updater" dev="MSC" vid="066F" pid="37FF"/> indicates the second phase of the burning process, the phase name is “Updater”, and a device named “MSC” should be connected with the USB pid “37FF” and vid “066F”.

**Notes: “BootStrap” support multi instances. For example:**

<STATE name="BootStrap" dev="MX6SL" vid="15A2" pid="0063"/>

<STATE name="BootStrap" dev="MX6D" vid="15A2" pid="0061"/>

<STATE name="BootStrap" dev="MX6Q" vid="15A2" pid="0054"/>

Host can auto choose dev type according to usb vid and pid.

## Update Command List

The tool uses Update Command List (UCL) to specify all the user tasks. The UCL contains a distinct list for each use case. Each list contains a set of command elements with attributes for the command type, body, and payload. The command element text provides a user interface message for the current operation.

Each UCL begins from <LIST name=”xx”, desc=”xxx”> while ending with </LIST>, the name of which can be specified by users. Parameter “desc” is used for comment purpose.

There are two types of commands: host specific commands and firmware specific commands. Host specific commands are parsed and executed by host tool while firmware specific commands are parsed and executed by firmware runs on targeted device.

### Host Specific Commands

The example below shows a typical command. “state” indicates the phase of the command executed and “type” specifies the type of a command. “body” is a parameter of the command. “file” is another parameter. “Loading Kernel” is a description of the command.

<CMD state="BootStrap" type="load" file="uImage" address="0x10800000"

loadSection="OTH" setSection="OTH" HasFlashHeader="FALSE" >Loading Kernel.</CMD>

|  |  |  |  |
| --- | --- | --- | --- |
| Command type | Body | Other parameters | Description |
| load |  | file  Address  loadSection  setSection  HasFlashHeader  CodeOffset | Download an image to RAM. It is strongly recommended to follow the example provided in release package since it involves ROM code parameters which may not be easy to understand.  file: specify the path and name of the image file.  Parameter “address” specifies the RAM address where the image is located.  loadSection: a parameter used by ROM code, should be set to “OTH”  setSection: a parameter used by ROM code, should be set to “OTH” if there are other images to be loaded; set to “APP” if the last image is loaded.  HasFlashHeader: set TRUE if the image contains a flash header, or set to FALSE.  CodeOffset: the address offset of first executed instruct within the image.  The command is only for Bulk-IO mode i.MX device except i.MX50 HID mode device. |
| jump |  |  | Notify ROM code to jump to the RAM image to run. The command must be followed after a load command in which setSection value is set to “APP”.  The command is only for Bulk-IO mode i.MX device except i.MX50 HID mode device. |
| boot | Recovery | File  if | Download an image to RAM. |

### Firmware Specific Commands

If a command is typed as “push”, which means the command is parsed and executed by the targeted device instead of host, the only thing host has to do is to send the command to the targeted device.

The commands actually executed by OS image are downloaded in [*Command lists*](#_Command_lists). As a result, each OS has its own commands.

|  |  |  |
| --- | --- | --- |
| Command | Arguments | Description |
| ? | None | Request to send the device identity information in XML form |
| ! | integer | Initiate the reboot depending on argument. 3 means reboot while other values will force a shutdown. |
| $ | string | Execute shell command Example: $ echo 'hello from utp' |
| flush | None | Wait for all data transfer to be finished and processed. |
| ffs | None | Partition the SD card and flash the boot stream to it. |
| mknod | device\_class, device\_item,node\_t o\_create, type | Create the device node by parsing sysfs entry. Example: mknod class/mtd,mtd0,/dev/mtd0 |
| read | string | Read the file specified by parameter and send it to the host. If there is no such file, the appropriate status will be returned. |
| send | None | Receive the file from the host. Subsequent shell commands can refer to the file received as $FILE. Example: <CMD type="push" body="send" file="stmp378x\_ta1\_linux.sb/> <CMD type=”push” body=”$ kobs-ng init -d $FILE" /> |
| selftest | None | Perform self-diagnostic; returns either pass or appropriate status. Implemented as empty function in current release. |
| save | string | Save the file received by command “send” to the file specified as parameter. |
| pipe | string  **require file attribute** | Execute shell command and read data from stdio pipe IN. mfg will send file to stdio pipe OUT. **It is useful for big data transfer, more than physical memory size**  <CMD type="push" body="pipe tar -xv -C /mnt/ubi0" file="files/rootfs.tar"/> <CMD type="push" body="flush">Finish Flashing NAND</CMD>  Note: The above two commands must be combined to use  Recommend: Please add below command prior to pipe command to free some memory.  <CMD type="push" body="$ echo 3 > /proc/sys/vm/drop\_caches">release memory</CMD> |
| wff | NONE | **Deprecate** Prepare Write firmware to flash. |
| wfs | NONE | **Deprecate** Prepare Write firmware to SD CARD. |
| ffs | NONE | Write firmware to SD. |
| wrf | NONE,   **require file attribute** | ubiformat nand with ubi image.  Example <CMD type="push" body="wrf" file="files/rootfs.tar"/> <CMD type="push" body="frf">Finish Flashing NAND</CMD> |
| wrs | number of sd partition **require file attribute** | Write rootfs image to sd card. Example <CMD type="push" body="wrs2" file="files/rootfs.ext2"/> <CMD type="push" body="frs">Finish Flashing NAND</CMD>  You can also use  <CMD type="push" body="pipe dd of=/dev/mmcblkp2 bs=1K" file="files/rootfs.ext2"/? <CMD type="push" body="frs">Finish Flashing NAND</CMD> |
| frf | NONE | same as flush |
| frs | NONE | same as flush |

#### OTP Bits Programming

Command below shows how to write a file to a disk:

<CMD state="Updater" type="push" body="$ ls /sys/fsl\_otp ">Showing HW\_OCOTP fuse bank</CMD>

<CMD state="Updater" type="push" body="$ echo 0x11223344 > /sys/fsl\_otp/HW\_OCOTP\_MAC0">write 0x11223344 to HW\_OCOTP\_MAC0 fuse bank</CMD>

<CMD state="Updater" type="push" body="$ cat /sys/fsl\_otp/HW\_OCOTP\_MAC0">Read value from HW\_OCOTP\_MAC0 fuse bank</CMD>

The fuse bank name (ex: HW\_OCOTP\_MAC0) should be set as needed.

### Condition execute

Commands are executed ccording to dev type. Dev, which was defined at BootStrap. Refer to .

To skip command, add ifdev in ucl command. For example:

<CMD state="BootStrap" type="load" file="firmware/uImage" address="0x12000000" loadSection="OTH" setSection="OTH" HasFlashHeader="FALSE" ifdev="MX6Q MX6D">Loading Kernel.</CMD>

If the dev variable matches with the valueof ifdev, command will be executed, otherwise command will be skipped. If you connect to mx6sl chip board, above command will be skipped.

### Use variable

Ucl support static variable. Format is %name%. For example:

<CMD state="BootStrap" type="boot" body="BootStrap" file ="firmware/u-boot-imx6q%board%\_sd.imx" ifdev="MX6Q">Loading U-boot</CMD>

There are two method set a actual value to variable.

1. By cfg.ini.

[variable]

board = sabresd

mmc = 0

1. By command line when launching mfgtool.exe with –s, command line will overwrite the values of same name variables in cfg.ini.

mfgtool2.exe -s "board=sabresd" -s "mmc=0"

#### Limitation of variable:

1. xml file parser once at software open. When change ucl.xml file, need close and reopen mfgtools.
2. All files declared in ucl.xml must be exited, even though command may be skipped by ifdev. For example:

<CMD state="BootStrap" type="boot" body="BootStrap" file ="firmware/u-boot-imx6q%board%\_sd.imx" ifdev="MX6Q">Loading U-boot</CMD>

<CMD state="BootStrap" type="boot" body="BootStrap" file ="firmware/u-boot-imx6dl%board%\_sd.imx" ifdev="MX6D">Loading U-boot</CMD>

If %board% is sabresd, both files u-boot-imx6qsabresd\_sd.imx and u-boot-imx6dlsabresd\_sd.imx must be existed.



