Quick Start Guide

Alif Security Toolkit

Table of Contents

1. Introduction

The Alif Security Toolkit (SETOOLS) contains several tools (written in Python3) required to program and provision an Alif Semiconductor SoC device.

Provisioning means being able to add images to the internal NVM storage (MRAM) as well as adding Security assets such as Keys to the One Time Programmable memory (OTP).

ALIF firmware images, known as System TOC (STOC), for the device are already provisioned into the MRAM, a user cannot change these, but can update to a new STOC version supplied by Alif. User images such as Application programs (binaries written for A32 or M55 cores) need to be packaged and written to the MRAM. The packages are called an Application Table of Contents (ATOC).

There are also pre-built binary images for the System Table of Contents (STOC) Package. The STOC contains the latest binary version of SERAM. The release also contains debug stubs that can execute on the Application cores to allow connection from the Debugger. These stubs can be loaded to Application CPUs that are not configured to run by appropriate entries in the ATOC configuration JSON file.

NOTE: These tools are designed to work in the following environment:

- Windows PC
- LINUX
- CPU Board with Alif Ensemble or Crescendo REV_A1 device populated

This Quick Start Guide will take you through the steps necessary to install the Alif Security Toolkit.

For a complete description of the tools and their usage, please refer to the version of the Alif Security Toolkit User Guide applicable to the release being used.

2. Installation Requirements

2.1 Python Executables

The Security tools are now executable images, previously these python scripts were delivered as source code. These executables do not require python or any pre-requisite libraries to be installed.

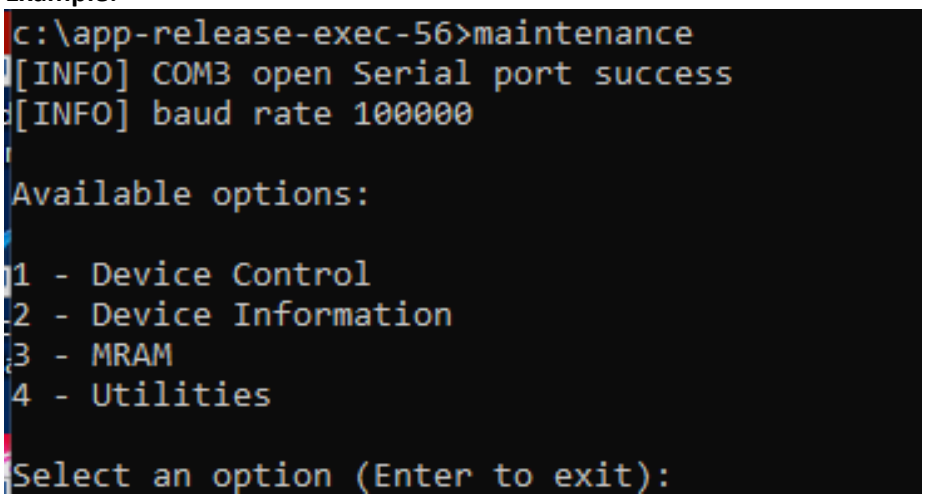
Previously you were required to run

```
$ python3 <tool-name>.py
```

Now you just run

```
$ <tool-name>
```

Example:



NOTE: Screenshots in other application notes or user guides might show the previous format but apply to this version or later releases if you enter just <tool-name> instead of “python3 <tool-name>.py”.

2.2 Windows

- Windows 10

- Tera Term (or similar) for SE-UART output

2.3 LINUX

- Ubuntu 18.04 (later versions of Ubuntu should work, but have not been tested)

- For a terminal emulator for SE-UART output, we have found that Minicom and picocom do not operate well with the 100000 baud rate used by the V54 SETOOLS. We recommend Gtkterm which works well

```
$ sudo apt install gtkterm
$ sudo gtkterm –device=/dev/ttyUSB0
```

2.4 Downloading the Alif Security Toolkit release

The latest version of the Alif Security Toolkit zip file archive is available on the "Kits & Software" tab of the Alif Ensemble page at www.alifsemi.com/ensemble. Download this archive and note the location where you save it.

3. Installation Guide

3.1 Release Unpacking

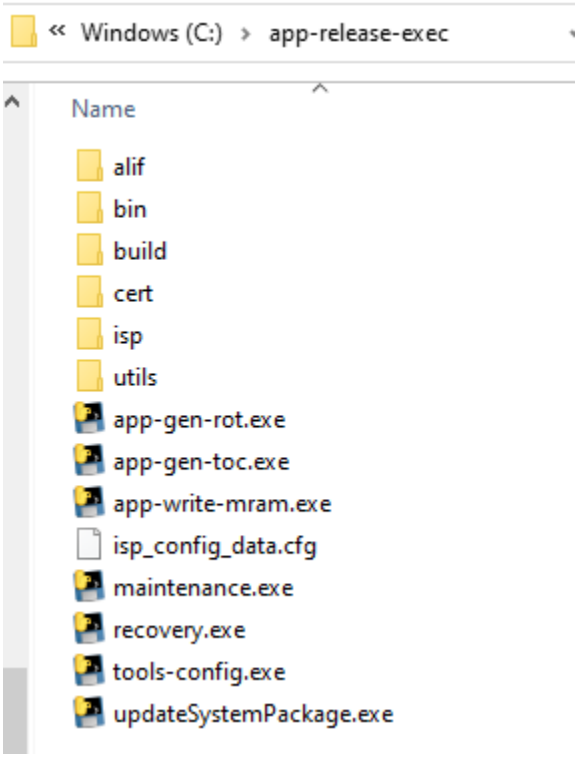
Unzip the Alif Security Toolkit Release to any directory on your Windows-PC or Linux machine.

The recommended <release-location> installation directories are:

Windows:	C:\app-release-exec
Linux:	/home/$USER/app-release-exec-linux

3.2 Release Contents

The directory structure is as follows:



root: utilities detailed in below table (Python3 executables)

alif/: folder with Alif Secure Enclave latest firmware for updates. The firmware **.bin** file is stored here.

bin/: internal utilities required by the tools

build/: The final **AppTocPackage** is in this folder. Sub-folder ***build/config/*** contains the configuration files for app-gen-toc and sub-folder ***build/images/*** contains all images declared in the configuration file.

cert/: all certificates (key1/key2 and image's content certificates) are generated here.

isp/: contains utilities used by the command scripts to do In System Programming

utils:/ this folder contains internal utilities as well as keys and configuration files.

The SETOOLS Release package contains the following (Windows executable names shown. Linux executables have no file extension):

tools-config.exe	This program shows the current configuration and allows the user to select supported options such as Part Number, device Revision, and MRAM Burner interface (jtag or isp)
app-gen-rot.exe	This program will generate the RoT (Root of Trust), corresponding Key 1/Key 2 certificates and Kceicv encryption key. For the first time, it will ask for a password for the generated keys and it will be saved for later use. If a new password is desired, delete the *utils/key/icv_keys_pass.pwd* file and run the script again. For convenience, **a sample RoT is provided so there is no need to run this tool**, unless the user wants to.
app-gen-toc.exe	This program will generate the **AppTocPackage.bin** image (in *build/* directory) based on the information provided in *build/app-cfg.json* configuration file. Images declared in this file must exist in this directory.
app-write-mram.exe	This program is used to write the **AppTocPackage.bin** image into the MRAM memory.
updateSystemPackage.exe	This program is used to update the **Alif Reserved Area in MRAM** (containing the *Secure Enclave firmware*) with a new release from Alif Semiconductor. These files are held in alif/ folder.
maintenance.exe	This program allows for the recovery of MRAM executable images as well as providing status information on installed applications and boot status.
recovery.exe	This program is used for SEROM recovery.

4. Target Device Selection

Beginning with the V1.107.00 release, the default tools configuration for the target device is the Ensemble E8 device used on the Ensemble E8 DevKit. If you are using any other kit or your own board, you will need to select the correct target device. There are two ways to accomplish this.

4.1 Use the “updateSystemPackage” command

Using this command is an easy way to ensure your target device’s internal firmware is upgraded to the current version to match your tools directory, and the tool will also detect if the target device does not match the default and ask if you want to change the default to match the target. If you reply “y”, your tools configuration will then be set properly.

4.2 Use A Command Line Version of “tools-config”

If you are using an Alif development or applications kit, you can enter the tools configuration in a command-line format.

Kit	**Command Syntax**
E7 DevKit (DK-E7)	tools-config -p "E7 (AE722F80F55D5LS) - 5.5 MRAM / 13.5 SRAM"
E7 AI/ML AppKit (AK-E7-AIML)	tools-config -p "E7 (AE722F80F55D5LS) - 5.5 MRAM / 13.5 SRAM"
E1C DevKit (DK-E1C)	tools-config -p “E1C (AE1C1F4051920PH) - 1.86 MRAM / 2.0 SRAM”
Balletto B1 DevKit (DK-B1)	tools-config -p “B1 (AB1C1F4M51820PH) - 1.8 MRAM / 2.0 SRAM”

4.3 Use the “tools-config” Menus

You can run “tools-config” and use the menus to select the family and device.

5. Common Commands and Usage

There are some common operations that users will often need to do with the SETOOLS.

5.1 Loading Debug Stubs To Prepare A Board For Debugging

To enable connection with a debugger such as ULINKpro or J-Link, the target core needs to be running. This is achieved by running debug stubs on the cores. With the factory Blinky application running, you can connect to the M55_HE core.

If the M55_HE core does not have an application loaded for it, the system will insert a debug stub for it. In either case, to debug a design for the M55_HP or A32 cores, the user must instantiate debug stubs by entering them in an ATOC configuration file.

Here are the steps needed to prepare the board for debugging based on the above documentation.

You can create an ATOC image with 3 SRAM debug stubs for the 3 CPU cores by opening a command prompt window in the SETOOLS directory (the default is C:\app-release-exec) and running

```
app-gen-toc -f build/config/app-cpu-stubs.json
```

Then erase the application part of MRAM by running

```
app-write-mram -e app
```

And then write the image with 3 debug stubs to MRAM by running

```
app-write-mram
```

Refer to the “Debug Stubs” section of the Alif Security Toolkit User Guide for more information.

5.2 Restoring the Default Factory “Blinky” Application

You can use a sequence of SETOOLS commands to restore your Beta Development Kit to the factory default Blinky application running on the M55_HE processor out of SRAM.
From a Windows Command Prompt window in the SETOOLS installation directory (the default is C:\app-release-exec), execute the following commands:

```
app-gen-toc          (this generates the binary ATOC image for the factory Blinkyprogram)
app-write-mram       (this programs the ATOC image into MRAM)
```

Your board should now be blinking the LEDs at a 1-second interval.

For a complete description of the tools and their usage, please refer to the version of the Alif Security Toolkit User Guide applicable to the release being used which is available in the Kits & Software section of the Technical Documentation web pages.

Document History

Version	Date	Change Log
1.0	12/17/2021	Initial release for Beta program
2.0	01/26/2022	Revised for January 2022 Beta updates including revised directory structure and commands
2.1	04/15/2022	Corrected internal section references and added links to supporting documents
2.2	05/06/2022	Added necessary installation of python3 keyboard package
3.0	08/15/2022	Revised for change to executables instead of Python source scripts
3.1	11/02/2022	Added summary of common commands
3.2	8/25/2025	Added section on selecting the tools configuration (target device)