User's Guide

UniFlash CC3x20, CC3x3x SimpleLink™ Wi-Fi® and Internet-on-a chip™ Solution ImageCreator and Programming Tool

ABSTRACT

The CC3x20, CC3x3x devices are part of the SimpleLink™ microcontroller (MCU) platform, which consists of Wi-Fi®, *Bluetooth*® low energy, Sub-1 GHz and host MCUs, which all share a common, easy-to-use development environment with a single core software development kit (SDK) and rich tool set. A one-time integration of the SimpleLink™ platform enables you to add any combination of the portfolio's devices into your design, allowing 100 percent code reuse when your design requirements change. For more information, visit www.ti.com/simplelink.

This user's guide describes the UniFlash CC3x20, CC3x3x SimpleLink™ Wi-Fi® and Internet-on-a chip™ Solution ImageCreator and Programming Tool from Texas Instruments™.

Table of Contents

1 Introduction....4
2 Terms and Concepts....5
3 Installation....6
4 Image Creator Application....7
5 Quick Start....9
5.1 Creating a New Project....9
5.2 Simple Mode....10
5.3 Adding the MCU Image....10
5.4 Adding the Service Pack....10
5.5 Creating and Programming Image from an Opened Project....11
6 Use....12
6.1 Creating a New Project....12
6.2 Opening a Recent Project....12
6.3 Managing Projects....13
6.4 Device Status and Settings....14
6.5 Simple Mode....16
6.6 Advanced Mode....17
6.7 Advanced Mode – General Settings....19
6.8 Advanced Mode – System Settings....19
6.9 Adding the Service Pack....30
6.10 Adding the Trusted Root-Certificate Catalog....31
6.11 Adding the Host Application File (CC32xx)....31
6.12 User Files....32
6.13 Device File Browser....38
6.14 Creating an Image From a Project....39
6.15 Creating an OTA....39
6.16 Saving an Image....40
6.17 Programming.bin and Programming.hex....40
6.18 Programming an Image From an Opened Project....40
6.19 Programming an Image Using a .sli File....40
6.20 Secured Image With Key....42
7 Command Line....43

7.1 Project Commands....43
7.2 Image Commands....50
7.3 Tools Commands.... 51
7.4 Device Commands....53
7.5 GUI Configure Commands....53
7.6 GUI Commands Additional Arguments.... 54
8 Tools....55
8.1 Certificate Sign Request (Only for CC323xS/SF Devices)....55
8.2 Sign File....55
8.3 Activate Image.... 56
9 Using CSR Utility.... 57
9.1 Get CSR From Device and Copy it to File 57
9.2 Replace CSR File in the Project....59
10 Default Power Values for LaunchPad at the Antenna.... 61
10.1 Defaults for CC3x35 Device....61
10.2 Defaults for CC3x35MOD.... 63
11 Revision History....64

List of Figures

Figure 1-1. Programming Using the Image Creator....4
Figure 4-1. Opening ImageCreator Through UniFlash (1 of 2)....7
Figure 4-2. Opening ImageCreator Through UniFlash (2 of 2)....8
Figure 5-1. New Project....9
Figure 5-2. Device Types....9
Figure 5-3. Simple Mode CC32xx....10
Figure 6-1. Open Recent Project.... 12
Figure 6-2. Project Management.... 13
Figure 6-3. Device Status: Disconnected.... 14
Figure 6-4. Device Status: Connected....15
Figure 6-5. Simple Mode CC31xx....16
Figure 6-6. Simple Mode CC32xx....17
Figure 6-7. Device Status Advanced Mode.... 18
Figure 6-8. Tool Tips.... 18
Figure 6-9. Example 16-Bit Key....19
Figure 6-10. Set Key Filename....19
Figure 6-11. PHY(2.4G) Calibration Mode....20
Figure 6-12. Regulatory Domain Table 2.4G (1 of 2)....20
Figure 6-13. Regulatory Domain Table 2.4G (2 of 2)....21
Figure 6-14. Regulatory Domain Table....23
Figure 6-15. Coexistence and Antenna Selection.... 25
Figure 6-16. Device Identity Configuration.... 26
Figure 6-17. Certificate Configuration....26
Figure 6-18. Certificate Sign Request Options.... 27
Figure 6-19. Self-Signed Certificate Options.... 28
Figure 6-20. HTTP Server.... 30
Figure 6-21. Vendor Certificate Catalog.... 31
Figure 6-22. OTP Section....31
Figure 6-23. MCU Image Advanced Mode.... 32
Figure 6-24. File Properties Dialog....34
Figure 6-25. Rename Filename.... 35
Figure 6-26. Delete File or Folder (1 of 2).... 36
Figure 6-27. Delete File or Folder (2 of 2).... 36
Figure 6-28. User File Action Monitor....37
Figure 6-29. Delete File.... 38
Figure 6-30. Get File....38
Figure 6-31. OTA Private Key File Name....39
Figure 6-32. Save Image.... 40
Figure 6-33. Program Image....40
Figure 6-34. Program Image....41
Figure 6-35. Program Image....42

Figure 8-1. Open Tools.......... 55
Figure 8-2. Certificate Sign Request.......... 55
Figure 8-3. Sign File.......... 56
Figure 8-4. Activate Image.......... 56

List of Tables

Table 2-1. Terms and Concepts.......... 5
Table 6-1. TX Parameters Table.......... 24
Table 6-2. Flags Options.......... 35
Table 6-3. Other File Properties.......... 35
Table 10-1. 2.4 GHz Default Values.......... 61
Table 10-2. 5 GHz Default Values.......... 62
Table 10-3. 2.4 GHz Default Values.......... 63
Table 10-4. 5 GHz Default Values.......... 64

Trademarks

SimpleLink™, Internet-on-a chip™, and Texas Instruments™ are trademarks of Texas Instruments.
Wi-Fi® are registered trademarks of Wi-Fi Alliance.
Bluetooth® is a registered trademark of Bluetooth SIG.
All other trademarks are the property of their respective owners.

1 Introduction

ImageCreator is a part of the UniFlash application used to create a programming image; the ImageCreator can also write the programming image into the SimpleLink™ CC3xx devices. The programming image is a file containing the SimpleLink™ device configurations and files required for the operation of the device. For the SimpleLink™ CC32xx wireless microcontroller (MCU), the programming image also includes the host application file.

A new SimpleLink™ device should first be programmed by a programming image. The image, created by the ImageCreator, can be programmed onto the device as part of the production procedure, or in development stage. The image can be programmed as follows:

- Using the ImageCreator tool through a UART interface
- Using an external off-the-shelf tool through a serial-flash SPI

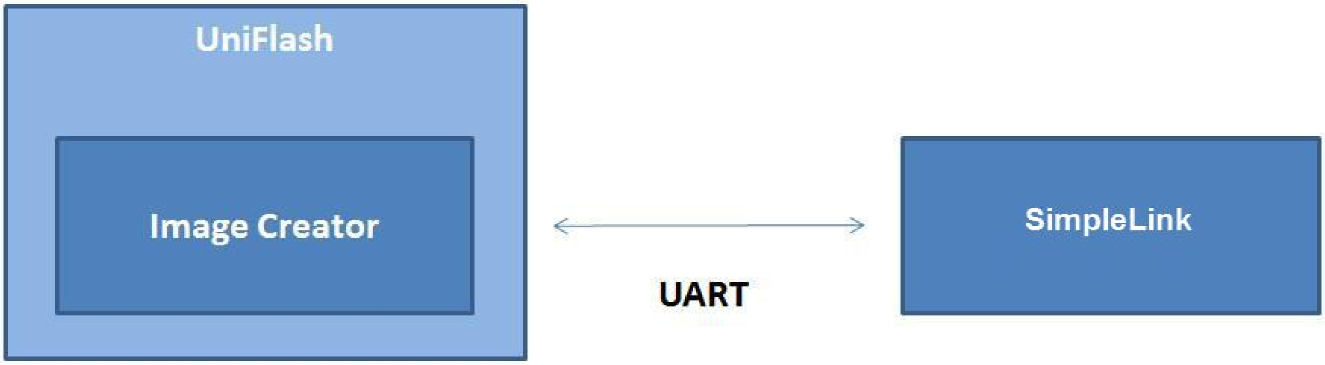


Figure 1-1. Programming Using the Image Creator

The main features of the ImageCreator are:

- Supports both the SimpleLink™ CC32xx and CC31xx devices
- Can create an image in either production mode or development mode. Production images restrict some of the features intended for development, such as the JTAG interface or access to individual files using this tool.
- Creates encrypted programming images
- As part of the programming image creation:
  - Applies the service pack and the certificate store
  - Defines the device configurations, such as Wi-Fi® mode, IP settings, provisioning, and more
  - Adds files and applies attributes per file, such as security settings and fail safe
- Connects to the device and retrieves its properties, such as device type, flash size, and MAC address
- When in development, the image supports on-line access to the device file system
- Programs the device, and can program using an image created by another instance of image creator
- Executes some operations using a command line interface
- Ability to manage projects: import an existing project, or export a project to another machine

This document describes the installation, operation, and usage of the SimpleLink™ ImageCreator tool as part of the UniFlash.

2 Terms and Concepts

Table 2-1 lists some of the terms and concepts used in this document.

Table 2-1. Terms and Concepts

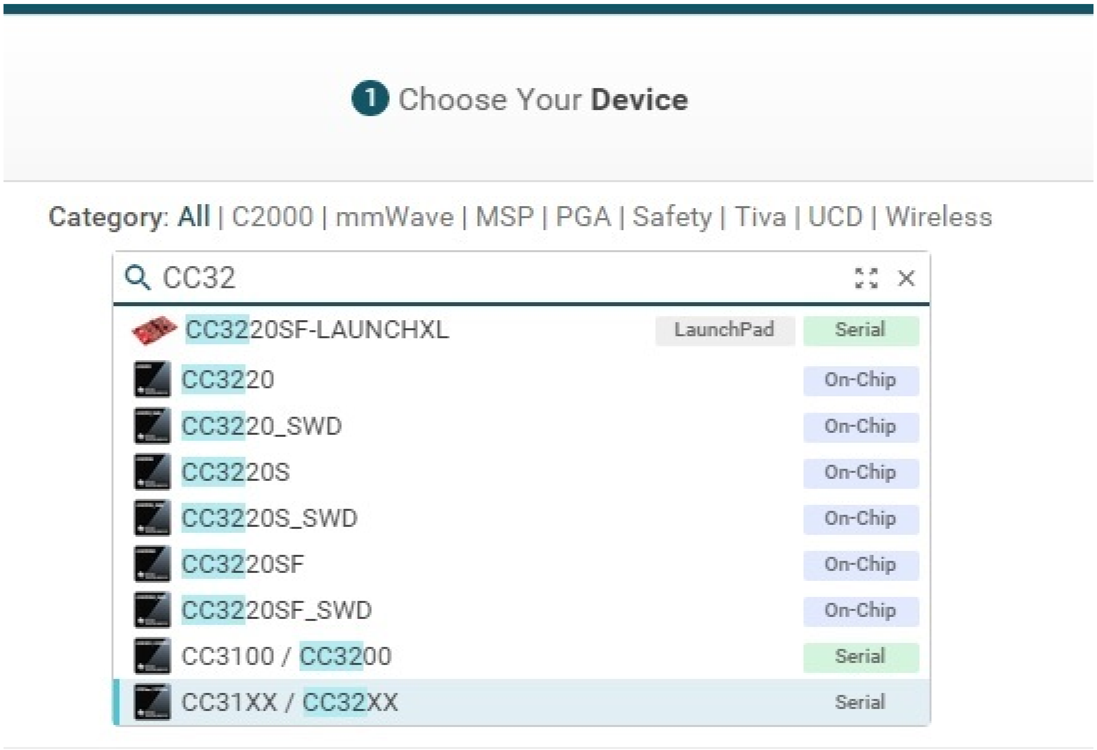
Term or Concept	**Description**
Image	Image is a packed file which contains the service pack, the system configuration files, the user files, and the host program (in case of the SimpleLink Wi-Fi CC32xx wireless MCU). The process of creating the programming image is an off-line process.
Project	Project is a workspace for creating an image file.
Connection	Users can connect to the device and get its attributes, such as its MAC address, security type, and so forth.
Key	The 16-byte key is used for image encryption.
Programming image file types	The image file is created in several encoding types: • Programming.bin and programming.hex, standard binary and intel hex files, are used for programming by an external serial flash programming tool. • Programming.ucf, (TI proprietary encoding) is used for programming by the host. • Programming.sli, (TI proprietary encoding) is used for programming by the image creator. • Notes – Encrypted images are named programming.encrypt.bin/hex/ucf/sli – The output files are under the image creator installation directory in *\projects\${project_name}\sl_image\Output*

3 Installation

ImageCreator is a part of the UniFlash application. Download and run the latest installer of the UniFlash application from http://www.ti.com/tool/UNIFLASH.

4 Image Creator Application

Run the UniFlash application. A list of all supported devices appears; choose *CC31xx / CC32xx* from the device list, as shown in Figure 4-1.



1 Choose Your **Device**

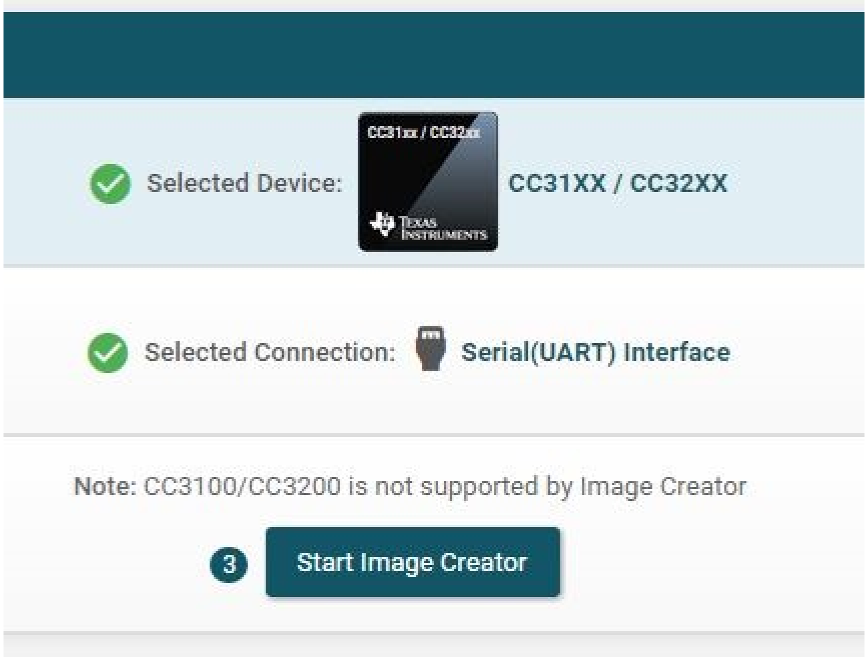
Category: All | C2000 | mmWave | MSP | PGA | Safety | Tiva | UCD | Wireless

CC32

CC3220SF-LAUNCHXL	LaunchPad Serial
CC3220	On-Chip
CC3220_SWD	On-Chip
CC3220S	On-Chip
CC3220S_SWD	On-Chip
CC3220SF	On-Chip
CC3220SF_SWD	On-Chip
CC3100 / CC3200	Serial
CC31XX / CC32XX	Serial

Figure 4-1. Opening ImageCreator Through UniFlash (1 of 2)

Then press on the Start Image Creator button, as shown in Figure 4-2.



✓ Selected Device: CC31XX / CC32XX

✓ Selected Connection: Serial(UART) Interface

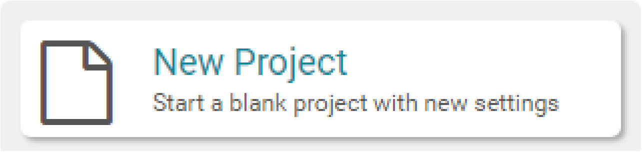
Note: CC3100/CC3200 is not supported by Image Creator

Figure 4-2. Opening ImageCreator Through UniFlash (2 of 2)

5 Quick Start

5.1 Creating a New Project

Click the New Project button on the Welcome page, as shown in Figure 5-1.

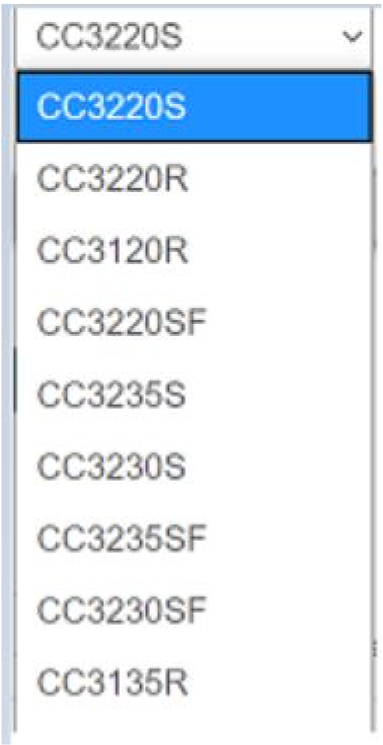


New Project
Start a blank project with new settings

Figure 5-1. New Project

The Create Project window appears.

- Project name – The unique project name (mandatory)
- Project Description – Short description of the project (not mandatory)
- Device Type – As shown in Figure 5-2.



CC3220S

CC3220S
CC3220R
CC3120R
CC3220SF
CC3235S
CC3230S
CC3235SF
CC3230SF
CC3135R

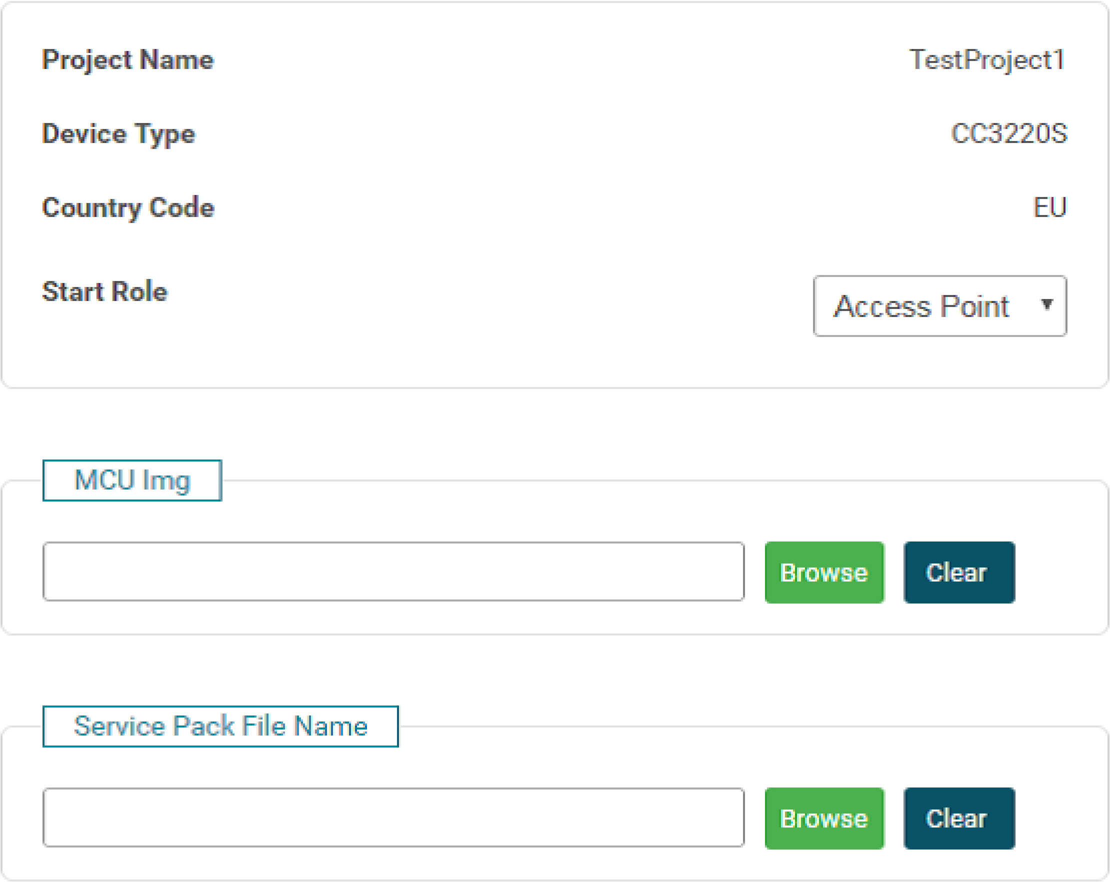
Figure 5-2. Device Types

- Device mode – (Production/Develop)
  - Production mode is the default mode. In this mode, the user cannot use IDEs for debug.
  - Develop or development mode is for the JTAG interface and access to individual files. The image created by this project is device-specific through the MAC address.

Users should fill out the relevant fields, and click the Create Project button. A new project with default parameters is then created.

5.2 Simple Mode

ImageCreator has two modes: simple and advanced. After the project is loaded or created, ImageCreator opens it in simple mode (see Figure 5-3).



Project Name: TestProject1

Device Type: CC3220S

Country Code: EU

Start Role: Access Point ▾

Figure 5-3. Simple Mode CC32xx

5.3 Adding the MCU Image

The SimpleLink ImageCreator lets the user add the host application file (MCU Image) for CC32xx devices. In simple mode, if the CC32xx device is secured, then when the MCU image is uploaded, it is automatically signed by the dummy root certificate.

After adding, the following name appears:

- For CC32xxR/RS: mcuimg.bin
- For CC32xxSF: mcuflashimg.bin

5.4 Adding the Service Pack

The service pack is used to upgrade the network peripheral internal firmware. The service pack file is provided by TI in the SDK package. The SP file name is sp_<release_versions_number>.bin, and it is placed in the <SDK_PATH>\tools\cc32xx_tools\servicepack-cc3xXX folder.

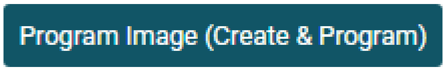
TI recommends adding the service pack to the programming image; this action, however, is not mandatory. If it is not programmed, the device uses its factory code.

When adding the service pack, the user selects the file location; however, the ImageCreator does not keep a link to the original file. To change the service pack, the new service pack file should be selected again.

5.5 Creating and Programming Image from an Opened Project

In production mode, there is no need to connect to the device before programming.

In development mode, the device’s mac address should be provided before creating an image, connect to the device by pressing the button before programming.

To program an image, click the  button: The program image also creates the image, so there is no need to create the image before the programming.

6 Use

6.1 Creating a New Project

See Section 5.1.

6.2 Opening a Recent Project

Open an existing project by clicking on the project name in the recent projects list, directly from the main "Welcome…" page. The main page can be navigated by clicking on the

(see Figure 6-1).

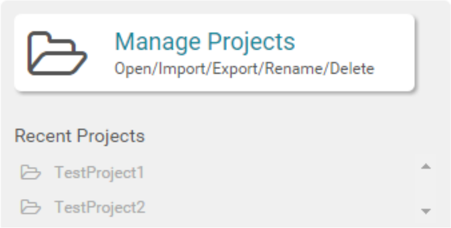


Figure 6-1. Open Recent Project

6.3 Managing Projects

Open the list of all projects by pressing on the Manage Projects button shown in Figure 6-1. From here, the screen appears as in Figure 6-2.

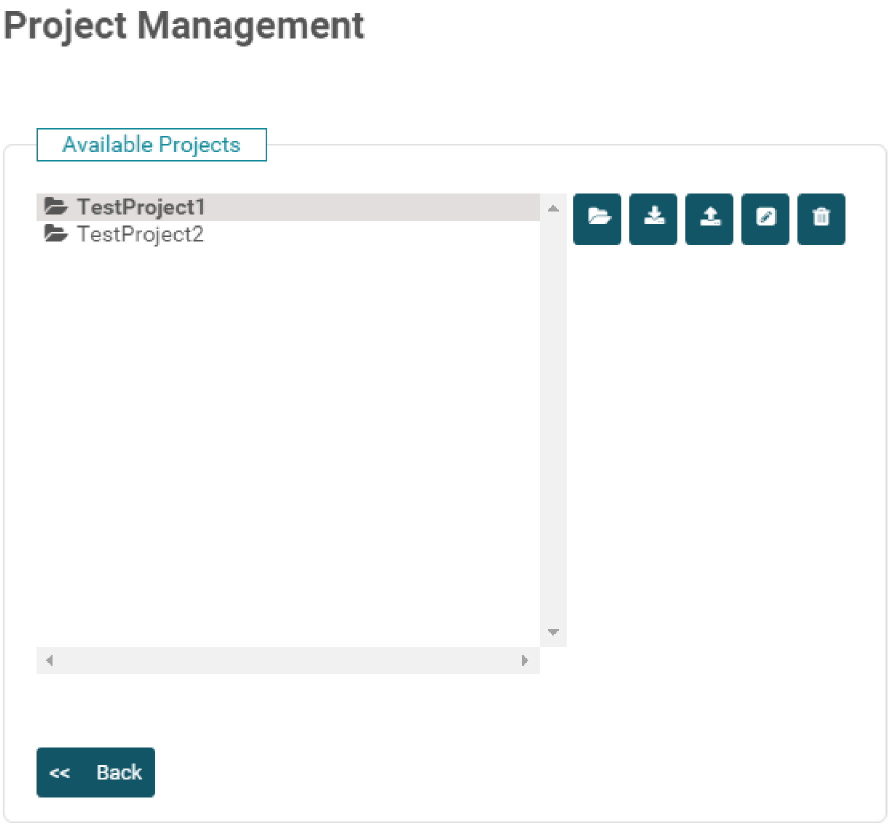


Figure 6-2. Project Management

The available operations are:

- Open project – Press 📂 to open the project.
- Import project from zip file – Press ⬇ to zip the file.
- Export project – Press ⬆ to export the project.
- Rename project – Press ✎ to rename the project.
- Delete project – Press 🗑 to delete the project.

6.4 Device Status and Settings

A programming image can be prepared and created while offline, for example, while the device is not physically connected to the Image Creator app computer (see Figure 6-3).

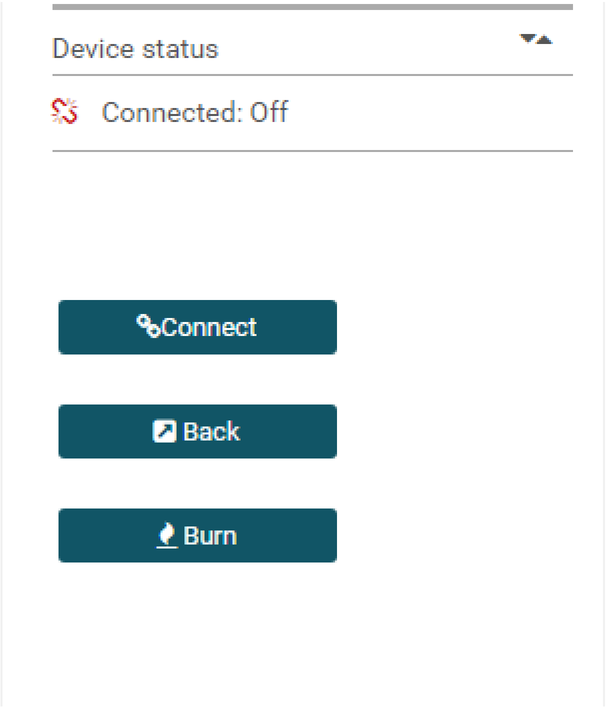
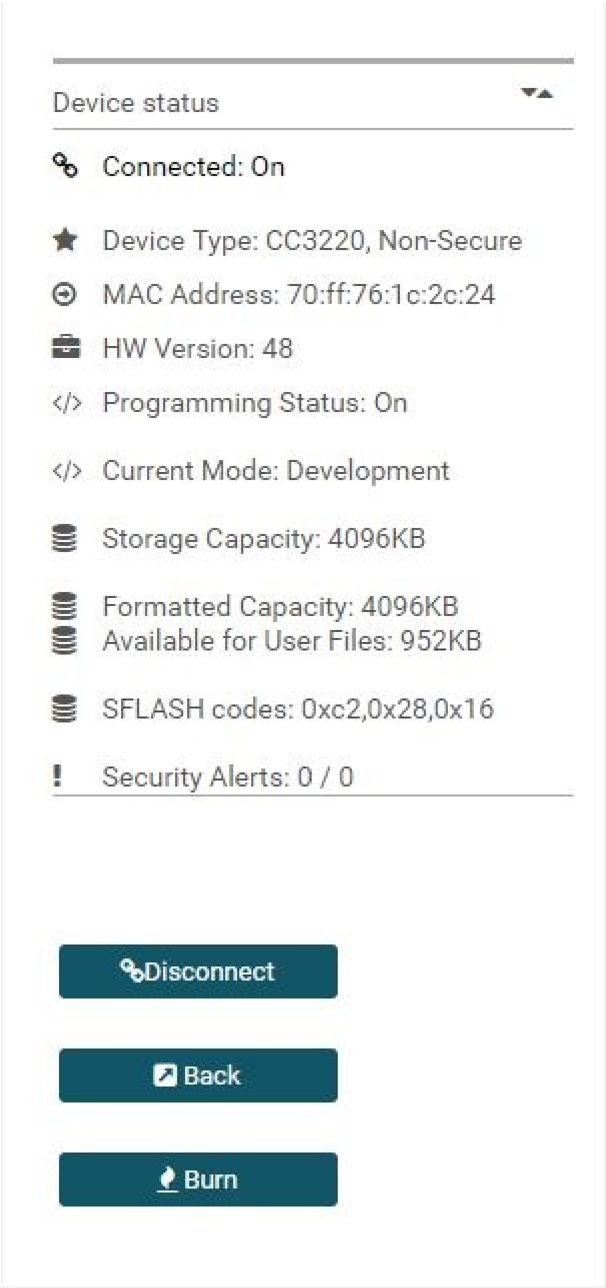


Figure 6-3. Device Status: Disconnected

If a device is physically connected by UART, the user can click the Connect button to automatically detect the device and perform an initial connection. The connect method retrieves the settings from the device and displays them, as shown in Figure 6-4.

The user can choose production or development mode:

- Production mode – The created image is programmable on any device.
  - For the device security, production mode exposes limited operations:
    - An online operation on the file system using the Image Creator is disabled.
    - Using JTAG (on the CC32xx device) is disabled.
- Development mode – Requires the target device MAC address to program it. The target device MAC address is set by the Image Creator setting window. This mode allows:
  - Browsing and modification of the device file system (see Section 6.13).
  - Using JTAG (CC32xx) is enabled.
  - The programming image file can be used to program only the device with the same MAC address as the one set into the image.



Device status

Connected: On

Device Type: CC3220, Non-Secure

MAC Address: 70:ff:76:1c:2c:24

HW Version: 48

Programming Status: On

Current Mode: Development

Storage Capacity: 4096KB

Formatted Capacity: 4096KB
Available for User Files: 952KB

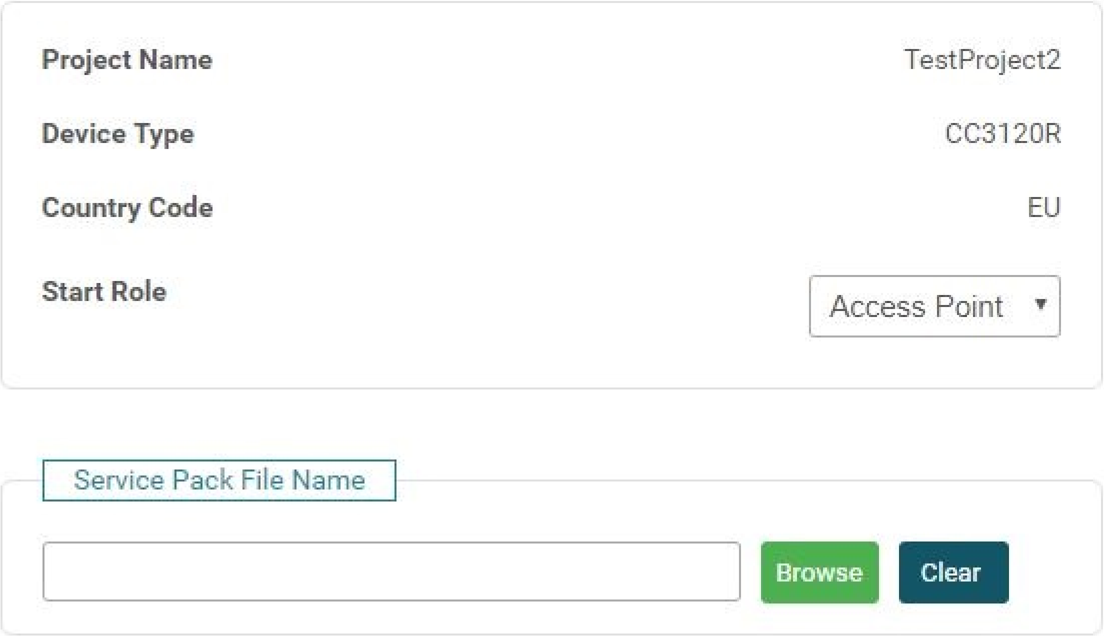
SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 0

Figure 6-4. Device Status: Connected

6.5 Simple Mode

ImageCreator has 2 show modes: Simple mode and advanced mode. After the project is loaded or created, ImageCreator opens it in simple mode (see Figure 6-5).



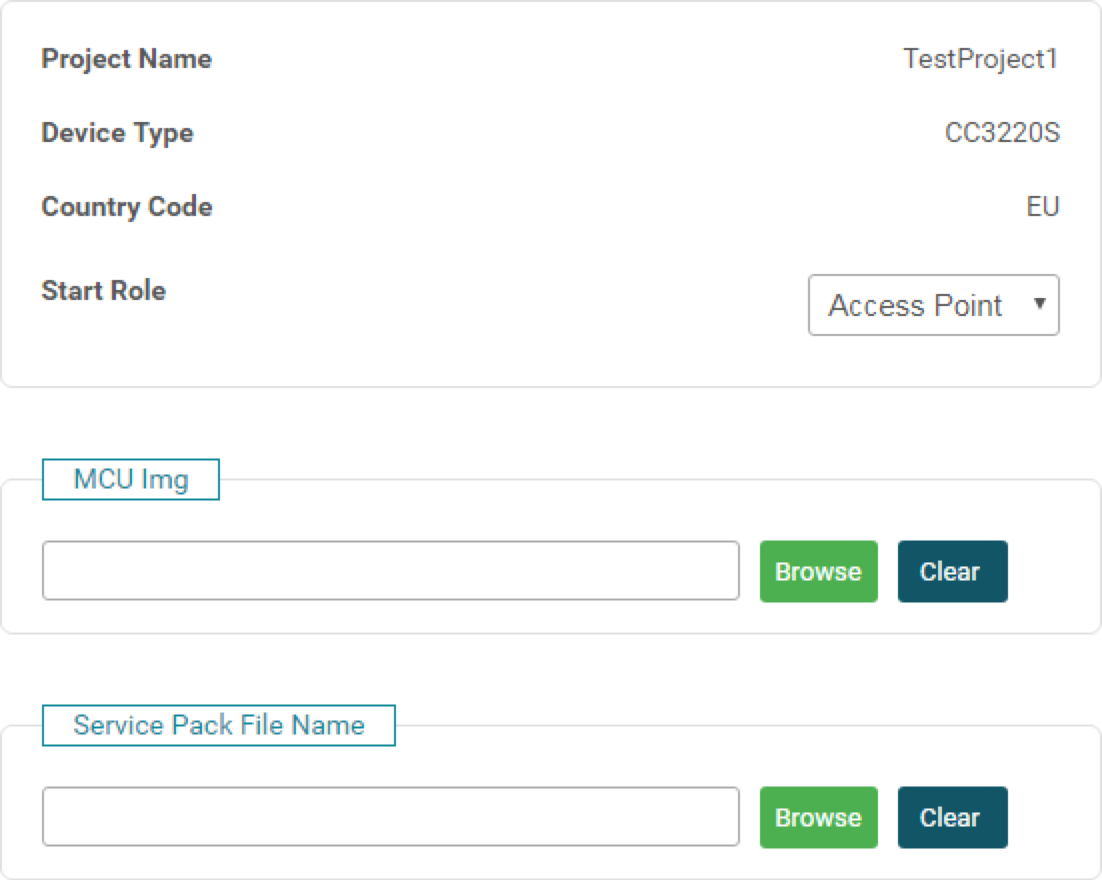
Project Name	TestProject2
Device Type	CC3120R
Country Code	EU
Start Role	Access Point ▾

Service Pack File Name

Figure 6-5. Simple Mode CC31xx

Simple mode provides an option to simplify mandatory project's parameters configuration.

In simple mode, if the CC32xx device is secured, then when the MCU image is uploaded, it is automatically signed by the dummy root certificate.



Project Name TestProject1

Device Type CC3220S

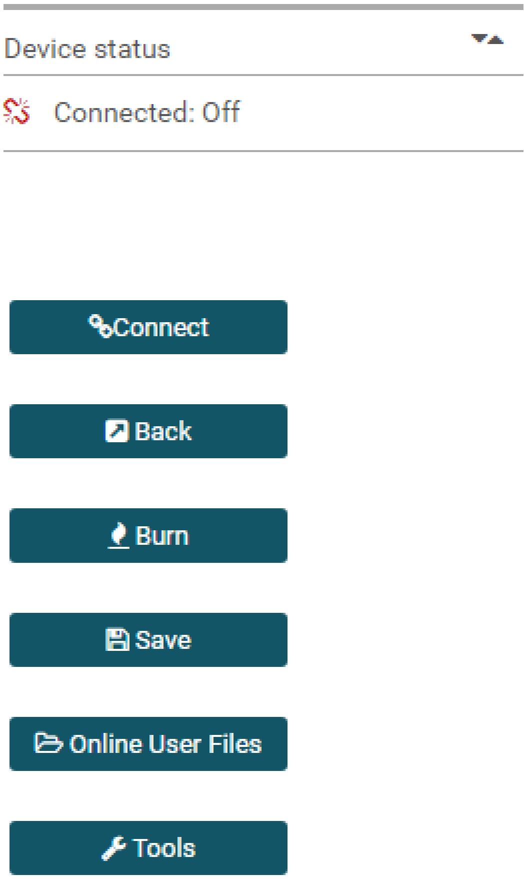
Country Code EU

Start Role Access Point

Figure 6-6. Simple Mode CC32xx

6.6 Advanced Mode

Advanced mode lets the user make more extensive changes and tunes for project parameters. After switching to advanced mode, more options appear on the right side bar (see Figure 6-7), in contrast with simple mode (See Figure 6-5 or Figure 6-6).

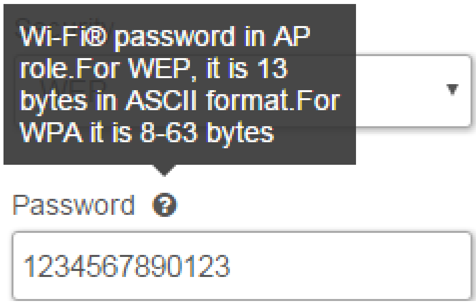


Device status

Connected: Off

Figure 6-7. Device Status Advanced Mode

The left side of the screen, under advanced mode, contains links to the configuration pages, organized in a tree structure. The tree structure enables quick navigation to any configuration page with a single click. Fields within the pages contain tool tips, with explanations that appear when the mouse is moved over the question mark icon, as shown in Figure 6-8.



Wi-Fi® password in AP role.For WEP, it is 13 bytes in ASCII format.For WPA it is 8-63 bytes

Password: 1234567890123

Figure 6-8. Tool Tips

6.7 Advanced Mode – General Settings

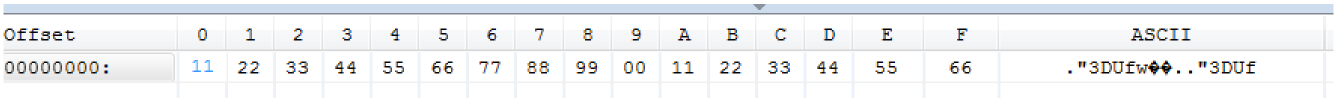
The user can configure general settings for the project:

- Change image mode (production/development). See Section 6.4.
- Set capacity and other defaults.
- Create encrypted image. See Section 6.7.1 and Section 6.20 .

6.7.1 Creating an Encrypted Image

ImageCreator lets the user create encrypted images (using AES-CTR encryption). An encrypted image can only be used with its key.

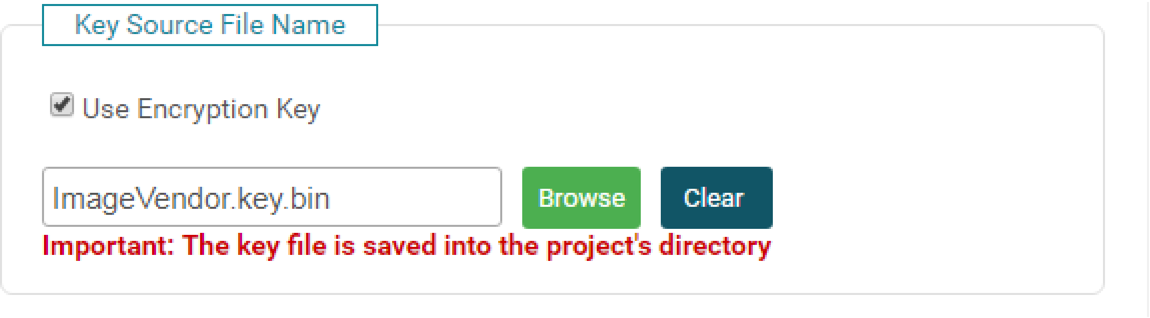
To create an encrypted image, create a binary file that contains a 16-byte key, such as the one shown in Figure 6-9.



```
Offset      0   1   2   3   4   5   6   7   8   9   A   B   C   D   E    F            ASCII
00000000:   11  22  33  44  55  66  77  88  99  00  11  22  33  44  55   66      ."3DUfw��.."3DUf
```

Figure 6-9. Example 16-Bit Key

Then set the key filename, as shown in Figure 6-10.



Key Source File Name

☑ Use Encryption Key

ImageVendor.key.bin

Important: The key file is saved into the project's directory

Figure 6-10. Set Key Filename

Following that, see Section 6.20.

6.8 Advanced Mode – System Settings

This section describes the options to configure the system settings.

6.8.1 Device

6.8.1.1 Radio Settings

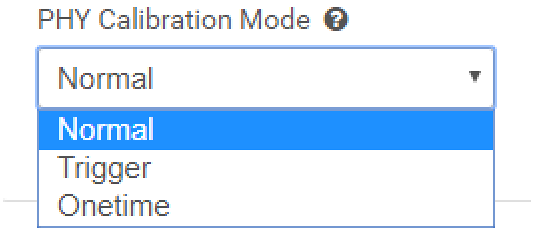
- PHY (2.4G) Calibration mode (see Section 6.8.1.1.1.1).
- Configuration options for 5G support devices:
  - 2.4G
    - TX power control (see Section 6.8.1.1.1.1.1)
  - 5G
    - PHY 5G Calibration Mode (see Section 6.8.1.1.2.1)
    - TX power control (see Section 6.8.1.1.2.2)
- Coexistence (see Section 6.8.1.1.3)
- Antenna selection (see Section 6.8.1.1.3)

6.8.1.1.1 RF 2.4G

6.8.1.1.1.1 PHY (2.4G) Calibration Mode

To modify calibration choose one of these modes (shown in Figure 6-11):

- Normal
- Trigger
- Onetime



PHY Calibration Mode

Normal

Figure 6-11. PHY(2.4G) Calibration Mode

For low-power applications, TI recommends choosing Trigger mode over Onetime mode, unless current peak limit is an absolute constraint.

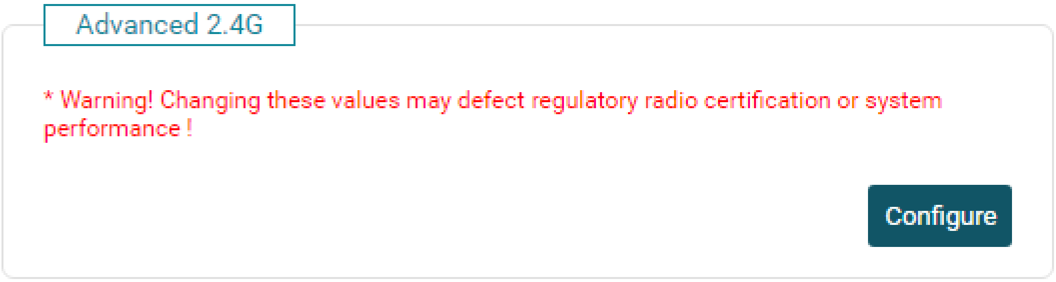
Trigger mode does not issue calibrations unless absolutely necessary, or manually triggered.

Normal calibration mode is used to achieve the best RF performance, or when the environment of the device is prone to changes (temperature changes).

6.8.1.1.1.1.1 TX Power Control (2.4G)

To modify output power, the tool lets the user configure 2.4-GHz band transmission power levels, per channel, for a defined regulatory region. This is useful for building a custom board (along with different BOM) or RF trace losses (this is different than the information presented in the TI reference design, although TI strongly recommends referring to the TI reference design instructions).

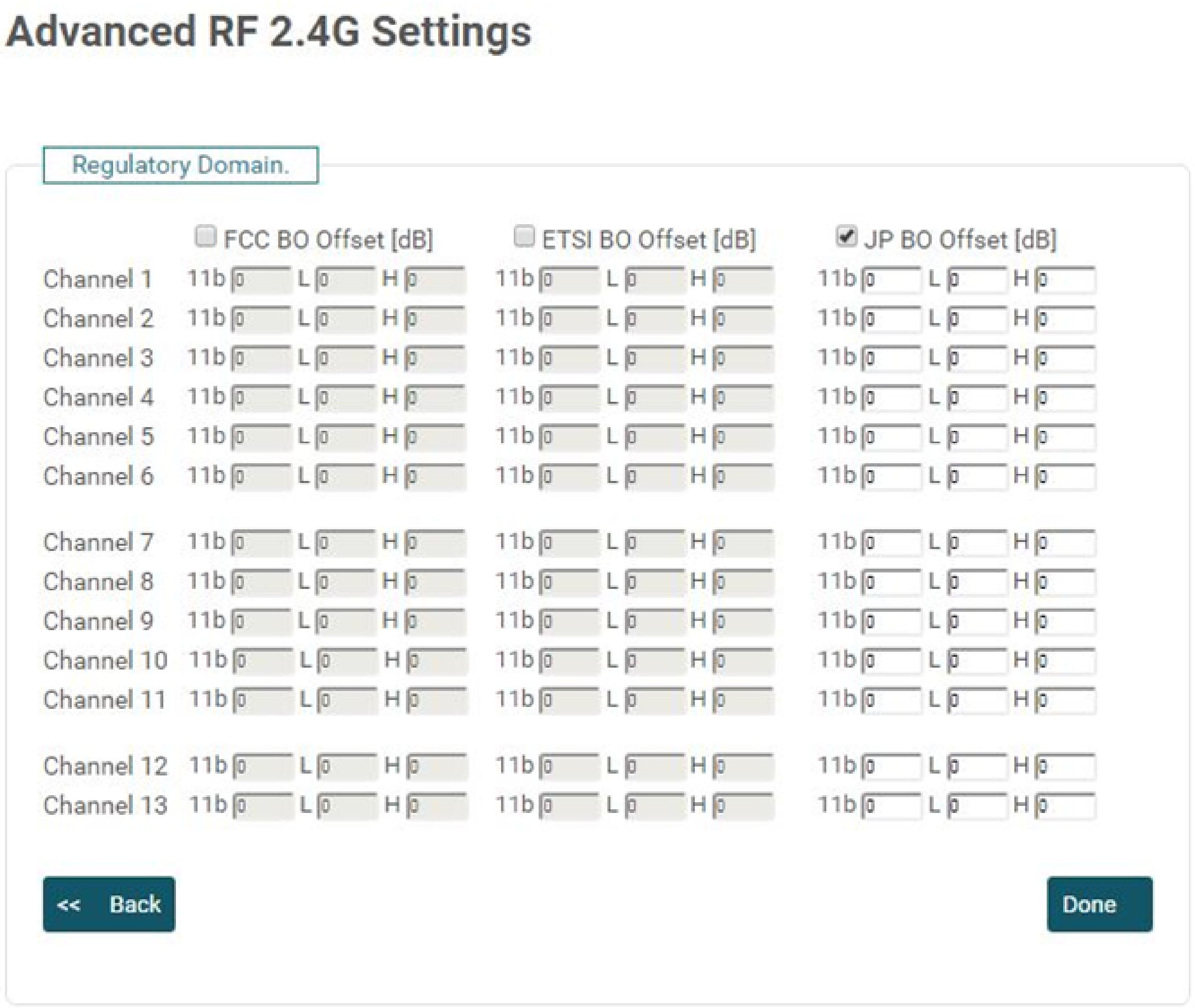
To open a regulatory domain table (shown in Figure 6-12 and Figure 6-13), press the Configure button in the RF 2.4G section.



Advanced 2.4G

* Warning! Changing these values may defect regulatory radio certification or system performance !

Figure 6-12. Regulatory Domain Table 2.4G (1 of 2)



Advanced RF 2.4G Settings

Regulatory Domain.

	☐ FCC BO Offset [dB]			☐ ETSI BO Offset [dB]			☑ JP BO Offset [dB]		
	11b	L	H	11b	L	H	11b	L	H
Channel 1	0	0	0	0	0	0	0	0	0
Channel 2	0	0	0	0	0	0	0	0	0
Channel 3	0	0	0	0	0	0	0	0	0
Channel 4	0	0	0	0	0	0	0	0	0
Channel 5	0	0	0	0	0	0	0	0	0
Channel 6	0	0	0	0	0	0	0	0	0
Channel 7	0	0	0	0	0	0	0	0	0
Channel 8	0	0	0	0	0	0	0	0	0
Channel 9	0	0	0	0	0	0	0	0	0
Channel 10	0	0	0	0	0	0	0	0	0
Channel 11	0	0	0	0	0	0	0	0	0
Channel 12	0	0	0	0	0	0	0	0	0
Channel 13	0	0	0	0	0	0	0	0	0

Figure 6-13. Regulatory Domain Table 2.4G (2 of 2)

The term Back-Off Offset (BO) determines that the value configured in dB is the power offset from the default TI design, limited to EVM and Mask constraints. The offset can be both positive and negative to allow power increase. To change specific regulatory domain BO, check the relevant box and change the offset according to wanted channel and rate group.

- 11b – only 11b rate
- H – High rates (MCS7, 54 Mbps, 48 Mbps)
- L – Low rates (all the rest)

Valid values are -6[dB] to +6[dB].

6.8.1.1.2 RF 5G

6.8.1.1.2.1 PHY (5G) Calibration Mode

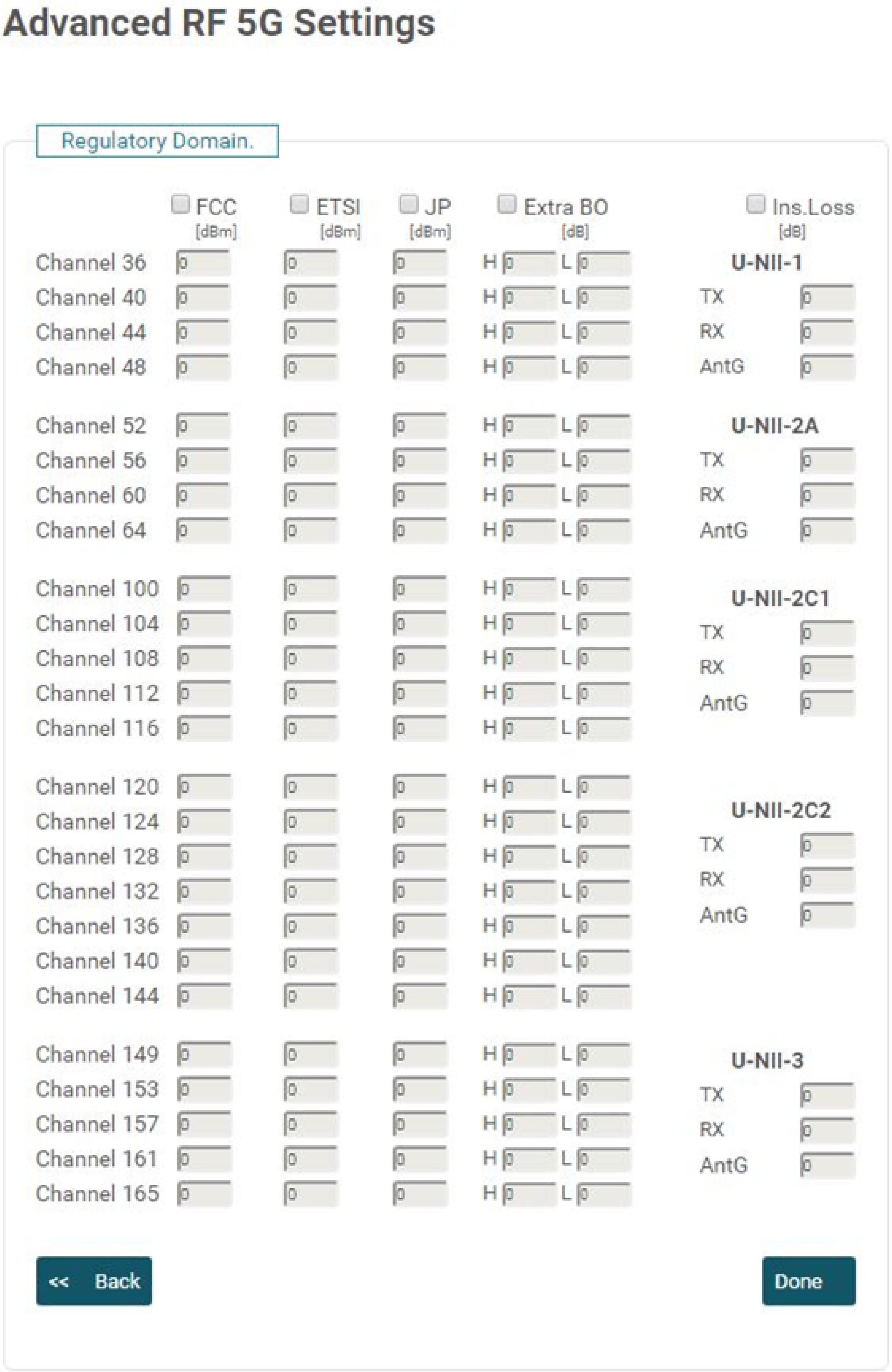
Only Normal option is supported.

6.8.1.1.2.2 TX Power Control

To modify output power, ImageCreator lets the user configure 5-GHz band transmission power levels, per channel, for a defined regulatory region. This is useful for building a custom board (along with different BOM) or

RF trace losses (this is different than the information presented in the TI reference design, although TI strongly recommends referring to the TI reference design instructions).

To open a regulatory domain table (shown in Figure 6-14 and described in Table 6-1), press the Configure button in the RF 5G section.



Advanced RF 5G Settings

Regulatory Domain.

	☐ FCC [dBm]	☐ ETSI [dBm]	☐ JP [dBm]	☐ Extra BO [dB] H	L
Channel 36	0	0	0	0	0
Channel 40	0	0	0	0	0
Channel 44	0	0	0	0	0
Channel 48	0	0	0	0	0
Channel 52	0	0	0	0	0
Channel 56	0	0	0	0	0
Channel 60	0	0	0	0	0
Channel 64	0	0	0	0	0
Channel 100	0	0	0	0	0
Channel 104	0	0	0	0	0
Channel 108	0	0	0	0	0
Channel 112	0	0	0	0	0
Channel 116	0	0	0	0	0
Channel 120	0	0	0	0	0
Channel 124	0	0	0	0	0
Channel 128	0	0	0	0	0
Channel 132	0	0	0	0	0
Channel 136	0	0	0	0	0
Channel 140	0	0	0	0	0
Channel 144	0	0	0	0	0
Channel 149	0	0	0	0	0
Channel 153	0	0	0	0	0
Channel 157	0	0	0	0	0
Channel 161	0	0	0	0	0
Channel 165	0	0	0	0	0

☐ Ins.Loss [dB]	
U-NII-1	
TX	0
RX	0
AntG	0
U-NII-2A	
TX	0
RX	0
AntG	0
U-NII-2C1	
TX	0
RX	0
AntG	0
U-NII-2C2	
TX	0
RX	0
AntG	0
U-NII-3	
TX	0
RX	0
AntG	0

Figure 6-14. Regulatory Domain Table

Table 6-1. TX Parameters Table

Column name	Description
FCC	Setting maximum output power limitation, in dBm at antenna level, after board trace loss and antenna gain, in countries regulated by FCC.
ETSI	Setting maximum output power limitation, in dBm at antenna level, after board trace loss and antenna gain, in countries regulated by ETSI.
JP	Setting maximum output power limitation, in dBm at antenna level, after board trace loss and antenna gain, in countries regulated by JP.
Extra BO	For TI use only. It applies extra power BO, to improve EVM and mask compliance at the expense of output power
o	H – High rates (MCS7, 54 Mbps, 48 Mbps).
o	L – Low rates (all the rest).
Ins. Loss	Setting board insertion loss, in dB.

Example of parameter change – the following example shows how to change the TX output power level at a certain channel, for a certain regulatory domain. To set channel 36 limit to 12[dBm], after antenna gain, for FCC regulatory domain:

1. Click on the FCC checkbox to enable editing.
2. Fill in all default values for all channels except channel 36, as described in Section 10.
3. Fill in 12 under channel 36.

Back to Default – to return to default values, uncheck the relevant checkbox. A checkbox that is not marked implies that default values are used, according to Section 10.

6.8.1.1.3 Coexistence and Antenna Selection

To allow maximum flexibility for every platform configuration, there are multiple choices for assigning the coexistence and antenna selection interface on the device's pins. These choices differ slightly based on device family (CC3135 versus CC3235x).

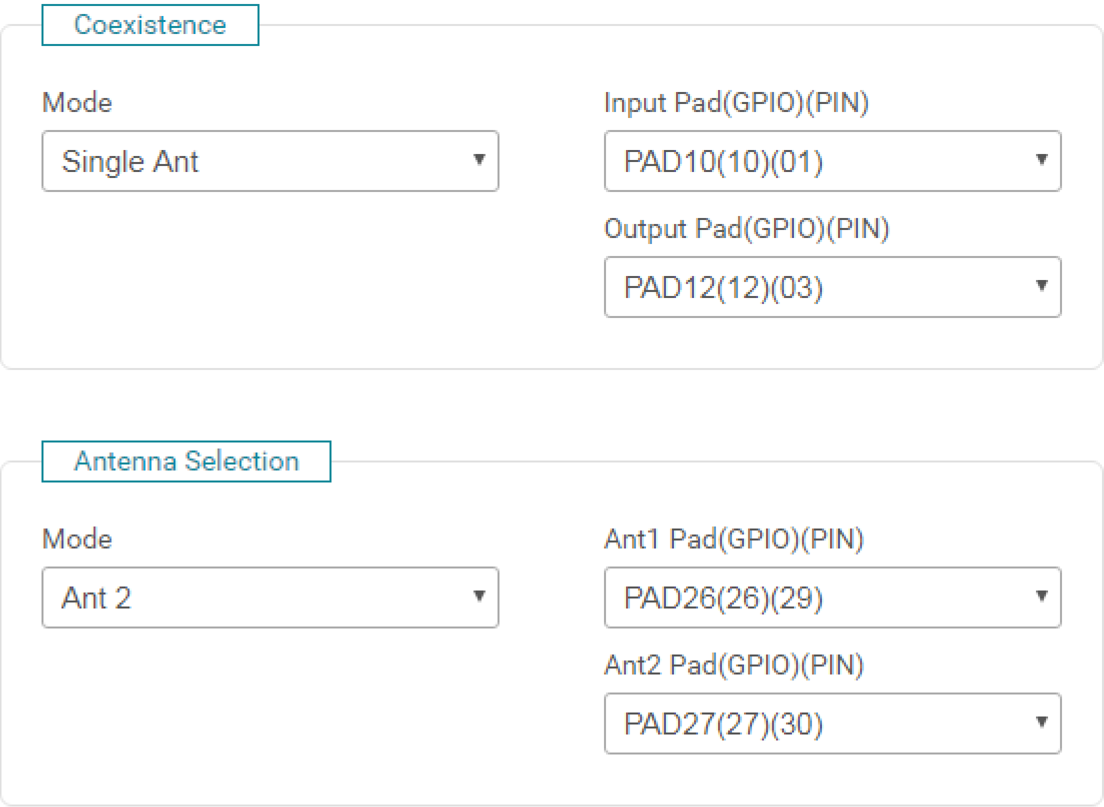
Coexistence modes:

- Off – BLE coexistence is not used (default)
- Single antenna – Choose this option when the platform includes an RF switch to share a single antenna between the BLE and Wi-Fi. This option requires the allocation of two GPIOs – one is input from the BLE as well as to the RF switch, the other is an output from the Wi-Fi to the RF switch.
- Dual antenna – Choose this option when the platform has separate antennas for BLE and Wi-Fi. In this mode, BLE signals Wi-Fi when it requires the channel, and Wi-Fi stops ongoing transmissions during those times. This mode requires the usage of just one I/O.

Antenna selection modes:

- Disable
- Ant1 – Statically select ant 1
- Ant2 – Statically select ant 2
- Auto select

See Figure 6-15.



Coexistence

Mode: Single Ant

Input Pad(GPIO)(PIN): PAD10(10)(01)

Output Pad(GPIO)(PIN): PAD12(12)(03)

Antenna Selection

Mode: Ant 2

Ant1 Pad(GPIO)(PIN): PAD26(26)(29)

Ant2 Pad(GPIO)(PIN): PAD27(27)(30)

Figure 6-15. Coexistence and Antenna Selection

6.8.1.2 Device Identity (DICE and CSR, Only for CC323xS/SF Devices)

The Device Identifier Composition Engine (DICE) is a security standard from the Trusted Computing Group (TCG). It is designed to help address the need for increased security in the Internet of Things (IoT) and targets devices such as microcontrollers. The DICE standard specifies a framework for hardware and software based on cryptographic device identity for authentication and attestation through a manufacturer's cloud servers (for example, Azure IoT cloud service).

The CC313x and CC323x implement DICE (Device Identifier Composition Engine), a protocol that provides foundations to enhance security and privacy without the need to add a costly TPM (Trusted Platform Module). The DICE implementation running on the CC313x and CC323x authenticates the individual chip identity and application code image with the manufacturer's cloud server, through the use of a client certificate chain in a TLS connection. This chain holds two certificates: the alias certificate signed by the device ID keys, and the device ID certificate.

The Certificate Signing Request (CSR) is the common way to create and sign a certificate. It can be created with the public key of the device and data that has been signed by the private key.

Texas Instruments simplifies the process of creating a CSR for a SimpleLink Wi-Fi device by providing a tool that enables the CSR to be generated internally by the device in PKCS #10 format.

Note

The CSR generation component of this feature can be used as an alternative to the CSR feature description in Section 9.

Note

The DICE features requires SP version 4.4.1.3_3.1.0.5_3.1.0.19 or higher.

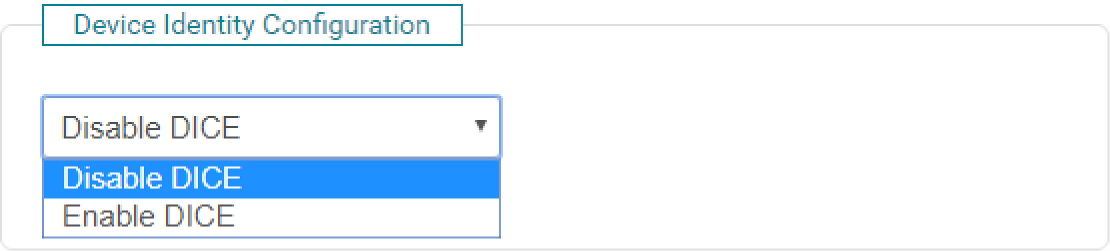
The Device Identity page holds the following sections:

- Device Identity Configuration (see Section 6.8.1.2.1)
- Certificate Configuration (see Section 6.8.1.2.2)
- Certificate Info (see Section 6.8.1.2.3)

6.8.1.2.1 Device Identity Configuration

To enable or disable the DICE feature, choose one of these modes:

- Enable DICE
- Disable DICE



Device Identity Configuration

Disable DICE ▾

Disable DICE

Enable DICE

Figure 6-16. Device Identity Configuration

Note

When choosing Enable DICE, the Certificate Configuration cannot be disabled.

6.8.1.2.2 Certificate Configuration

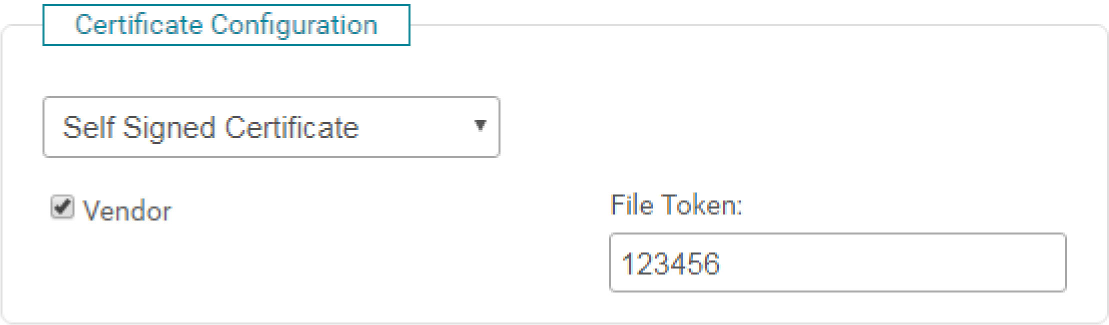
The Certificate configuration can be set to one of the following modes. Each mode opens specific fields in the certificate info.

- Certificate Sign Request (see Section 6.8.1.2.3.1)
- Self-Signed Certificate (see Section 6.8.1.2.3.2)
- Disabled (Available only when Disable DICE is chosen)

When the Certificate Configuration is enabled and the Device Identity Configuration is set to Disable DICE, the CSR part of this feature can be used instead of CSR from Section 9.

If DICE is enabled, either CSR or Self-Signed Certificate should be specified for the Certificate Configuration.

There is also an option to add a token in the Certificate Configuration, which allows the certificate file to be rewritten.



Certificate Configuration

Self Signed Certificate ▾

☑ Vendor

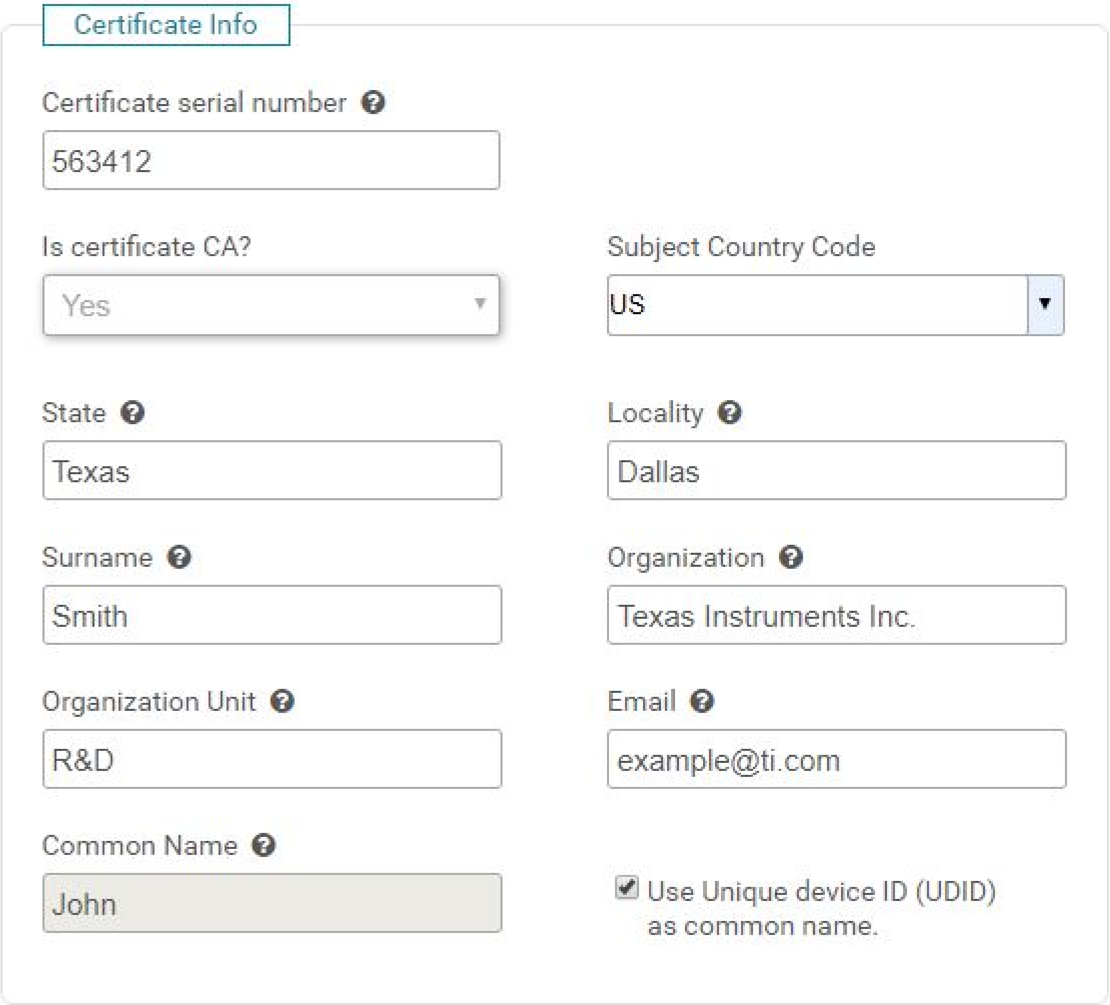
File Token: 123456

Figure 6-17. Certificate Configuration

6.8.1.2.3 Certificate Info

6.8.1.2.3.1 Certificate Sign Request Options

1. Serial number
2. Is certificate Client Authentication (when DICE is chosen, value is set to Yes)
3. Country Code
4. State
5. Locality
6. Surname
7. Organization
8. Organization unit
9. Email
10. Common Name
11. Use unique device ID (UDID) as common name



Certificate Info

Certificate serial number
563412

Is certificate CA?
Yes

Subject Country Code
US

State
Texas

Locality
Dallas

Surname
Smith

Organization
Texas Instruments Inc.

Organization Unit
R&D

Email
example@ti.com

Common Name
John

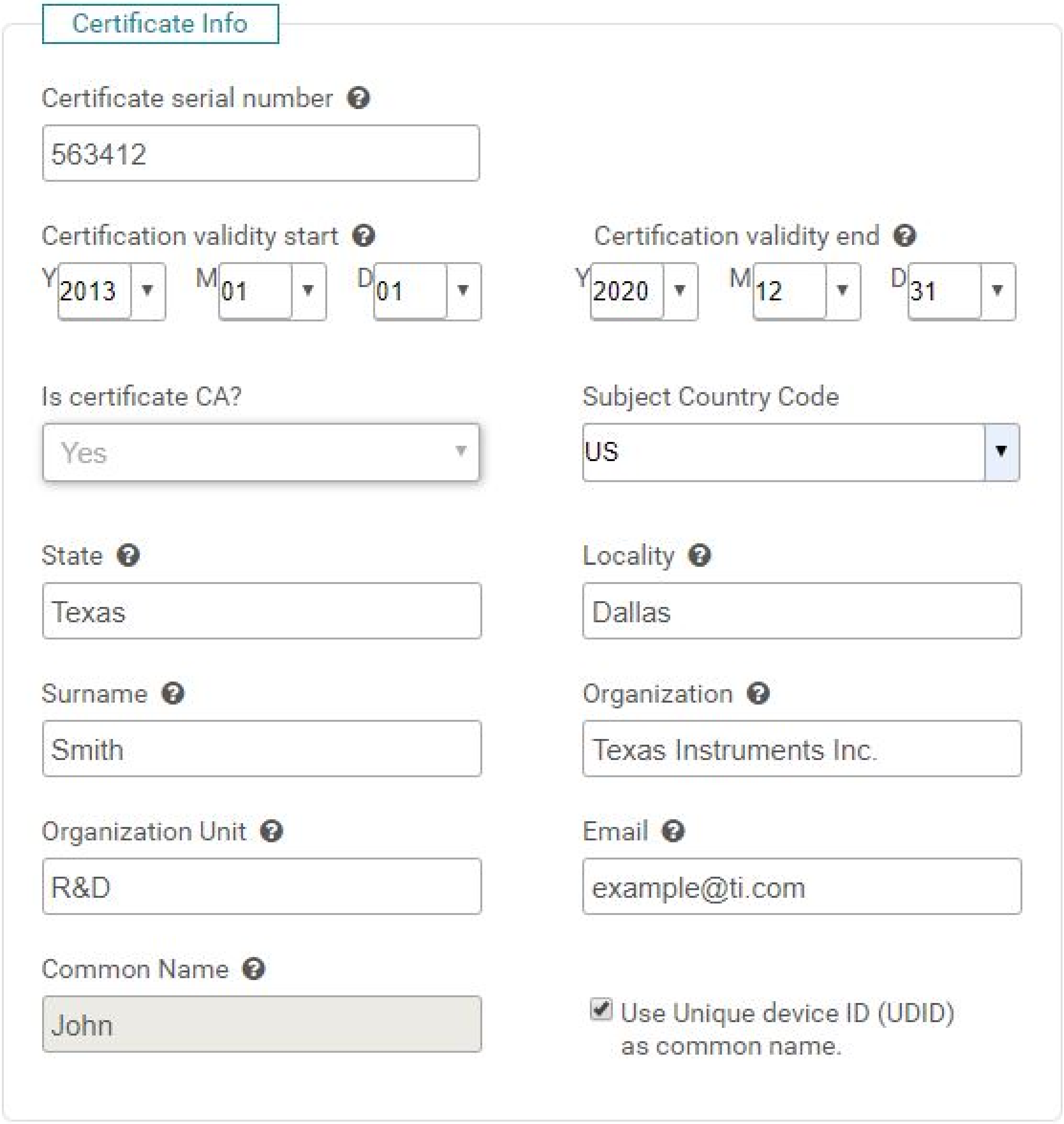
☑ Use Unique device ID (UDID) as common name.

Figure 6-18. Certificate Sign Request Options

6.8.1.2.3.2 Self-Signed Certificate Options

1. Serial number
2. Validity start/end

3. Is certificate Client Authentication
4. Country Code
5. State
6. Locality
7. Surname
8. Organization
9. Organization unit
10. Email
11. Common Name
12. Use unique device ID (UDID) as common name



Certificate Info

Certificate serial number: 563412

Certification validity start: Y 2013 M 01 D 01

Certification validity end: Y 2020 M 12 D 31

Is certificate CA?: Yes

Subject Country Code: US

State: Texas

Locality: Dallas

Surname: Smith

Organization: Texas Instruments Inc.

Organization Unit: R&D

Email: example@ti.com

Common Name: John

☑ Use Unique device ID (UDID) as common name.

Figure 6-19. Self-Signed Certificate Options

6.8.1.2.4 CSR ONLY Usage Example

This section shows how to create a CSR file.

1. According to Section 6.8.1.2.1, Disable DICE must be chosen.
2. According to Section 6.8.1.2.2 Certificate Sign Request must be chosen.
3. Fill the fields according to Section 6.8.1.2.3.1.
4. Program the device in Simple mode (Section 6.5) or Advance mode.
5. Use the Read CSR tool from Section 8.1 to retrieve the CSR file. The output file name is “csr.der”.
6. Sign the CSR file.
7. Write the signed certificate back to the device using the write certificate tool from Section 8.1.
8. Continue with the application that has been programmed.

6.8.1.2.5 DICE Usage Example

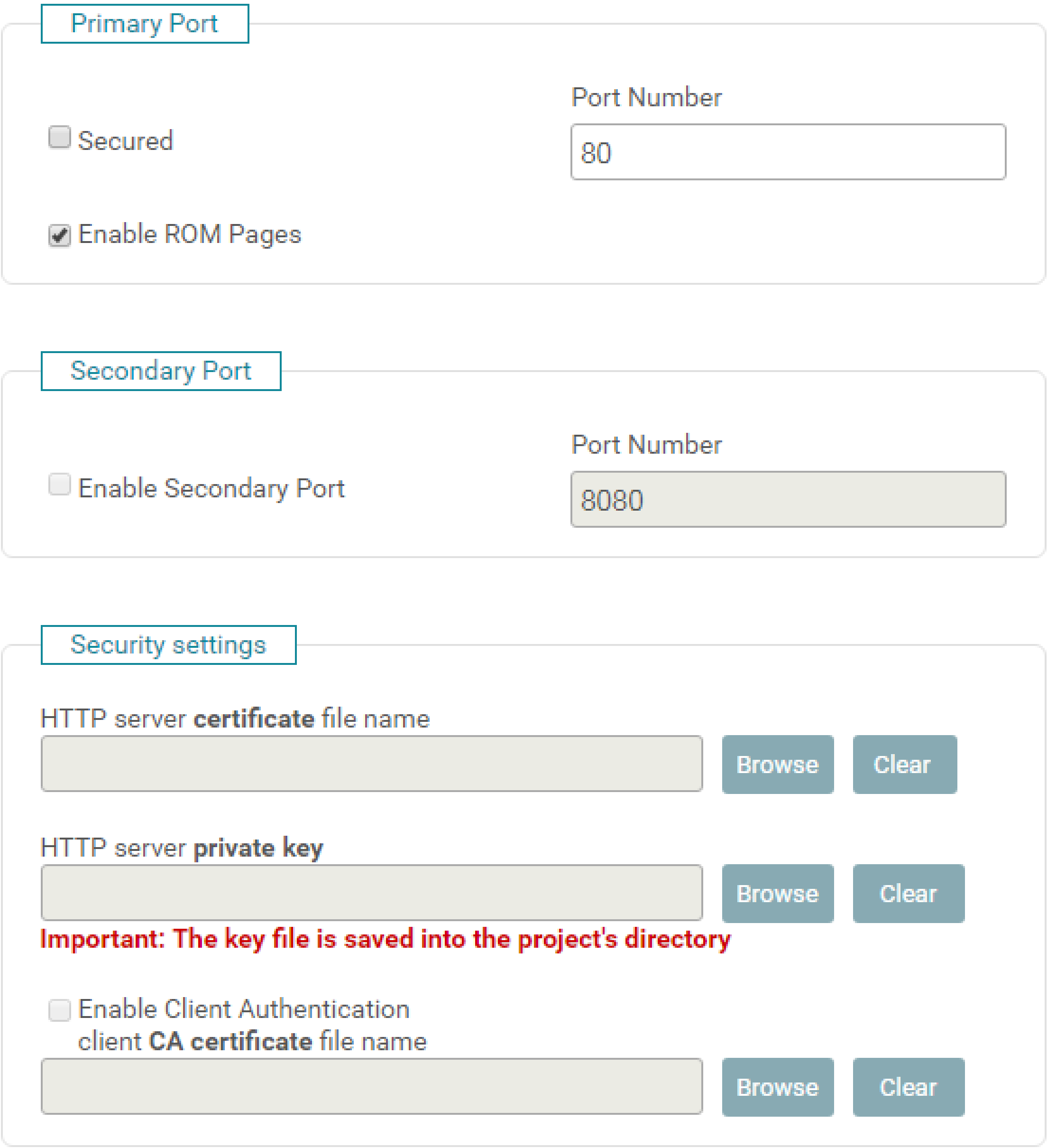
1. According to Section 6.8.1.2.1, Enable DICE must be chosen.
2. According to Section 6.8.1.2.2, choose Certificate Sign Request to work with the manufacturer’s cloud server that supports DICE.
3. Fill the fields according to Section 6.8.1.2.3.1.
4. Program your device in Simple mode (Section 6.5) or Advance mode.
5. Use the Read CSR tool from Section 8.1 to retrieve the CSR file. The output file name is “csr.der” (the name can be changed when using the read CSR through CLI commands: see Section 7.3.1).
6. Sign the CSR file using the manufacturer’s cloud server.
7. Write the signed certificate back to the device using the write certificate tool from Section 8.1.
8. To create a secure connection to the manufacturer’s cloud server, use:
   - The DICE certificate chain created with the name aliascert.pem and stored in the root directory.
   - The private key of the DICE certificate chain with the name tempkey02.der, and stored in the sys directory.
9. Continue with the application that has been programmed, and connect to the manufacturer’s cloud server.

6.8.2 Role Settings

- General Settings
  - Device mode:
    - Start role (AP/P2P/Station)
    - Country code
  - Device name
  - Connection policy:
    - Auto connect
    - Fast connect
    - Wi-Fi direct
    - Auto provisioning
  - Auto provisioning external confirmation
- STA/Wi-Fi Direct device
  - WLAN settings
  - Network settings
- AP/Wi-Fi Direct Go
  - WLAN settings
  - Network settings

6.8.3 HTTP Server

The options to configure the HTTP Server through ImageCreator are shown in Figure 6-20.



Primary Port

☐ Secured

Port Number

80

☑ Enable ROM Pages

Secondary Port

☐ Enable Secondary Port

Port Number

8080

Security settings

HTTP server **certificate** file name

HTTP server **private key**

Important: The key file is saved into the project's directory

☐ Enable Client Authentication

client **CA certificate** file name

Figure 6-20. HTTP Server

6.9 Adding the Service Pack

The service pack is used to upgrade the device software. The service pack file is provided by TI in the SDK package. The SP file name is sp_<release_version_number>.bin, and it is placed in the <SDK_PATH>\tools \cc32xx_tools\servicepack-cc3xXX folder. TI recommends adding the service pack to the programming image; this action, however, is not mandatory. If it is not programmed, the device uses its factory code.

When adding the service pack, the user selects the file location; however, the ImageCreator does not keep a link to the original file. To change the service pack, the new service pack file should be selected again.

6.10 Adding the Trusted Root-Certificate Catalog

The trusted root-certificate is a file provided by TI. The store contains a list of known and trusted root CAs and a list of revoked certificates. The list of the CAs supported by TI can be found in the *CC3x20, CC3x3x SimpleLink™ Wi-Fi® Internet-on-a chip™ solution built-in security features Application Report*.

The ImageCreator installation has a default trusted root-certificate catalog used by the ImageCreator. The default trusted root-certificate can be overridden by selecting a different file and its signature file. The ImageCreator has no link to the selected trusted root-certificate original file. To change the trusted root-certificate content, select a new file.

6.10.1 Vendor Certificate Catalog

The alternate bootloader allows a customer to use a self-signed certificate when signing their own firmware image.

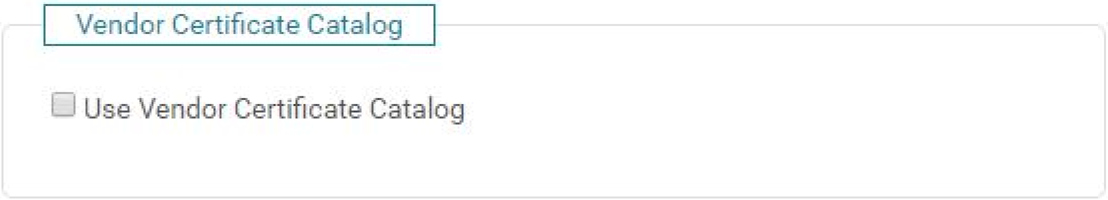


Figure 6-21. Vendor Certificate Catalog

6.10.2 OTP

The root-of-trust is kept one-time programming (OTP) memory, thus it cannot be replaced after it is programmed. In this section, the vendor can add device-specific information and sign it, to authenticate that the hardware platform is authentic and produced by the vendor. For more information, refer to the *Vendor Device Authentication With SimpleLink™ WiFi® Devices User's Guide*.

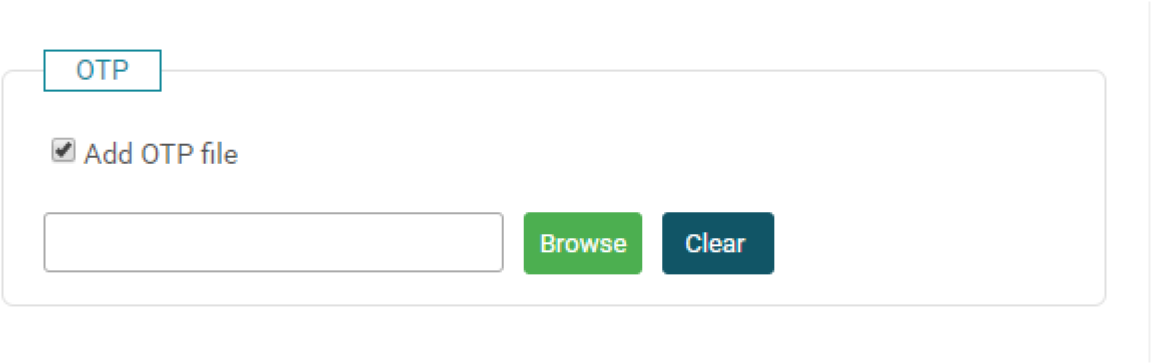


Figure 6-22. OTP Section

6.11 Adding the Host Application File (CC32xx)

The SimpleLink ImageCreator allows adding the host application file for CC32xx devices. For adding the MCU Image file in simple mode, see Section 6.5. In advanced mode, on the User Files, open the action drop menu and select the Select MCU Image, as shown in Figure 6-23.

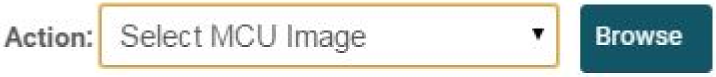


Figure 6-23. MCU Image Advanced Mode

Press the Browse button and select the MCU Image file from the local drive. The File properties dialog appears. See Section 6.12.3.

Configuring the host application file properties should be followed by clicking on the Save button. The host application file is created, with the following name on the device:

- For CC32xx R/RS: /sys/mcuimg.bin
- For CC32xxSF: /sys/mcuflashimg.bin

For secure devices, the host file must be created with the flags secure-signed. To enable future updates of the file (by OTA), the user must open it with the public-write flag. In addition, the maximum size of the file should consider the future growth of the file, as the maximum size of a file cannot be changed after the file creation.

6.11.1 Host Application for the CC32xxSF Devices

The CC3220SF application requires adding 20 bytes of SHA1 to the beginning of the host file. The SHA1 is the result of a hash algorithm calculated on the host file content. The file signature is calculated on the host file content, including the SHA1. The creation steps are as follows:

1. Add SHA1 to the beginning of the file (host_final).
2. Calculate the host signature (signature of the host_final file).

The ImageCreator offers two methods of adding the host application:

- The application file, including the SHA1, is created by the user and the file signature. The input is the host_final and the signature of the host_final.
- The host_file, without the SHA1, is set by the user:
    1. The ImageCreator calculates the SHA1 and the host_final.
    2. The file signature of the host_final file is calculated by the ImageCreator, using the private key in DER format as an input.

Note

Increase the File Maxsize setting of the host application by 20 bytes, to include the file SHA1.

6.12 User Files

A user file can be added to the image because the ImageCreator supports files operations, including adding or removing a file, creating a directory, and viewing file properties.

File operations are available while moving the cursor over the file/directory icon. After a file is chosen, the file is saved as part of the project.

The ImageCreator files are not linked to the original selected files. To change a file, content in the file should be deleted and added again.

6.12.1 Secure Signed User Files

For secure signed files, the ImageCreator must receive a signed certificate and the file signature. For more information regarding how to retrieve a signed certificate and how to create a file signature, refer to the secure file system chapter in the user manual (search for the sl_FsClose () function).

When the certificate is chained to another certificate, the name of the chained certificate should be in the certificate “issued to” field. All the certificates in the chain are added to the project before adding the file signed by them.

ImageCreator adds secure signed files using the following methods:

- Sets a file signature.
- Receives the private key; using the key, the ImageCreator creates the file signature.

Note

In both methods, a signed certificate containing the public key must be supplied.

To enable future updates of the file (by OTA), add the file with the public write flag. Another option is to use the vendor token flag and define the file master token.

6.12.2 Adding a File

To add a file, users should click the 🗎 icon, or drag-and-drop the desired file from the appropriate folder.

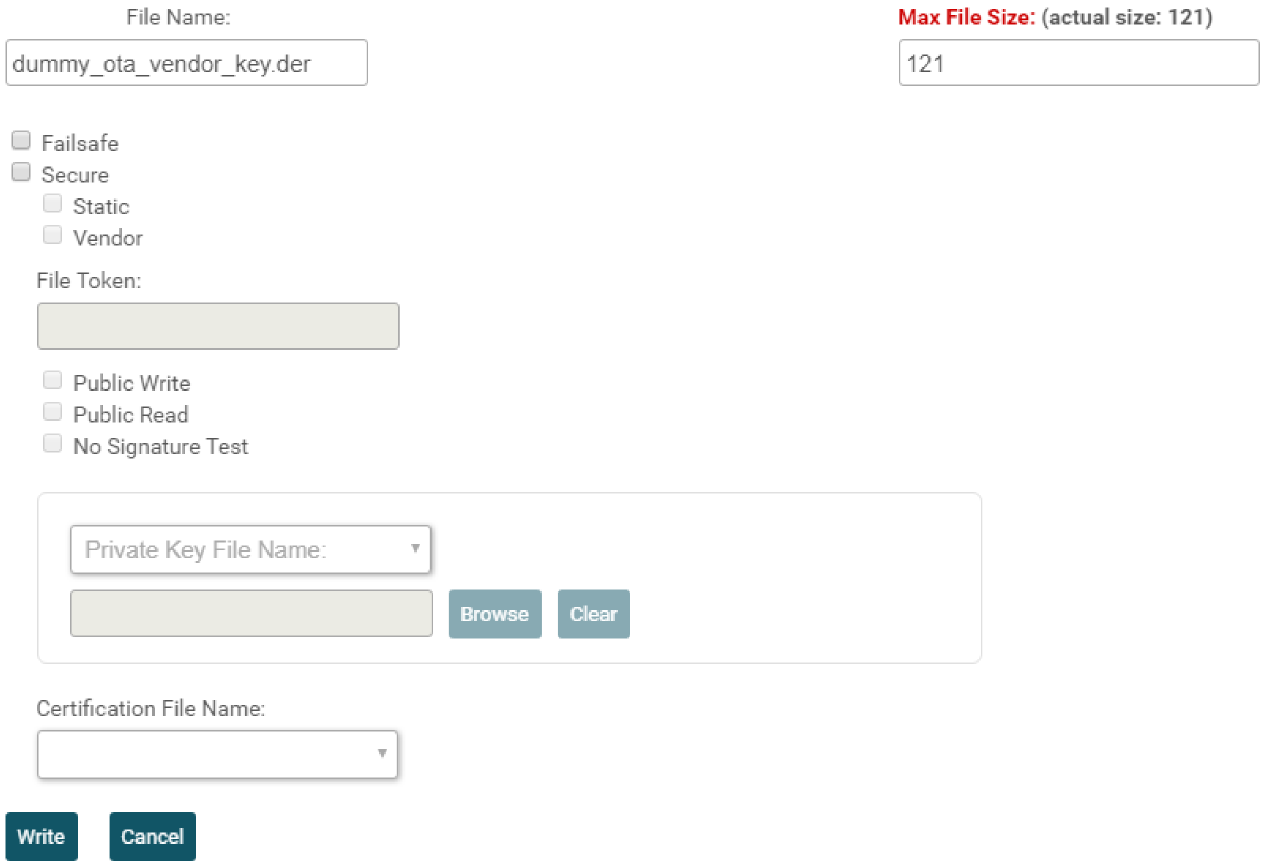
After a file is added, the File properties dialog appears. See Figure 6-24.

Note

Specification requires a certificate where the last line ends the UNIX end line format (only with the “\n”); any other symbol may cause unexpected behavior (for example, DOS end line format “\r\n” is prohibited).

6.12.3 Editing a File

To edit the properties of a file, select the file and press 📝. The File properties dialog appears, as shown in Figure 6-24.



File Name: dummy_ota_vendor_key.der

Max File Size: (actual size: 121) 121

□ Failsafe

□ Secure

□ Static

□ Vendor

File Token:

□ Public Write

□ Public Read

□ No Signature Test

Private Key File Name:

Certification File Name:

Figure 6-24. File Properties Dialog

Table 6-2 lists the flag options.

Table 6-2. Flags Options

Flag Option	Description
FailSafe	Editing the file is fail-safe. For more details, refer to the secure file system chapter in the user's manual, search for: SL_FS_CREATE_FAILSAFE
Secure	File is encrypted on the serial flash. For more details, refer to the secure file system chapter in the user's manual, search for: SL_FS_CREATE_SECURE
No Signature Test	Relevant only for secure files. By default, secure files require a signature. For more details, refer to the secure file system chapter in the user's manual, search for: SL_FS_CREATE_NOSIGNATURE
Static	Relevant only for secure files. Tokens are not replaced each time a file is open for write. For more details, refer to the secure file system chapter in the user's manual, search for: SL_FS_CREATE_STATIC_TOKEN
Vendor	Relevant only for secure files. The master token is set by the vendor. For more details, refer to the secure file system chapter in the user's manual, search for: SL_FS_CREATE_VENDOR_TOKEN
Public Write	Relevant only for secure files. The file can be written without a token. For more details, refer to the secure file system chapter in the user's manual, search for: SL_FS_CREATE_PUBLIC_WRITE
Public Read	Relevant only for secure files. The file can be read without a token. For more details, refer to the secure file system chapter in the user's manual, search for: SL_FS_CREATE_PUBLIC_READ
No Signature Test	Relevant only for secure files. By default, secure files require a signature. For more details, refer to the secure file system chapter in the user's manual, search for: SL_FS_CREATE_NOSIGNATURE

Table 6-3. Other File Properties

File Property	Description
File token	Relevant only when using the vendor flag. Token for secured file.
Signature filename	Relevant only when using the secure-signed flag, the signature file should be picked by browsing it on the local machine. The filename should then appear in the test box.
Certification filename	Relevant only when using the secure-signed flag, the list of previously added certificates should appear. Users should pick the relevant file from the list.
Maximum size	The size of the storage to allocate for the file. By default, the ImageCreator sets the maximum size as the actual size of the file. To enable future updates of the file (by OTA) set the maximum size to the maximum future growth of the file, (maximum size of a file cannot be changed for the existing file). The maximum size of a file will be rounded up by the device to correlate the serial flash block size (4096 bytes); for more information, see the secure file system chapter in the user manual.

To rename a file, use the File Name field on the on the File Properties dialog, as shown in Figure 6-25.

File Name: ca-cert.der

Figure 6-25. Rename Filename

6.12.4 Adding a Folder

To add a new folder, locate the desired position on the root folder and click the 🗀 icon.

6.12.5 Deleting a File or Folder

Check the files or folders to be deleted. In the Action drop-down menu, choose Remove Selected (see Figure 6-26), then click Apply (see Figure 6-27).

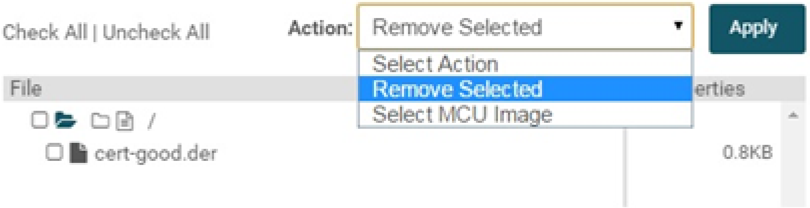

Action: Remove Selected

Figure 6-26. Delete File or Folder (1 of 2)

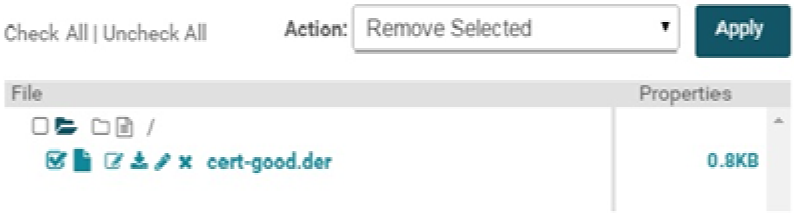

Action: Remove Selected

Figure 6-27. Delete File or Folder (2 of 2)

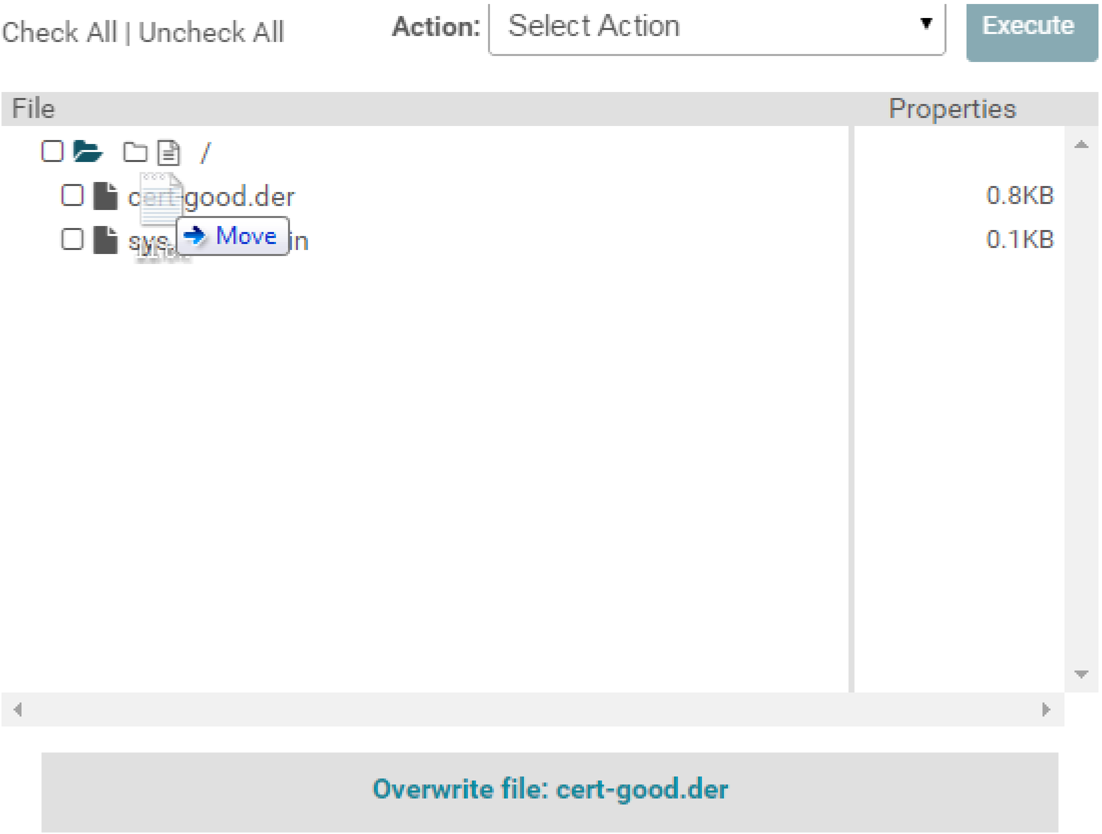
The user can also delete a single file or folder by directly clicking its X (delete) button.

6.12.6 Overwriting a File

The user can overwrite the file by clicking on the pencil button (✏).

6.12.7 User File Action Monitor

The user can drag a file and drop it on the user file area, as shown in Figure 6-28. During the action, the user file action monitor shows the action itself.



Overwrite file: cert-good.der

Figure 6-28. User File Action Monitor

6.13 Device File Browser

A device programmed with a development image allows for browsing its files, user files, and system files, and can perform several operations listed in the following subsections.

Most system files can not be viewed in the file list, and have no read/write access.

The online browser edits the file on the device serial flash; the changes do not affect the programming image file, which is prepared with the offline browser.

To view the online file browser, press the following icon: 🗁 Online User Files

6.13.1 Adding a File

To add a file, click the 🗎 icon.

The File properties dialog appears. See Section 6.12.

6.13.2 Editing a File

Move the cursor over a filename, and click the 📝 icon.

The File properties dialog appears. See Figure 6-24.

6.13.3 Adding a Folder

To add new folder, locate its position on the root folder and click the 🗀 icon.

6.13.4 Deleting a File

To delete a file, point to the file to be deleted and click the Delete button, as shown in Figure 6-29. If the file is secure, a prompt for the file token will appear.

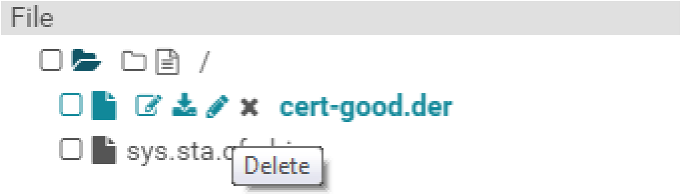


Figure 6-29. Delete File

6.13.5 Retrieving a File

To retrieve (upload) a file from the device, move the cursor over the file and click the Get File button, as shown in Figure 6-30. If the file is secure, a prompt for the file token will appear.

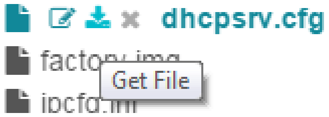


Figure 6-30. Get File

6.14 Creating an Image From a Project

When the configuration phase is over, creating an image is done by clicking on the Generate Image button Burn.

In this phase, all types of programming image files are created, and can be found under the ImageCreator installation directory.

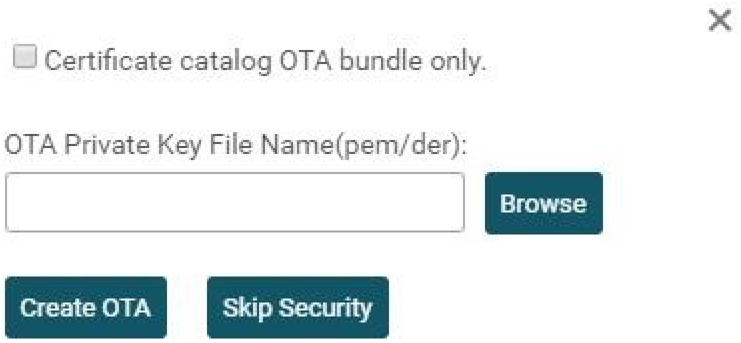
To create an image, click the Create Image button: Create Image.

For creating an encrypted image, see Section 6.7.1.

6.15 Creating an OTA

To create an OTA image, click the Create OTA button Create OTA.

The OTA security dialog appears as in Figure 6-31.



□ Certificate catalog OTA bundle only.

OTA Private Key File Name(pem/der):

Figure 6-31. OTA Private Key File Name

6.15.1 Creating an OTA With a Security Sign

1. Click on the Browse button and load the OTA private file name (pem or der).
2. Click the Create OTA button.

6.15.2 Creating an OTA Without a Security Sign

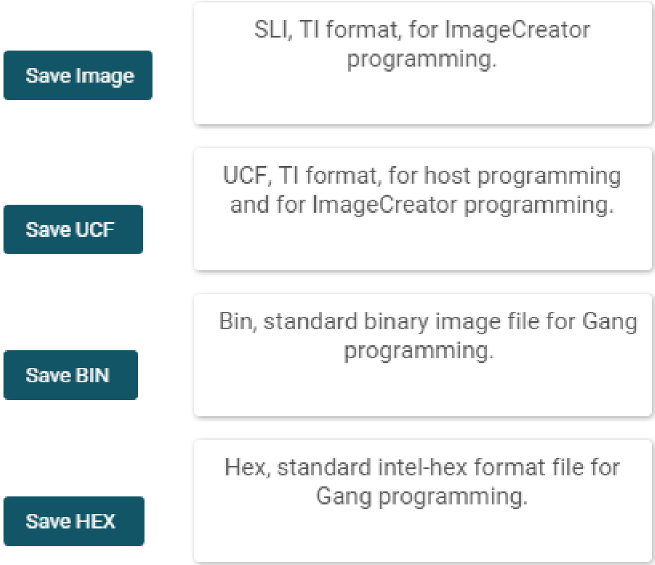
Click Skip Security button.

6.15.3 Use Certificate Catalog OTA Bundle Only

If the certificate catalog must be updated, OTA should be performed in two steps. First, the tar file should include only the certificate catalog and its signature. In this case, the check box Certificate catalog OTA bundle only should be marked. No other file can be included in the tar file once certificate catalog is present. The second step includes a tar file with all other files that are required, such as MCU image, service pack, and so forth.

6.16 Saving an Image

The Save Image buttons become clickable upon a successful image creation.



Save Image

SLI, TI format, for ImageCreator programming.

Save UCF

UCF, TI format, for host programming and for ImageCreator programming.

Save BIN

Bin, standard binary image file for Gang programming.

Save HEX

Hex, standard intel-hex format file for Gang programming.

Figure 6-32. Save Image

6.17 Programming.bin and Programming.hex

Standard binary and intel hex files are used for programming by an external Sflash programming tool.

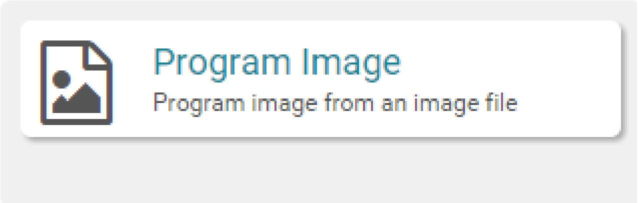
- Programming.ucf (TI proprietary encoding) is used for programming by the host.
- Programming.sli (TI proprietary encoding) is used for programming by the image creator.

6.18 Programming an Image From an Opened Project

See Section 5.5.

6.19 Programming an Image Using a .sli File

When an .sli file is created, it can be used by any instance of the image creator to program the device. To use an existing .sli file, click the Program Image button on the Welcome page, as shown in Figure 6-33.



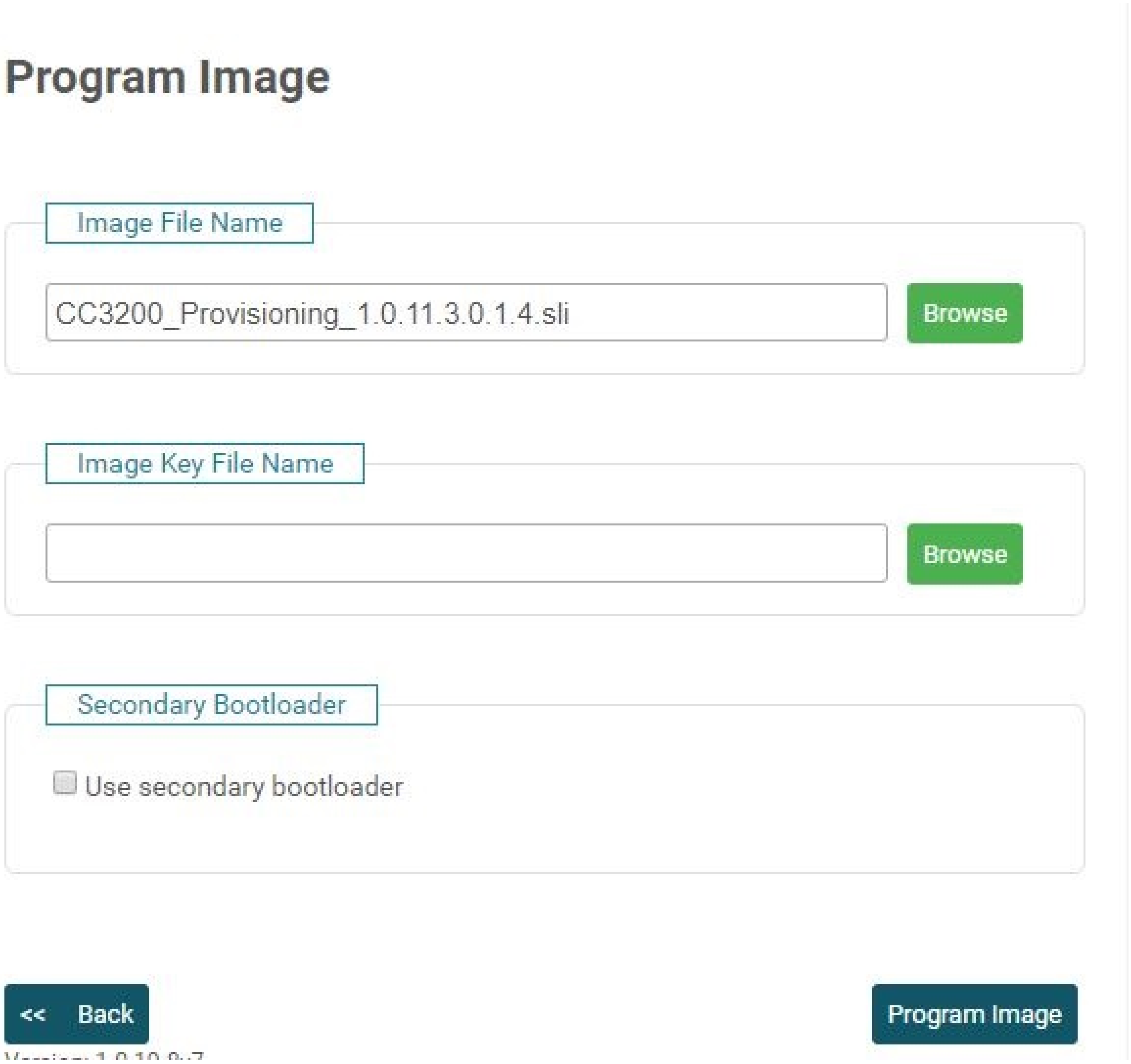
Program Image

Program image from an image file

Figure 6-33. Program Image

The Program image dialog appears. Set the .sli file to be programed, and click on the Program Image button.

For programming a secured .sli file image, set the key file used for the image encrypting (see Section 6.7.1), then click the Program Image button (see Figure 6-34).



Program Image

Image File Name: CC3200_Provisioning_1.0.11.3.0.1.4.sli

Image Key File Name:

Secondary Bootloader

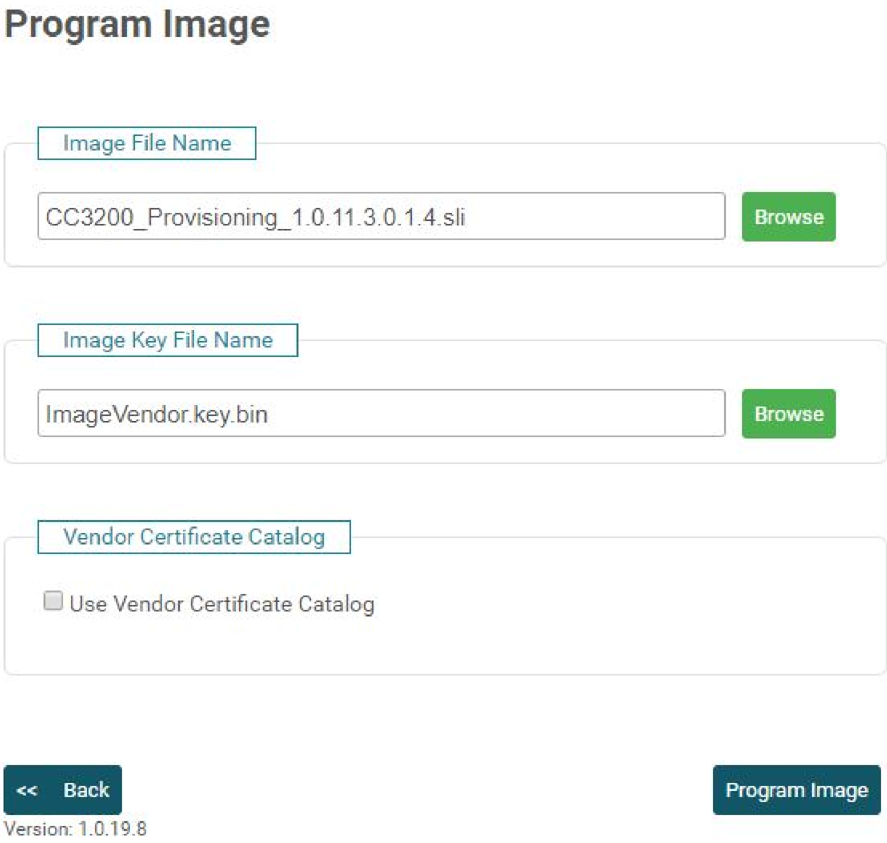
□ Use secondary bootloader

Figure 6-34. Program Image

A sign file tool is in the tools section. After the user chooses a file to sign and a private key (see file system user manual for more information about supported key formats), the user can get the signed file as either binary or base 64 (see Section 8.3).

6.20 Secured Image With Key

For programming a secured .sli file image, set the key file which was used for the image encrypting (see Section 6.7.1), then click the Program Image button (see Figure 6-35).



Program Image

Image File Name

CC3200_Provisioning_1.0.11.3.0.1.4.sli

Image Key File Name

ImageVendor.key.bin

Vendor Certificate Catalog

☐ Use Vendor Certificate Catalog

Version: 1.0.19.8

Figure 6-35. Program Image

7 Command Line

Navigate to the UniFlash install directory:

- For Windows:

```
cd  c:\ti\uniflash_X.X
```

- For Linux:

```
cd /home/YOUR_USER/ti/uniflash_4.0
```

- For Mac OS X:

```
cd /Users/YOUR_USER/ti/uniflash_X.X
```

Use the dslite shell script to send commands in cc31xx/cc32xx mode:

- For Windows:

```
dslite.bat --mode cc31xx COMMAND
dslite.bat --mode cc32xx COMMAND
```

- For Linux and Mac OS X:

```
./dslite.sh --mode cc31xx COMMAND
./dslite.sh --mode cc32xx COMMAND
```

7.1 Project Commands

7.1.1 Add a File or Set an MCU Image

Basic command:

```
project add_file --name PROJECT_NAME --file MCU_FILENAME.bin --mcu
       -OR-
project add_file --name PROJECT_NAME --file MCU_FILENAME.bin --fs_path /path/filename.ext
```

Required arguments:

--name PROJECT_NAME	Name of the project to use
--file FILENAME	File to add
--fs_path /path/filename.ext -OR- --mcu	File path or name in the SimpleLink file system The file is an MCU image

Optional arguments:

--sign SIGNATURE_FILENAME -OR- --priv PRIVATE_KEY_FILENAME	Signature file used to sign the MCU image file Private key to be used to generate a signature for the file
--flags flag1,flag2,...	File flags, available values: failsafe, secure, nosignaturetest, static, vendor, publicread, publicwrite, nofailsafe, nopublicwrite. The last two are negative flags. The default is that failsafe and publicwrite are true. To disable that, set negative flags.
--token TOKEN_NUMBER	File token a 32-bit unsigned integer, used in conjunction with the vendor flag
--max_size MAX_SIZE_IN_BYTES	Maximum size in bytes to be allocated for the file in the SimpleLink file system
--cert CERT_NAME	Certificate file to use (from the device file system); does not change if omitted. Use --cert "" to erase.
--overwrite	Force overwrite in case the file already exists
--project_path PROJECT_PATH	Path to the projects folder
--cfg_json PATH_CFG_JSON_FILE	Full path to the cfg.json file

Notes:

- In case of "--mcu", the security properties and maximum file size are selected automatically in accordance to the project type, but can also be overridden with the "--flags" and "--max_size" options.
- The command prints an error and exits if the file already exists in the project; use "--overwrite" to force an overwrite.

Examples:

Set MCU image:

```
project add_file --name MY_PROJECT --file MCU_FILENAME.bin --mcu
```

Add a file:

```
project add_file --name MY_PROJECT --file MY_TEXT_FILE.txt --fs_path /mydir/myfilename.txt --
flags "failsafe,publicwrite"
```

Set secure MCU:

- Add certificate file first (certificates always reside in the SimpleLink file system root directory):

```
project add_file --name MY_PROJECT --file MY_CA_CERT.der --fs_path CA_CERT
```

- Set MCU image and sign with private key with project from non-default folder:

```
project add_file --name MY_PROJECT --project_path FULL_PROJECTS_PATH --file MCU_FILENAME.bin --
mcu --priv MY_PRIVATE_KEY_FILENAME.key --
cert CA_CERT
```

7.1.2 Set Service Pack

The service pack file is provided by TI in the SDK package. The SP file name is sp_<release_version_number>.bin, and it is placed in the <SDK_PATH>\tools\cc32xx_tools\servicepack-cc3xXX folder.

Basic command:

```
project set_sp --name PROJECT_NAME --file SP_FILENAME.bin
```

Required arguments:

--name PROJECT_NAME	Name of the project to use
--file FILENAME	Service pack .bin file

Optional arguments:

--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

7.1.3 Program Image (From a Project)

Basic command:

```
project program --name PROJECT_NAME
```

Required arguments:

--name PROJECT_NAME	Name of the project to use

Optional arguments:

--vendor_cert	Use Vendor Certificate Catalog

--otp_file OTP_INF_FILE	OTP file
--port COM<port_number>	COM port to use
--reconfig RECONFIG_FILENAME.json	Apply reconfiguration file on the fly (without saving to project)
-dev	Confirm programming in case the project is in development mode
--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file
--script_path SCRIPT_PATH	Full path to the power_on/off scripts folder

Notes:

- If the project is in development mode, programming is not allowed without passing the "--dev" argument.
- There is an option to call to COM port as parameter. In this case, the user should provide and use power_off_com.py/ power_on_com.py to reset the device.

Example:

```
project program --name PROJECT_NAME --project_path PROJECT_PATH --script_path SCRIPT_PATH --port
COM11
```

7.1.4 Set Trusted Root-Certificate Catalog

Basic command:

```
project set_certstore --name PROJECT_NAME --file CERT_STORE.lst --sign certstore.lst.signed
```

Required arguments:

--name PROJECT_NAME	Name of the project to use
--file CERT_STORE.lst	Trusted Root-Certificate Catalog file
--sign CERT_STORE.lst.signed	Trusted Root-Certificate Catalog signature file

Optional arguments:

--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- Specifing both "--file" and "--sign" as empty switches back to using the default files provided. For example: project set_certstore --name PROJECT_NAME --file "" --sign ""

7.1.5 Export Project

Basic command:

```
project export --name PROJECT_NAME --file EXPORTED_PROJECT.zip
```

Required arguments:

--name PROJECT_NAME	Name of the project to use
--file EXPORTED_PROJECT.zip -OR- --path PATH	Exported project archive file name Write archive to path with time stamped filename

Optional arguments:

--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- Using "--path" generates a time stamped filename for the archive, and creates the file in the given path.

7.1.6 Import Project

Basic command:

```
project import --file EXPORTED_PROJECT.zip
```

Required arguments:

--file EXPORTED_PROJECT.zip	An exported project archive file

Optional arguments:

--overwrite	Force overwriting existing project with the same name
--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- The command refuses to import if a project with the same name exists; use "--overwrite" to force overwriting.

7.1.7 Clone Project

Basic command:

```
project clone --name PROJECT_NAME --new NEW_PROJECT_NAME
```

Required arguments:

--name PROJECT_NAME	Name of the project to clone
--new NEW_PROJECT_NAME	New project name for the cloned project

Optional arguments:

--overwrite	Force overwriting existing project with the same name
--with_key	Copy encryption key to the cloned project
--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- The command refuses to import if a project with the same name exists; use "--overwrite" to force overwriting.
- If the project has an encryption key, the command does not copy it to the cloned project unless "--with_key" is used.

7.1.8 New Project

Basic command:

```
project new --name NEW_PROJECT_NAME
```

Required arguments:

--name NEW_ PROJECT_NAME	Name of the project to create

Optional arguments:

--device	Device type (CC3220S, CC3220R, CC3120R, CC3220SF, CC3235S, CC3235SF, or CC3135R)

--mode	Mode development/production
--description	Project description
--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Example:

```
project new --name PROJ_NAME --device DEVICE_TYPE --mode MODE --description DESC
```

Notes:

- Default values are CC32xx Production without description

7.1.9 Create Image (From Project)

Basic command:

```
project create_image --name PROJECT_NAME --sli_file IMAGE_FILENAME.sli
```

Required arguments:

--name PROJECT_NAME	Name of the project to use
--sli_file IMAGE_FILENAME.sli	Image file name to write as sli file

Optional arguments:

--ucf_file IMAGE_UCF_FILENAME.ucf	Image UCF file name to write
--bin_file IMAGE_BIN_FILENAME.bin	Image bin file name to write
--hex_file IMAGE_HEX_FILENAME.hex	Image hex file name to write
--reconfig RECONFIG_FILENAME.json	Apply reconfiguration file on the fly (without saving to project)
--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

7.1.10 Reconfigure Project

Basic command:

```
project reconfig --name PROJECT_NAME --file RECONFIG_FILENAME.json
```

Required arguments:

--name PROJECT_NAME	Name of the project to use
--file RECONFIG_FILENAME.json	Apply reconfiguration file

Optional arguments:

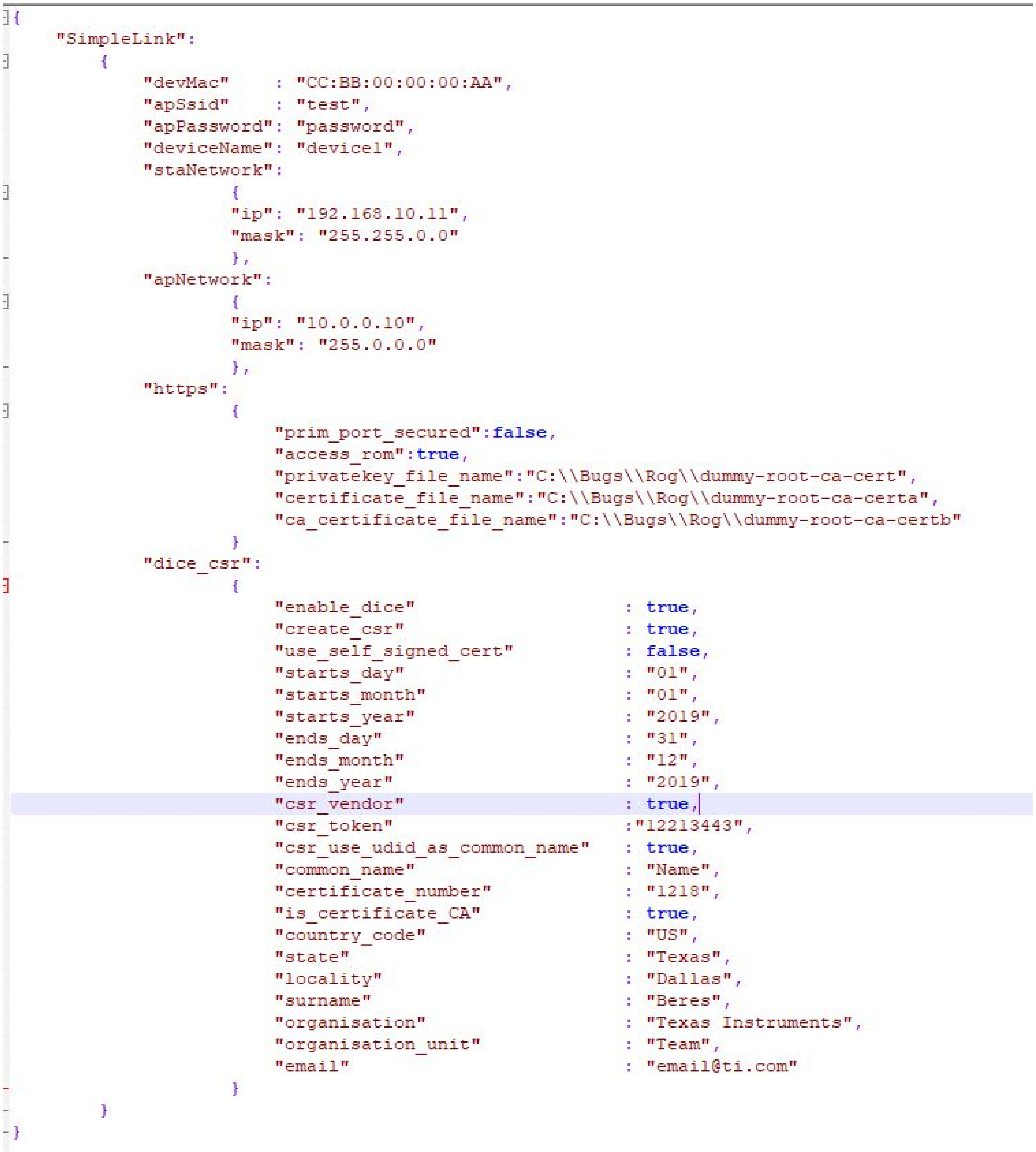
--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- The reconfiguration is applied and saved into the project.
- reconfig.json supported arguments:
  - “macAddress”
  - "startRole"
  - "countryCode"
  - “devMac “
  - “apSsid”

- “apPassword”
- “deviceName”
- “staNetwork”
  - “ip”
  - “mask”
  - “gateway”
  - “dns”
  - “dhcp”
- “apNetwork”
  - “ip”
  - “mask”
  - “gateway”
  - “dns”
  - “startIp”
  - “lastIp”
- "https"
  - "prim_port_secured"
  - "access_rom"
  - "prim_port_val"
  - "sec_port_enable"
  - "sec_port_val"
  - "access_ca_cert"
  - "privatekey_file_name"
  - "certificate_file_name"
  - "ca_certificate_file_name"
- "dice_csr"
  - "enable_dice"
  - "create_csr"
  - "use_self_signed_cert"
  - "starts_day"
  - "starts_month"
  - "starts_year"
  - "ends_day"
  - "ends_month"
  - "ends_year"
  - "csr_vendor"
  - "csr_token"
  - "csr_use_udid_as_common_name"
  - "common_name"
  - "certificate_number"
  - "is_certificate_CA"
  - "country_code"
  - "state"
  - "locality"
  - "surname"
  - "organisation"
  - "organisation_unit"
  - "email"

– each file should start with SimpleLink name. Example:



```
{
    "SimpleLink":
        {
            "devMac"    : "CC:BB:00:00:00:AA",
            "apSsid"    : "test",
            "apPassword": "password",
            "deviceName": "device1",
            "staNetwork":
                    {
                    "ip": "192.168.10.11",
                    "mask": "255.255.0.0"
                    },
            "apNetwork":
                    {
                    "ip": "10.0.0.10",
                    "mask": "255.0.0.0"
                    },
            "https":
                    {
                        "prim_port_secured":false,
                        "access_rom":true,
                        "privatekey_file_name":"C:\\Bugs\\Rog\\dummy-root-ca-cert",
                        "certificate_file_name":"C:\\Bugs\\Rog\\dummy-root-ca-certa",
                        "ca_certificate_file_name":"C:\\Bugs\\Rog\\dummy-root-ca-certb"
                    }
            "dice_csr":
                    {
                        "enable_dice"                       : true,
                        "create_csr"                        : true,
                        "use_self_signed_cert"              : false,
                        "starts_day"                        : "01",
                        "starts_month"                      : "01",
                        "starts_year"                       : "2019",
                        "ends_day"                          : "31",
                        "ends_month"                        : "12",
                        "ends_year"                         : "2019",
                        "csr_vendor"                        : true,
                        "csr_token"                         :"12213443",
                        "csr_use_udid_as_common_name"       : true,
                        "common_name"                       : "Name",
                        "certificate_number"                : "1218",
                        "is_certificate_CA"                 : true,
                        "country_code"                      : "US",
                        "state"                             : "Texas",
                        "locality"                          : "Dallas",
                        "surname"                           : "Beres",
                        "organisation"                      : "Texas Instruments",
                        "organisation_unit"                 : "Team",
                        "email"                             : "email@ti.com"
                    }
        }
}
```

7.1.11 List Available Projects

Basic command:

```
project list
```

Optional arguments:

--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- Prints out all available projects
- Unless specified in a cfg.json file, the project directory is situated at:
  - Win 7/10 – c:\Users\the_user\.SLImageCreator\projects
  - Linux – /home/the_user/.SLImageCreator/projects
  - Mac OS X – /Users/the_user/.SLImageCreator/projects

7.1.12 Create OTA Archive From the Project

Basic command:

```
project create_ota --name PROJECT_NAME --file TAR_FILE
```

-OR-

```
project create_ota --name PROJECT_NAME --path TAR_PATH
```

Required arguments:

--name PROJECT_NAME	Name of the project to use
--file -OR- --path	File path for destination tar file Destination path – <PROJECT_NAME>.tar file will be created

Optional arguments:

--project_path PROJECT_PATH	Path to the projects folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file
--priv	Private key file name for secured OTA
--cert_catalog	Set certificate catalog only, ignore other files

7.2 Image Commands

7.2.1 Program Image

Basic command:

```
image program --file IMAGE_FILENAME.sli
```

Required arguments:

--file IMAGE_FILENAME.sli	Image file name

Optional arguments:

--vendor_cert	Use Vendor Certificate Catalog
--otp_file <OTP_FILE>	Provide OTP file
--key KEY_FILENAME	Key file name

--port COM<port_number>	COM port to use
--script_path SCRIPT_PATH	Path to the power_on/off scripts folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- There is an option to call to COM port as parameter. In this case, the user should provide and use power_off_com.py/ power_on_com.py to reset the device.

7.3 Tools Commands

7.3.1 Read CSR File (Only for CC323xS/SF Devices)

This command can be used instead of CSR from Section 9.

Basic command:

```
tools get_csr --out_file OUT_FILENAME
```

Required arguments:

--out_file OUT_FILENAME	CSR out file name

For CSR configuration, see Section 7.1.10.

7.3.2 Write Certificate File (Only for CC323xS/SF Devices)

This command can be used instead of CSR from Section 9.

Basic command:

```
tools set_csr --file CERT_FILENAME
```

Required arguments:

-- file CERT_FILENAME	Certificate file name for writing to device

7.3.3 Sign File

Basic command:

```
tools sign --file FILENAME --priv PRIVATE_KEY_FILENAME --out_file OUT_FILENAME
```

Required arguments:

--file FILENAME	File to sign
--priv PRIVATE_KEY_FILENAME	Private key to use for signing
--out_file SIGNATURE_FILENAME	Signature file name

Optional arguments:

--fmt	"BINARY_SHA1"/"BINARY_SHA2"/"BASE64"

7.3.4 Activate Image

Basic command:

```
tools activate --key KEY_FILENAME
```

Required arguments:

--key KEY_FILENAME	Key file name

Optional arguments:

--port COM<port_number>	COM port to use
--script_path SCRIPT_PATH	Path to the power_on/off scripts folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- There is an option to call to COM port as parameter. In this case, the user should provide and use power_off_com.py/ power_on_com.py to reset the device.

7.3.5 Create the otp.meta Section

Basic command:

```
tools meta --cert CERT_FILE_NAME --out_file OUTPUT_FILE --mac "112233445566" -usechain
```

Required arguments:

--cert CERT_FILE_NAME	Vendor certificate file name
--out_file OUTPUT_FILE	Output file name

Optional arguments:

--mac MAC_ADDRESS	Mac address. Format 12 hex digits
--usechain	Use second signature

7.3.6 Create the otp.inf File

Basic command:

```
tools inf --algo 2 --sign1 SIGNATURE --sign2 SIGNATURE_2 --meta META_FILE --out_file OUTPUT_FILE
```

Required arguments:

--algo ALGO	ALGO values : 1 - RSASHA1, 2 - RSASHA256
--sign1 SIGNATURE	Self-signature file name
--meta META_FILE	Meta section file name
--out_file OUTPUT_FILE	Output file name

Optional arguments:

--sign2 SIGNATURE2	Vendor chain signature file name. Use chain flag from meta command (see Section 7.3.5) should be set. Otherwise this signature is ignored.

7.3.7 Create cert Catalog

Basic command:

```
tools make_cert_catalog --cert_folder CERT_FOLDER --out_file OUTPUT_FILE
```

Required arguments:

--cert_folder	Vendor certificate file name
--out_file OUTPUT_FILE	Output file name

Notes:

- All the certificates must be in DER format to create the certificate store file.

7.4 Device Commands

7.4.1 Get Device Information

Basic command:

```
device info
```

Optional arguments:

--json	Print to stderr in JSON format
--port COM<port_number>	COM port to use
--script_path SCRIPT_PATH	Path to the power_on/off scripts folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- There is an option to call to COM port as parameter. In this case, the user should provide and use power_off_com.py/ power_on_com.py to reset the device.
- Using "--json" prints a JSON object with the information into stderr, allowing a script to easily capture and deserialize the information.

Example:

```
device info --json
```

- Capture stderr
- If the return code is 0, then deserialize the text captured from stderr as a JSON object.

7.4.2 Restore to Factory Image

Basic command:

```
device restore
```

Optional arguments:

--defaults_only	Restore defaults only
--port COM<port_number>	COM port to use
--script_path SCRIPT_PATH	Path to the power_on/off scripts folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- There is an option to call to COM port as parameter. In this case, the user should provide and use power_off_com.py/ power_on_com.py to reset the device.

7.5 GUI Configure Commands

7.5.1 Configure GUI

Basic command:

```
gui_cfg
```

Optional arguments:

--project_path PROJECT_PATH	Path to the projects folder
--port COM<port_number>	COM port to use
--script_path SCRIPT_PATH	Path to the power_on/off scripts folder
--cfg_json CFG_JSON_PATH	Full path to the cfg.json file

Notes:

- This command allows setting the com port for GUI.
- There is an option to call to COM port as parameter. In this case, the user should provide and use power_off_com.py/ power_on_com.py to reset the device.
- Cfg.json supports next parameters:

projectDir	Path to the projects folder
tempDir	Path to the log folder
scriptDir	Path to the power_on/off scripts folder

7.6 GUI Commands Additional Arguments

7.6.1 Get ImageCreator Version

- v or –version

Example:

```
SLImageCreator.exe  -v
```

7.6.2 Quiet Print Mode

-q or --quiet

-q or --quiet should be before command

Example:

```
SLImageCreator.exe  -q -v
```

8 Tools

Click on the Tools button on the welcome page, as shown in Figure 8-1.

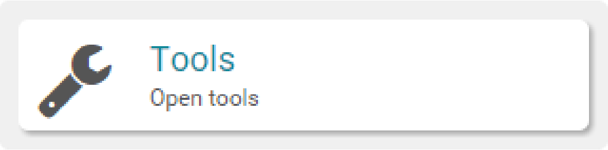


Figure 8-1. Open Tools

Or click on the Tools button inside the project.

8.1 Certificate Sign Request (Only for CC323xS/SF Devices)

These commands can be used instead of the CSR from Section 9.

With this tool, the user can read the CSR file from the device or write the final signed certificate. The tool supports both der/pem formats.

Note

When using the Read CSR option, the file that will be downloaded is named as csr.der.

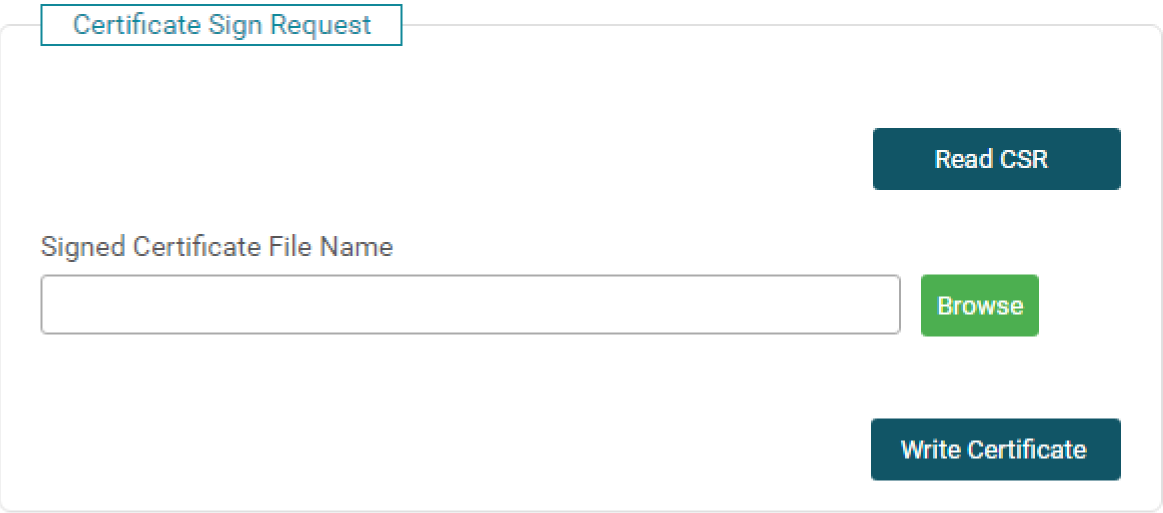


Figure 8-2. Certificate Sign Request

For CSR configuration, see .Section 6.8.1.2.3

8.2 Sign File

Using the tool shown in Figure 8-3, the user can sign a file with a private key and get, as output, a signed file as binary (SHA2) or base-64. For creating a binary SHA1 signature, use the command line (see Section 7.3.3).

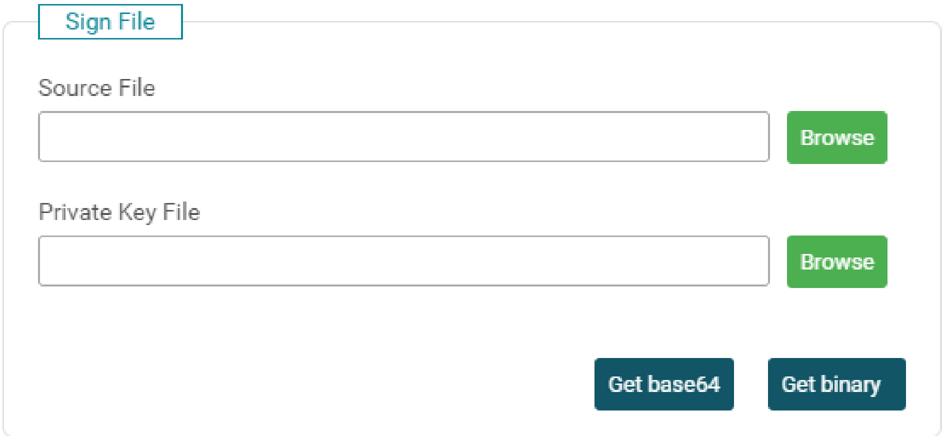


Figure 8-3. Sign File

8.3 Activate Image

Using the tool shown in Figure 8-4, the user can activate a programmed encrypted image.

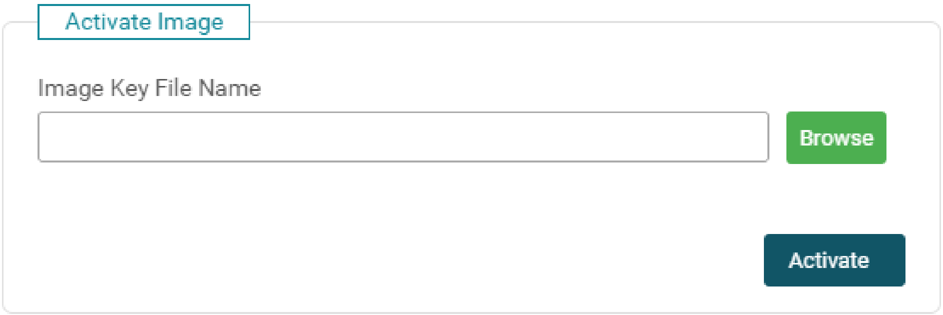


Figure 8-4. Activate Image

9 Using CSR Utility

One of the special security features of the SimpleLink™ Wi-Fi ® is the unique key-pair per device. This feature enables crypto utilities such as sign and verify, without requiring direct access to the private key of the device from the host application.

This unique key pair could also be used for mutual authentication in the TLS handshake. For that ability, it is not enough to have a unique key-pair for the device, but the device must have a certificate signed by an authority or chain of trust that is accepted by the server. To create this certificate, in most cases, access to the public key of the device is not enough. The common way to create and sign a certificate is to use certificate signing request (CSR), which requires a signature of some data with the private key during the creation.

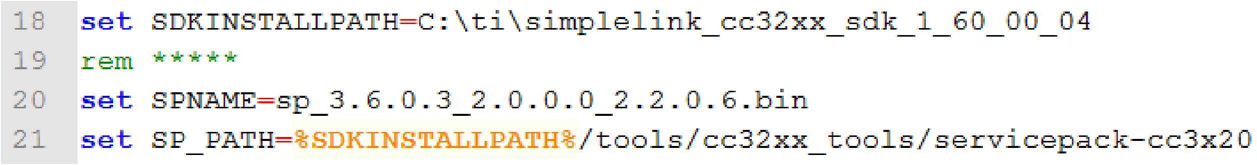
Texas Instruments simplifies this process and provides a tool to get the CSR in PKCS #10 format generated internally by the device. This appendix describes how to get the CSR from the device and how to program the signed certificate.

9.1 Get CSR From Device and Copy it to File

To create a csr file, use the get_csr.bat file. For Linux/mac os versions, use ./get_csr.sh.get_csr.* is a script that creates a project (according to device type), programs it to the device, and executes a CSR utility.

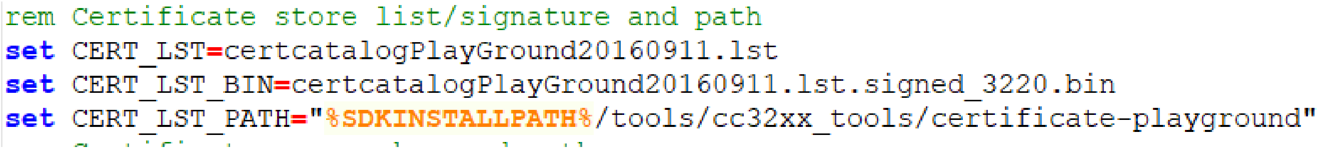
9.1.1 Edit get_csr.bat

1. Set SDK path and service pack name.



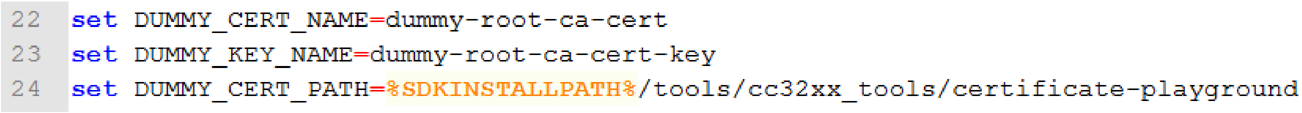
```
set SDKINSTALLPATH=C:\ti\simplelink_cc32xx_sdk_1_60_00_04
rem *****
set SPNAME=sp_3.6.0.3_2.0.0.0_2.2.0.6.bin
set SP_PATH=%SDKINSTALLPATH%/tools/cc32xx_tools/servicepack-cc3x20
```

2. Set certificate list and signature.



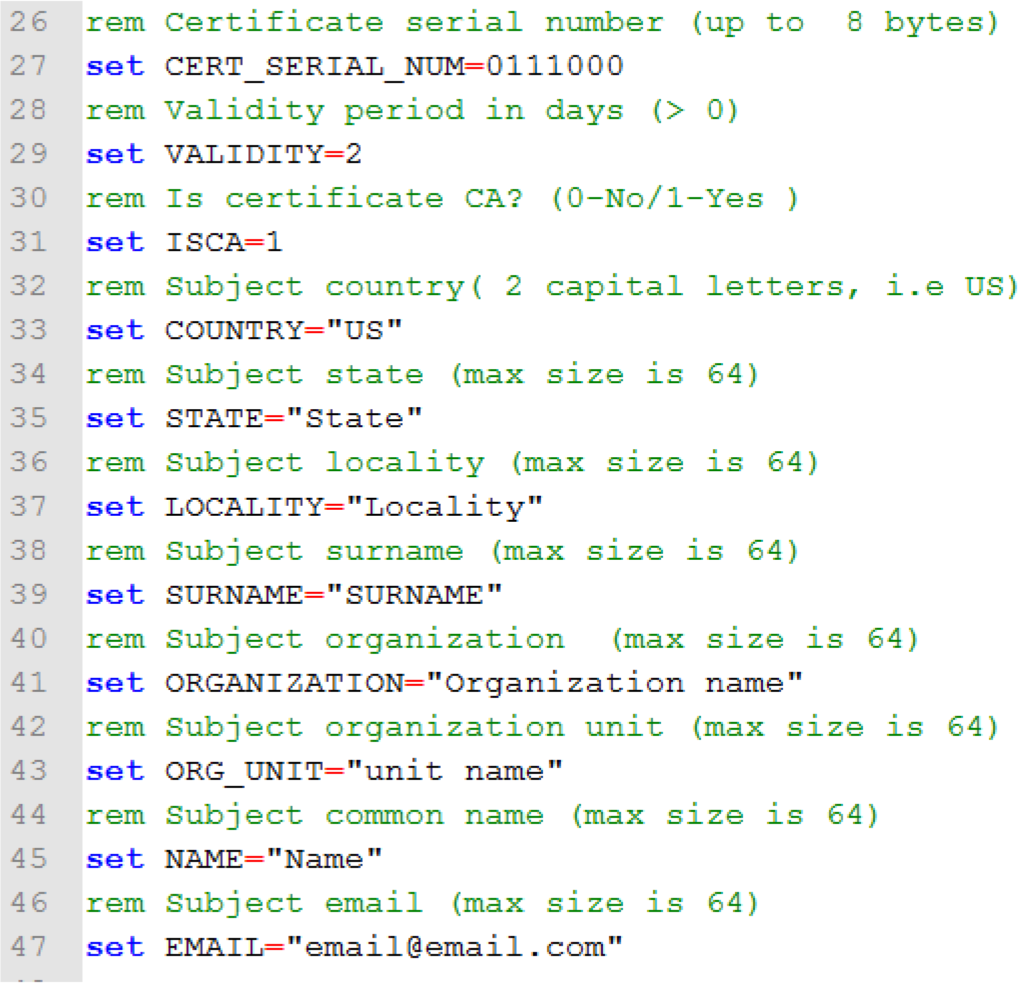
```
rem Certificate store list/signature and path
set CERT_LST=certcatalogPlayGround20160911.lst
set CERT_LST_BIN=certcatalogPlayGround20160911.lst.signed_3220.bin
set CERT_LST_PATH="%SDKINSTALLPATH%/tools/cc32xx_tools/certificate-playground"
```

3. Set certificate and key names and path.



```
set DUMMY_CERT_NAME=dummy-root-ca-cert
set DUMMY_KEY_NAME=dummy-root-ca-cert-key
set DUMMY_CERT_PATH=%SDKINSTALLPATH%/tools/cc32xx_tools/certificate-playground
```

4. Set parameters to csr certificate.

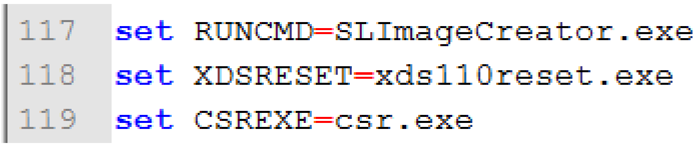


```
26 rem Certificate serial number (up to  8 bytes)
27 set CERT_SERIAL_NUM=0111000
28 rem Validity period in days (> 0)
29 set VALIDITY=2
30 rem Is certificate CA? (0-No/1-Yes )
31 set ISCA=1
32 rem Subject country( 2 capital letters, i.e US)
33 set COUNTRY="US"
34 rem Subject state (max size is 64)
35 set STATE="State"
36 rem Subject locality (max size is 64)
37 set LOCALITY="Locality"
38 rem Subject surname (max size is 64)
39 set SURNAME="SURNAME"
40 rem Subject organization  (max size is 64)
41 set ORGANIZATION="Organization name"
42 rem Subject organization unit (max size is 64)
43 set ORG_UNIT="unit name"
44 rem Subject common name (max size is 64)
45 set NAME="Name"
46 rem Subject email (max size is 64)
47 set EMAIL="email@email.com"
```

5. Verify paths and parameters.

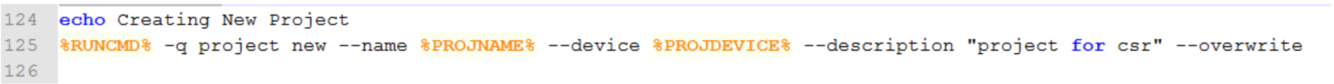
   From line 55 to line 122.

6. Set executables.



```
117 set RUNCMD=SLImageCreator.exe
118 set XDSRESET=xds110reset.exe
119 set CSREXE=csr.exe
```

7. Create a new ImageCreator project.

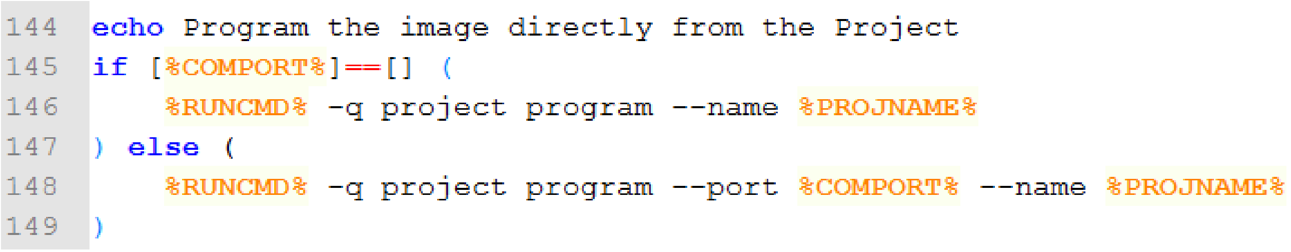


```
124 echo Creating New Project
125 %RUNCMD% -q project new --name %PROJNAME% --device %PROJDEVICE% --description "project for csr" --overwrite
126
```

8. Set ServicePack, certificates, and MCU image.

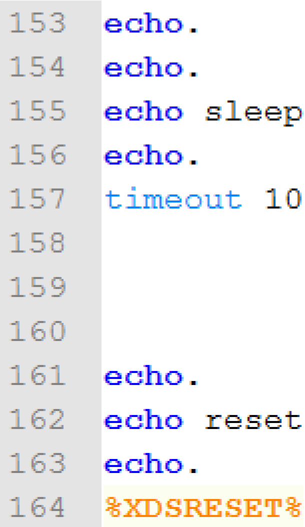
   From line 129 to line 145.

9. Program new project.



```
144 echo Program the image directly from the Project
145 if [%COMPORT%]==[] (
146     %RUNCMD% -q project program --name %PROJNAME%
147 ) else (
148     %RUNCMD% -q project program --port %COMPORT% --name %PROJNAME%
149 )
```

10. Reset device and wait 10 seconds.

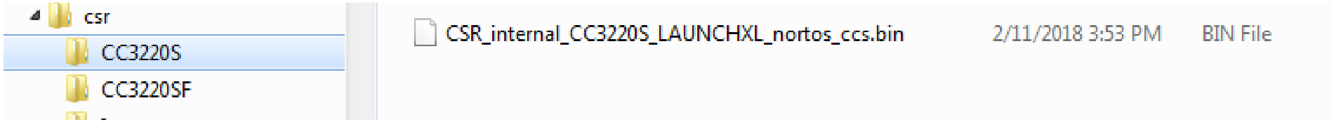


```
153 echo.
154 echo.
155 echo sleep 10
156 echo.
157 timeout 10 >nul
158
159
160
161 echo.
162 echo reset device
163 echo.
164 %XDSRESET%
```

11. Run csr utility.

After programming, the script executes the csr utility (csr.exe/csr, lines 176-186). This utility interacts with the device over RS232, and sends the inputs from the script for creating a csr.pem file at the output folder.

The relevant mcu files exist in the CC3220S/SF and CC3235S/SF folders.

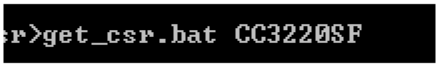


CSR_internal_CC3220S_LAUNCHXL_nortos_ccs.bin 2/11/2018 3:53 PM BIN File

9.1.2 Use get_csr.bat

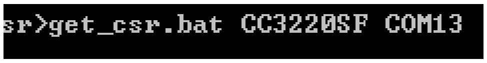
Call to get_csr.bat:

Provide device type (CC3220SF/ CC3220S/ CC3235SF/ CC3235S)



```
r>get_csr.bat CC3220SF
```

To avoid using auto detection for the com port, provide a com port as parameter, so that the call to get_csr.bat is:

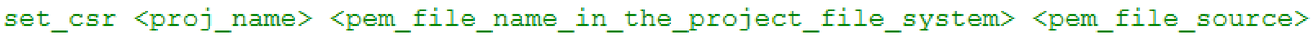


```
sr>get_csr.bat CC3220SF COM13
```

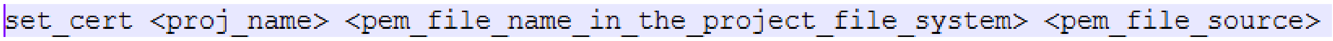
9.2 Replace CSR File in the Project

To replace or add a new csr.pem file, use the set_csr.bat (./set_csr.sh) file. This batch deletes old files from the project (if they exist) and adds a new one.

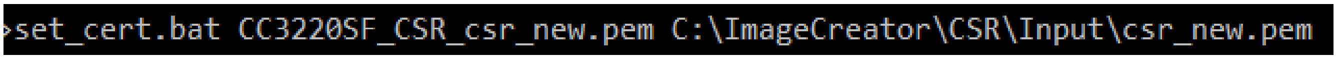
9.2.1 Usage



```
set_csr <proj_name> <pem_file_name_in_the_project_file_system> <pem_file_source>
```

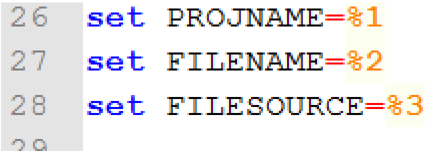


```
set_cert <proj_name> <pem_file_name_in_the_project_file_system> <pem_file_source>
```



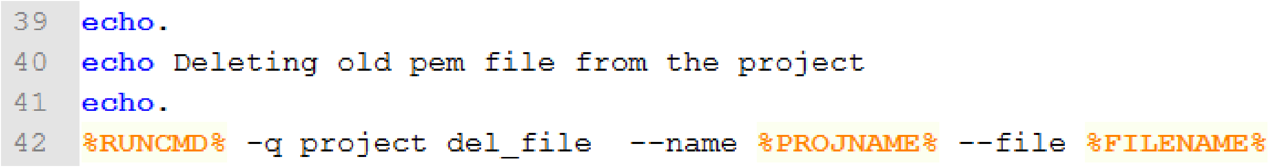
```
>set_cert.bat CC3220SF_CSR_csr_new.pem C:\ImageCreator\CSR\Input\csr_new.pem
```

9.2.2 Script Parameters



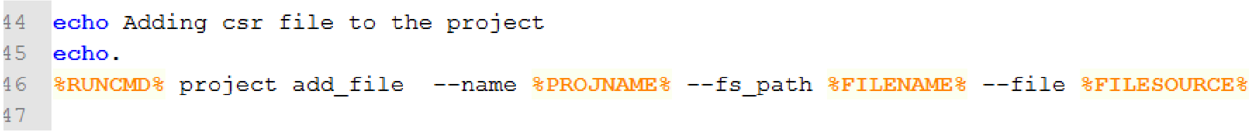
```
set PROJNAME=%1
set FILENAME=%2
set FILESOURCE=%3
```

9.2.3 Delete Old User Files From the Project



```
echo.
echo Deleting old pem file from the project
echo.
%RUNCMD% -q project del_file  --name %PROJNAME% --file %FILENAME%
```

9.2.4 Add New File



```
echo Adding csr file to the project
echo.
%RUNCMD% project add_file  --name %PROJNAME% --fs_path %FILENAME% --file %FILESOURCE%
```

10 Default Power Values for LaunchPad at the Antenna

10.1 Defaults for CC3x35 Device

Table 10-1. 2.4 GHz Default Values

Channel	FCC BO Offset [dB]			ETSI BO Offset [dB]			JP BO Offset [dB]		
	11b	L	H	11b	L	H	11b	L	H
1 [2412 MHz]	0	0	0	0	0	0	0	0	0
2 [2417 MHz]	0	0	0	0	0	0	0	0	0
3 [2422 MHz]	0	0	0	0	0	0	0	0	0
4 [2427 MHz]	0	0	0	0	0	0	0	0	0
5 [2432 MHz]	0	0	0	0	0	0	0	0	0
6 [2437 MHz]	0	0	0	0	0	0	0	0	0
7 [2442 MHz]	0	0	0	0	0	0	0	0	0
8 [2447 MHz]	0	0	0	0	0	0	0	0	0
9 [2452 MHz]	0	0	0	0	0	0	0	0	0
10 [2457 MHz]	0	0	0	0	0	0	0	0	0
11 [2462 MHz]	0	0	0	0	0	0	0	0	0
12 [2467 MHz]	0	0	0	0	0	0	0	0	0
13 [2472 MHz]	0	0	0	0	0	0	0	0	0

Table 10-2. 5 GHz Default Values

Channel	SubBand	FCC [dBm]	ETSI [dBm]	JP [dBm]	Extra Backoff [dB]		Insertion Loss [dB]		
					High	Low	TX	RX	Antenna Gain
36 [5180 MHz]	U-NII1	14.5	0	0	0	0	4.2	4.2	3.2
40 [5200 MHz]		16.125	0	0	0	0			
44 [5220 MHz]		16	0	0	0	0			
48 [5240 MHz]		16.25	0	0	0	0			
52 [5260 MHz]	U-NII-2A	16.125	0	0	0	0	4.2	4.2	3.2
54 [5280 MHz]		16	0	0	0	0			
60 [5300 MHz]		14.5	0	0	0	0			
64 [5320 MHz]		13.625	0	0	0	0			
100 [5500 MHz]	U-NII-2C1	13.5	0	0	0	0	4.2	4.2	3.2
104 [5520 MHz]		17	0	0	0	0			
108 [5540 MHz]		17.125	0	0	0	0			
112 [5560 MHz]		17	0	0	0	0			
116 [5580 MHz]		0	0	0	0	0			
120 [5600 MHz]	U-NII-2C2	0	0	0	0	0	4.2	4.2	3.2
124 [5620 MHz]		0	0	0	0	0			
128 [5640 MHz]		0	0	0	0	0			
132 [5660 MHz]		0	0	0	0	0			
136 [5680 MHz]		0	0	0	0	0			
140 [5700 MHz]		12.375	0	0	0	0			
144 [5720 MHz]		0	0	0	0	0			
149 [5745 MHz]	U-NII-2C3	16.5	0	0	0	0	4.2	4.2	3.2
153 [5765 MHz]		17	0	0	0	0			
157 [5785 MHz]		0	0	0	0	0			
161 [5805 MHz]		0	0	0	0	0			
165 [5825 MHz]		15.125	0	0	0	0			
169 [5845 MHz]		0	0	0	0	0			

10.2 Defaults for CC3x35MOD

WARNING

As configured, this device has been granted US Federal Communications Commission (FCC) equipment authorization, FCC Identifier: Z64-CC3235MOD. Any modifications to the device software or configuration can cause the device performance to vary beyond the scope of the currently referenced FCC authorization. If you modify the device software or configuration, you may be required to seek FCC and other regulatory authorizations before distributing or marketing the devices or products.

Table 10-3. 2.4 GHz Default Values

Channel	FCC BO Offset [dB]			ETSI BO Offset [dB]			JP BO Offset [dB]		
	11b	L	H	11b	L	H	11b	L	H
1 [2412 MHz]	0	0	0	0	0	0	0	0	0
2 [2417 MHz]	0	0	0	0	0	0	0	0	0
3 [2422 MHz]	0	0	0	0	0	0	0	0	0
4 [2427 MHz]	0	0	0	0	0	0	0	0	0
5 [2432 MHz]	0	0	0	0	0	0	0	0	0
6 [2437 MHz]	0	0	0	0	0	0	0	0	0
7 [2442 MHz]	0	0	0	0	0	0	0	0	0
8 [2447 MHz]	0	0	0	0	0	0	0	0	0
9 [2452 MHz]	0	0	0	0	0	0	0	0	0
10 [2457 MHz]	0	0	0	0	0	0	0	0	0
11 [2462 MHz]	0	0	0	0	0	0	0	0	0
12 [2467 MHz]	0	0	0	0	0	0	0	0	0
13 [2472 MHz]	0	0	0	0	0	0	0	0	0

Table 10-4. 5 GHz Default Values

Channel	SubBand	FCC [dBm]	ETSI [dBm]	JP [dBm]	Extra Backoff [dB]		Insertion Loss [dB]		
					High	Low	TX	RX	Antenna Gain
36 [5180 MHz]	U-NII1	14.5	0	0	0	0	3	3	4.5
40 [5200 MHz]		16.125	0	0	0	0			
44 [5220 MHz]		16	0	0	0	0			
48 [5240 MHz]		16.25	0	0	0	0			
52 [5260 MHz]	U-NII-2A	16.125	12.75	12.75	0	0	3	3	4.5
54 [5280 MHz]		16	12.75	12.75	0	0			
60 [5300 MHz]		13	12.75	12.75	0	0			
64 [5320 MHz]		12.125	12.75	12.75	0	0			
100 [5500 MHz]	U-NII-2C1	13.5	12.75	0	0	0	3	3	4.5
104 [5520 MHz]		17	12.75	0	0	0			
108 [5540 MHz]		17.125	12.75	0	0	0			
112 [5560 MHz]		17	12.75	0	0	0			
116 [5580 MHz]		14.25	12.75	0	0	0			
120 [5600 MHz]	U-NII-2C2	0	12.75	0	0	0	3	3	4.5
124 [5620 MHz]		0	12.75	0	0	0			
128 [5640 MHz]		0	12.75	0	0	0			
132 [5660 MHz]		0	12.75	0	0	0			
136 [5680 MHz]		0	12.75	0	0	0			
140 [5700 MHz]		12.375	12.75	0	0	0			
144 [5720 MHz]		0	0	0	0	0			
149 [5745 MHz]	U-NII-2C3	16.5	0	0	0	0	3	3	4.5
153 [5765 MHz]		17	0	0	0	0			
157 [5785 MHz]		0	0	0	0	0			
161 [5805 MHz]		0	0	0	0	0			
165 [5825 MHz]		15.125	0	0	0	0			
169 [5845 MHz]		0	0	0	0	0			

11 Revision History

Changes from Revision G (February 2020) to Revision H (August 2020) Page

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265