

Intel® Platform Flash Tool

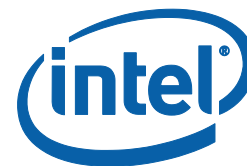
User guide

June 2020

Revision 5.10.0.0

Intel Confidential

Document Number: N/A



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

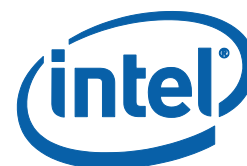
45-nm products are manufactured on a lead-free process. Lead-free per EU RoHS directive July 2006. Some E.U. RoHS exemptions may apply to other components used in the product package. Residual amounts of halogens are below November 2007 proposed IPC/JEDEC J-STD-709 standards.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2016, Intel Corporation. All rights reserved.

Document Number: N/A



Contents

1	Introduction	12
1.1	Terminology	12
2	Platform flash tool prerequisites	13
2.1	Operating Systems.....	13
2.2	Required Drivers.....	13
3	Installation	14
3.1	Installation on Windows OS.....	14
3.2	Installation on Linux OS.....	15
3.3	Installation on Macintosh OS	16
4	Usage	17
4.1	Launching the tool	17
4.2	Select flashing ingredients	19
4.2.1	Use a standalone flash file	19
4.2.2	Flash file editor	20
4.3	Start the flashing operation.....	25
4.3.1	'Blank' flash configuration	25
4.3.2	Fastboot or OTA flash file.....	27
4.4	Secure Tokens.....	27
4.5	Push Data and Provisioning	29
4.6	Backup and restore	30
4.6.1	Backup procedure.....	30
4.6.2	Restore procedure	32
4.7	Tool Options.....	34
4.7.1	The General tab of the option window	34
4.7.2	The external tool tab of the option window	35
4.7.3	The Log tab of the option window	36
4.7.4	The Advanced tab of the option window.....	39
4.8	Using the tool in command line.....	43
4.9	Embedded command line tools	46
4.9.1	DnX Firmware Downloader.....	46
4.10	Updating the tool	48
4.11	Edit USB Scan interval.....	48
5	FAQ	49
5.1	PID/VID.....	49
5.2	Conflicts with other USB devices	49
5.3	Fail to bind status server	49



Figures

Figure 3-1: Windows Setup Wizard	14
Figure 3-2: Platform Flash Tool Package Installer.....	15
Figure 3-3: Macintosh Setup Wizard.....	16
Figure 4-1: Platform Flash Tool Icon	18
Figure 4-2: Main window of Platform Flash Tool	18
Figure 4-3: Local Flash file tab	20
Figure 4-4: Select an xml flash file in the local flash tab	20
Figure 4-5: Select an archive flash file in the local flash tab	20
Figure 4-6: Flash editor tab, basic view	21
Figure 4-7: Flash editor: Adding a predefined sequence.....	22
Figure 4-8: Modem only flash.....	23
Figure 4-9: Flash blank configuration on a single device in MOS or POS	25
Figure 4-10: Flash blank configuration on a single device powered off	25
Figure 4-11: Flash blank configuration in parallel.....	26
Figure 4-12: Security tab.....	27
Figure 4-13: Access Security user guide	28
Figure 4-14: Provisioning tab	29
Figure 4-15: Backup/Restore contextual menu	30
Figure 4-16: Device backup window.....	30
Figure 4-17: Full backup.....	31
Figure 4-18: Device restore window.....	32
Figure 4-19: Full restore.....	33
Figure 4-20: General tab of the option window	34
Figure 4-21 External tools options	35
Figure 4-22: Log tab of the Options window.....	36
Figure 4-23: Advanced tab of the option window	39

Tables

Table 1-1: Terminology	12
Table 4-1: List of keyword for the logger options	38
Table 4-2: Status server messages structure (version 1.0)	41
Table 4-3: Status server event list.....	42



Revision History

Document Number	Revision Number	Description	Author	Revision Date
N/A	4.0.0	<ul style="list-style-type: none"> Updates for PFT 4.0.0 release 	Broquère, Xavier	May 13 th , 2013
N/A	4.1.0	<ul style="list-style-type: none"> Updates for PFT 4.1.0 release 	Broquère, Xavier	May 27 th , 2013
N/A	4.1.2	<ul style="list-style-type: none"> Updates for PFT 4.1.2 release 	Broquère, Xavier	June 24 th , 2013
N/A	4.1.4	<ul style="list-style-type: none"> Updates for PFT 4.1.4 release 	Broquère, Xavier	July 18 th , 2013
N/A	4.1.5	<ul style="list-style-type: none"> Updates for PFT 4.1.5 release 	Schohn, Cédric	August 22 th , 2013
N/A	4.2.0	<ul style="list-style-type: none"> Updates for PFT 4.2.0 release 	Broquère, Xavier	October 7 th , 2013
N/A	4.2.1	<ul style="list-style-type: none"> Updates for PFT 4.2.1 release 	Broquère, Xavier	October 23 th , 2013
N/A	4.2.2	<ul style="list-style-type: none"> Updates for PFT 4.2.2 release 	Broquère, Xavier	Nov. 20 th , 2013
N/A	4.3.0	<ul style="list-style-type: none"> Updates for PFT 4.3.0 release 	Broquère, Xavier	Jan. 15 th , 2014
N/A	4.4.0	<ul style="list-style-type: none"> Updates for PFT 4.4.0 release 	Broquère, Xavier	Feb. 26 th , 2014
N/A	4.4.1	<ul style="list-style-type: none"> Updates for PFT 4.4.1 release 	Broquère, Xavier	Feb. 28 th , 2014
N/A	4.4.2	<ul style="list-style-type: none"> Updates for PFT 4.4.2 release 	Broquère, Xavier	March 12 th , 2014
N/A	4.4.4	<ul style="list-style-type: none"> Updates for PFT 4.4.4 release 	Broquère, Xavier	April 17 th , 2014
N/A	4.4.5	<ul style="list-style-type: none"> Updates for PFT 4.4.5 eng Update xFSTK to 1.5.3 support windows registry hack 	Broquère, Xavier	April 30 th , 2014
N/A	5.1.0	<ul style="list-style-type: none"> Updates for PFT 5.1.0 Major changes Add support for JSON flash file Update xFSTK to 1.7.1 Add support for JSON flash file 	Broquère, Xavier	July 28 th , 2014
N/A	5.2.0	<ul style="list-style-type: none"> Updates for PFT 5.2.0 Update adb to 1.0.32 (from imin legacy) Update fastboot from imin legacy Enhancements and fixes 	Broquère, Xavier	Oct. 20 th , 2014
N/A	5.2.1	<ul style="list-style-type: none"> Updates for PFT 5.2.1 Update xFSTK API Enhancements and fixes 	Broquère, Xavier	Nov. 12 th , 2014
N/A	5.2.2	<ul style="list-style-type: none"> Updates for PFT 5.2.2 Update Intel SOC Drivers to 1.2.2 Update Intel Android USB Drivers to 1.8.1 Update xfstk API to 1.8.0 Enhancements and fixes 	Broquère, Xavier	Feb. 20 th , 2015
N/A	5.2.3	<ul style="list-style-type: none"> Updates for PFT 5.2.3 Bug Fix in command line interface for group activation 	Broquère, Xavier	March 2 nd , 2015



Introduction

N/A	5.2.5	<ul style="list-style-type: none"> Updates for PFT 5.2.5 JSON flash file format 3.0 support 	Broquère, Xavier	May 15 th , 2015
N/A	5.3.0	<ul style="list-style-type: none"> Updates for PFT 5.3.0 Add section for secure tokens 	Dartigue, Cyril	June 10 th , 2015
N/A	5.3.1	<ul style="list-style-type: none"> Updates for PFT 5.3.1 Update Broxton downloader 	Broquère, Xavier	June 30 th , 2015
N/A	5.3.2	<ul style="list-style-type: none"> Updates for PFT 5.3.2 Update ADB/fastboot 	Broquère, Xavier	July 27 th , 2015
N/A	5.3.3	<ul style="list-style-type: none"> Updates for PFT 5.3.3 Support dnxFwDownloader tool for Broxton devices Update ADB/fastboot on Mac OS to SDK 24.3.4 (platform tools 23.0.1) Update dfu-util to 0.8.1 (Windows & Linux) & 0.7.1 (OSX) Remove deprecated dldCli tool (replaced by dnxFwDownloader) Update downloadTool (Sofia device downloader) to 2.5 	Broquère, Xavier	October 8 th , 2015
N/A	5.3.4	<ul style="list-style-type: none"> Updates for PFT 5.3.4 Add Atlas Edge support Fix clear RPMB command in dnxFwDownloader Remove JLink version check at startup 	Broquère, Xavier	October 14 th , 2015
N/A	5.4.0	<ul style="list-style-type: none"> Updates for PFT Add flash editor example to create flash file for DnX FW flash 	Dartigue, Cyril	December 16 th , 2015
N/A	5.4.0	<ul style="list-style-type: none"> Updates for PFT New tool name: Platform Flash Tool Adb/fastboot update Dfu-utils update to 0.8.2 (parallel flash support) Update Dldr API to 1066 	Broquère, Xavier	January 20 th , 2016
N/A	5.4.1	<ul style="list-style-type: none"> Updates for PFT 5.4.1 Fix cflasher with no X server Dfu-util updated to 0.8.3 	Broquère, Xavier	Feb. 1 st , 2016
N/A	5.4.2	<ul style="list-style-type: none"> Updates for PFT 5.4.2 Do not specify -path option to dfu command by default 	Broquère, Xavier	Feb. 3 rd , 2016
N/A	5.5.0	<ul style="list-style-type: none"> Updates for PFT 5.5.0 Adb/fastboot updated Support PCIe Modem devices DownloadTool/FisTool updated New parameter editor feature (JSON flash file v3.1) New flash history menu 	Broquère, Xavier	March. 9 th , 2016
N/A	5.5.1	<ul style="list-style-type: none"> Updates for PFT 5.5.1 Update dfu-util to fix -path option on Ubuntu 12.04 	Broquère, Xavier	March. 18 th , 2016

Introduction



N/A	5.6.0	<ul style="list-style-type: none"> Updates for PFT 5.6.0 Update Qt to 5.5.1 Linux: package is now installed in /opt/intel/platformflashtool 	Broquère, Xavier	May 04 th , 2016
N/A	5.6.1	<ul style="list-style-type: none"> Updates for PFT 5.6.1 Update BXT token template 	Broquère, Xavier	July 04 th , 2016
N/A	5.7.1	<ul style="list-style-type: none"> Updates for PFT 5.7.1 Update DLDR API to 1108 Use new DnX drivers (DnX drivers for legacy platforms ; Flash Drivers for BXT-based platforms) Update BXT-based secure tokens to cope with CSS convention 	Dartigue, Cyril	August 18 th , 2016
N/A	5.8.0	<ul style="list-style-type: none"> Updates for PFT 5.8.0 Add "ABL" knob to BXT OEM Unlock token 	Dartigue, Cyril	September 1 st , 2016
N/A	5.8.1	<ul style="list-style-type: none"> Updates for PFT 5.8.1 Update DownloadTool to 2.64 Update FlsTool to 2.47 	Broquère, Xavier	September 13 th , 2016
N/A	5.8.2	<ul style="list-style-type: none"> Updates for PFT 5.8.2 Bug fixes for running on Ubuntu 14.04 	Deep, Ashesh	February 3 rd , 2017
N/A	5.8.3	<ul style="list-style-type: none"> Updates for PFT 5.8.3 Support for macOS 10.12.3 Support for Ubuntu 16.04 	Deep, Ashesh	February 24 th , 2017
N/A	5.8.4	<ul style="list-style-type: none"> Updates for PFT 5.8.4 Update DLDR API to 1171 V4 DnX support of larger FW images Update adb to version 1.0.39 Update flasboot to version 3db08f2c6889-android Update Crashtooluploader to 1.3.2 Fixed Auto-updater and artifactory issue 	Deep, Ashesh	May 02 nd , 2017
N/A	5.8.5	<ul style="list-style-type: none"> Updates for PFT 5.8.5 Support of ioc_flash_server in OEM version for Ubuntu 	Deep, Ashesh	October 03 rd , 2017
N/A	5.8.6	<ul style="list-style-type: none"> Updates for PFT 5.8.6 Update for CSE firmware version 3.1.50.2222 Update for ABL Payload knob (32-bit selection) Update for token signing with remote-HSM (User credentials are not mandatory) 	Deep, Ashesh	December 19 th , 2017
N/A	5.8.7	<ul style="list-style-type: none"> Updates for PFT 5.8.7 Added new platform KSL in token generation. 	Deep, Ashesh	March 29 th , 2018



Introduction

		<p>Updated CSE firmware 7063 (LKF support).</p> <p>Added binary to xml token generation feature for BXT</p> <p>Added product id for LKF.</p>		
N/A	5.8.8	<ul style="list-style-type: none"> Updates for 5.8.8 <p>Updated DownloadTool V2.89</p> <p>Added Token generation and signing support for LKF platform</p> <p>Added support for ICL platform in security</p> <p>Added IoC_flash_server 1.6.3 support for internal release</p> <p>Change EDSS key UI for getting the user permission for token</p>	Deep, Ashesh	July 6th, 2018
N/A	5.8.9	<ul style="list-style-type: none"> Updates for 5.8.9 <p>Added IOC flash server support for Windows OEM</p> <p>Updated IOC flash server app version 1.6.3 for OEM</p> <p>Added ABL and BIOS Secure Boot knobs in Intel unlock token and IDLM unlock token for Broxton</p>	Deep, Ashesh	August 9th, 2018
N/A	5.9.0	<ul style="list-style-type: none"> Updates for 5.9.0 <p>Upgraded QT to 5.11.2</p> <p>EDSS Connection timer in token signing has been increased</p> <p>Token ID related changes in LKF token XML</p> <p>Support added for CNL and CDF platforms for token generation & signing</p> <p>Added ICL Product ID</p> <p>Two new DnX commands added in DnXFWDownloader CLI</p> <p>DnX capabilities knob for LKF tokens added</p> <p>CSME Kit version 13.30.0.1012 integrated</p> <p>Added USBScanner Delay</p> <p>Added support for Lakefield tokens</p> <p>Removed ISH GDB Debug knob for Lakefield</p> <p>Added Lakefield OEM unlock token template</p>	Deep, Ashesh	February 21 st , 2019
N/A	5.9.1	<ul style="list-style-type: none"> Updates for 5.9.1 <p>Updated DNX kit to version 13.30.0.1022 (Includes multiple device support for IFWI flashing).</p>	Khan, Hamshan	May 6 th , 2019



		<p>Fix for PT-3653 (BTG knob values not shown).</p> <p>Upgraded Qt to 5.12.3 to make it complaint with latest OSPDT guidelines.</p> <p>Added new platforms KBL and LBG.</p> <p>Added new knob in Broxton token "Cancel OEM Authentication".</p> <p>Upgraded SignFile to 4.0.78.</p> <p>Added feature to get device data from Binary file.</p> <p>Removed other tokens from drop down of PFT OEM except OEM tokens.</p> <p>Added ISH debug knob to Lakefield tokens: Intel unlock, OEM unlock token and IDLM unlock token</p> <p>Modification in OEM unlock token template.</p> <p>Added USBScanner utility to OEM version of PFT.</p> <p>XFSTK downloader support disabled.</p>		
N/A	5.9.2.0	<ul style="list-style-type: none"> Updates for 5.9.2.0: Upgraded the Fastboot/adb version to 28.0.3 OEM unlock tokens added for platforms CDF, CNL, LBG AND KBL Fix for PT-3667 -Mismatch in Serial Number between USBScanner.exe and dnxFwDownloader.exe. Fix for PT-3653 BTG knob values not shown Fix for PT-3654 PFT crashes while generating OEM token for Broxton platform Fox for PT-3657 KBL and LBG - Can't load Part ID bin file 	Magadum, Mahesh	June 12 th , 2019
N/A	5.9.2.1	<ul style="list-style-type: none"> Updates for 5.9.2.1 Added Fastboot dependency files 	Magadum, Mahesh	June 28, 2019
N/A	5.9.3.0	<ul style="list-style-type: none"> Updates for 5.9.3.0 Support for EHL secure debug tokens. Support for TGL secure debug tokens. New knob "CSE Tracing" added to all the platforms. New Knob "Enable Debug Interface". Few other alignments made in knobs with token architecture diagram. Merrifield Moorefield support withdrawn from this release, as these platforms are EOled. 	Magadum, Mahesh	July 26, 2019



Introduction

		<p>Renamed of all the platform acronyms to full names.</p> <p>Changed the Boot Guard options as per the latest implementation and aligned it across all the platform in PFT except Broxton</p>		
N/A	5.9.4.0	<ul style="list-style-type: none"> Updates for 5.9.4.0 Support 3K keys for TGL token signing Added token warning message for Intel and IDLM tokens Upgraded fastboot to 29.0.4 Added libcpp, dependency for fastboot commands into PFT toolchain Upgraded openssl to 1.0.2s Upgraded 7zip to 16.04 Upgraded zlib to 1.2.11 Removed libxml2 from the installer Added separate Security User Guide for OEM version 	Khan, Hamshan	September 24 th 2019
N/A	5.9.5	<ul style="list-style-type: none"> Updates for PFT 5.9.5 Upgraded openssl to 1.0.2t Integrated DnX Version 13.30.0.1052 Added support for CometLake-V platform Excluded fastboot-ethernet.exe from PFT toolchain as it was having critical BDBA issues Upgraded 7zip to 19.0.0 Fix for PT-3705 PID file failed to load for TGL platforms Support for DG1 platform has been added Changed the Jira link to point to new jira url Changed manifest header values for 3K keys signing Intel Bios Payload has been added for TGL Intel tokens Support for a new device type "ufs_hpb" has been added in dnxfwdownloader utility Added remaining flags for LKF whose position is set but are not being used 	Khan, Hamshan	November 29 th 2019
N/A	5.9.6	<ul style="list-style-type: none"> Completed all SDL requirements Upgraded Qt to 5.14.1 Upgraded OpenSSL to 1.1.1d 	Khan, Hamshan	April 3 rd 2020

Introduction



		Added PSIRT email for any security vulnerability reporting		
N/A	5.9.6.3	<ul style="list-style-type: none">Changes made for Ubuntu 18.04 build Removing libicu libraries from ubuntu package as these were having critical BDBA issues Upgrading openssl to 1.1.1g	Khan, Hamshan	May 15 th 2020
N/A	5.10.0.0	<ul style="list-style-type: none">Adding support for 3k signing in CLI using a key-size switch Added ISH GDB Debug and aligned EHL and TGL token knobs Updated FlashUsbDriver to version 1.0.3.0 Added Cancel OEM authentication support to Lakefield Platform Fix for PT-3728 Added options in the DnX Capabilities knob	Khan, Hamshan	June 3 rd 2020



1 Introduction

This document is a technical document that provides instructions on the installation and usage of the Platform Flash Tool for Windows*, Macintosh* and Linux* hosts.

The Platform Flash Tool Users Guide is targeted at engineers using the Platform Flash Tool for flashing firmware, OS image and Modem, pushing data and installing Android* Application Package (APK) files on multiple devices.

1.1 Terminology

Term	Description
APK	Android* Application Package file used for distributing Android* software
OS	Operating System
xFSTK	Cross-Platform Firmware Software Tool Kit
USB	Universal Serial Bus
GUI	Graphical User Interface
PFT	Platform Flash Tool

Table 1-1: Terminology



2 *Platform flash tool prerequisites*

2.1 **Operating Systems**

This tool supports the following OS:

- Microsoft Windows XP (32 bits);
- Microsoft Windows 7 (32/64 bits);
- Microsoft Windows 8 (32/64 bits);
- Microsoft Windows 8.1 (32/64 bits);
- Microsoft Windows 10 (32/64 bits);
- Ubuntu 16.04 LTS 64 bits
- Ubuntu 18.04 LTS 64 bits
- Mac OS X 10.9 (Mavericks)
- Mac OS X 10.10 (Yosemite)
- Mac OS X 10.11 (El Capitan)
- Mac OS X 10.12 (Sierra)

2.2 **Required Drivers**

On Windows, the Intel SOC drivers and the Intel Android USB drivers are required. Those drivers are included in the Intel® Platform Flash Tool package and they will be installed or updated during PFT installation. If newer drivers are required, check the Intel Download Center (http://downloadcenter.intel.com/Detail_Desc.aspx?agr=Y&DwnldID=21217) or contact the Intel Customer team.

3 *Installation*

This chapter describes the installation steps.

3.1 **Installation on Windows OS**

Run the installation package.

Microsoft Visual C++ 2012 Redistributable (x86), Intel® Android USB Drivers and Intel® SOC USB Drivers may be installed by the installer.

Setup wizard will start (see Figure 3-1). Click "Next" to complete the installation.



Figure 3-1: Windows Setup Wizard



3.2 Installation on Linux OS

Only 64bit package is available.

Ubuntu 16.04 LTS 64bits and later

The x86_64 package installs a native 64bits version of Platform flash tool

Prerequisites on Ubuntu 16.04 64bits and later:

```
sudo apt-get install gdebi libncurses5:i386 libstdc++6:i386
```

Installation:

The "Ubuntu Software Center" will handle the installation, double-click on the .deb file and then click "Install Package" (see Figure 3-2) and enter the password. The IPL license must be accepted.

Note: You can also install the package using the command line (replace with the correct name of the deb file):

```
sudo dpkg -i [PlatformFlashTool.deb]
```

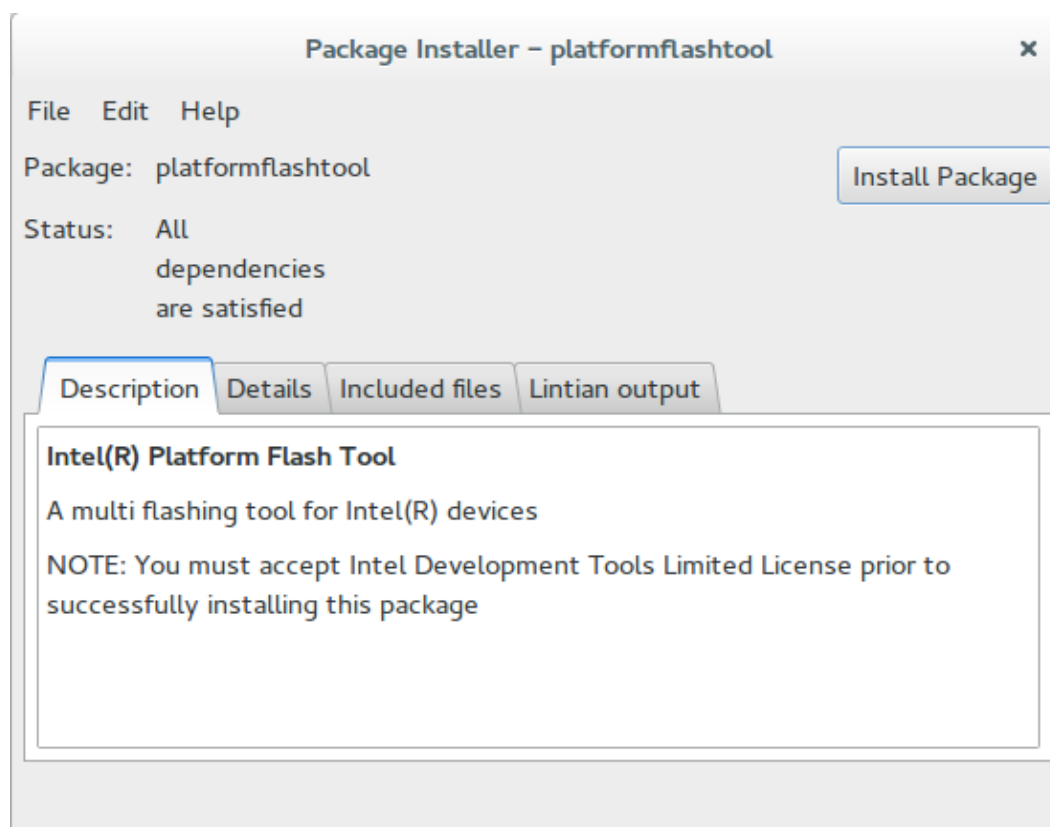


Figure 3-2: Platform Flash Tool Package Installer



3.3 Installation on Macintosh OS

Run the installation package.

Setup wizard will start (see Figure 3-3). Click "Continue" to complete the installation.

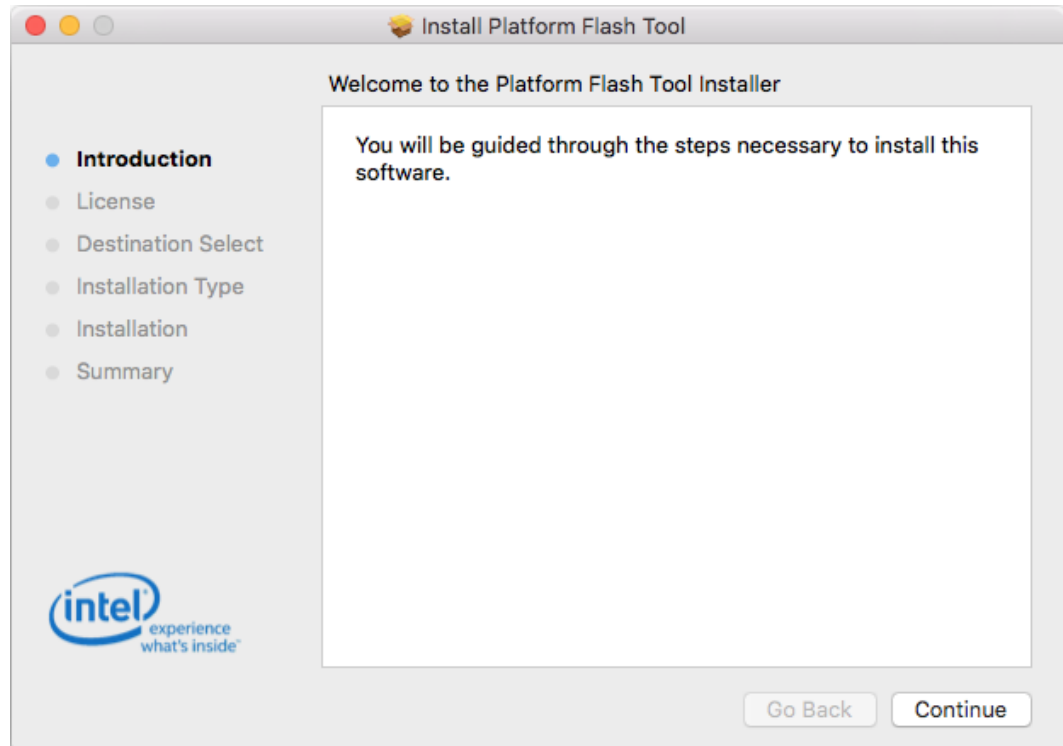


Figure 3-3: Macintosh Setup Wizard



4 Usage

Platform Flash Tool can be used with the graphical user interface or in command line. The first part of this chapter will describe the graphical part. The command line mode is detailed in section 4.8.

4.1 Launching the tool

1. Double-click the desktop shortcut (Figure 4-1).
2. The main GUI of the tool is shown in Figure 4-2.



Figure 4-1: Platform Flash Tool Icon

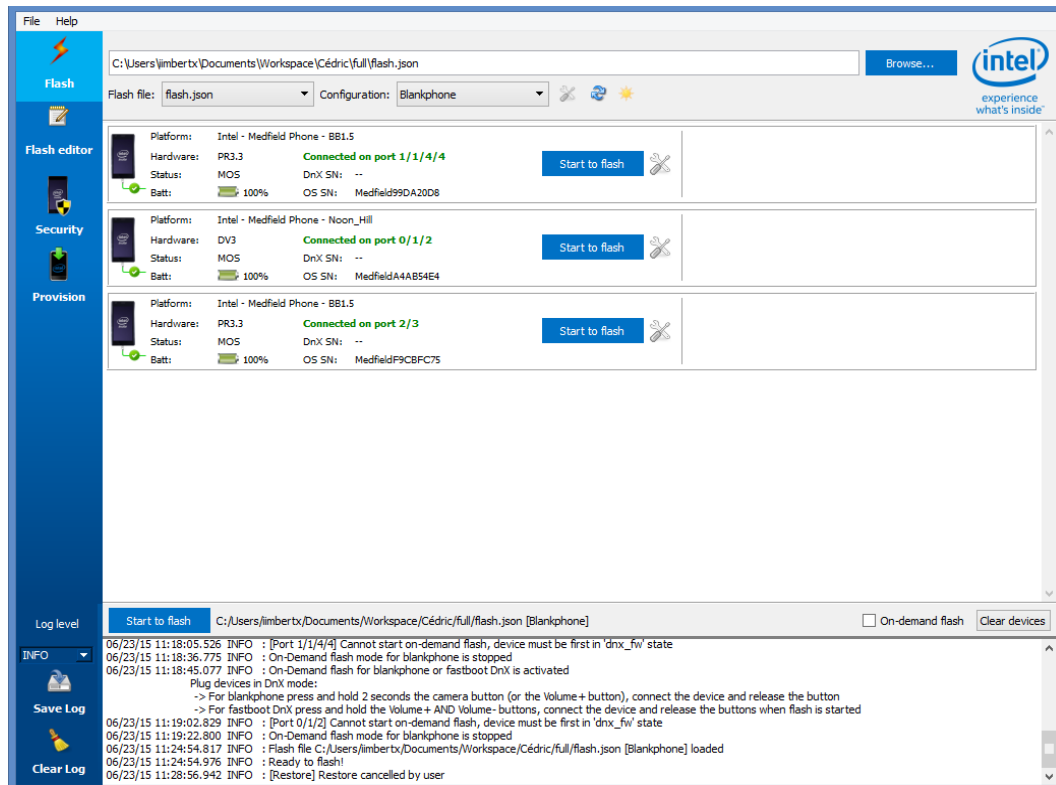


Figure 4-2: Main window of Platform Flash Tool

4.2 Select flashing ingredients

The user has to select the local flash file using the browse button. The flash file validity is then checked, and the flash operation can be started only if the selected flash file is valid. The details of the loaded flash file are printed in the log area in the DEBUG log level.

4.2.1 Use a standalone flash file

You can also specify a local archive *.zip *.tgz file or *.xml or *.json flash file. Select the flash file or the archive using the "Browse" button (Figure 4-3).

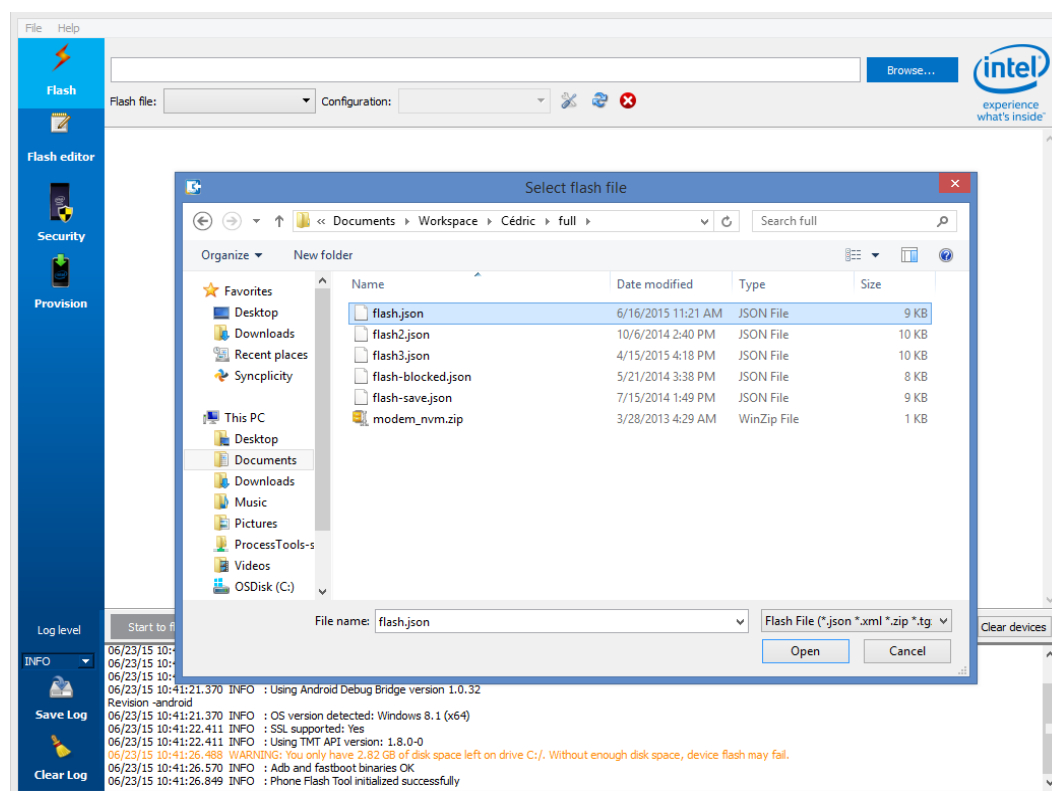


Figure 4-3: Local Flash file tab

4.2.1.1 Flash from a local flash file

When a flash file is selected, Platform Flash Tool checks the file validity and activates the “Start to flash” buttons (Figure 4-4).

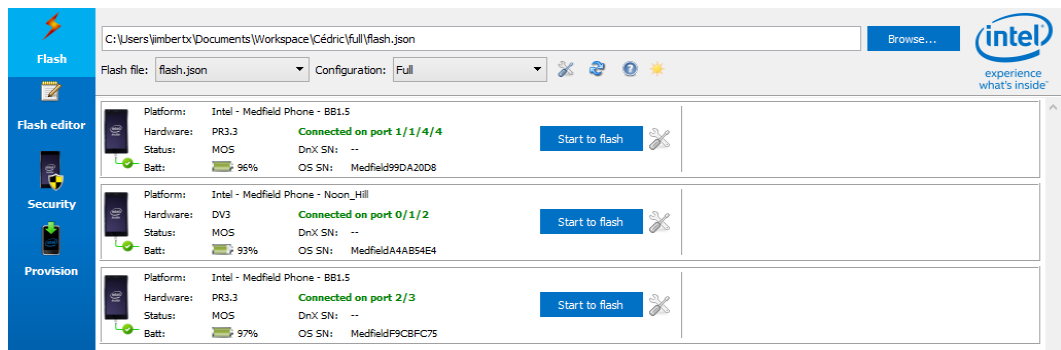


Figure 4-4: Select an xml flash file in the local flash tab

4.2.1.2 Flash from a local archive file

If a flash archive (*.zip) file is selected, you can select the flash file (Figure 4-5). When the file is selected, Platform Flash Tool checks the file validity and activates the “Start to flash” buttons.

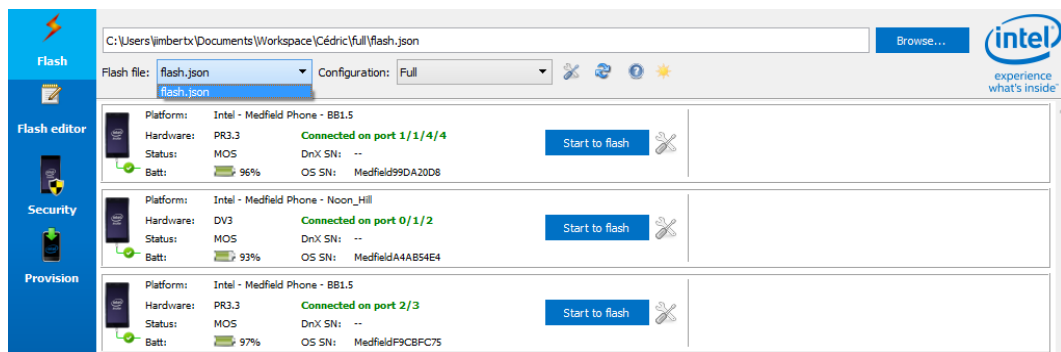


Figure 4-5: Select an archive flash file in the local flash tab

4.2.2 Flash file editor

This section presents the “Flash Editor” tab (Figure 4-6). This tab helps the user to generate a customized JSON flash file.

In this JSON file, the path of the selected files is an absolute path. The generated JSON flash file can be saved with the “Save file” button.



This interface can also load an XML or JSON flash file using the “Load file” button, once the file is loaded the user can then modify the file. If an XML file is loaded, the file will be converted to the JSON format on “Save”.

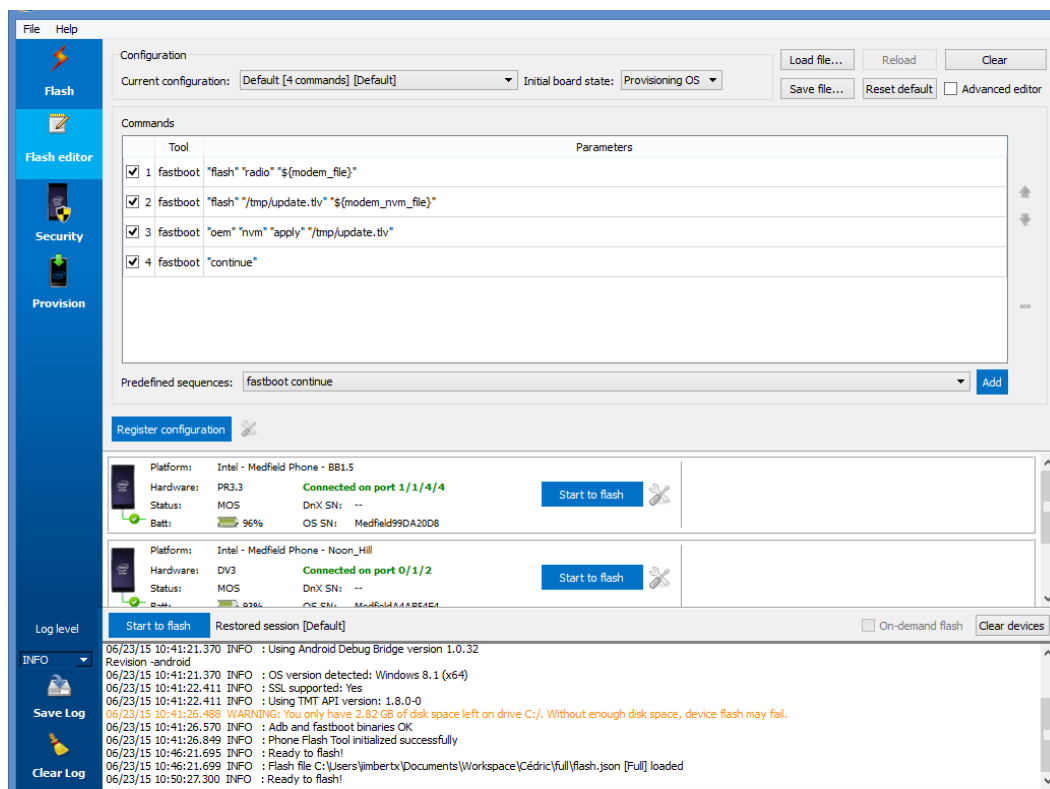


Figure 4-6: Flash editor tab, basic view

4.2.2.1 Basic steps to generate a custom flash file from scratch

First the user must select the start state of the flash configuration (DnX, DnX Provisioning OS, Provisioning OS, Test OS, Main OS ...). The start state corresponds to the state in which the device must be before executing the first command. If the selected device is not in the required start state, Platform Flash Tool will restart the device in the required state.

Details of the main start states:

- **DnX FW:** Flash device from SOC interface (SCU or CSE DnX).
- **Provisioning OS:** Flash device rebooting in android boot loader. The generated flash file will contain only “fastboot” or “adb” commands. This state corresponds to the old “fastboot” flash file type.

Once the start state is selected, the user can add predefined command sequences (see Figure 4-7).

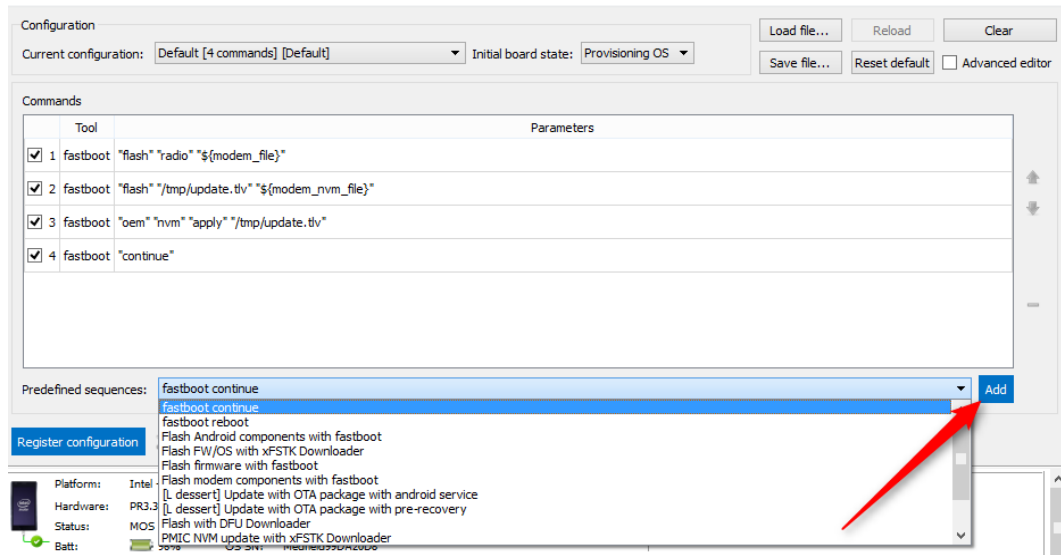


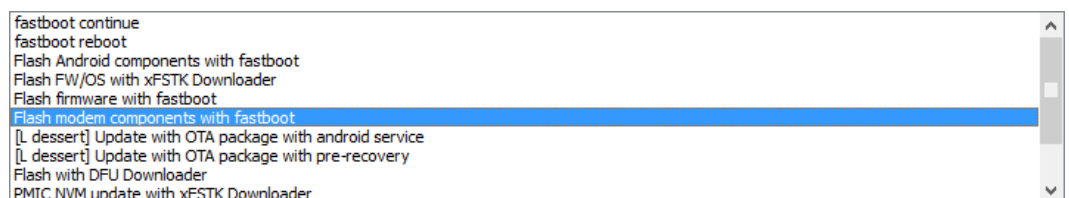
Figure 4-7: Flash editor: Adding a predefined sequence

The parameters of the commands can be edited in the advanced view of the editor (check the box "Advanced editor") and by double-clicking on the corresponding parameter in the table. The commands can also be re-ordered and removed from the list.

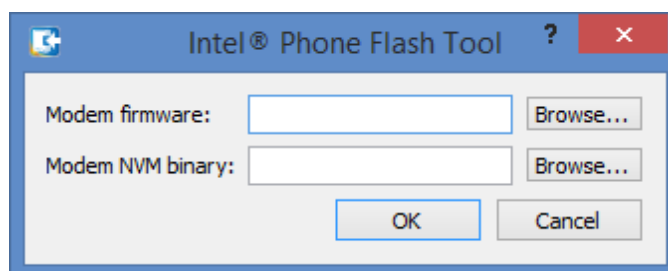
Once the custom flash file is OK, it must be registered (loaded in the internal flash manager of the tool) using the Register flash file button. If the auto-register check box is checked, the flash file is automatically registered.

4.2.2.2 Example: Flashing the modem only

1. Click "**Reset Default**" button if the flash editor contains some data
2. Select the "**Provisioning OS**" start state;
3. Select the "**Flash Modem components with fastboot**" sequence and click on the "Add" button on the right;



4. Fill the component you want to flash in the opened dialog and click **OK**. The required Fastboot commands will be automatically added to the flash configuration;



5. Register the flash file with the **"Register configuration"** button. The sentence "Ready to flash!" is printed in the log area (Figure 4-8);
6. Start the flash operation on the desired device in the same way as the "Flash" tab (see next section 4.3).

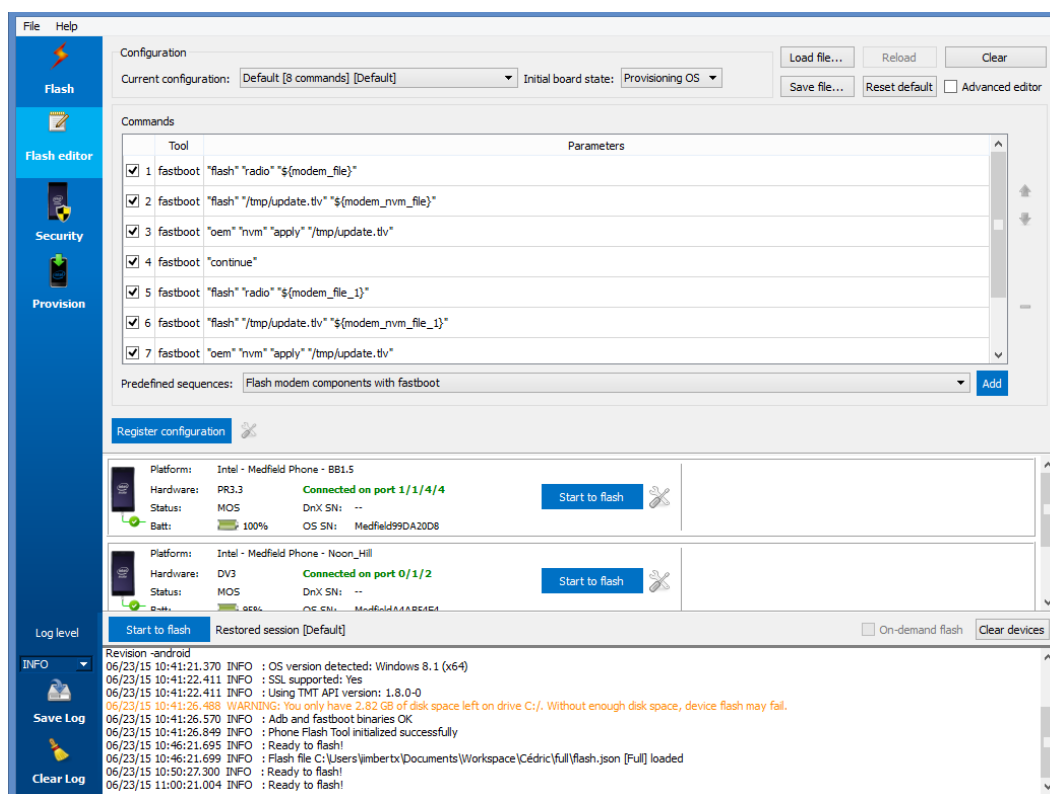
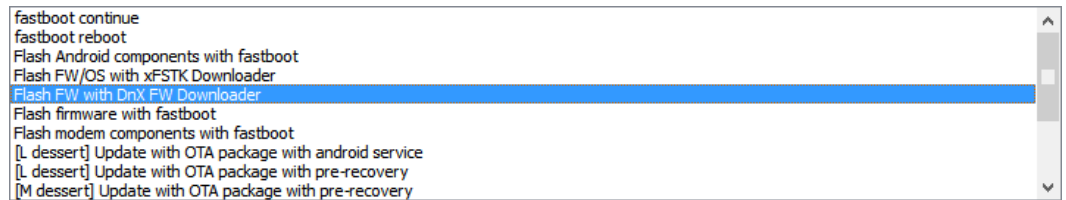


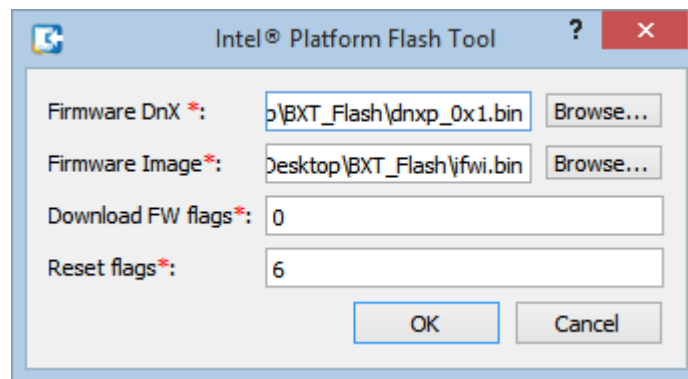
Figure 4-8: Modem only flash

4.2.2.3 Example: Creating Flash file for CSE FW flash with DnX

1. Click **"Reset Default"** button if the flash editor contains any data
2. Set **"Initial board state"** to DNX_FW
3. Select **"Flash FW with DnX FW Downloader"** sequence and click "Add"

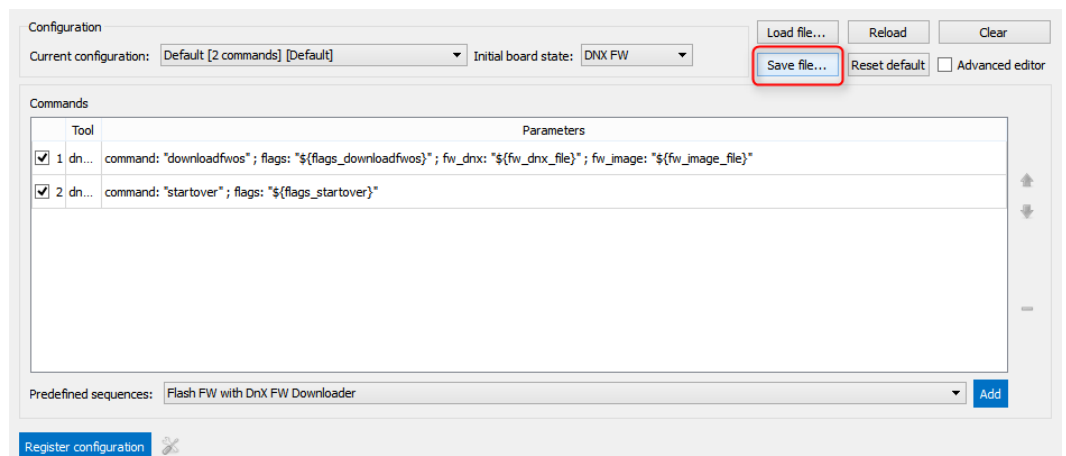


4. After clicking **"Add"**, a new Window will open:



- In the **"Firmware DnX"** field, select the path to the DnX module. The DnX module is part of the IFWI kit, provided by Intel (Usually named "dnxp_0x1.bin")
- In the **"Firmware Image"** field, select the path to the IFWI image you would like to flash

5. Click **"OK"** and then **"Save file"**



6. After saving the file, you can load it in later time to flash the image or, you can click the **"Start to flash"** button to flash the image now

4.3 Start the flashing operation

4.3.1 'Blank' flash configuration

4.3.1.1 Single device

- Case1: The device is plugged in Android (MOS) or provisioning (POS) mode:

Click "Start to flash" in the widget of the device (Figure 4-9);

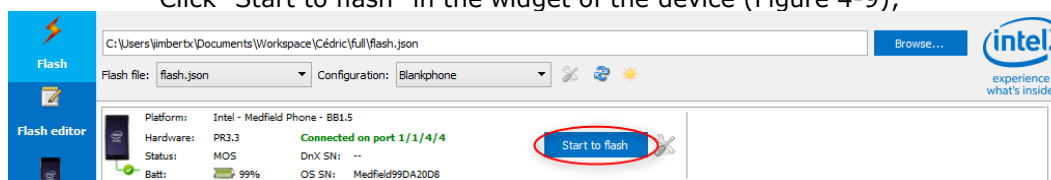


Figure 4-9: Flash blank configuration on a single device in MOS or POS

- Case2: The device will be connected powered off:

First click the main "start to flash" button and connect the powered off device (Figure 4-10).

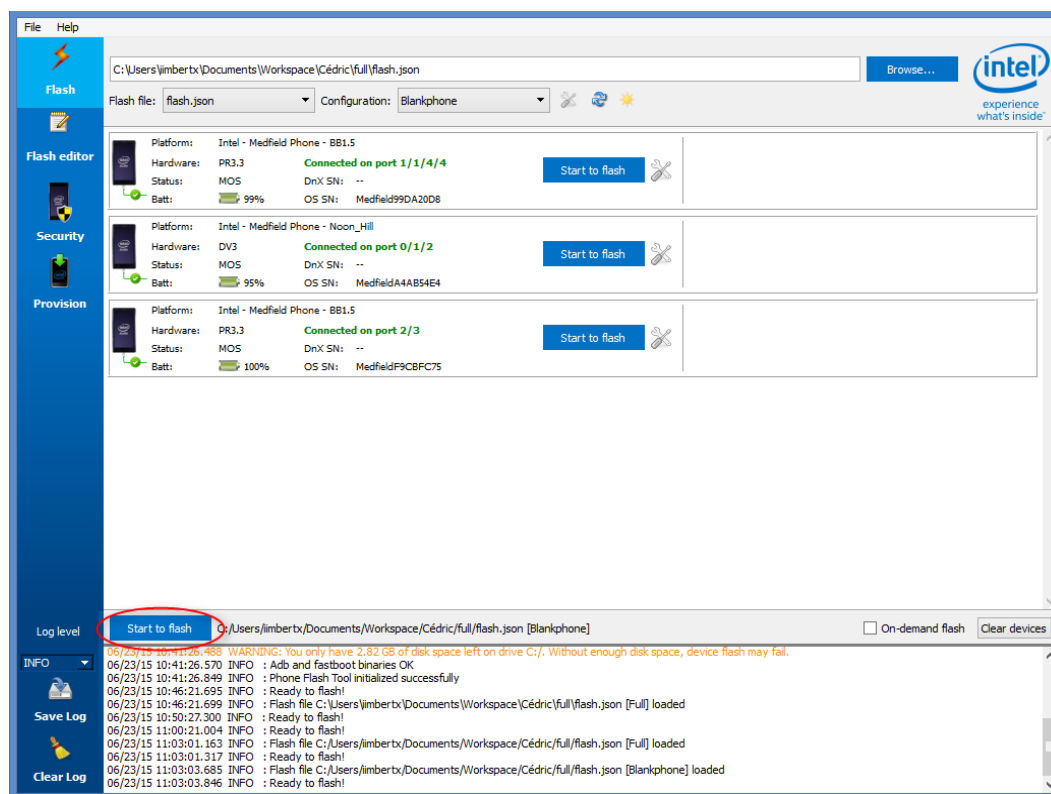


Figure 4-10: Flash blank configuration on a single device powered off



4.3.1.2 Asynchronous multiple devices

For parallel flash with more than four devices, it's recommended to use powered USB hub (up to 8 devices in parallel with two USB hub). Platform Flash Tool is not designed for manufacturing usage.

For multiple flashes, devices must be in infinite DnX timeout:

1. Select the "On-demand flash mode"; (To flash multiple devices simultaneously, On-demand flash needs to be checked. Else, flashing will be sequential)
2. Click the main "Start to flash" button;
3. The tool is waiting for new device in infinite DnX mode (on BB1.5/PRX, press and hold 2 sec. the camera button (or Volume UP for CTP), connect the device and release the camera button);
4. When a device is plugged, the flash procedure starts automatically (Figure 4-11).

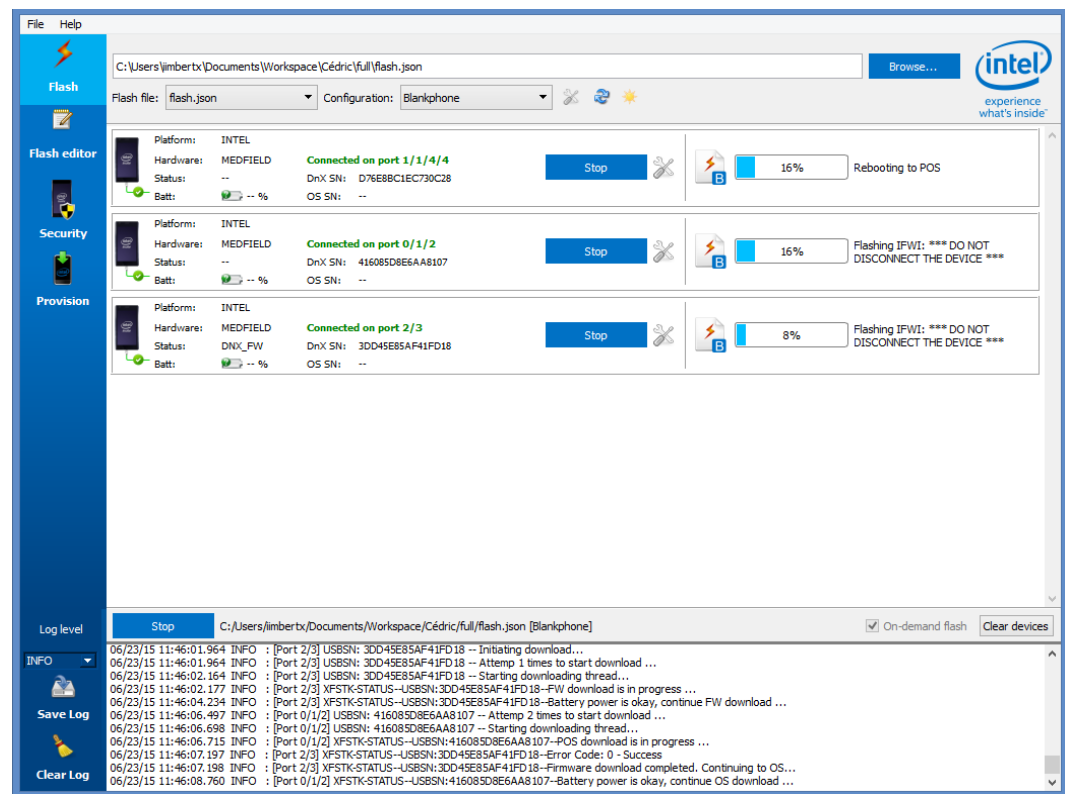


Figure 4-11: Flash blank configuration in parallel

4.3.2 Fastboot or OTA flash file

- Method 1: Click "Start to flash" in the widget of the device;
- Method 2: Click the main "Start to flash" button: that will start flash of all connected devices.

4.4 Secure Tokens

Secure tokens are handled through the "Security" tab, as shown in Figure 4-12. A new "Security" menu will appear in the main toolbar.

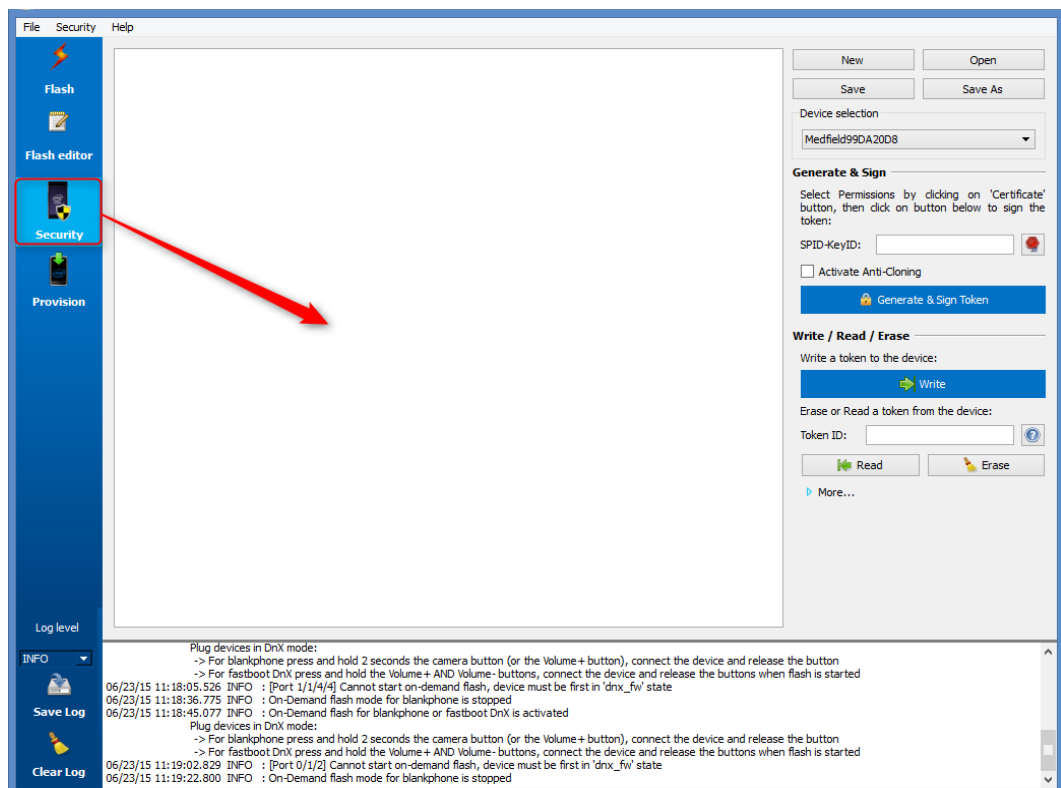


Figure 4-12: Security tab



For more details on how to use secure tokens, please refer to specific user guide, accessible from 'Security' menu (see Figure 4-13).

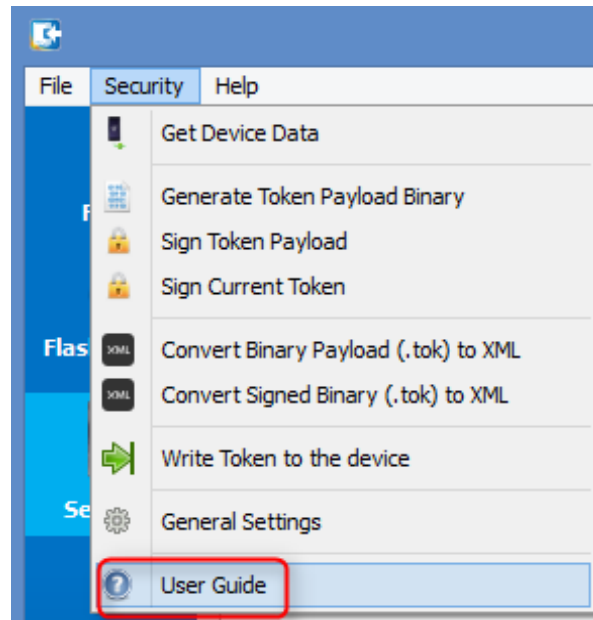


Figure 4-13: Access Security user guide



4.5 Push Data and Provisioning

After flashing, now we can install APKs and push data to the device. Click “Provision” button on the left panel, the Android* devices are listed in the “Android devices Connected”. Add your desired APKs and the local data files to the list and click “Install Data and APK” (see Figure 4-14).

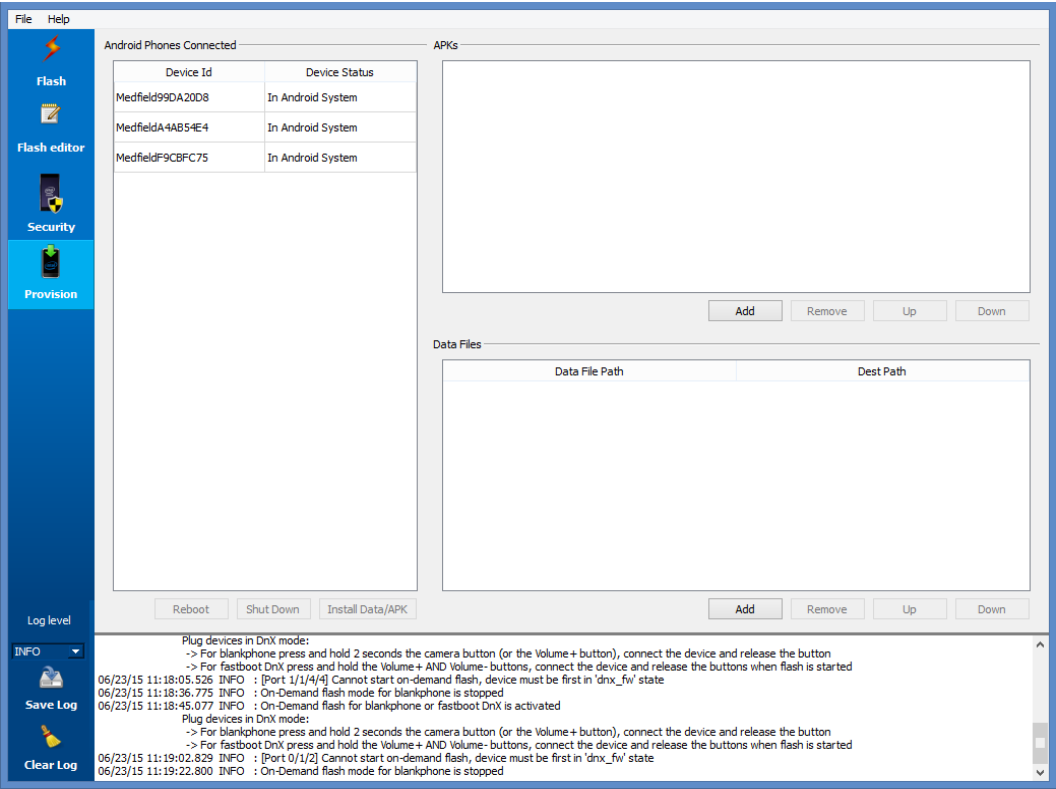


Figure 4-14: Provisioning tab

4.6 Backup and restore

This section describes how to backup or restore Intel Android devices. This feature is only available in the GUI tool. The device status must be "MOS" (android booted).

4.6.1 Backup procedure

1. Right click on the "device" icon of the device's widget (see Figure 4-15);

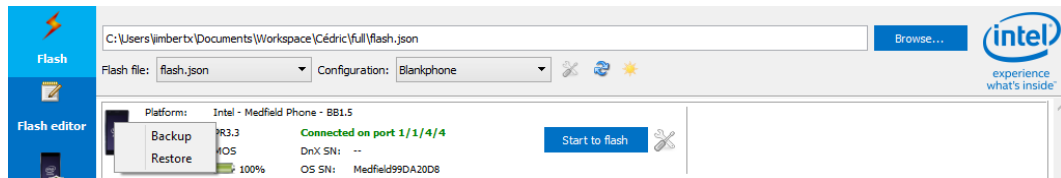


Figure 4-15: Backup/Restore contextual menu

2. Select the "Backup" action, a dialog box appears (see Figure 4-16);

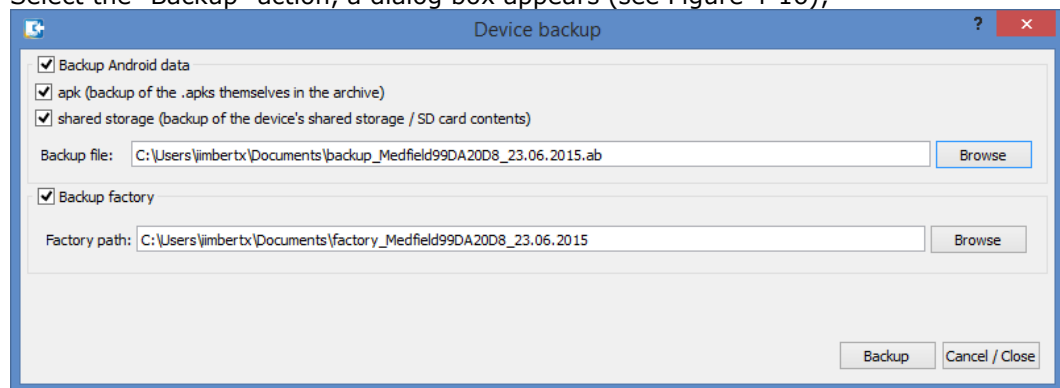


Figure 4-16: Device backup window

3. Click on the "Backup" button and press *"Back up my data"* on the device side to start the backup process (Figure 4-17).

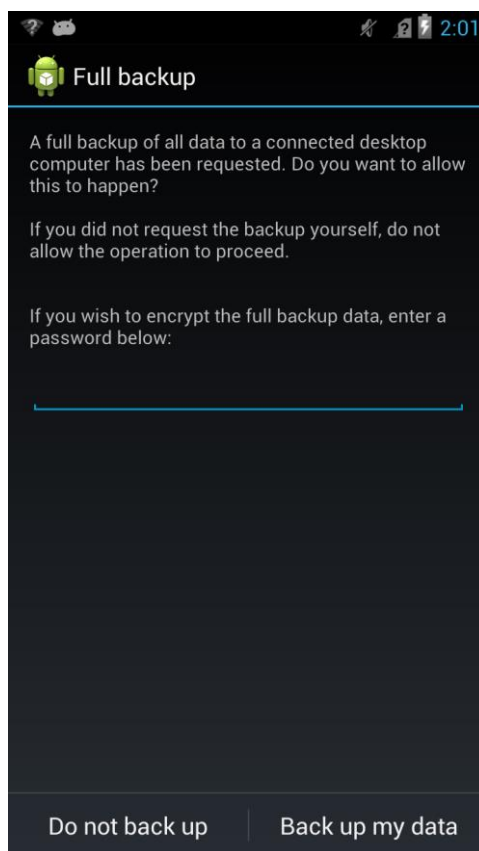


Figure 4-17: Full backup

4.6.2 Restore procedure

1. Right click on the "device" icon of the device's widget (see Figure 4-15);
2. Select the "Restore" action and then the backup file. Note that the latest backup file used is already selected (see Figure 4-18);

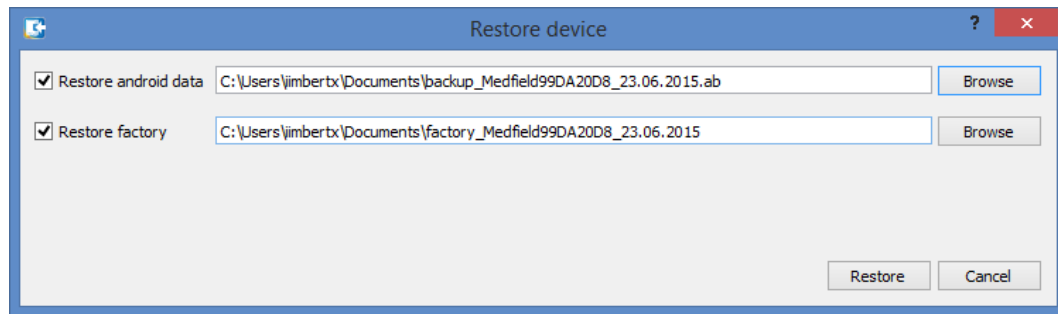


Figure 4-18: Device restore window

3. Click on the "Restore" button and press "*Restore my data*" on the device side to start the restore process (Figure 4-19).

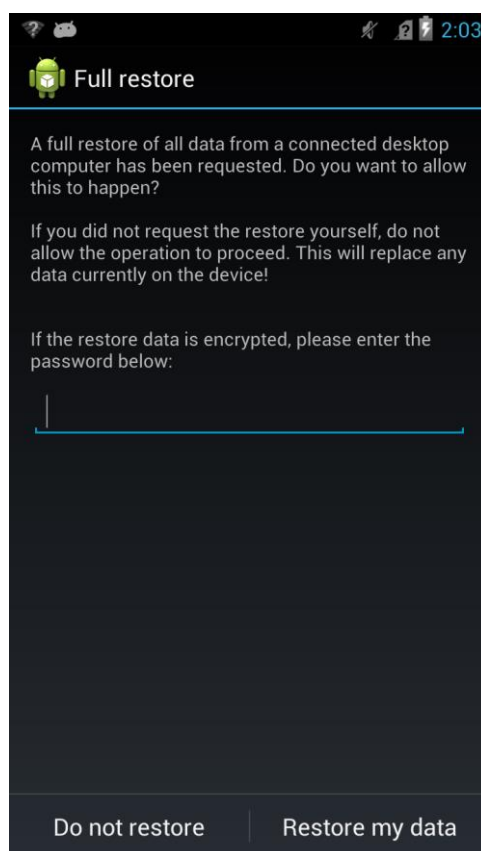


Figure 4-19: Full restore

4.7 Tool Options

This section describes the options of the tool with the Graphical User Interface (GUI). The option panel is accessible via the main menu File → Options.

The option windows contains a General tab (Figure 4-20), an external tool tab (Figure 4-21), a Result Log tab (Figure 4-22) and an Advanced tab (Figure 4-23).

4.7.1 The General tab of the option window

In this tab, the user can configure the directories used by Platform Flash Tool and the path of the fastboot and adb binary files.

"Always deflate local flash archives" option:

- If the "Always deflate local flash archives" is checked, the zip files always will be unzipped by the tool.

"Reload flash file at startup" option:

- If checked, the latest loaded flash file in the local tab is reloaded at tool startup (if it still exist).

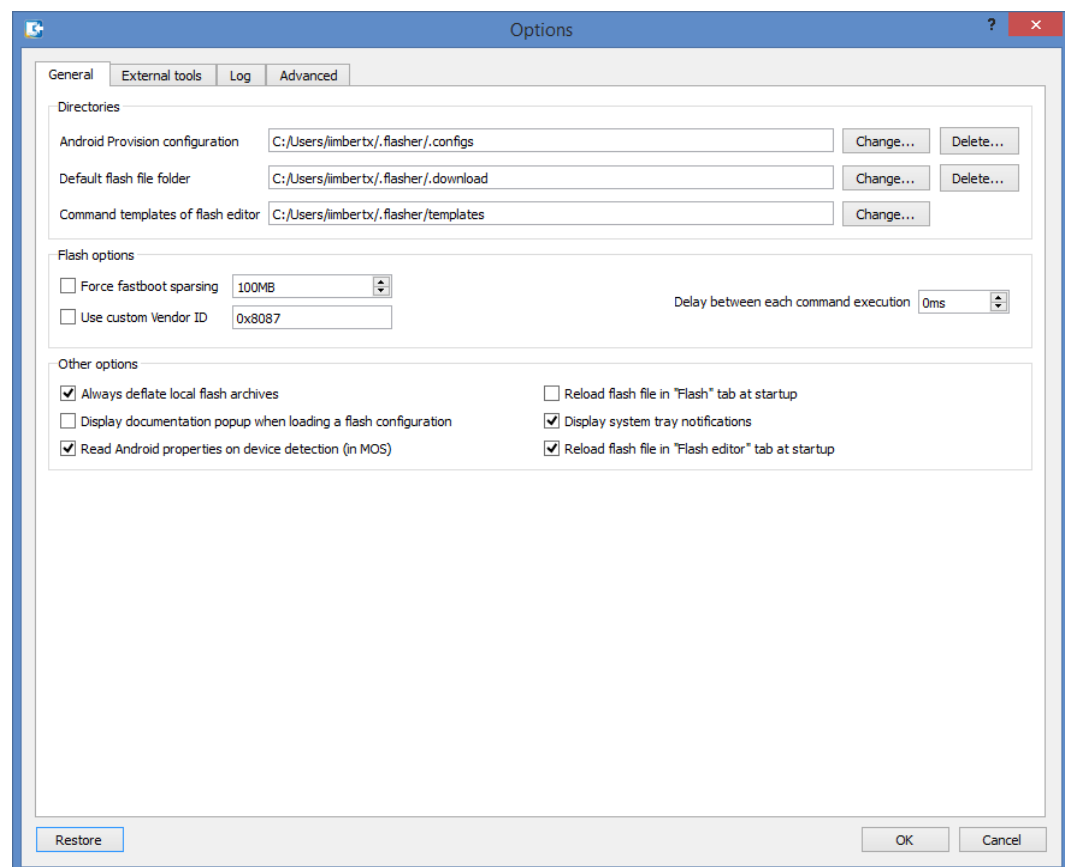


Figure 4-20: General tab of the option window



4.7.2 The external tool tab of the option window

In the tab, the user can set the path of the fastboot and adb binary files and also add custom tool binaries that can be used in JSON flash files or in the flash editor.

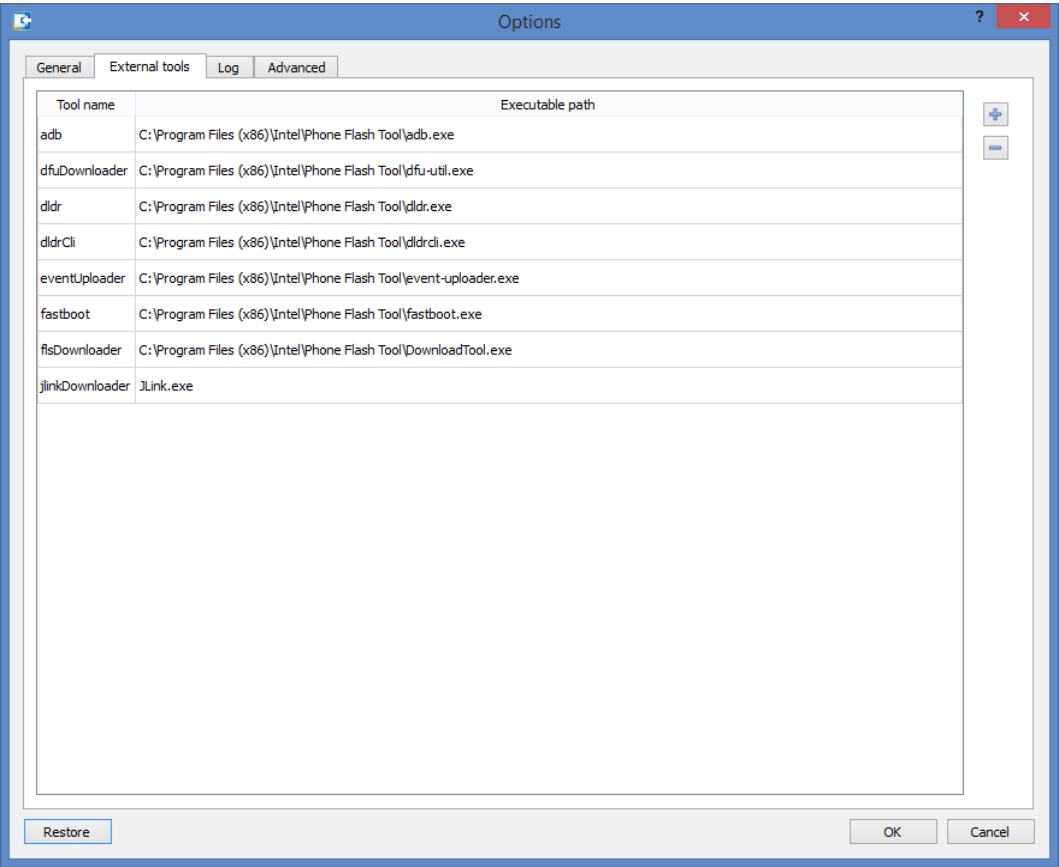


Figure 4-21 External tools options

4.7.3 The Log tab of the option window

This tab is used to customize the result log file of each flash and the application log file.

All fields except “Log directory” have a substitution feature (more details in 0 section below).

The “Log directory” option allow you to set the root directory of all logs files (Flash result log files and Application log files)

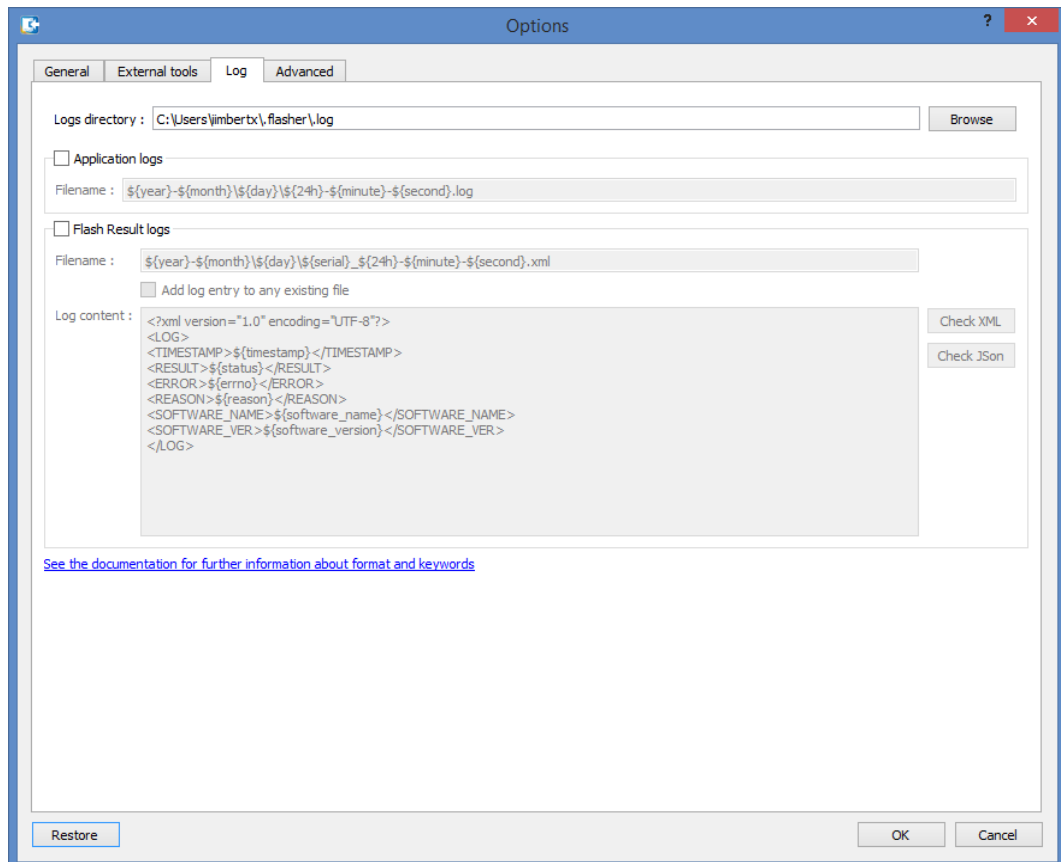


Figure 4-22: Log tab of the Options window



4.7.3.1 Flash result logs

You just have to check the "*Flash Result logs*" checkbox to enable the logger (see Figure 4-22).

The date and time used in the substitution is the end of flash time.

"*Filename*" option:

- The template filename of the log files

"*Add log entry to any existing file*" option:

- If checked, if a log file already exists the log will be appended at the end of the file. Otherwise, the existing file is erased when the new log is written.

"*Log content*" option:

- The template log file content, supported in xml & JSon format. Valid format can be verified by clicking on Check XML or Check JSon respectively.

4.7.3.2 Platform Flash Tool logs

When this option is enabled, all the log entries are stored in the file pointed by the Filename field. The "Log level" drop-down list in the main window (see Figure 4-2) does not filter the entries of this file.

You just have to check the "*Application logs*" checkbox to enable the logger (see Figure 4-22). Use the "*Filename*" option to set the template filename of the log file.

The date and time used in the substitution is Platform Flash Tool start time.



4.7.3.3 The substitution feature

Each variable begins with a dollar followed by a keyword that contains only alpha numeric characters and the underscore character. All unknown substitute keywords will not be touched.

To remove any ambiguity, surround the keyword by curly brackets.

Keyword	Description	Context	Substitute result examples
\$	Dollar sign. Use this to escape the dollar sign	All	\$
errno	Error number Different of zero if the flash failed	Result logs	0
command_response	Last ran command response	Result logs	Flash success
reason	Reason of the flash ending	Result logs	INTERRUPTED SUCCESS
status	"PASS" if the flash succeeded "FAIL" if the flash failed	Result logs	PASS FAIL
soc_serial	SoC serial number If not available, the value is 'unknown_soc'	Result logs	D6BE922F464FA94B
android_serial	Android* serial number If not available, the value is 'unknown_os'	Result logs	RHBEC244201465
day	Day (2 digits)	All	05
month	Month (2 digits)	All	04
year	Year (4 digits)	All	2013
24h	Hour (2 digits, 24 hours format)	All	17
12h	Hour (2 digits, 12 hours format)	All	05
ampm	"AM" or "PM" value depending on the hour	All	AM PM
minute	Minutes (2 digits)	All	09
second	Seconds (2 digits)	All	07
msecond	Milliseconds (3 digits)	All	015
timestamp	Number of seconds since the epoch (01-01-1970 00:00:00 UTC)	All	1368632654
software_name	Name of the flash software	All	Intel Flash Platform Tool
software_version	Version of the flash software	All	4.1.0-0

Table 4-1: List of keyword for the logger options

4.7.4 The Advanced tab of the option window

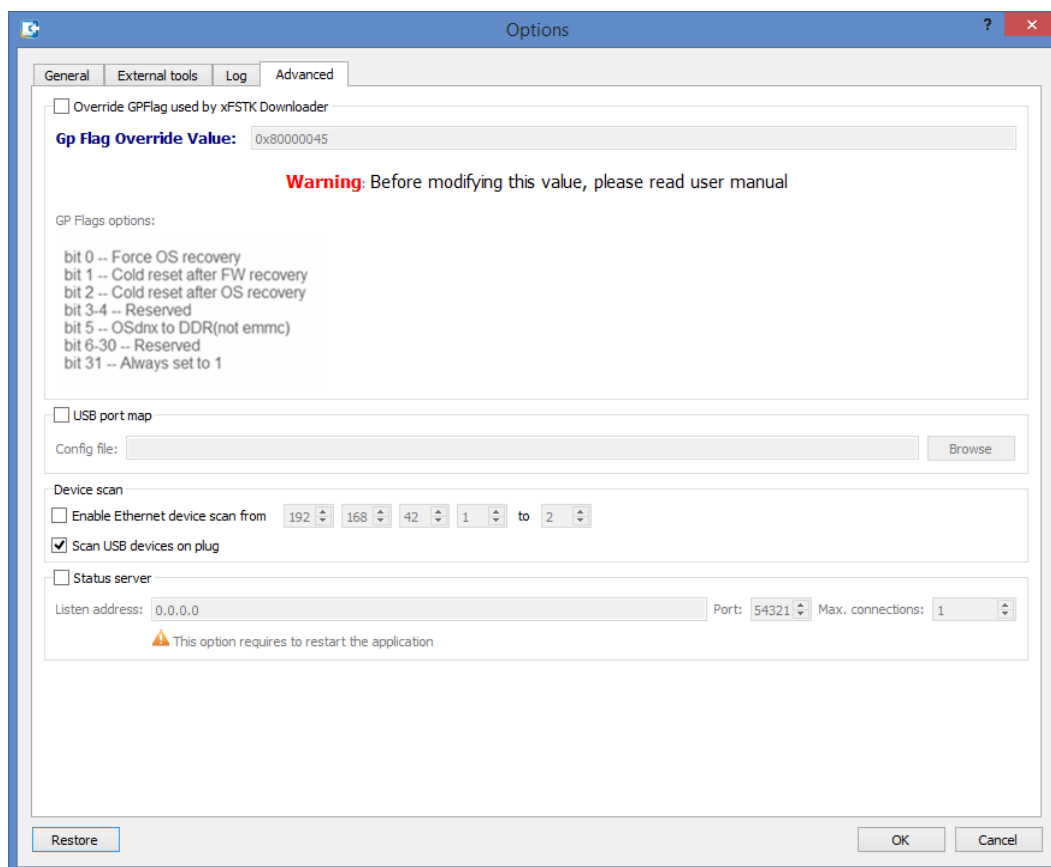


Figure 4-23: Advanced tab of the option window

4.7.4.1 GP flags override setting

GP flag is very important in FW/OS flashing, it specifies overwrite mode during flashing.

If you are flashing a zipped image file, this flag is stored in a flash .xml file, but you still can configure your own flag for your own purpose, you can choose to use GP flag in flash xml file, or use GP flag you custom.

4.7.4.2 USB port map settings

This option provides a workaround for the USB3.0 support for Windows 7. During the "blank" configuration flash with a USB3.0 cable connected on a USB 3.0 port of a USB3.0 ready device, there is a protocol switch (from USB 2.0 to USB 3.0) because the ROM code only support USB 2.0. To avoid this problem, Platform Flash Tool is able to load a text file generated manually by the user that describes the USB port map.

USB map file example:



```
[USB_MAP]
7=3
8=4
```

Logic:

[USB_MAP]

```
USB3_PORT_ID_1 = USB2_PORT_ID_1
USB3_PORT_ID_2 = USB2_PORT_ID_2
```

How to get the port ids?

Plug a device in MOS or POS with a USB 3.0 cable on a USB3.0 port. The port id is displayed in the platform flash tool logs in the *DEBUG* level:

```
10/02/13 15:42:14.760 DEBUG : Android device detected -> serial: RYHDNOD465 status
(id=4): POS_STATUS port: 1/4 (port_id=72057594037927955) type: USB 2.0
```

Then replace the USB3.0 cable by a USB2.0 one. The new port id should be printed in the platform flash tool logs.

4.7.4.3 Device scan

By default, Ethernet device scan is deactivated to prevent useless scan. In figure 4-23, the scan will be performed from the IP 192.168.42.1 to 192.168.42.2

On Windows, the "Scan USB devices on plug" will detect your devices when you plug it. This reduces the detection time.

4.7.4.4 Status server

Platform Flash Tool can act as an event publisher. If this feature is enabled, a unidirectional TCP server will be started by Platform Flash Tool.

In Platform Flash Tool, the server will start if the "Status server" is enabled at startup. This settings do not affect the command-line behavior.

The "Listen address" field value can be:

- "0.0.0.0" to allow clients to contact server on all active IPv4 interfaces
- "127.0.0.1" to allow clients to contact server only on IPv4 loopback interfaces
- "::" to allow clients to contact server on all active IPv6 interfaces
- "::1" to allow clients to contact server only on IPv6 loopback interfaces
- Any valid IPv4/IPv6 address allocated to one of the network interfaces
- Any DNS name that resolves into an IPv4/IPv6 address allocated to one of the network interfaces

The port number can be any TCP port number. Be aware that the operating system will refuse application started by low-privileged users to listen to port from 1 to 1024. You might have to use port number from 1025 to 65535.

You can prevent access to more than "Max. connections" active clients.



The messages sent by Platform Flash Tool are JSON objects. They are separated by a null character. You can pick the format in table 4-2.

The complete event list with the event-specific parameters are listed in table 4-3.

```

message ::= {
  ["devices" : <device-map> ,]
  "event" : <event-object> ,
  "software" : <software-version> ,
  "version" : <message-structure-version>
}

device-map ::= { [ <device-map-element> [(, <device-map-element>)*]] }
device-map-element ::= <device-port-id> : <device-object>
device-port-id ::= <number>
device-object ::= {
  "battery" : <number> ,
  "flashStatus" : <string> ,
  "flashing" : <boolean> ,
  "hardware" : <string> ,
  "hardware_id" : <string> ,
  "osSerial" : <string> ,
  "osStatus" : <os-status-enum> ,
  "platform_family" : <string> ,
  "platform_family_id" : <string> ,
  "product_line" : <string> ,
  "product_line_id" : <string> ,
  "socSerial" : <string> ,
  ("usbPath" : <string> , | "ip" : <ip-address> ,)
  "vendor" : <string> ,
  "vendor_id" : <string>
}
os-status-enum ::= "MOS_STATUS" | "COS_STATUS" | "POS_STATUS" | "ROS_STATUS" | ""
ip-address ::= <string>
event-object ::= {
  ["device" : <device-port-id> , ]
  "id" : <event-id> ,
  "time" : <string>
  [(, "event-param-name-1" : "event-param-value-1")*]
}
event-id ::= <string>

```

Table 4-2: Status server messages structure (version 1.0)

Legend:

Square brackets: elements inside the brackets can be omitted

Round brackets: grouping elements

Less-than/greater-than "brackets": type of element (non-native types are depicted in the file)

::= sign: Type definition

Pipe (vertical bar): Choice, left part or right part?

Star: element can be repeated (1 to N times)

Everything else is fixed data, space might be removed accordingly to JSON format specification.

The field "time" of the event-object is an ISO 8601 compliant string.

The event-id values are listed below in table 4-3.



Event identifier (event-id)	Description	Device related	Device list	Event parameters	
				Name	Type
flashStarted	Flash starts	✓	✓	type	enum (Blank, update...)
				components	map of filenames
ifwiFlashStarted	IFWI flash starts	✓	✓		
ifwiFlashFinished	IFWI flash ends	✓	✓	success	boolean
flashProgress	Flash progression	✓	✓	done	number
				total	number
				deviceStatus	string
flashFinished	Flash ends	✓	✓	error	number (0 = success)
				message	string
flashStopped	Flash stopped by user	✓	✓		
deviceDetected	New board detected	✓	✓		
deviceRemoved	Board lost	✓	✓		
deviceUpdated	Board data updated	✓	✓		
deviceReset	Board data reset	✓	✓		
newAndroidDeviceReady	Device on Android* ready to be flashed	✓	✓		
newSocDeviceReady	Device on SOC ready to be flashed	✓	✓		
batteryTooLow	Battery too low	✓	✓		
connectionSucceed	Connection succeed	-	✓		
connectionFailed	Connection failed	-	-	message	string
connectionClosed	Platform Flash Tool is exiting	-	-		

Table 4-3: Status server event list

Device-related column indicate if the device identifier is present in event object

Device list column indicate if the map of devices is present in the message object



4.8 Using the tool in command line

On windows OS the binary file for the command line tool is the *cflasher.exe* file. On linux OS, *cflasher* is a script that executes the "*platformflashtool --cli*" command.

The -f option is a mandatory option, this command line tool is designed to flash a single device. Multiple instance of the *cflasher* tool can be launch in parallel for multi flash but in this case the user has to provide the android serial number (example: Medfield95A45IE7) and/or the SOC serial number with the respective options --os-sn and --soc-sn.

When *cflasher* is executed without parameter, the following help message is displayed:

```
Usage of the UI tool:
    platformflashtool [option]
Options:
    -h
    --help
        Display this information
    --version
        Display the Platform Flash Tool version
```

```
Usage of the UI tool:
    platformflashtool [option]
Options:
    -h
    --help
        Display this information
    --version
        Display the Platform Flash Tool version

Usage of the CLI tool:
    cflasher [options]

General options:
    -h
    --help
        Display the CLI information
    --version
        Display the Platform Flash Tool version
    -l <level>
    --log-level
        Set log level, [0-5]
        0: no log, 3: including xfstk log, 5: including all debug info
    --timeout <timeout_value>
        Set the timeout value in minutes for CLI, if not specified the default value
        is 30min.
```



After <timeout_value> minutes elapsed, all ongoing flashes are stopped and the tool exits with 0x4 code

If <timeout_value> <= 0, the timeout is deactivated

--non-interactive
Prevent Platform Flash Tool to prompt user

Flash options:

- f <file>
Flash platform by .zip/.tgz/.xml/.json file
- x <file>
--flash-filename <file>
(Optional)specify the flash file in archive, default is flash.json
- c <configuration-id>
--configuration <configuration-id>
(Optional)specify the flash configuration ID to be selected in the flash file
- enable-group <group-id>
(Optional)Activate the flash group ID (Group of commands are only supported in Json flash file)
- disable-group <group-id>
(Optional)Deactivate the flash group ID (Group of commands are only supported in Json flash file)
- soc-sn <serial>
Soc serial number for flashing specific SOC devices (flash configuration with DNX FW start state)
- os-sn <serial>
OS serial number for flashing specific Android devices
- adb <path>
Set adb.exe file path
- fastboot <path>
Set fastboot.exe file path
- always-unzip
Always extract archives
- clean-emmc --fw-dnx <fw_dnx> --fw-image <fw_image>
Clean the eMMC of the specified soc device in --soc-sn (xFSTK wipe out ifwi option)
- fru-value <hex_value>
Set the FRU hexadecimal value (mandatory for FRU flash files)
- reboot <value>
Set the reboot behavior before starting the flash procedure (flash configuration with DNX FW start state)
 - 1 : auto, reboot is executed only if a device is connected
 - 0 : no reboot: even if a device is connected, the reboot cmd is not executed
 - 1 : force reboot, the "adb reboot" and "fastboot reboot" cmds are always executed at startup

Board status server options:

- status-server [<server-address>]
Starts the board status server
If <server-address> is not provided, the server will listen on all active IPv4 interfaces



--status-server-port <server-port>

Set the server port

If this option is not provided, the server will use the default port 54321

*** WARNING: Port numbers up to 1024 require elevated privileges. Use port number between 1025 and 65535 to prevent it. ***

--wait-client-timeout <value>

Optional, when used the tool will wait for a client during <value> ms to be connected to the status server.

If the timeout is reached without client connection, an error is raised and the tool exits.

--status-server-max-clients <clients-number>

Set the number of allowed clients

If this option is not provided, the server will use the default value 1

Single-action options:

*** The options in this section immediately exit. You have to use them one by one without any other options. ***

*** It must *NOT* be used with flash commands otherwise they will be ignored. ***

--enable-ethernet <value>

Enable ethernet device support if <value>=1 or disable it if <value>=0

This setting is persistent.

--set-ethernet-ip <value>

Set the ethernet ip used during the ethernet device scan to <value>

This setting is persistent.

--purge-cache <age>

Delete all downloaded flash files not used in the last <age> days.

-If <age> is positive, only the archive files and their extracted folder will be deleted if any xml flash files have been accessed since the requested age

-If <age> is zero, the whole download directory will be cleaned including any non-downloaded files

-If <age> is negative, an invalid argument error will be raised

--get-error-code <value>

Return the error code description



4.9 Embedded command line tools

Platform Flash Tool comes with command line tools that are used during flash sequence. Some of them are specific to certain Intel platforms. Each tool is identified by a key name to be used in JSON flash files.

Some tools can be used separately in command line.

Here is the list of available tools within PFT package:

Tool name	Key name in JSON	Usable in command line
ADB	adb	Yes
Fastboot	fastboot	Yes
DownloadTool	flsDownloader	Yes
xFSTK	xfstkDownloader	No
DFU	dfuDownloader	Yes
DnX Firmware Downloader	dnxFwDownloader	Yes

4.9.1 DnX Firmware Downloader

This command line tool provides means to interact with Intel® CSE firmware (Broxton platform). It allows configuring partitions, flashing, getting information, etc.

Usage is like:

dnxFwDownloader --command [command] [options]

List of options varies from one command to another, but some are used for all of them:

Option	Description
--sn	USB device serial number

For all command's description, please use '*dnxFwDownloader --help*'.
For more details, please refer to CSE FAS ("DnX Support" appendix).

Main commands supported can be used as follows.

4.9.1.1 Configure partitions

In order to configure emmc and/or ufs partitions of DnX supported platforms, '*configpart*' command shall be used.



Options needed are:

Option	Description
--fw_dnx	path to the firmware recovery module
--path	path to configuration file (XML format)
--device	device type (emmc = 2, ufs = 3)
--idx	device index (optional)

4.9.1.2 Flash firmware

dnxFwDownloader can also be used to flash the firmware image to the platform. 'downloadfwos' shall be used for that.

Options needed are:

Option	Description
--fw_dnx	path to the firmware recovery module
--fw_image	path to the firmware image
--flags	firmware download command flags (optional)
--os_dnx	path to OS recovery module (optional)
--os_image	path to OS image (optional)

4.9.1.3 Reset the platform

Along with 'downloadfwos' command, the 'startover' command can be used to reset the platform to a particular state.

Options needed are:

Option	Description
--flags	command flags

4.9.1.4 Get token part ID

'gettokenpid' command is used to get hardware information from the platform.

Options needed are:

Option	Description
--fw_dnx	path to the firmware recovery module



--flags	Flags (0 = no anti-replay protection needed, 1 = anti-replay needed) (Optional)
---------	---

4.10 Updating the tool

Please contact our Customer Support Center

4.11 Edit USB Scan interval

PFT uses USB port scanner to scan the devices connected to the host and lists them in the flash tab. The scanner continuously scans after a set interval of time, which can be configured from the General tab of the Options menu.

File > Options > General

Default value of USB scan interval is 2000ms (2 seconds).

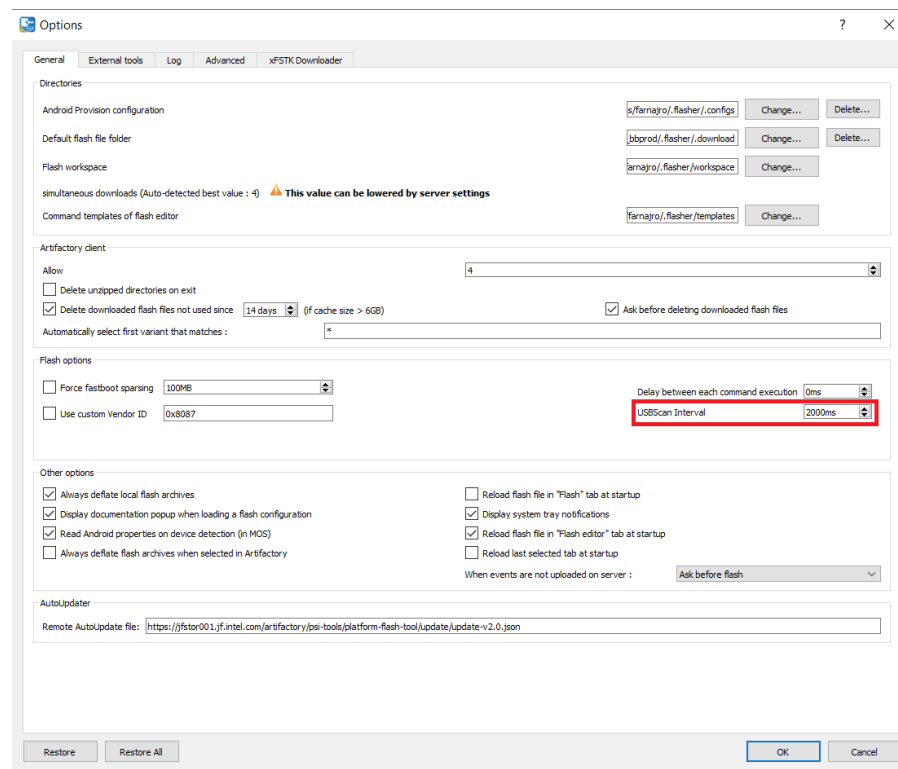


Figure 4-24: General tab of the options window for configuring USB scan interval

Note: User must **Restart** PFT for the changes to get reflected.



5 FAQ

5.1 PID/VID

Intel devices are using Intel PID/VID, which requires driver installation (see Installation chapter).

1. On Linux, if you install Platform Flash Tool, everything is handled by Platform Flash Tool, no actions are required.
2. On Windows, you need to install new xFSTK iSOC driver and new Intel Android Platform driver, please contact your Intel UMG partners for more details.

5.2 Conflicts with other USB devices

The fastboot tool may fail to recognize Intel Android devices (command "fastboot devices" does not end). This problem can happen when other USB devices like webcams are connected.

One solution is to remove the extra devices and/or reconnect USB to another USB port.

5.3 Fail to bind status server

On Windows, Platform Flash Tool may fail to release the status server port. This happens when ADB server is not already running.

To prevent this, you must launch ADB manually by running "adb start-server" in a command line.

If you encounter this issue, you can recover the feature by running "adb kill-server" in a command line.