



# XCTU

Configuration and Test Utility Software

---

User Guide

## Revision history—90001458-13

---

Revision	Date	Description
P	September 2019	6.4.4.X release.
R	November 2019	6.5.0.X release.
S	May 2020	6.5.1.X release.
T	August 2020	6.5.2.X release.
U	October 2020	6.5.3.x release.

### Trademarks and copyright

Digi, Digi International, and the Digi logo are trademarks or registered trademarks in the United States and other countries worldwide. All other trademarks mentioned in this document are the property of their respective owners.

© 2020 Digi International Inc. All rights reserved.

### Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. Digi provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

### Warranty

To view product warranty information, go to the following website:

[www.digi.com/howtobuy/terms](http://www.digi.com/howtobuy/terms)

### Customer support

**Gather support information:** Before contacting Digi technical support for help, gather the following information:

- Product name and model
- Product serial number (s)
- Firmware version
- Operating system/browser (if applicable)
- Logs (from time of reported issue)
- Trace (if possible)
- Description of issue
- Steps to reproduce

**Contact Digi technical support:** Digi offers multiple technical support plans and service packages. Contact us at +1 952.912.3444 or visit us at [www.digi.com/support](http://www.digi.com/support).

## Feedback

To provide feedback on this document, email your comments to

[techcomm@digi.com](mailto:techcomm@digi.com)

Include the document title and part number (XCTU User Guide, 90001458-13 U) in the subject line of your email.

# Contents

---

## Download and install XCTU

XCTU requirements .....	11
Operating systems .....	11
System requirements .....	11
Supported RF modules .....	11
Install XCTU - Windows .....	12
XCTU updates .....	12
Install XCTU - Linux .....	12
XCTU updates .....	13
Install XCTU - OSX .....	13
XCTU updates .....	13
Optional: Manually install USB drivers .....	13
Install USB drivers for cellular modems .....	14
Drivers for XBee 3 Cellular LTE CAT 1 .....	14
Drivers for XBee 3 Cellular LTE-M/NB-IoT .....	14

## RF concepts and terminology

RF modules .....	16
XBee RF modules .....	16
XTend RF modules .....	16
XLR PRO radio solutions .....	16
Radio firmware .....	16
Radio communication protocols .....	17
Radio module operating modes .....	17
AT operating mode .....	18
API operating mode .....	18
API escaped operating mode .....	19
API frames .....	19
AT settings or commands .....	20
Configuring in AT mode .....	20
Configuring in API mode .....	20
Local radio modules .....	21
Remote radio modules .....	21

## XCTU overview

Menu bar .....	22
Main toolbar .....	23
Devices list .....	23

Working area .....	24
Status bar .....	24
XCTU preferences .....	24
Appearance .....	24
Automatic updates .....	25
Consoles .....	25
Firmware updates .....	25
Network .....	25
Radio firmware library .....	27
XCTU working modes .....	27
Configuration working mode .....	27
Consoles working mode .....	27
Network working mode .....	28

## Add radio modules to XCTU

Add a radio module manually .....	30
Add a programmable radio module .....	33
Discover local radio modules .....	34
Radio module information panel .....	36
Module Icon .....	37
New firmware indicator .....	39
Remove a radio module .....	40
Expand/Collapse radio modules list .....	40
Module information box .....	41
Organize your modules .....	41
Find radio modules .....	41
Sort radio modules .....	41
Clear radio modules .....	41
Discover remote radio modules .....	41
Find radio modules .....	43
Search expressions .....	43

## Configure your modules

Configuration working mode .....	46
Configuration toolbar .....	46
Firmware information panel .....	47
Firmware settings .....	47
Setting status .....	49
Setting types .....	50
Special functions .....	51
Read radio module configuration .....	54
Write module settings .....	54
Load default firmware settings .....	54
Update firmware .....	55
Cellular modem firmware updates .....	56
Remote firmware updates .....	56
Bootloader updates .....	57
Work with configuration profiles .....	57
Create a configuration profile .....	57
Apply a configuration profile .....	65
Load and edit a configuration profile .....	66
Search for a firmware setting .....	67

Configure remote modules securely .....	67
Secure session .....	67
Secure Remote Commands .....	70
Configure a Wi-Fi access point .....	71
Enable and configure Bluetooth .....	73
View firmware release notes .....	74

## Communicate with your modules

Consoles working mode .....	77
Console status .....	77
Consoles toolbar .....	78
Line status indicator .....	78
Console overview .....	79
Connect and disconnect the console .....	79
Record a console session .....	80
Attach and detach the console .....	80
Communicate with modules running in API or API escaped mode .....	81
API console .....	81
Filter sent and received frames .....	85
Add an API frame .....	89
Manage API frames .....	91
Send a single API frame .....	93
Send a sequence of API frames .....	94
Save an API console session .....	95
Communicate with modules running in AT mode .....	96
AT console .....	96
Add a data packet .....	99
Manage data packets .....	101
Send a single packet .....	102
Send a sequence of packets .....	103
Save an AT console session .....	104
Console log files .....	104
Console session records .....	104
Data records .....	105

## View your radio network

Network working mode .....	107
Network toolbar .....	107
Network scan status .....	108
Graph view .....	108
Table view .....	112
Scan the network for available modules .....	114
Search for network nodes .....	114
Change network perspective .....	114
Set network layout .....	115
Filter network connections by quality .....	115
Take a screenshot of the network .....	116
Set zoom level in Graph view .....	117
Export a network table .....	117

## Configure XCTU

Set general preferences .....	119
Set appearance preferences .....	120
Set automatic software update preferences .....	121
Set console preferences .....	123
Set firmware update preferences .....	124
Set network discovery preferences .....	125
Set network appearance preferences .....	127
Set radio firmware library preferences .....	129
Set MicroPython Terminal preferences .....	130

## Update software

Update radio firmware library .....	132
Install legacy radio firmware .....	133
Install XCTU updates .....	134
Open change log .....	135

## Use the XCTU command line

Understanding the XCTU command line interface .....	137
List all commands .....	137
Program arguments .....	137
List ports via command line .....	137
Options .....	138
Load profile via command line .....	138
Options .....	138
Examples .....	139
Update firmware via command line .....	139
Set options .....	140
Examples .....	140

## XCTU tools

Frames generator tool .....	143
XBee API Frame generator dialog .....	144
ASCII/HEX conversion .....	146
Generate an API frame .....	146
Frames interpreter tool .....	148
XBee API Frames interpreter dialog .....	149
Decode a frame .....	151
XBee recovery tool .....	153
Recover a radio module dialog .....	154
Recover a radio module .....	156
Load console session tool .....	157
Console session viewer dialog .....	158
Load a console session .....	161
Range test tool .....	162
Radio range test dialog .....	163
Device Selection .....	164
Range Test Configuration .....	166
Data representation .....	167

Supported products .....	168
Special considerations .....	168
Range test .....	169
Perform a range test .....	170
Firmware explorer tool .....	171
Firmware explorer dialog .....	172
Firmware selection panel .....	173
Firmware toolbar .....	174
Firmware settings panel .....	174
Inspect a firmware version .....	174
Serial console tool .....	175
Serial console dialog .....	176
Open a serial console session .....	178
Configure the serial port settings .....	178
Open a serial console session .....	179
Spectrum analyzer tool .....	180
Spectrum analyzer dialog .....	181
Device selection .....	182
Analysis configuration .....	182
Channels Chart .....	183
Channel summary values .....	184
Number of samples .....	185
Analyze the spectrum of a radio band .....	185
Throughput tool .....	185
Throughput dialog .....	187
Device selection .....	187
Throughput session configuration .....	188
Data representation .....	192
Supported products .....	193
Special considerations .....	193
Measure the transfer ratio between two radio modules .....	194
MicroPython Terminal tool .....	196
Configure the serial port settings - MicroPython Terminal tool .....	198
Open a MicroPython Terminal session .....	199
File system manager tool .....	200
File system manager dialog .....	201
File System Manager sections .....	202
Interact with XBee file system .....	202
Update the file system of remote devices .....	204
Profile Editor tool .....	208
Profile Editor dialog .....	208
Profile Editor toolbar .....	209

## How-to articles and videos

How to update the firmware of your modules .....	212
Step 1: Add the module to XCTU .....	212
Step 2: Update the firmware .....	213
Over-the-air firmware update considerations .....	214
How to visualize your network .....	214
Step 1: Scan the network .....	214
Step 2: Explore the network .....	215
	216



## Troubleshooting for XCTU

Troubleshooting: General .....	218
Troubleshooting: Networking .....	219
Troubleshooting: Firmware update .....	219
Troubleshooting: Add radio module .....	220

## Known issues

## Download and install XCTU

---

This section contains download and install instructions based on operating system. XCTU is compatible with Linux, OSX, and Windows. It may be necessary to configure your system prior to installing XCTU for the first time.

XCTU requirements .....	11
Install XCTU - Windows .....	12
Install XCTU - Linux .....	12
Install XCTU - OSX .....	13
Optional: Manually install USB drivers .....	13
Install USB drivers for cellular modems .....	14

## XCTU requirements

### Operating systems

XCTU is compatible with the following operating systems:

- Windows Vista/7/8/10 (32-bit or 64-bit versions)
- Mac OS X v10.6 and higher versions (64-bit only)
- Linux with KDE or GNOME window managers (32-bit or 64-bit versions)

### System requirements

Property	Minimum	Recommended
HDD space	500 MB	1 GB
RAM memory	2 GB	4 GB
CPU	Dual-core processor	Quad-core processor

### Supported RF modules

XCTU supports configuration and communication for most Digi RF modules. XCTU uses a serial link to interact with these radio modules, providing an easy-to-use and intuitive graphical interface. The following is a complete list of XCTU-compatible RF modules:

- XBee®/XBee-PRO® RF Module Family
  - XBee SX
  - XBee-PRO SX
  - XBee 802.15.4
  - XBee-PRO 802.15.4
  - XBee ZB
  - XBee-PRO ZB
  - Programmable XBee-PRO ZB
  - XBee ZB SMT
  - XBee-PRO ZB SMT
  - Programmable XBee-PRO ZB SMT
  - XBee-PRO 900HP
  - Programmable XBee-PRO 900HP
  - XBee-PRO XSC
  - XBee-PRO 900
  - XBee-PRO DigiMesh 900
  - XBee DigiMesh 2.4
  - XBee-PRO DigiMesh 2.4
  - XBee-PRO 868

- XBee Wi-Fi
- XBee 865LP
- Programmable XBee 865LP
- XBee Cellular
- XBee 868LP SX
- XBee Thread
- XBee3 (Zigbee, DigiMesh 2.4, 802.15.4 and Cellular)
- XTend® RF Module family
- XLR PRO radio solution
- XLR Module

## Install XCTU - Windows

Follow the steps below to download and install XCTU on your computer.

1. Visit [www.digi.com/xctu](http://www.digi.com/xctu).
2. Click **Download XCTU**.
3. Under **Utilities**, click the Windows installer link.
4. When the file has finished downloading, run the executable file and follow the steps in the XCTU Setup Wizard.

A “What’s new” dialog appears when XCTU opens the first time after the installation.

## XCTU updates

You may be notified about XCTU software updates once XCTU has loaded. You should always update XCTU to the latest available version. See [Install XCTU updates](#).

## Install XCTU - Linux

By default, access to the serial and USB ports in Linux is restricted to root and dialout group users. To access your XBee devices and use XCTU to communicate with them, your Linux user must belong to this group.

To add your Linux user to the dialout group:

1. Open a terminal console.
2. Execute this command:

---

```
sudo usermod -a -G dialout <user>
```

---

where <user> is the user you want to add to the dialout group.

3. Log out and log in again with your user in the system.

Then download and install XCTU:

4. Visit [www.digi.com/xctu](http://www.digi.com/xctu).
5. Click **Download XCTU**.
6. Under **Utilities**, click the Linux installer link.
7. When the file has finished downloading, run the executable file and follow the steps in the XCTU Setup Wizard.

A “What’s new” dialog appears when XCTU opens the first time after the installation.

## XCTU updates

You may be notified about XCTU software updates once XCTU has loaded. You should always update XCTU to the latest available version. See [Install XCTU updates](#).

## Install XCTU - OSX

OSX version 10.8 (Mountain Lion) and greater only allows you to install applications downloaded from the Apple Store. To install XCTU, you must temporarily disable this setting.

Follow these steps to enable installation of "unsigned" software:

1. Click the **Apple** icon in the top-left corner of your screen and choose **System Preferences**.
2. Click the **Security & Privacy** icon.
3. To edit security settings, click the **padlock** icon in the bottom left of the window.
4. Enter your Mac credentials and click **Unlock**. The **Allow applications downloaded from** dialog appears.
5. Click the **Anywhere** radio button and, in the confirmation window, click **Allow From Anywhere**.

---

**Note** We recommend you set this option back to **Mac App Store** or **Mac App Store and identified developers** once you have finished installing XCTU.

---

Then download and install XCTU:

6. Visit [www.digi.com/xctu](http://www.digi.com/xctu).
7. Click **Download XCTU**.
8. Under **Utilities**, click the OSX installer link.
9. When the file has finished downloading, unzip and run the executable file and follow the steps in the XCTU Setup Wizard.

A “What’s new” dialog appears when XCTU opens the first time after the installation.

## XCTU updates

You may be notified about XCTU software updates once XCTU has loaded. You should always update XCTU to the latest available version. See [Install XCTU updates](#).

## Optional: Manually install USB drivers

When you connect the XBee board to your computer for the first time, USB drivers are installed automatically. There are times when this does not occur, and you need to install device drivers

manually:

1. Find the appropriate USB drivers on the [Digi support site](#).
2. Choose your operating system.
3. Download and run the file.
4. Follow the steps in the installation wizard.

## Install USB drivers for cellular modems

XCTU requires additional drivers to update the modems of newer XBee 3 Cellular devices. If you have not installed them, use the following instructions to do so depending on the devices you want to program.

---

**Note** This step is only required if you are programming the modem of [XBee 3 Cellular LTE CAT 1](#) or [XBee 3 Cellular LTE-M/NB-IoT](#) devices.

---

### Drivers for XBee 3 Cellular LTE CAT 1

1. Go to the [Telit drivers page](#).
2. Select the **Telit Windows Desktop Drivers Installer**.
3. Run the executable file.
4. Follow the steps in the installation wizard.

### Drivers for XBee 3 Cellular LTE-M/NB-IoT

1. Download the [u-blox drivers](#).
2. Uncompress the file and run the executable.
3. Follow the steps in the installation wizard.
4. When prompted select the **ETHERNET-DHCP** option.
5. After driver installation completes, reboot your computer.



**CAUTION!** If you are using Windows 7 or Vista, we highly recommend that you disable the drivers installation from Windows Update in order to speed up the modem update process. For more information on how to do this, go to <https://support.microsoft.com/en-us/help/2500967/how-to-stop-windows-7-automatically-installing-drivers>.

---

## RF concepts and terminology

---

This section contains concepts regarding radio frequency modules and the XCTU application itself. Understanding these concepts will help you work most effectively with XCTU.

RF modules .....	16
Radio firmware .....	16
Radio communication protocols .....	17
Radio module operating modes .....	17
API frames .....	19
AT settings or commands .....	20
Local radio modules .....	21
Remote radio modules .....	21

## RF modules

A radio frequency (RF) module is a small electronic circuit used to transmit and receive radio signals on different frequencies. Digi produces a wide variety of RF modules to meet the requirements of almost any wireless solution, such as long-range, low-cost, and low-power modules. The most popular wireless products are the XBee RF modules.

XCTU is compatible with Digi's XBee and XTend RF modules and XLR PRO. For a complete list of XCTU-compatible modules, see [XCTU requirements](#).

### XBee RF modules



XBee is the brand name of a family of RF modules produced by Digi. They are modular products that make deploying wireless technology easy and cost-effective. Digi has made multiple protocols and RF features available in the popular XBee footprint, giving customers enormous flexibility to choose the best technology for their needs.

XBee RF modules are available in two form-factors, Through-Hole and Surface Mount, with different antenna options. One of the most popular features of these modules is that almost all of them are available in the Through-Hole form factor and share the same footprint.

### XTend RF modules



XTend family devices are long-range RF modules produced by Digi that provide unprecedented range in a low-cost wireless data solution. They were engineered to provide customers with an easy-to-use RF solution that provides reliable delivery of critical data between remote devices. These modules transfer standard asynchronous serial data streams, operate within the ISM 900 MHz frequency band, and sustain up to 115.2 Kbps data throughput.

### XLR PRO radio solutions



The XLR PRO is an ultra long-range, rugged 900MHz radio solution designed for optimal performance even in the most challenging RF environments. Leveraging Digi's patent-pending Chirp Spread Spectrum technology, the XLR PRO provides industry-leading receive sensitivity and interference immunity, making it ideal for deployments in noisy RF environments like oil fields. The XLR PRO includes 2 Ethernet ports and 1 serial port, enabling wireless data communications between Ethernet and/or serial devices up to distances of over 90 miles.

## Radio firmware

Radio firmware is program code stored in a radio module's persistent memory that provides the control program for the device. Digi periodically releases new radio firmware versions to fix bugs or improve functionality. You may need to add these firmware files to XCTU's radio firmware library. You can use XCTU to update or change the firmware of a module if, for example, you want to change the role of a device or you want to use the latest firmware version.

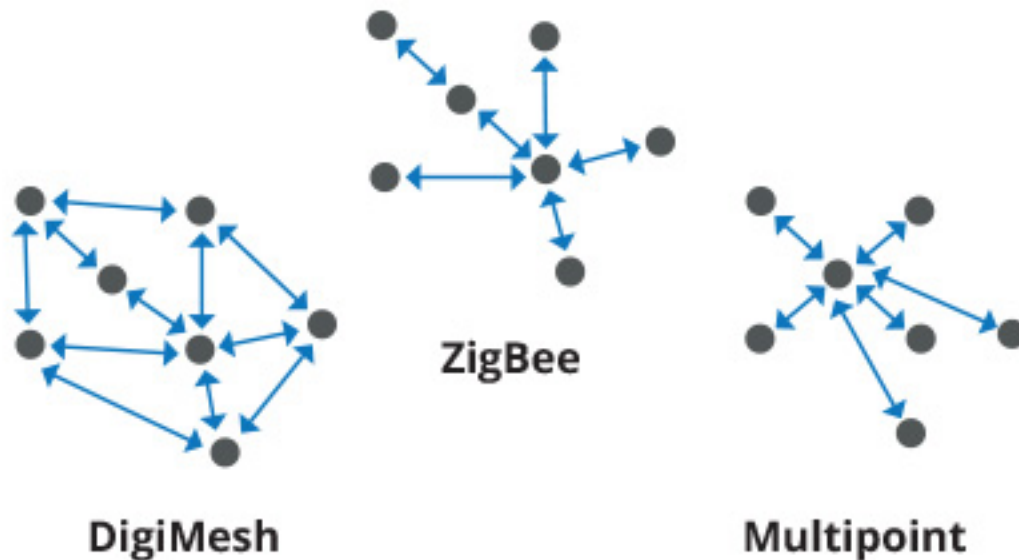


## Radio communication protocols

A radio communication protocol is a set of rules for data exchange between radio devices. An RF module supports a specific radio communication protocol depending on the module and its radio firmware.

The following is the complete list of protocols supported by the XBee radio modules:

- IEEE 802.15.4
- ZigBee
- ZigBee Smart Energy
- DigiMesh (Digi proprietary)
- ZNet
- IEEE 802.11 (Wi-Fi)
- Point-to-multipoint (Digi proprietary)
- XSC (XStream-compatible)



Not all XBee devices can run all listed communication protocols. The combination of XBee hardware and radio firmware determines the protocol that an XBee device can execute. For more information about the available XBee RF modules and the protocols they support, see [XBee RF Family Comparison Matrix](#).

## Radio module operating modes

The operating mode of an XBee radio module establishes the way a user or any microcontroller attached to the XBee communicates with the module through the Universal Asynchronous Receiver/Transmitter (UART) or serial interface.

Depending on the firmware and its configuration, radio modules can work in three different operating modes:

- Application Transparent (AT) operating mode
- API operating mode
- API escaped operating mode

In some cases, the operating mode of a radio module is established by the firmware version, which determines whether the operating mode is AT or API, and the AP setting of the firmware, which determines if the API mode is escaped (2) or not (1). In other cases, the operating mode is only determined by the AP setting, which allows you to configure the mode to be AT (AP=0), API (AP=1), or API escaped (AP=2).

## AT operating mode

In AT (Application Transparent) or transparent operating mode, all serial data received by the radio module is queued up for RF transmission. When RF data is received by the module, the data is sent out through the serial interface.

To configure an XBee module operating in AT, you must put it in command mode to send the configuration commands.

### AT Command mode

When the radio module is working in AT operating mode, you must use the command mode interface to configure settings.

To enter AT command mode, send the three-character command sequence (usually "+++") within one second. Once AT command mode has been instigated, the module sends an "OK\r", the command mode timer is started, and the radio module is able to receive AT commands.

### AT command structure

The structure of an AT command is:

---

```
AT[ASCII command][Space (optional)][Parameter (optional)][Carriage return]
```

---

For example:

---

```
ATNI MyDevice\r
```

---

If no valid AT commands are received within the command mode timeout, the radio module automatically exits AT command mode. You can also exit command mode by issuing the CN command:

---

```
(ATCN\r)
```

---

## API operating mode

API (Application Programming Interface) operating mode is an alternative to AT mode. API operating mode requires that communication with the module be done through a structured interface. In other words, data is communicated via API frames.

The API specifies how commands, command responses, and module status messages are sent and received from the module using the serial interface. With API operating mode, you can:

- Configure the XBee module itself.
- Configure remote modules in the network.
- Manage data transmission to multiple destinations.

- Receive success/failure status of each transmitted RF packet.
- Identify the source address of each received packet.

Depending on the AP parameter value, the radio module can operate in one of two modes: API (AP=1) or API escaped (AP=2) operating mode.

### API escaped operating mode

API escaped operating mode (AP=2) is similar to API mode except that when working in API escaped mode, some bytes of the API frame specific data must be escaped. Since XCTU is compatible with both API and API escaped operating modes, you do not need to manually escape characters.

API escaped operating mode increases the reliability of RF transmission by preventing conflicts with special characters such as the start-of-frame byte (0x7E). API non-escaped (API=1) operation relies solely on the start delimiter and length bytes to differentiate API frames. In API escaped mode, on the other hand, those special bytes are escaped. Since 0x7E can only appear at the start of an API packet, a module can always "assume" that a new packet has started if 0x7E is received at any time while in API escaped mode.

### Escape characters

When sending or receiving an API frame in API escaped mode, specific data values must be escaped (flagged) so they do not interfere with the data frame sequence.

To escape a data byte, insert 0x7D and follow it with the byte to be escaped XOR'd with 0x20. The data bytes that need to be escaped are as follows:

- 0x7E: Frame delimiter
- 0x7D: Escape
- 0x11: XON
- 0x13: XOFF

---

**Note** XCTU automatically escapes the appropriate characters when interacting with API escaped radio modules.

---

## API frames

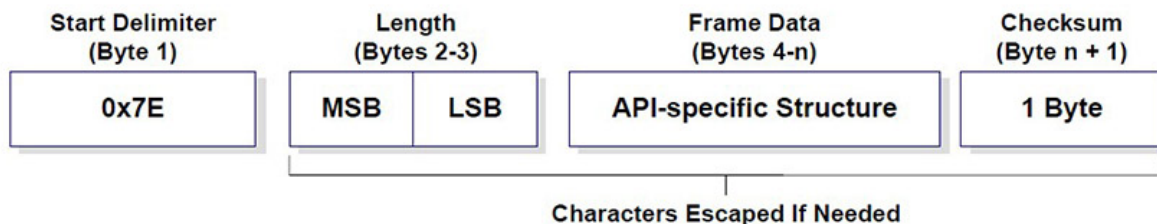
An API frame is the structured data sent and received through the serial interface of the radio module when it is configured in API or API escaped operating modes. API frames are used to communicate with the module or with other modules in the network. An API frame has the following structure:



Field	Description
Start delimiter	The first byte of a frame consisting of a special sequence of bits which indicate the beginning of a data frame. Its value is always 0x7E. This allows for easy detection of a

Field	Description
	new incoming frame.
Length	Specifies the total number of bytes included in the frame data field. Its two-byte value excludes the start delimiter, the length, and the checksum.
Frame data	Composed by the API identifier and the API identifier-specific data. The content of the specific data depends on the API identifier (also called API frame type).
Checksum	The last byte of the frame. It helps test data integrity and is calculated by taking the hash sum of all the API frame bytes that came before it, excluding the first three bytes (start delimiter and length).

If your module is operating in API escaped operating mode, some bytes in the Length, Frame data, and Checksum frame fields may need to be escaped. XCTU automatically performs this step and escapes the appropriate characters. See [API escaped operating mode](#).



**Note** There are many different types of API frames. You can use the Frames generator tool to learn the specific data contained within a determined API frame as well as to build and fill any type of API frame. See [Frames generator tool](#).

## AT settings or commands

The firmware running in RF modules contains a set of settings and commands that can be configured to change the behavior of the module or to perform any action related to it. Depending on the protocol, the number of settings and their meanings varies, but all XBee RF modules can be configured with AT commands.

All firmware settings or commands are identified with two ASCII characters. Applications and documents refer to them as either AT settings or AT commands.

The configuration process of these AT settings varies depending on the operating mode of the RF module.

### Configuring in AT mode

In AT operating mode, you must put the module in a special mode called command mode so it can receive AT commands. For more information about configuring RF modules working in AT operating mode, see [AT operating mode](#).

### Configuring in API mode

To configure or execute AT commands when the RF module is in API operating mode, you must generate an AT command API frame containing the AT setting identifier and the value of that setting,

and send it to the RF module. See [API frames](#).

## Local radio modules

A local radio module is any module added to the device list using the Add a radio module or Discover radio modules buttons.

XCTU communicates directly with local modules, and they are physically attached to the PC through a serial or USB port. A local radio module can discover remote modules in the same network if their protocol is ZigBee or DigiMesh. A local module is configurable if Configuration working mode is active, and you can communicate with a local module through its console when Consoles working mode is active.

## Remote radio modules

You can locate remote radio modules in the same network as a local module. A remote module is not physically attached to your computer. Remote modules are displayed in a sub-list under the local module, and that local module functions as an interpreter; without it, XCTU is unable to communicate with the remote module. See [Discover remote radio modules](#).

Communication between XCTU and a remote module takes place in two stages: serial communication from XCTU to the local module, and wireless communication between the local module and the remote module. XCTU uses the serial port to send a message intended for the remote module, along with delivery specifics, to the local module. The local module then transmits the message wirelessly to the remote module.

If the local device containing remote modules is configured in AT (transparent) operating mode, you cannot configure its remote radio modules due to a protocol limitation. If the local radio module is configured in API operating mode, you can configure its remote radio modules like any local module.

Since a remote radio module is not physically connected to the PC, it does not have a communication console in Consoles working mode. For the same reason, you also cannot obtain a remote radio module's network topology in Network working mode. See [Consoles working mode](#) and [Network working mode](#).

## XCTU overview

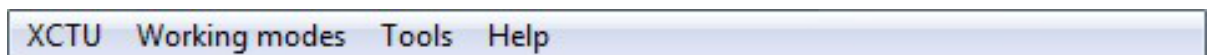
---

XCTU is divided into five main sections: the menu bar, main toolbar, devices list, working area, and status bar.



### Menu bar

The menu bar is located at the top of the application. You can use the menu bar to access all XCTU features, tools, and working modes.



## Main toolbar

The main toolbar is located at the top of the application and is divided into three sections.



- The first section contains two icons used to add radio modules to the radio modules list. See [Add radio modules to XCTU](#).



- The second section contains the static XCTU functionality that does not require a radio module. This section includes the XCTU tools, the XCTU configuration, the feedback form, and the help and updates functions. See [XCTU tools](#) and [Configure XCTU](#).



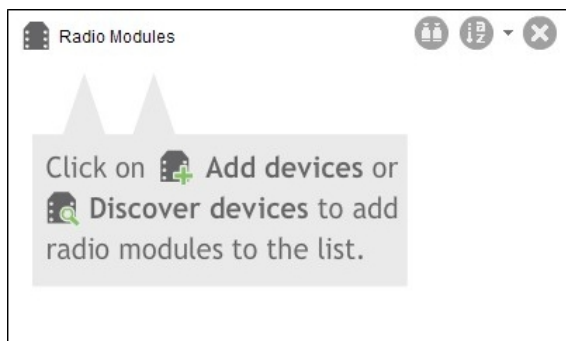
- The third section contains tabs corresponding to the three XCTU working modes. To use this functionality, you must have added one or more radio modules to the list. See [XCTU working modes](#).



## Devices list

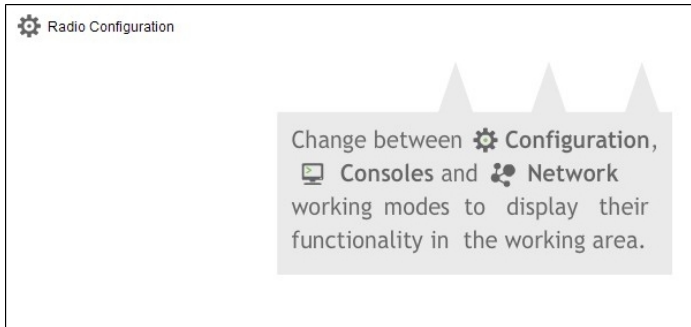
The radio modules list, or devices list, is located on the left side of the tool and displays the radio modules that are connected to your computer. If you know the serial port configuration of a radio module, you can add it to the list directly. You can also use the discovery feature of XCTU to find radio modules connected to your PC and add them to the list. See [Add radio modules to XCTU](#).

Depending on the protocol of the local radio modules added, you can also add remote radio modules to the list using the module's search feature.



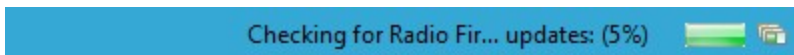
## Working area

The working area is the largest section and is located at the right side of the application. The contents of the working area depend on the working mode selected in the toolbar. To interact with the controls displayed in the working area, you must have added one or more radio modules to the list and one of the modules must be selected.



## Status bar

The status bar is located at the bottom of the application and displays the status of specific tasks, such as the firmware download process.



## XCTU preferences

To configure XCTU settings, click the **Preferences** button  on the XCTU toolbar.

Configuration preferences are grouped into categories listed on the left-hand side of the preferences dialog box. You can configure settings in the following XCTU categories:

### Appearance

You can configure some graphic aspects of the tool and how some elements are displayed.

Field	Description
Font size	Change all the XCTU texts size in percentage, from 50 to 120%.
Show top bar menu	Displays an application top bar menu with texts.
Show text on toolbar actions	Displays the name of the action below each toolbar element for a better understanding of the meaning of each action.
Use reduced toolbars size	Changes the size of the application toolbars reducing them.



## Automatic updates

Field	Description
Automatically find new updates and notify me	Enables or disables automatic XCTU updates. Uncheck if you do not want XCTU to update automatically.
Update schedule	Sets a schedule to search for updates or to update when XCTU is started. If you select <b>Look for updates on the following schedule</b> , you must also specify the search interval and hour.
Download options	Establishes when new updates should download and sets permissions for whether updates are automatically downloaded.
When updates are found	Sets the frequency of update notification.

## Consoles

Field	Description
API console	Configures the maximum number of API frames that can be displayed in the frames log during a session. When the maximum limit is reached, the session starts overwriting frames.
AT console	Configures the maximum number of bytes that can be displayed during a session. When the maximum limit is reached, the session starts overwriting bytes.

---

MadCap:autonum="<span style="color: #84c361;" class="mcFormatColor"><b>Note </b></span>">If you set high values for maximum API frames and/or AT bytes, you may notice performance issues in the consoles.

---

## Firmware updates

Field	Description
Remote firmware update timeout	Configures the remote firmware update timeout in milliseconds. This value is the maximum time the application will wait for answers sent by the remote node during remote firmware update before concluding that there was an error during the process.

## Network

You can configure Network view in the **Network preferences** dialog. The first four options are common to all networks:

Field	Description
Always clear the Network view before starting	Clears Network view before each new network scan.
Remove nodes if they were not discovered in the last performed scan	Removes any nodes not discovered in the last scan.
Stop after scan	Sets the number of scans to perform before stopping the discovery process. A value of '0' means the process will not stop automatically.
Time between scans	Sets the duration of time XCTU waits before starting a new network scan. The value must be between 0 and 300 seconds.

The remainder of the options are specific to 802.15.4, DigiMesh, and ZigBee network types:

Field	Description
Discovery mode	<p>Sets the method used by the network discovery process.</p> <ul style="list-style-type: none"> <li>▪ Flood: The neighbor discovery process is performed for every node at the moment it is found. Several discovery processes may be running at the same time. This method may be faster, but it may also generate a lot of traffic and saturate the network.</li> <li>▪ Cascade: The neighbor discovery process is performed for every node as soon as the discovery process finishes. Only one discovery process runs at a time. This method may be slower, but it is likely to generate less traffic.</li> </ul>
Neighbor discovery timeout	<p>Sets the maximum duration, in seconds, the discovery process should spend finding neighbors of a module. Value must be between 5 and 1800 seconds (30 minutes).</p> <p>This timeout is highly dependent on the nature of the network. For DigiMesh, the value should be greater than the highest NT (Node Discover Timeout) and include enough time to let the message propagate, depending on the sleep cycle of your devices.</p>
Time between requests	<p>Sets the wait time between node neighbor requests. The value must be between 0 and 300 seconds (5 minutes).</p> <p>For the Cascade method, this is the number of seconds to wait after completion of the neighbor discovery process of the previous node.</p> <p>For the Flood method, this is the minimum time to wait between each radio module's neighbor requests.</p>

---

**Note** The Cascade discovery method is recommended for large networks.

---

### **Network appearance**

You can configure how node links are represented in Network graphic view.

Field	Description
Connection default color	Defines the default color of the node's connection lines.
Show colored connections based on their quality	Enables or disables the coloring of node connection lines based on their link quality.
DigiMesh / ZigBee network	<p>Enables you to modify the maximum and minimum values and RGB colors for each quality range. Click in the cell, type the value, and click <b>Enter</b> to change a value.</p> <p>Ranges include minimum values but not maximum values. When you change the minimum value of a quality range, the maximum value of the next range adopts a corresponding value.</p>

### Radio firmware library

You can instruct XCTU to look for new radio firmware when it starts up by checking **Automatically update the XBee Firmware Library each time XCTU is started**. If this option is disabled, you can only check for firmware updates manually.

## XCTU working modes

XCTU operations are grouped into three working modes—Configuration, Consoles, and Network. The selected working mode determines which specific operations you can perform with a radio module or modules in your device list. You can only select one working mode at a time. By default, XCTU launches in Configuration mode.

### Configuration working mode



Use configuration working mode to configure a radio module selected from your device list. See [Configure your modules](#).

### Consoles working mode



Use consoles working mode to interact or communicate with the selected radio module. See [Communicate with your modules](#).

## Network working mode



Use network working mode to discover and visualize the topology and interconnections of your network. See [View your radio network](#).

## Add radio modules to XCTU

---

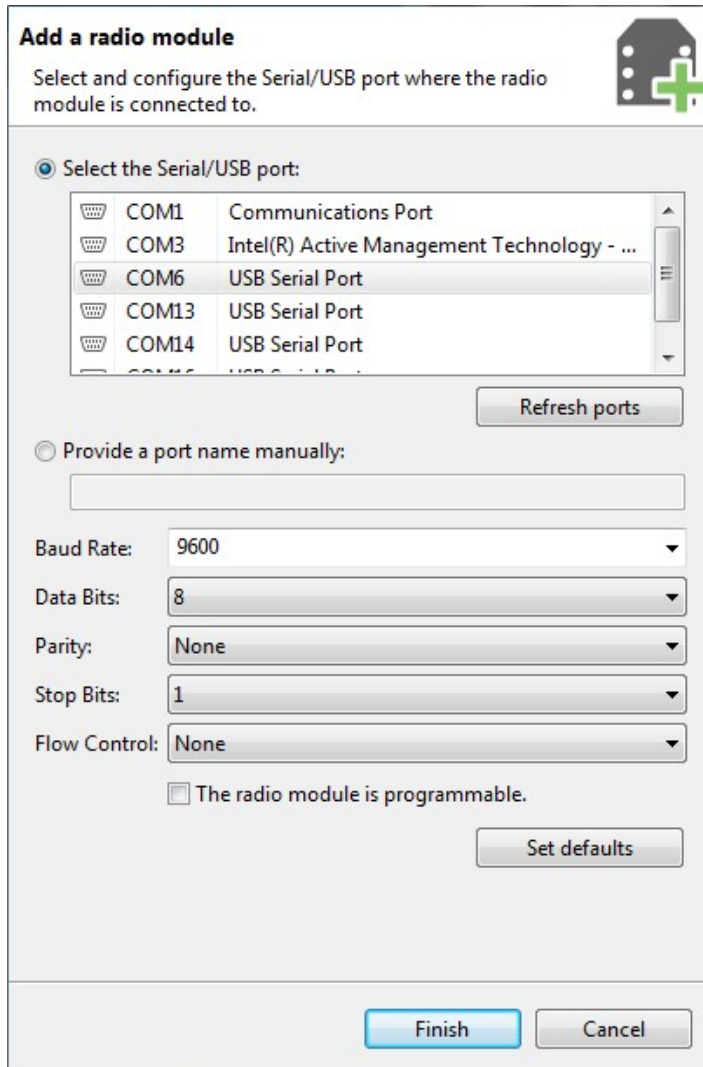
This section describes how to add, discover and organize your radio modules in XCTU.

Add a radio module manually .....	30
Add a programmable radio module .....	33
Discover local radio modules .....	34
Radio module information panel .....	36
Organize your modules .....	41
Discover remote radio modules .....	41
Find radio modules .....	43

## Add a radio module manually

If you know the serial configuration of your radio module, you can add it to the list manually.

1. Click the **Add a radio module** button  from the toolbar. The **Add a radio module** dialog opens.



**Add a radio module**

Select and configure the Serial/USB port where the radio module is connected to.

Select the Serial/USB port:

COM1	Communications Port
COM3	Intel(R) Active Management Technology - ...
COM6	USB Serial Port
COM13	USB Serial Port
COM14	USB Serial Port

Provide a port name manually:

Baud Rate: 9600

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: None

The radio module is programmable.

Refresh ports

Set defaults

Finish Cancel

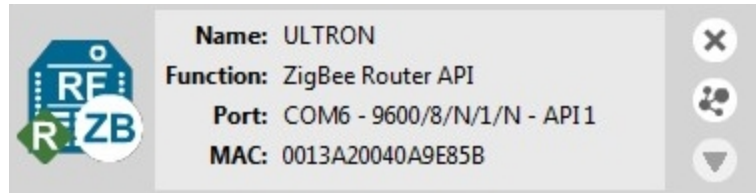
2. Select the serial port where the radio module is connected (or enter its name manually) and configure the serial settings of the port.

---

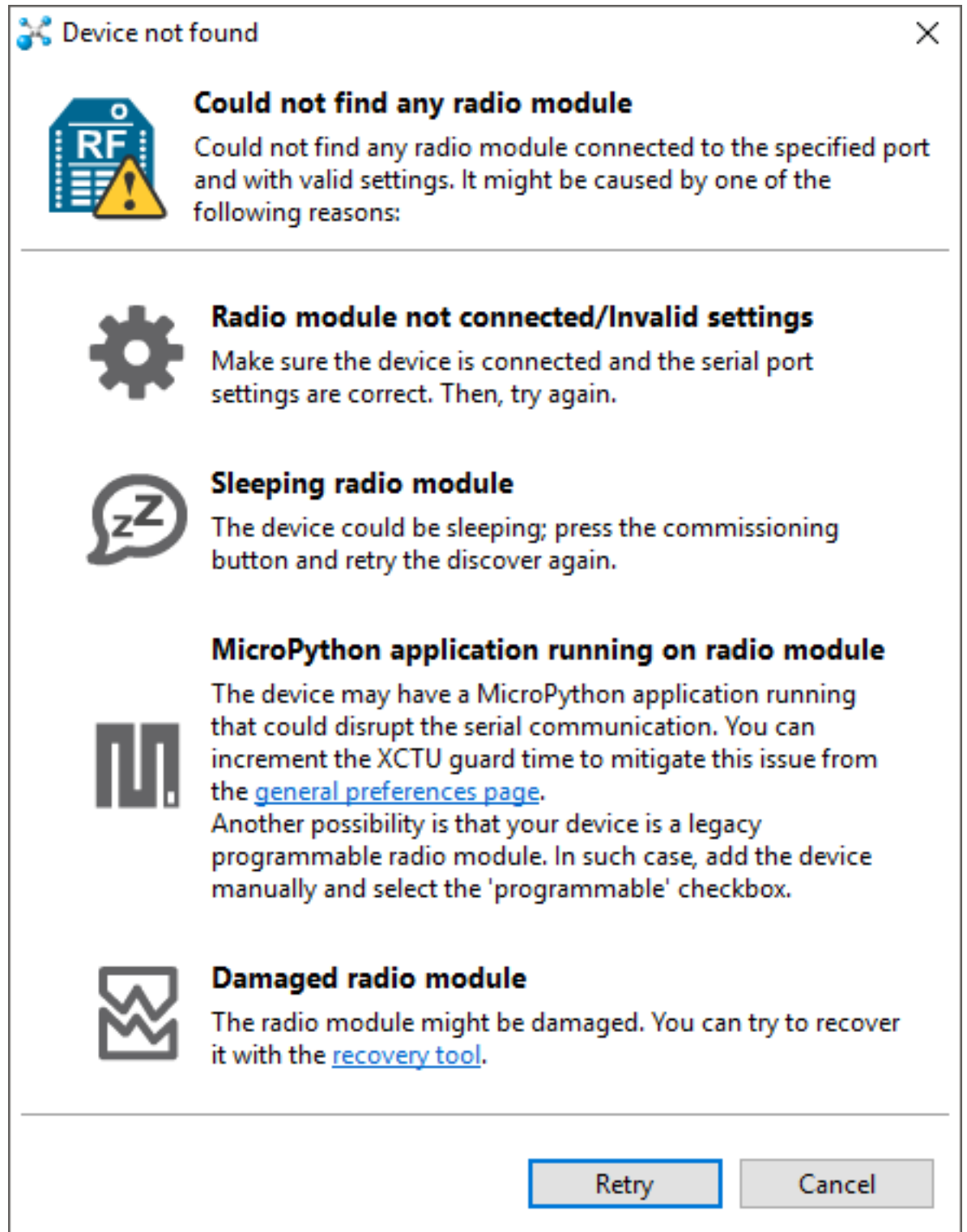
**Note** Custom baud rates can only be typed under Windows OS.

---

3. Click **Finish** to add the radio module to the list of radio modules.
4. If the settings were configured correctly and the radio module was connected to the selected port, the module is displayed in the device list. For more information about the device list, see [Radio module information panel](#).



5. If the settings were configured incorrectly, an **Action Required** dialog asks you to reset the module. Reset the module. The action required dialog should close and your module should be added to the list.
6. If your module could not be found, XCTU displays the **Could not find any radio module** dialog providing possible reasons why the module could not be added. To resolve the issue, see [Troubleshooting for XCTU](#).



**Note** You can also use the Add a radio module dialog to add programmable radio modules. See [Add a programmable radio module](#).



For more information, see [Local radio modules](#) and [Radio module information panel](#).

## Add a programmable radio module

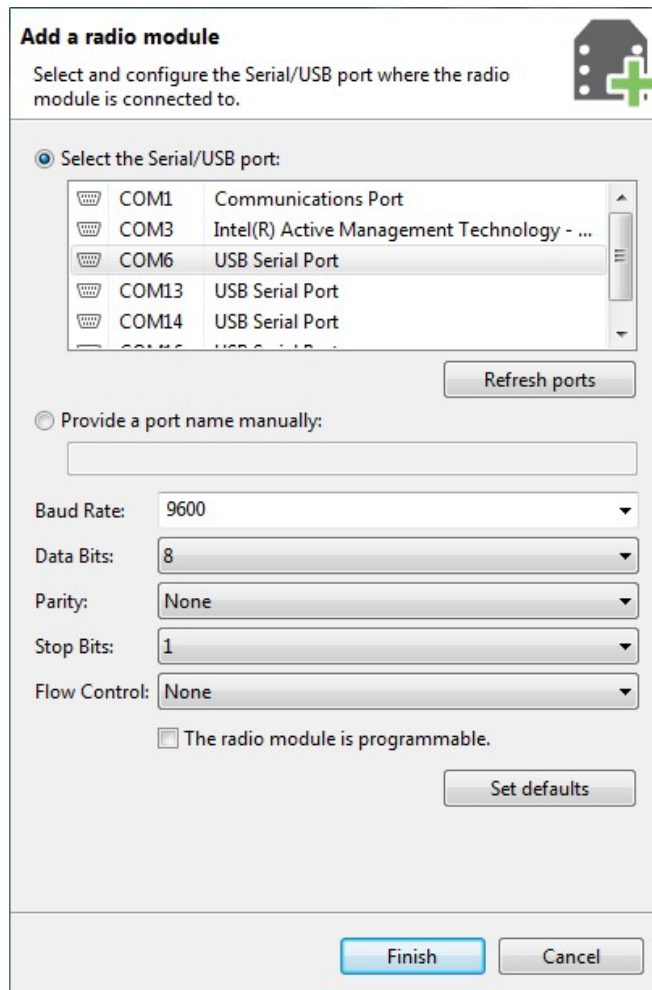
Some radio module variants are programmable, which means they are able to run applications written in C. Normally, they are known as Programmable XBee modules and can be identified by a part number ending in B on the back label.



XBee-PRO modules are often confused with Programmable XBee modules. The -PRO suffix does not mean that the module is programmable.

To add a programmable radio module:

1. Click the **Add a radio module** button  from the toolbar. The **Add a radio module** dialog opens.



**Add a radio module**

Select and configure the Serial/USB port where the radio module is connected to.

Select the Serial/USB port:

COM1	Communications Port
COM3	Intel(R) Active Management Technology - ...
COM6	USB Serial Port
COM13	USB Serial Port
COM14	USB Serial Port

Provide a port name manually:

Baud Rate: 9600

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: None

The radio module is programmable.

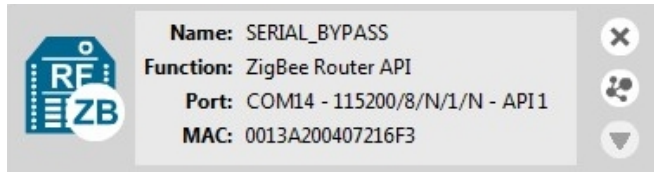
Refresh ports

Set defaults

Finish Cancel

2. Select the serial port to which the radio module is connected (or enter its name manually) and configure the serial settings of the port.
3. Check the **My radio module is programmable** setting.


4. Click **Finish**.
5. Reset your radio module when prompted. The module appears in the device list.

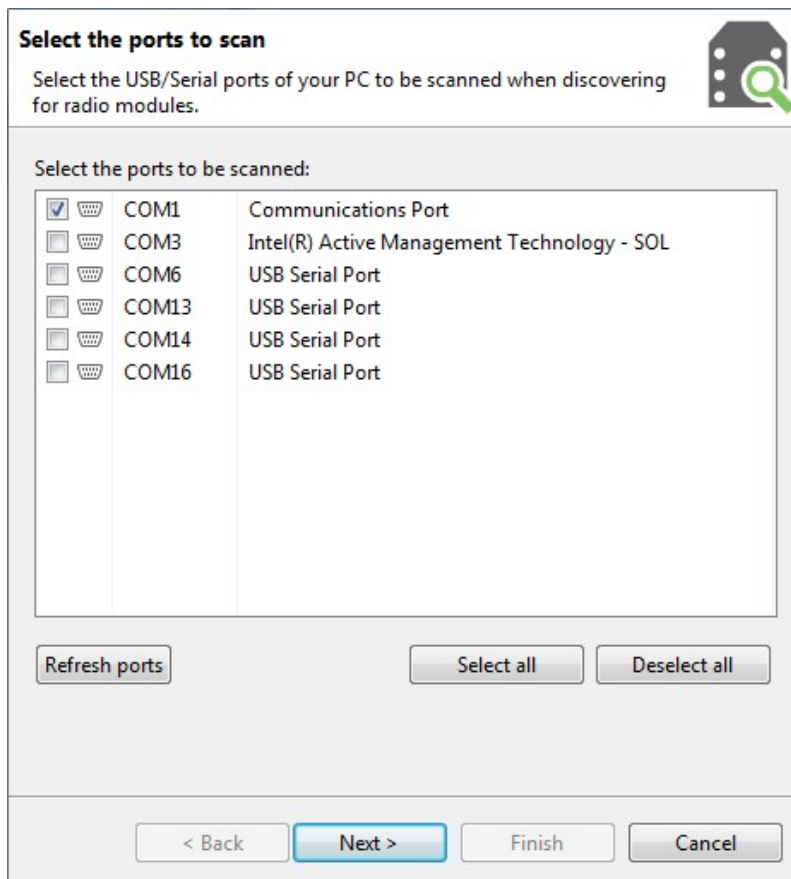


For more information, see [Local radio modules](#) and [Radio module information panel](#).

## Discover local radio modules

XCTU can discover radio modules that are connected directly to your computer. You can use the discovery tool if you don't know the serial configuration of your radio module, don't know the port it is connected to, or want to add multiple modules at once.

1. Click the **Discover radio modules** button  on the XCTU toolbar. The **Discover radio modules** dialog box opens.



2. Select the serial ports you would like to scan for radio modules. Click **Next**.

3. Select any port parameters you would like to include in the search process.

**Note** XCTU displays estimated discovery time in the **Set port parameters** dialog. Adding more port parameters to the search increases discovery time.

**Set port parameters**  
Configure the Serial/USB port parameters to discover radio modules.

**Baud Rate:**

- 1200
- 2400
- 4800
- 9600
- 19200
- 38400

**Data Bits:**

- 7
- 8

**Parity:**

- None
- Even
- Mark
- Odd
- Space

**Stop Bits:**

- 1
- 2

**Flow Control:**

- None
- Hardware
- Xon/Xoff

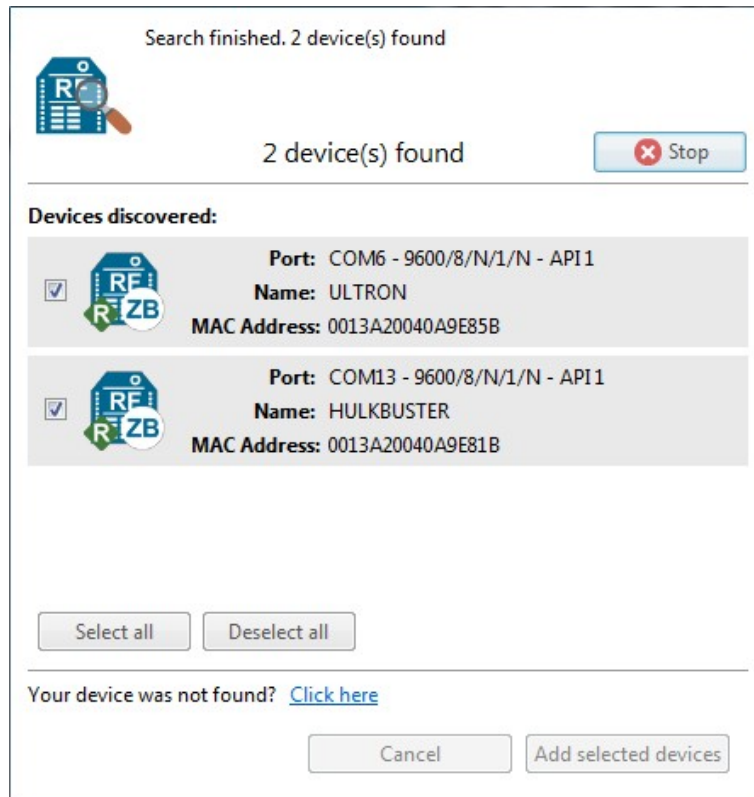
Buttons: **Select all**, **Deselect all**, **Set defaults**

Estimated discovery time: 00:10

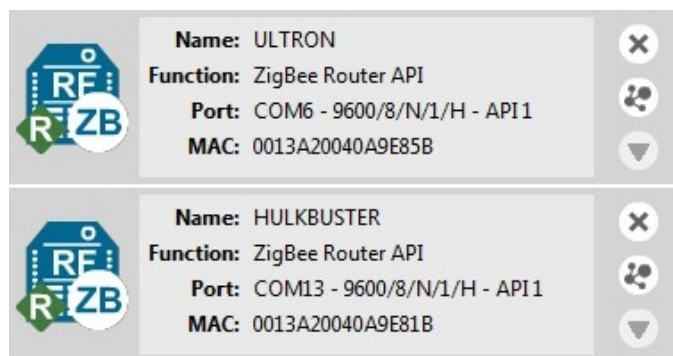
Navigation: **< Back**, **Next >**, **Finish**, **Cancel**

4. Click **Finish** to initiate the discovery scan.

A new dialog opens, displaying devices found and estimated time remaining. You can click **Stop** to halt the discovery process at any time. For example, you can stop the process if the modules you were looking for are already found.



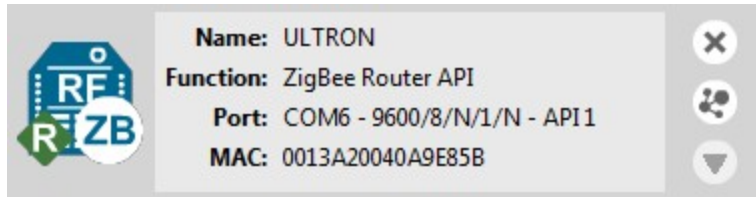
5. Select the box next to the module(s) you want to add to your device list and click **Add selected devices**. The modules appear in the device list.



For more information, see [Local radio modules](#) and [Radio module information panel](#).

## Radio module information panel

Local radio modules appear as big buttons in the modules list. Each module displays identifying information about itself.















To work with a radio module, you must select it from the list of devices. When you hover over a module, the background color changes to yellow.



Selecting a radio module refreshes the contents of the working area, displaying the information or actions you can perform on the selected module. The contents of the working area depend on the active working mode.

### Module Icon




An icon on the left side of the information panel identifies the module type and protocol.

Icon	Module type	Protocol
	XBee	ZigBee
	XBee	DigiMesh (Digi's proprietary protocol)
	XBee	802.15.4
	XBee	Point-to-multipoint
	XBee	Smart Energy




Icon	Module type	Protocol
	XBee	ZNet
	XBee	Wi-Fi
	XBee	XStream Compatibility
	XTend	XTend native
	XTend	DigiMesh (Digi's proprietary protocol)
	XLR	XLR-PRO native
	XLR Module	XLR Module
	XBee	SX

Icon	Module type	Protocol
	XBee	Cellular
	XBee	Thread

Each icon may contain a small image in the lower-right corner that identifies the role of that module within its network:

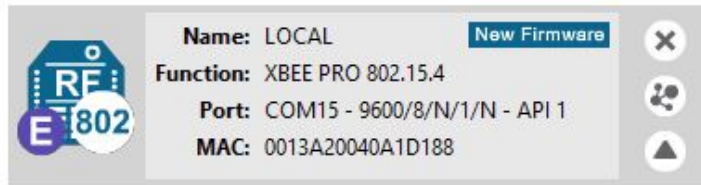
Icon	Description
	Coordinator
	Router
	End device

Buttons along the right-hand side of the module information panel perform actions on the selected module:

Button	Name	Description
	Remove	Removes the radio module from the list of devices. Also removes the associated communication console and network view of that module. Removes the local module and any associated remote modules.
	Search	Discovers remote radio modules in the same network. Dependent on radio module protocol. Performs an SSID discovery for Wi-Fi modules, and for any other protocol performs a discovery of remote radio modules. See <a href="#">Discover remote radio modules</a> and <a href="#">Active scan</a> .
	Expand/collapse	Expands and collapses the remote radio modules list.


### New firmware indicator

When a radio module is not running the latest firmware version it is able to run, the **New Firmware** control displays next to the name label for the information panel.



This control indicates that there is a newer firmware version that can be flashed in the radio module. If you click **New Firmware**, the **Update firmware** dialog is automatically opened for that radio module. All of the available firmware versions for the radio module are listed in the dialog.

## Remove a radio module


Click **Remove**  to remove a selected radio module from the list of devices. Removing a module from the list also disables the associated communication console and network view of the module.

---

**Note** Clicking the **Remove** button of a remote module removes only that radio module from the sub-list of remote modules. Click the **Remove** button of a local module with a sub-list of remote modules removes the local module as well as all of its remote modules.

---

## Expand/Collapse radio modules list

If the protocol of the radio module is ZigBee or DigiMesh and you have found remote modules in the same network, you can use the **expand/collapse**  button to expand or collapse the list of remote modules under the local device.

---

**Note** This button is only enabled for local radio modules.

---



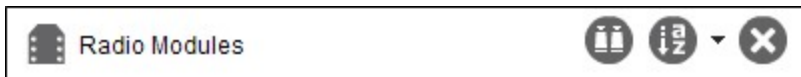
## Module information box

When you hover over the icon, XCTU displays additional information about the selected module, including module type, family, protocol, device type, firmware, and hardware.


<b>Module type:</b>	XB24-ZB
<b>Family:</b>	XBEE
<b>Protocol:</b>	ZigBee
<b>Device type:</b>	Router
<b>Firmware:</b>	23A7
<b>Hardware:</b>	0x19

## Organize your modules


The Radio Modules view contains a toolbar with options to manage radio modules in the list. These options are only enabled when the list contains at least one radio module.



### Find radio modules

Click the **Find radio modules** button  to find local and remote radio modules in your list. For details, see [Find radio modules](#).

### Sort radio modules


Radio modules are displayed in the order in which you added them to XCTU. Click the **Sort radio modules list** button  to sort radio modules by name, function, serial port, or MAC address. Or you can select a specific device and move it up/down in the list.

---

**Note** The sorting feature affects both local and remote radio modules.

---


### Clear radio modules

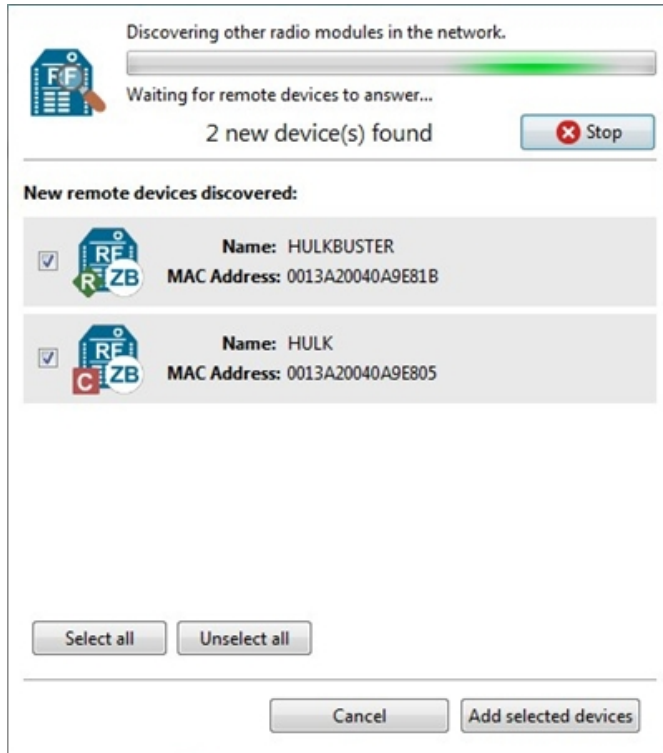
Click **Clear radio modules list**  to remove all discovered modules from the radio modules list. For more information, see [Local radio modules](#) and [Remote radio modules](#).

## Discover remote radio modules

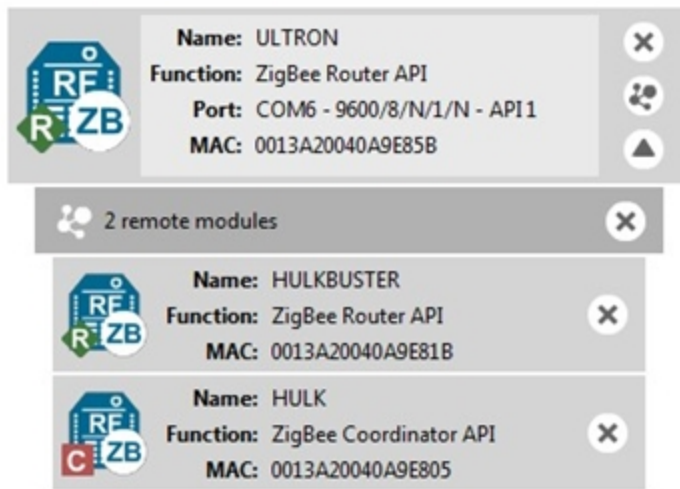
You can execute a discovery process to locate remote radio modules in the same network as the local (selected) module. To discover remote modules:

1. Select a module from your device list. If you do not have any modules in the list, see [Add a radio module manually](#) or [Discover local radio modules](#).

2. Click the **Discover radio nodes in the same network** button . As XCTU discovers new remote radio modules, they appear in the discovery process dialog box.



3. Click **Stop** to halt the discovery process at any time.
4. Check the box next to the module(s) you want to add to your device list and click **Add selected devices**. The discovered remote modules appear in the list of remote modules.



---


**Note** XBee Wi-Fi modules do not support the remote radio modules discovery feature. Instead, they can look for access points.

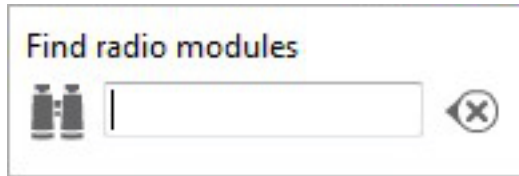
---

For more information, see [Remote radio modules](#).

## Find radio modules

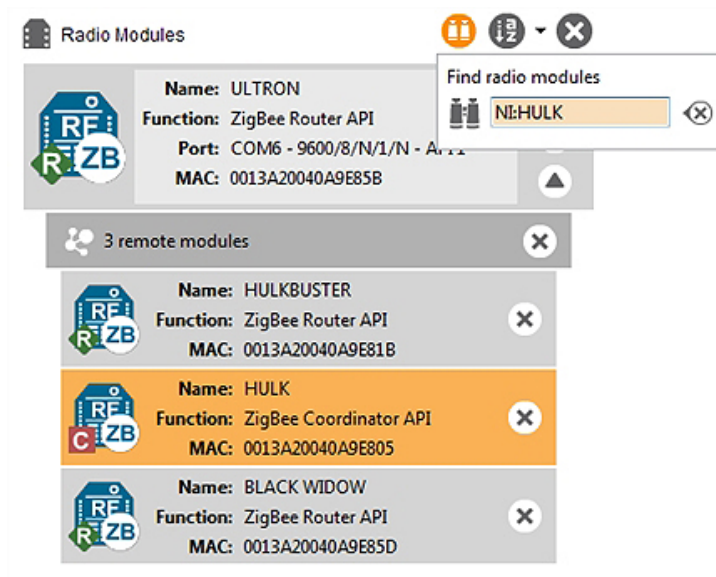
To find local or remote radio modules, you must have already discovered the network they are on. You can then use the Find radio modules search box to find radio modules by MAC address, name, network address, and other search expressions.

1. On the Radio Modules toolbar, click the **Find radio modules**  button. The **Find radio modules** search box appears.



2. Enter your search expression to find one or more modules.  
For a list of search prefixes and wildcards, as well as sample searches, see [Search expressions](#).
3. Press **Enter**.

The background color of the search box and search icon indicates status. Yellow indicates matches found by XCTU, and red indicates no matches. Modules found along the list are also highlighted in yellow.



## Search expressions

You can enter the following search prefixes in the Find radio modules dialog box.

Search prefix	Search by
MAC: (or no prefix)	MAC address

Search prefix	Search by
SH:	Serial Number High
SL:	Serial Number Low
NI:	Node Identifier (only available in 802.15.4 and DigiMesh)
MY:	16-bit Network Address (only available in 802.15.4 and ZigBee)

You can also use a wildcard if you do not want to specify the entire parameter or if you want to find more than one node.

Wildcard	Equals
*	any string
?	any character
\	escape for literals (i.e. *, ?, or \)

### Sample searches:

- To search for a module with node identifier (NI) "NODE1" and network address (MY) 0831, enter

---

NI:NODE1, or MY:0831

---

- To find all nodes whose MAC starts with 00 and ends with B, enter

---

MAC:00\*B

---

## Configure your modules

---

This section describes how to use Configuration working mode to configure your modules and change application settings once you have added a radio module or modules to your list of devices.

Configuration working mode .....	46
Read radio module configuration .....	54
Write module settings .....	54
Load default firmware settings .....	54
Update firmware .....	55
Work with configuration profiles .....	57
Search for a firmware setting .....	67
Configure remote modules securely .....	67
Configure a Wi-Fi access point .....	71
Enable and configure Bluetooth .....	73
View firmware release notes .....	74

## Configuration working mode

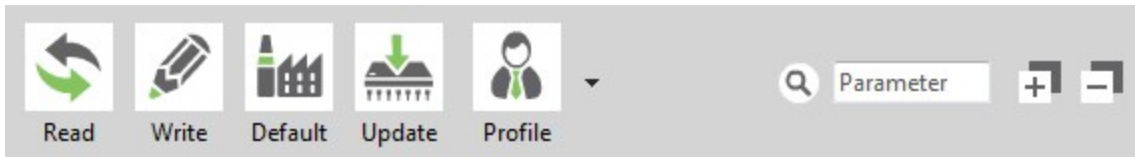
When you launch XCTU, Configuration working mode opens as the default operating mode.






The Configuration working mode allows you to configure any radio module that has been added to your device list. When you select a module from the list, XCTU loads the firmware information of the selected radio module and displays the firmware settings in the working area. It automatically reads the values and fills in the fields.

### Configuration toolbar

The Configuration toolbar presents the configuration actions you can perform with the selected radio module and firmware settings.



Button	Name	Description
	Read module settings	Reads the firmware settings for the selected radio module.
	Write module settings	Writes new firmware values to the selected radio module.
	Load default firmware settings	Loads the default firmware values on to the selected radio module but does not write them.
	Update firmware	Opens the <b>Update the radio module firmware</b> dialog, displaying available compatible firmware for the selected radio module.

Button	Name	Description
	Load/save configuration profile	Opens the <b>Load configuration profile</b> or <b>Save configuration profile</b> dialog.
	Search	Enables you to search firmware settings by AT parameter.
	Expand/collapse settings	Expands or collapses all firmware settings sections.

### Firmware information panel

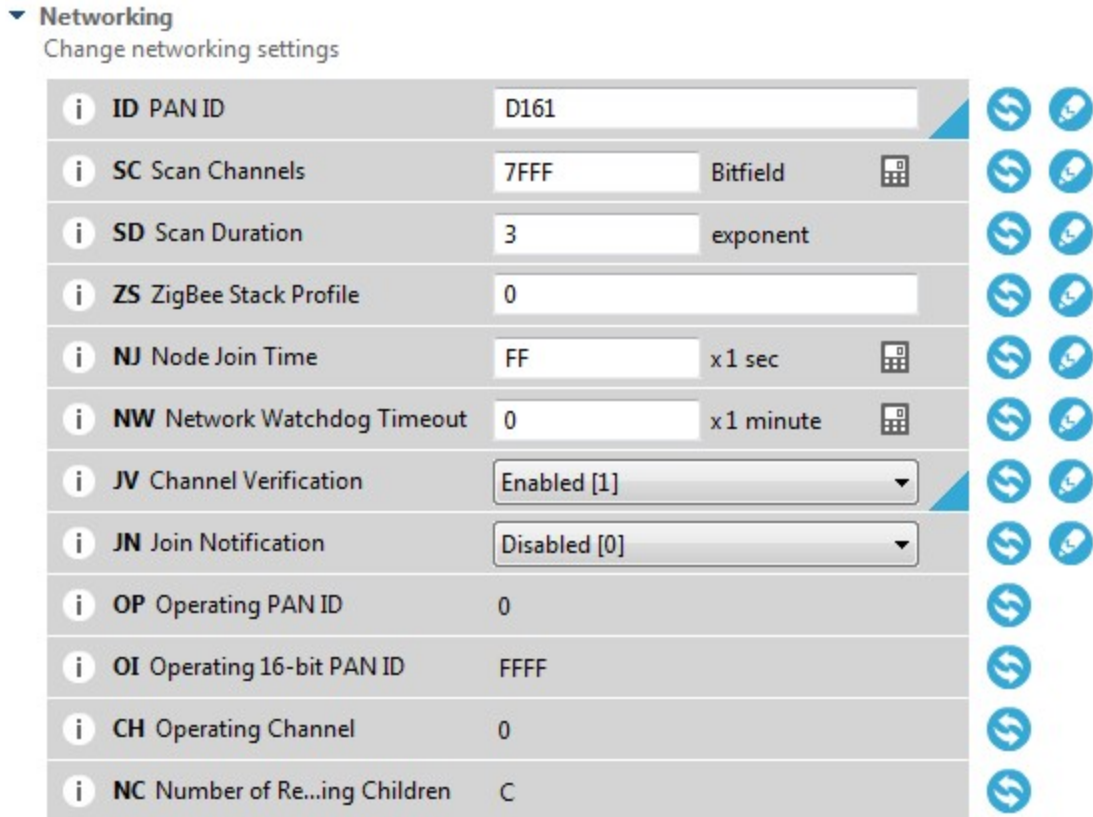
The firmware information panel is located below the Configuration toolbar and displays information about the firmware running in the selected radio module.

**Product family:** XBP24BZ7      **Function set:** ZigBee Router API      **Firmware version:** 23A7







---

### Firmware settings

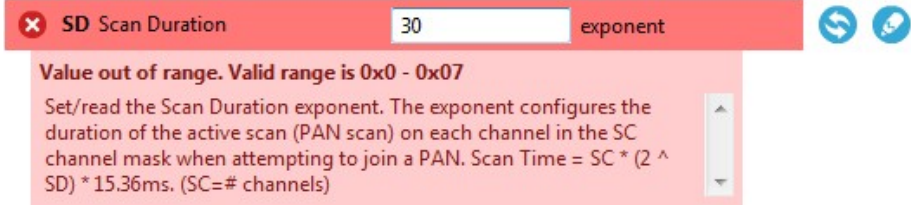
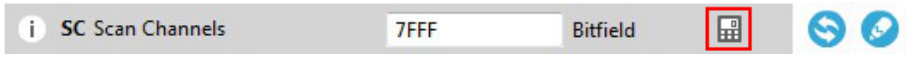
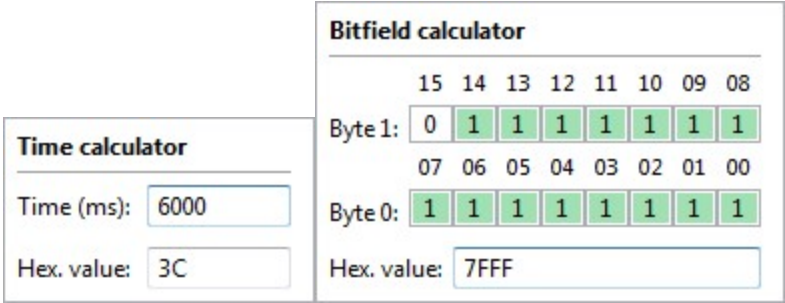


XCTU displays firmware settings of the radio module below the firmware information panel. They are divided into sections or categories with a short description in each. Read-only settings are displayed with a gray label.



The following table provides descriptions for the setting controls.






Field	Description
Information button	<p> <b>SD</b> Scan Duration 3 exponent  </p> <p>Clicking the information button displays a short description of the setting, including the default value and the valid range, if the setting is numeric.</p> <p> <b>SD</b> Scan Duration 3 exponent  </p> <p><b>Range: 0x0 - 0x07 (Default: 3)</b> Set/read the Scan Duration exponent. The exponent configures the duration of the active scan (PAN scan) on each channel in the SC channel mask when attempting to join a PAN. Scan Time = SC * (2 ^ SD) * 15.36ms. (SC=# channels)</p>
AT parameter	<p>Displays the associated AT parameter of the setting. Some settings, such as actions, may not have an associated AT parameter.</p> <p>In the example above, the AT parameter is <b>SD</b>.</p>
Setting name	<p>Displays the name of the setting.</p> <p>In the example above, the setting name is <b>Scan Duration</b>.</p>



Field	Description
<p>Setting configuration control</p>	<p>Contains the text box or combo box where the setting value must be entered or configured. You can hover over the text box of a numeric setting to display the valid range for that setting. Always enter numeric values in hexadecimal format (without the '0x' prefix), unless the hover text indicates that an ASCII value is required.</p> <p>If the setting is configured with an invalid value, an explanation for the error appears.</p> 
<p>Units label</p>	<p>Displays the units of measure for that setting. Not all settings have a units label.</p> <p>In the example above, the unit is <b>exponent</b>.</p>
<p>Value calculator icon</p>	 <p>Clicking the value calculator icon launches a time or bitfield calculator for numeric settings that are difficult to compute. The content of the calculator panel depends on the setting.</p> 
<p>Refresh and write buttons</p>	<p>Clicking the refresh button  or write button  allows you to read or write the value of the setting. Some settings, such as the read-only settings, do not have a write button.</p>

### Setting status

XCTU delineates the status of each setting with background color and/or the color of a triangle located next to the setting value. These are the possible statuses of a setting:

	Green triangle	The value of the setting has changed but it has not been written in the radio module yet.
	Blue triangle	The value of the setting is written in the radio module but is different from the default value.
	Gray background	The value of the setting is written in the radio module and matches the default value.
	Yellow background	Indicates that the setting is highlighted because it has been found using the <b>Search</b> parameter control.
	Red background	The value of the setting is not valid.

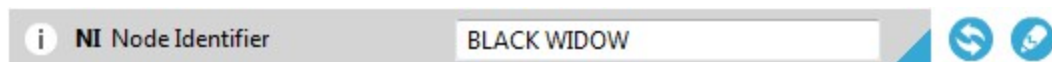
## Setting types

There are different kind of settings that you can configure in a radio firmware. Depending on the setting type they display different controls and options.

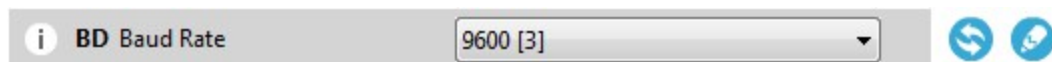
- Numeric settings:** These settings must always be configured with a numeric value in hexadecimal format (without the '0x' prefix). Hovering over the text box of a numeric setting displays the valid range for the setting. There are several types of settings:



- Text settings:** Text settings are very similar to the numeric settings, but they can be configured with hexadecimal or ASCII characters. If you hover over the text box of a text setting, a dialog displays the minimum and maximum characters and whether they must be an ASCII or hexadecimal value.



- Combo settings:** A combo box displays all the possible values of the setting with symbolic text, to help you to choose the correct option.



- Read-only settings:** These settings cannot be modified. They can only be read from the radio module and their values are displayed in a label.



- Action settings:** These settings can be neither read nor written. The main purpose of the action settings is to execute a task or process in XCTU that implies some interaction with the radio module. To learn more about the Action settings see the Special functions topic.



## Special functions

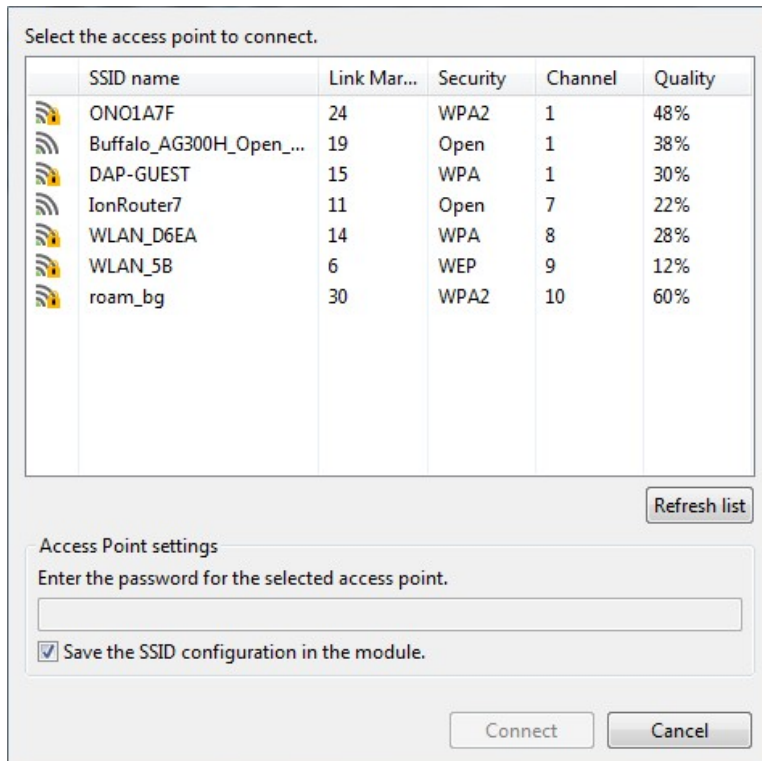
Some settings within XCTU cannot be read or written. Instead, they execute tasks or processes in XCTU related to interaction with the radio module. The processes that these settings execute are called special functions. At this time, XCTU has two special functions: **Active scan** and **Bluetooth Authentication**.

### Active Scan

The **Active Scan** function discovers and configures the access point for an XBee Wi-Fi module.



When you click the **Active Scan** button, XCTU reads the SSID configuration of the Wi-Fi module. If the module has an SSID already configured, you need to clear the configuration and perform a new SSID discovery. If the SSID configuration is empty, the nearby SSIDs are displayed in a new dialog.



The dialog displays all the nearby access points as well as their security protocols and signal quality. Select the Access Point you want the Wi-Fi module to connect to and, if necessary, configure the password of the Access Point. The Access Point settings also have a check box that allows you to

permanently save the SSID configuration in the Wi-Fi module. If you uncheck this option, the next time you reset the module the SSID configuration is cleared.

Click **Connect** to connect the Wi-Fi module to that Access Point and refresh the settings of the radio module.

### **Enable BLE and configure the BLE password**

The Bluetooth Authentication function allows you to configure the security parameters of your XBee device with Bluetooth Low Energy (BLE) support. This step is mandatory if you want to enable Bluetooth on the module.



XBee BLE security is based on the Secure Remote Password protocol. When you click the **Configure** button, XCTU displays a dialog asking you to either:

1. Type a password, which you are asked for when you connect to the XBee via Bluetooth Low Energy.
2. Enter a random salt and a password verifier generated from a password and that salt (as specified in the SRP protocol).



**CAUTION!** Only use the advanced configuration if you are familiar with the SRP protocol. You must know the password which originated the verifier since you will be asked for it when you connect to the XBee via BLE.

Basic configuration

*i* Password:

Advanced configuration

*i* Salt:

*i* Verifier:


Note that you must know the password which originated this verifier. Connecting to the XBee via BLE requires the known password.

When you enter the password or the salt/verifier combination, click **OK** to save the settings to the module.

For more information about Bluetooth security, see the user guide for your XBee device.

## Read radio module configuration


You can refresh a radio module's firmware settings once you have added the module to your device list. To read a module's configuration settings:

1. Switch to Configuration working mode .
2. Select a radio module from the device list. XCTU displays the current firmware settings for that module.

---


**Note** If the selected module is remote and the Remote AT Command password (**KZ**) is set, you may be asked to enter that password in order to access the module configuration.

---


3. From the configuration toolbar, click the **Read module settings** button  to refresh the selected radio module's firmware settings.

## Write module settings

You can configure a radio module's firmware settings once you have the module to your device list. To configure a radio module:

1. Switch to Configuration working mode .
2. Select a radio module from the device list. XCTU displays the current firmware settings for that module.


---

**Tip** To refresh the selected radio module's firmware settings, click the **Read module settings** button  on the configuration toolbar.

---




**Note** If the selected module is remote and the Remote AT Command password (**KZ**) is set, you may be asked to enter that password in order to access the module configuration.

---

3. Change the value of the setting or settings to be configured.
4. Click the **Write module settings** button  to write any newly configured firmware values to the module.



## Load default firmware settings

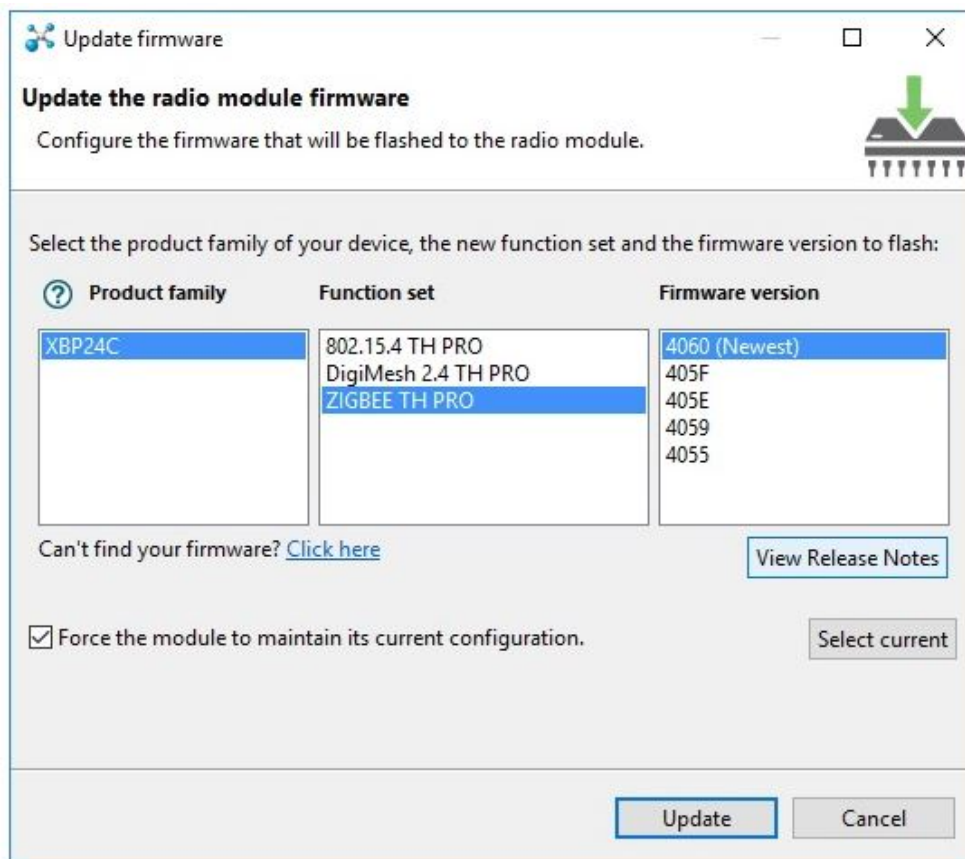
You can load default radio firmware settings in a module in your device list.

1. Switch to Configuration working mode .
2. Select a radio module from the device list. XCTU displays the current firmware settings for that module.
3. On the Configuration toolbar, click the **Load default firmware settings** button  to load the default values established by the firmware.
4. Firmware settings are loaded but not written to the radio module. In order to write them in the module, click the **Write module settings** button  on the toolbar.

## Update firmware

You can use XCTU to update a module's radio firmware.

1. Switch to Configuration working mode .
2. Select a radio module from the device list.
3. Click the **Update firmware** button . A dialog box appears displaying the available and compatible firmware for the selected module.



4. Choose the firmware family, function, and version.

---

**Note** If you do not remember the firmware version that is currently installed in your radio module, click **Select current** to automatically select it.

---

5. Click **Update**. A dialog box displays update progress. You can click the **Show details** button to view a detailed progress log, and **Hide details** to hide it.

---

**Note** During the firmware update process, XCTU attempts to obtain the module information again, as some critical settings such as the operating mode could have changed. If the **Maintain current module configuration** setting is checked, XCTU writes the old configuration to the module and then reads the setting's values.

---

## Cellular modem firmware updates

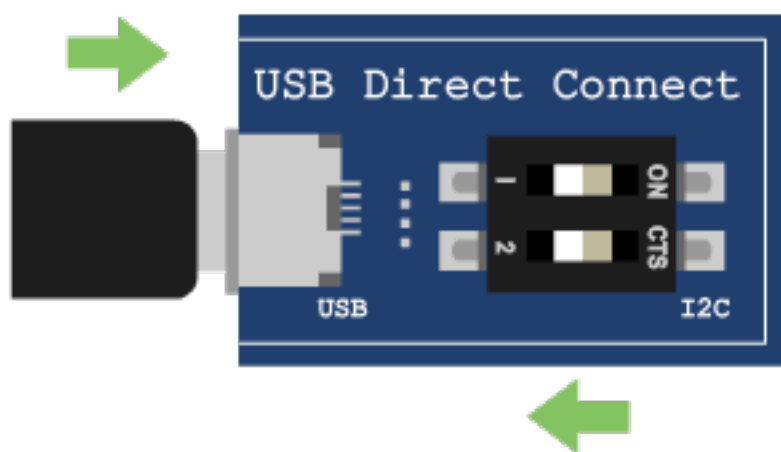
When you update the firmware of an XBee Cellular device, XCTU may need to update the modem firmware. This process is completely transparent, but XCTU requires an internet connection to download the modem firmware.

---

**Note** The system prompts you for confirmation when the device requires a modem firmware update, as this process may take up to 30 minutes in old devices.

---

For newer XBee 3 Cellular devices (LTE CAT 1 and LTE-M/NB-IoT), this process is done over USB and is much faster, but only works in Windows. You need to have the appropriate [USB drivers](#) installed on your computer and either a [Digi XBIB-CU-TH](#) development board or your own hardware that makes a USB port available to a PC. Before starting the update, XCTU asks you to connect the USB micro-B cable to the USB Direct Connect port and set the switches to OFF.



## Remote firmware updates

You can use XCTU to perform firmware updates on remote modules because once you add a remote module to XCTU's device list, the update process is exactly the same whether you are updating a local module or a remote module. To perform a remote firmware update, the local module must be working in API or API escaped operating mode.

Remote firmware updates can be performed on the following radio modules:

- XBee/XBee PRO SX
- XLR Module
- XBee/XBee PRO 802.15.4 (S2C module versions only)
- XBee/XBee-PRO DigiMesh 2.4 (S2C module versions only)
- XTend RF Module Family (SX module versions only)
- XLR PRO Radio Solution
- XBee/XBee-PRO ZB and Programmable XBee-PRO ZB
- XBee/XBee-PRO ZB SMT and Programmable XBee-PRO ZB SMT
- XBee-PRO 900HP and Programmable XBee-PRO 900HP
- XBee 865LP and Programmable XBee 865LP



- XBee 868LP SX
- XBee3 (Zigbee, DigiMesh 2.4, and 802.15.4)

---

**Note** If something goes wrong during an over-the-air firmware update on a remote module—for example, communication is lost because the remote device is disconnected—you must perform a manual recovery. See [Recover a radio module](#).

---

## Bootloader updates

Some firmware versions require a specific bootloader version in order to work properly. The bootloader is the software running in the module that, among other things, launches the firmware as soon as the module boots.

The bootloader update takes place during the firmware update process. If the firmware to be flashed requires a bootloader update, XCTU updates the bootloader first and then the module's firmware.

---

**Note** The system prompts you for confirmation when the firmware requires a bootloader update, as this process erases the configuration of the module. You must accept it in order to continue with the firmware update process. If you decline the bootloader update, the overall firmware update process is canceled.

---

## Work with configuration profiles

A configuration profile is a snapshot of a specific radio firmware configuration, including firmware, settings values and other configuration information. XCTU allows you to create, inspect and apply configuration profiles to the radio module. This feature is useful in a production environment when you need to set the same parameters on multiple radios.

These are the actions that XCTU allows you to perform related to configuration profiles:

- [Create a configuration profile](#)
- [Apply a configuration profile](#)
- [Load and edit a configuration profile](#)

### Create a configuration profile

You can create a configuration profile from two different places within XCTU. You can either create a profile based on the configuration of an XBee device or create a profile from scratch.

#### ***Create a profile based on the configuration of an XBee device***


The first way to create a profile is to create a representative model upon which to base the configuration profile and then save the profile. To do this, you need to have a radio module added in XCTU.

---

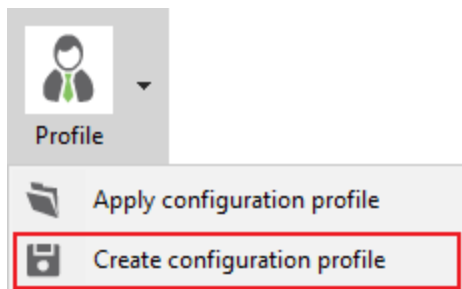
**Note** You only need to configure the values; it is not necessary to write the settings to the module.

---

To do so follow these steps:

1. Switch to **Configuration working mode** .
2. Select a radio module from the device list.
3. Configure the radio module with your desired values.

- Click the **Configuration profile's** drop-down menu on the Configuration toolbar and select **Create configuration profile**.



- [Create profile wizard](#) displays.

### **Create a new profile from scratch**

It is also possible to create a new profile without any pre-configured firmware or settings. To create a new profile from scratch:

- Launch the Profile editor tool by selecting **Profile Editor** from the **Tools** drop-down menu on the main toolbar.
- Click the **Create** button from the toolbar. The [Create profile wizard](#) displays.

In either case, the Create profile wizard displays. Follow the steps provided there to create the configuration profile.

- [Create profile wizard](#)

### **Create profile wizard**

The **Create profile wizard** allows you to create a new configuration profile to be used later in XCTU or other XBee management applications. You can access this wizard either from the **Profile editor** tool or from the toolbar of the **Configuration working mode** view. See [Create a configuration profile](#) for more information.

Once the wizard is open, follow these steps to configure and create a new configuration profile:

- In the first page of the profile you can set the profile **description** and the configuration features:
  - Flash radio firmware.** With this setting you can decide if the profile will flash the firmware in the XBee module or not. If you check it you can choose one of the following flash options:
    - Flash if firmware is different.** Firmware is flashed in the module only if the firmware version that the module is running is different from the one specified in the profile.
    - Flash always.** Firmware is always flashed regardless of the version running in the device.
  - Reset module to factory defaults before applying settings.** Check this option if you want to reset the XBee module to its default values (equivalent to send an **RE** command) before configuring the firmware parameters.
  - Format existing file system.** You can check this option to format the existing file system of the device when applying the profile. If you choose to flash a new file system too, the

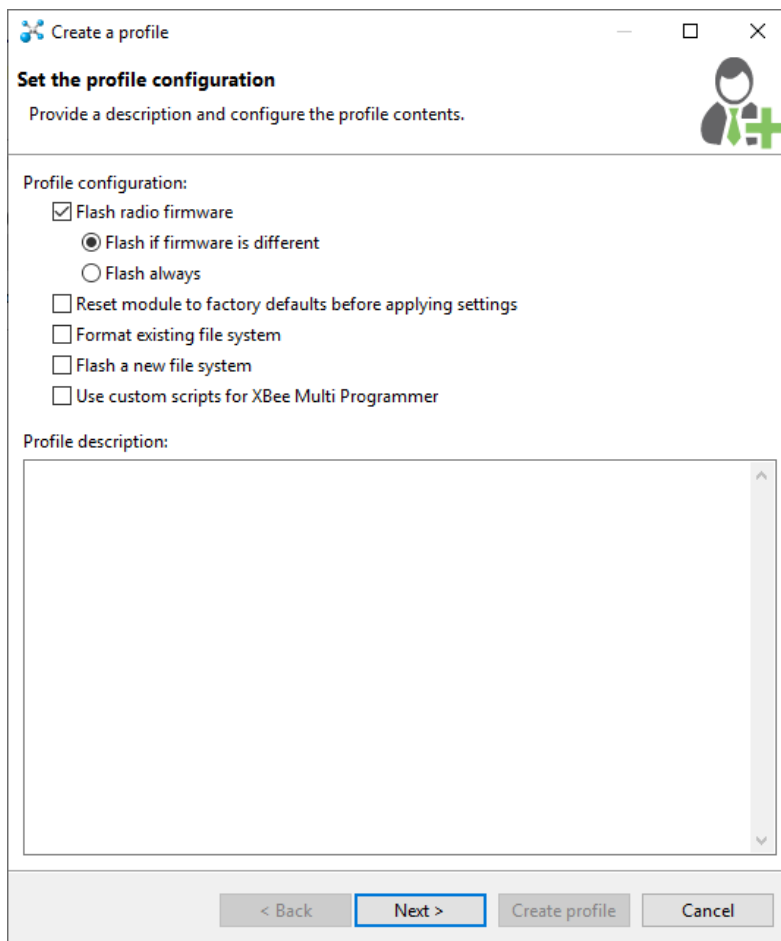
format step will be performed before flashing the new file system.

- d. **Flash a new file system.** With this setting you can attach a file system to the profile that will be transferred to the XBee module when applying the profile. Notice that the **Format existing file system** option is automatically checked when you check this one.
- e. **Use custom scripts.** When creating a profile with this setting, you can indicate a pre-processing script or a post-processing script, or both to be used when programming devices with XBee Multi Programmer.
  - This feature allows you to write your own scripts using your own libraries or the ones provided by Digi to interact with your XBee devices.
    - **Pre-processing script:** This script is run before applying the profile to the XBee device. Use it to read specific information of your device before applying the configuration profile.
    - **Post-processing script:** This script is run after applying the profile to the XBee device, as a final step. With it you can set specific configurations to your devices, for example, use it to set a different Node Identifier on each of your XBee devices.

---

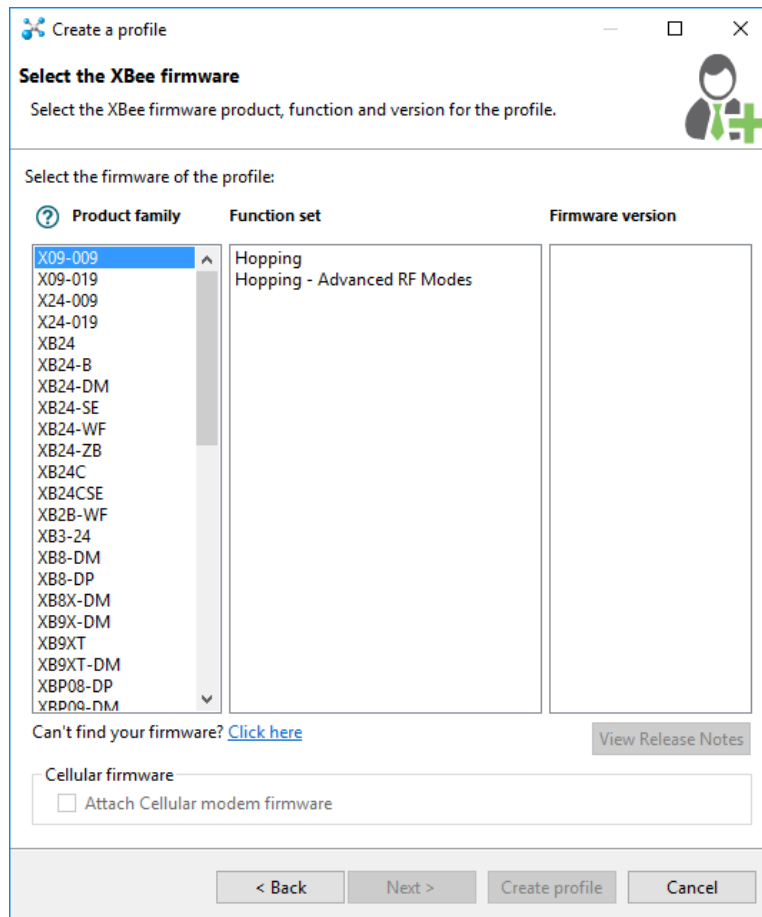
**Note** This feature can only be used by XBee Multi Programmer. Other tools will skip the custom scripts steps when applying a profile to an XBee device.

---



The screenshot shows a dialog box titled "Create a profile" with a sub-header "Set the profile configuration". Below the sub-header is the instruction "Provide a description and configure the profile contents." and a user icon. The "Profile configuration:" section contains several options: "Flash radio firmware" (checked), "Flash if firmware is different" (selected with a radio button), "Flash always" (unselected), "Reset module to factory defaults before applying settings" (unchecked), "Format existing file system" (unchecked), "Flash a new file system" (unchecked), and "Use custom scripts for XBee Multi Programmer" (unchecked). The "Profile description:" section is a large empty text area. At the bottom are buttons for "< Back", "Next >" (highlighted), "Create profile", and "Cancel".

2. Click **Next** to configure the XBee firmware of the profile. Choose the desired firmware selecting its corresponding properties:
  - a. **Product family** of the firmware.
  - b. **Function set** that determine the functionality of the firmware (mode of operation, special sensor and adapter settings, and so forth).
  - c. **Firmware version**. The most recent firmware version appears at the top of the list.

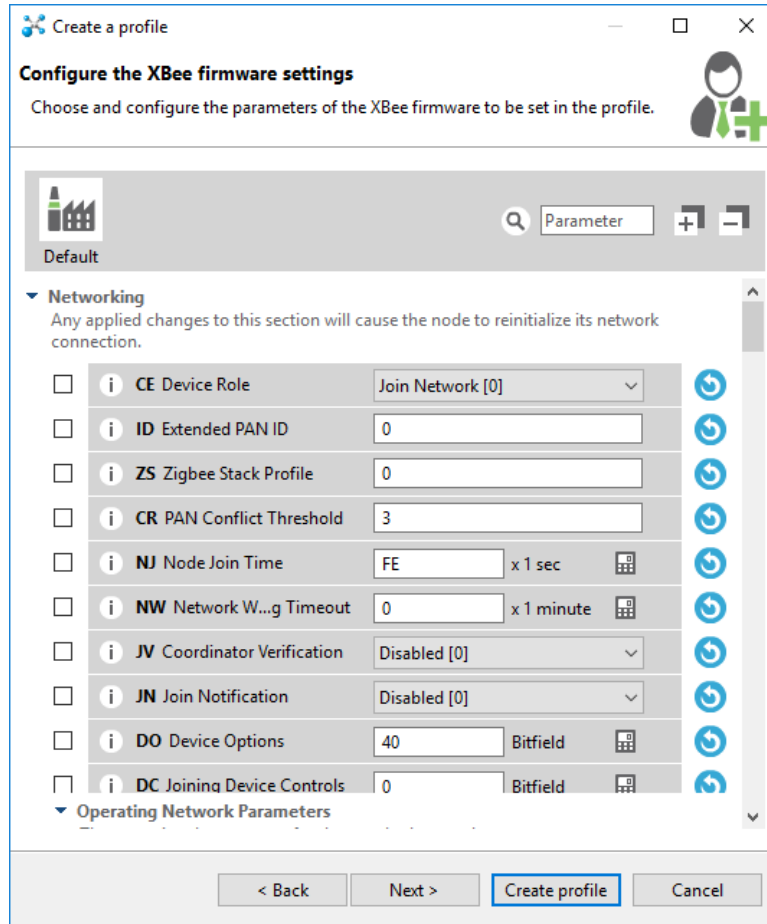



---

**Note** If you accessed the wizard from the **Configuration working mode**, the firmware should already be selected with the firmware that the XBee device is running.

---

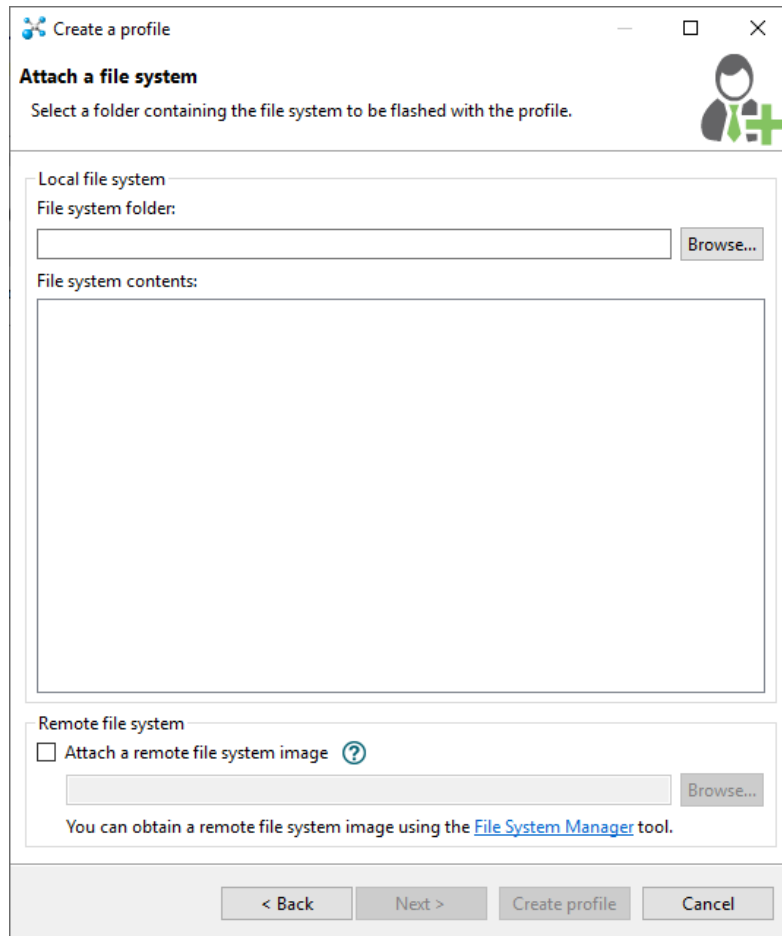
3. Click **Next** to configure the XBee firmware parameters. This page displays all the firmware settings split in categories. Each configurable setting has a checkbox at the left side. Check and edit the value of those settings you want to be configured by the profile.



**Note** If you accessed the wizard from the **Configuration working mode**, the firmware settings should be already checked and selected with the ones configured in the XBee device.

- Click **Next** to set a file system to the profile. If the profile is configured to attach a file system—the first page of the wizard—you can use the **Browse** button to specify it here. Updating the file system of remote devices must be performed through a remote file system image. So, if the target firmware supports it and you want the profile to flash a file system in remote devices, you must check the **Attach a remote file system image** option and then specify the image using the corresponding **Browse** button. If the target firmware does not support OTA file system updates, the **Attach a remote file system image** option will not be displayed.

**Note** If you do not know how to obtain a remote file system image, see [Update the file system of remote devices](#).




5. Click **Next** to set the pre and/or post scripts to be executed when flashing the profile with XBee Multi Programmer.

This feature allows you to run a script/program before and/or after applying a profile to the target device. This wizard page is split into two sections, one for the pre-processing script, and the other for the post-processing script. Depending on whether you want to run the pre-processing script or the post-processing script, or both, fill in the appropriate information on each section respectively. Each section contains fields to indicate:

- a. **The command to execute.** Use whatever scripting language you want, but make sure it can be used by the machine that applies the profile. For example, if the command to run is: `pyhton set_node_id.py`, it is mandatory that the correct Python interpreter is installed and accessible in the machine that applies the profile.
- b. **The maximum timeout for the script.** By default, it is five minutes. This is the maximum time XBee Multi Programmer should wait for the script to finalize its execution. When this time has passed, if the script has not finished, XBee Multi Programmer will assume your program got locked, and will stop the execution of the script returning a Timeout Exception. This parameter is specified in minutes, and its valid range of values goes from 1 to 300 minutes.
- c. **Optionally, the folder containing the script/program to run.** This is a directory containing the script to be run and all of the resources it might need during its execution.


It is not mandatory to attach any script to the profile, as long as it is accessible by the machine that applies the profile. To indicate the script/program to run, use one of the following:

- Attach the folder that contains the script to the profile. To do so, click the **Browse** button of the corresponding section. Select the complete folder that contains your script—this will be the execution path. It does not matter that in order to run the script needs multiple files, as long as they are all included in the selected folder. If external resources are required by the script, make sure they are properly handled within your script, or include them in the same folder as your script so they are attached to the profile.
- Use absolute paths to call a script that already exists in the machine that will apply the profile. The first field of each section (the command) can be called using absolute paths. Bear in mind that such script needs to handle all paths in an absolute manner, as well. In this case, the execution path will be the path where the XBee Multi Programmer was installed.

 Create a profile
— □ ×

### Attach your custom pre/post scripts

Set the paths of the custom scripts to be executed before and/or after loading the profile onto a target XBee device





Pre-processing script command:	Script timeout (minutes):
<input type="text"/>	<input type="text" value="5"/>
Pre-processing script folder:	
<input type="text"/>	<input type="button" value="Browse..."/> <input type="button" value="Clear"/>
<div style="border: 1px solid #ccc; width: 100%; height: 100%;"></div>	
Post-processing script command:	Script timeout (minutes):
<input type="text" value="python set_node_id.py --nametable resources/names.xml"/>	<input type="text" value="3"/>
Post-processing script folder:	
<input type="text" value="C:\profile_post_script"/>	<input type="button" value="Browse..."/> <input type="button" value="Clear"/>
<div style="border: 1px solid #ccc; padding: 5px;"> <div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;"> <span>▼</span> <span style="color: #ffc107;">📁</span> <span>resources</span> </div> <div> <span style="color: #6c757d;">📄</span> names.xml  <span style="color: #6c757d;">📄</span> set_node_id.py                 </div> </div> </div>	

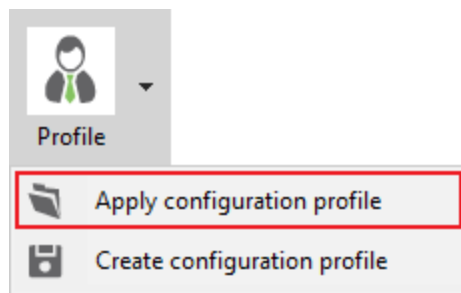
6. When you have all the profile parameters configured, click **Create profile** to open the save file dialog.
7. Select the destination path and click **Save** to create the profile.



## Apply a configuration profile

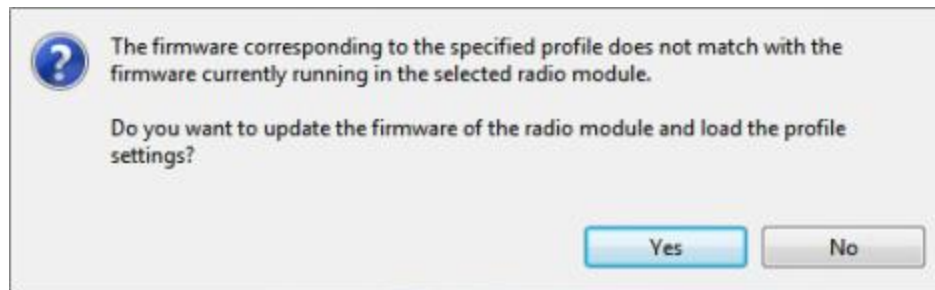
Follow these steps to apply a configuration profile to an XBee device:

1. Switch to the **Configuration working mode** .
2. Select a radio module.
3. Click the **Configuration profiles**  drop-down menu on the Configuration toolbar and select **Apply configuration profile**.

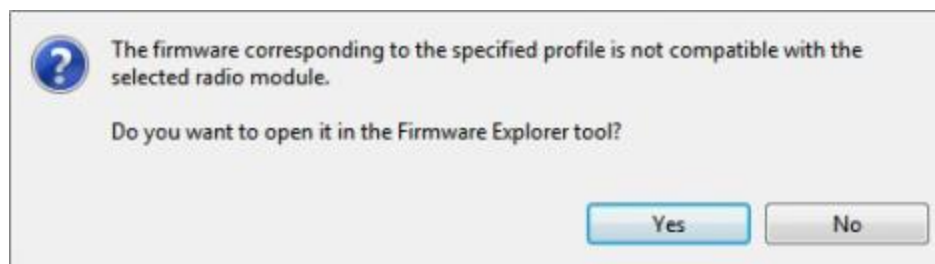


An **Open file** dialog appears, asking for the configuration profile file to load.

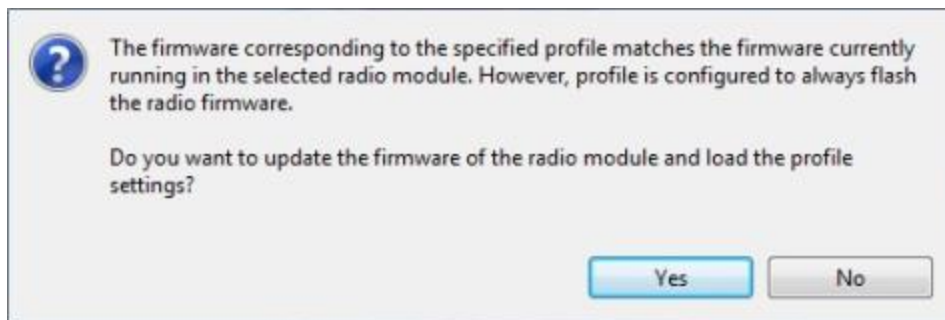
4. Locate the configuration profile (an **XPRO** file) and click **Open**. Depending on the firmware compatibility with your radio module and the firmware flash action configured in the profile, a dialog box may appear:
  - If the firmware of the configuration profile does not match the firmware running in the radio module but is compatible with the module and the profile is configured to always flash the firmware or flash when it is different, click **Yes** to update the module's firmware so the profile will load correctly.



- If the firmware of the configuration profile is not compatible with the radio module, click **Yes** to open the profile in the **Profile editor** tool.



- If the firmware of the configuration profile matches the firmware running in the radio module but the profile is configured to always flash the firmware, click **Yes** to update the module's firmware so the profile will load correctly.




- If the firmware of the profile you are loading and the firmware running in the radio module match or the profile is configured to don't flash the radio firmware, XCTU applies the settings saved in the profile to the radio module.

## Load and edit a configuration profile

**Note** This topic explains how to load a configuration profile to inspect and modify it using the **Profile editor** tool. For an explanation of how to apply a configuration profile in a module, see [Apply a configuration profile](#) instead.

Follow these steps to load and inspect a configuration profile:

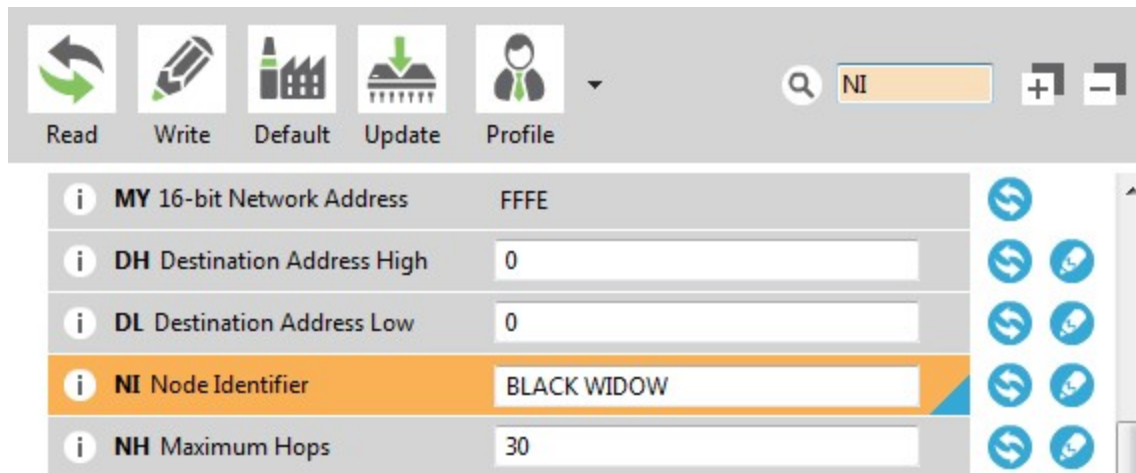
1. Launch the **Profile editor** tool by selecting **Profile Editor** from the **Tools** drop-down menu  on the main toolbar.
2. Click **Load** from the toolbar.
3. Look for the configuration profile (**xpro** extension) you want to load. Click **Open**.
4. The **Profile editor** tool loads the profile information and displays it along the different tabs of the view:
  - a. **Configuration**. Contains the configured profile features and description.
  - b. **XBee firmware**. Displays the product family, function set and version of the firmware corresponding to the profile.
  - c. **Firmware settings**. Contains the list of firmware settings configured for the profile.
  - d. **File system**. If the profile is configured to include a file system, this tab displays its structure.
  - e. **Custom scripts**. Displays the pre and post-processing custom scripts commands, timeout and structure for each script, if this profile is configured to use custom scripts.

If you want to modify the firmware, change any setting you want and click **Save**. You can then choose between one of the following **Save** options:

- **Save profile**. Overwrites the loaded profile.
- **Save as new profile...** Allows you to save a new profile with the changes made in the loaded one.

## Search for a firmware setting

The configuration toolbar includes a search box. To search for a firmware setting in the list of settings, search for the AT parameter associated with the setting. If the setting is found, it is highlighted in yellow.



## Configure remote modules securely

New XBee firmware versions allow you to add extra protection layers to encrypted networks in order to prevent an intruder from hijacking a device and making changes in the network and to protect against external man-in-the-middle attacks. You can use different protection methods depending on the XBee product:

- [Secure session](#) (XBee 3 only)
- [Secure Remote Commands](#) (XBee SX only)

### Secure session

A **secure session** is a secure connection between two XBee nodes of a network—a server and a client—where the pair can send and receive encrypted data that only they can decrypt. In a secure session, the XBee devices can act as client or server:

- **Client.** The device that is attempting to log in and send secured data/commands.
- **Server.** The device that is being logged into and will receive secured data/commands.

Secure sessions provide a way to password-protect communication between two nodes on a network above and beyond the security of the network itself. With secure sessions, a device can 'log in', or create a session with another device that is encrypted and only readable by the two nodes involved. By restricting certain actions—such as remote AT commands or OTA firmware updates—to be only allowed over one of these secure sessions, you can make it so access to the network does not allow network configuration.

---

**Note** To learn more about this feature and how it works, refer to the user guide for your XBee RF module.

---

### **Enable and disable the secure session**

The Secure session feature is enabled by default in the firmware, but you can disable it with the **SR** (SRP Secure Session disable) setting in order to allow the host MCU to play the authentication role instead:


- Bit 0 - Disable SRP authentication on the client side of the connection.
- Bit 1 - Disable SRP authentication on the server side of the connection.

Set all other option bits to 0.

### **Set the password of a secure session**

For a device to act as a secure session server, it must have a password configured. The password is configured on the server in the form of a salt and verifier used for the SRP authentication process. The salt and verifier values are calculated when you set your password.

Follow these steps to configure the secure session password of the server:

1. Switch to the **Configuration** working mode .
2. Select a Secure Session capable radio module from the **Radio Modules** list.
3. Browse the **Secure Session Authentication** setting in the list of settings (inside the **Secure Session** category) and click the **Configure** button. The **Configure Secure Session Authentication** dialog appears.

**Basic configuration**

**i** Password:

**Advanced configuration**

**i** Salt:

**i** Verifier:

Note that you must know the password which originated this verifier.  
Opening a Secure Session requires the known password.

- In the **Password** field, type the password for the device. As you type, the **Salt** and **Verifier** fields are automatically calculated and populated in the dialog as shown above. This is the password that will be requested when you try to configure the device remotely.

### **Configure the secured features**

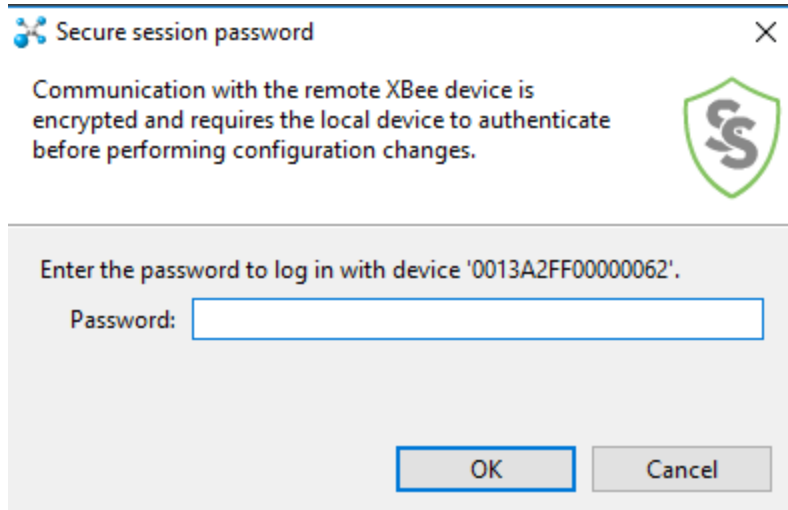
Secured access to the different features of the XBee device is enabled by setting bits of the **SA** (Secure Access Options):

- Bit 0 - Unused.
- Bit 1 - Remote AT Commands.
- Bit 2 - Over-the-air file system updates.
- Bit 3 - Over-the-air firmware updates.
- Bit 4 - Serial Data.

Set all other bits to 0.

### Start a secure session

XCTU detects automatically if a secure session needs to be established with a remote XBee device in order to configure or communicate with it. When this happens, XCTU displays the Secure session password dialog, where you have to enter the password configured in the remote XBee device you want to communicate with.



### Secure Remote Commands

XBee SX devices include a feature in the firmware called Secure Remote Commands that allows you to protect an XBee device on the network from external configurations.

To enable the secured remote configuration, set a password with the **KZ** command. To do so, type the old password (**0** by default), followed by a blank space and the new password. For example, if your password is **1234** and you want to change it to **7a\$b3l1**, you should type:

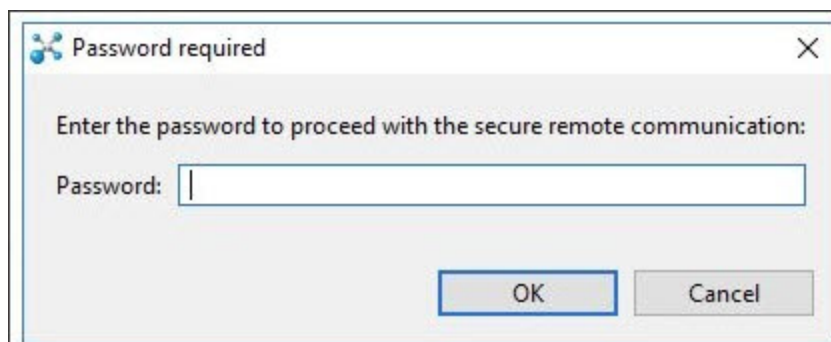
---

```
1234 7a$b3l1
```

---

After the password has been set, the **Password required** dialog appears when you try to configure a remote device and the password is set on your local one (**KZ** is not set to **0**). The same dialog appears if you perform a range or throughput test with a remote device.

You must enter the password and click **OK** before you can configure the device.




---

**Note** To learn more about this feature and how it works, refer to the user guide for your XBee RF module.

---

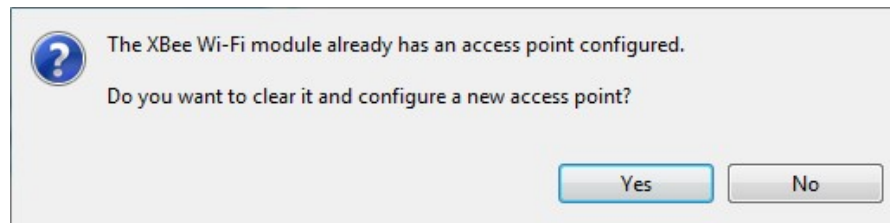
## Configure a Wi-Fi access point

The Active scan special function discovers and configures the access point for the XBee Wi-Fi module. The feature is only enabled for Wi-Fi modules.

1. Switch to Configuration working mode .
2. Select a Wi-Fi-enabled radio module from the device list.
3. Click the **Active scan** button. XCTU reads the SSID configuration of the Wi-Fi module.

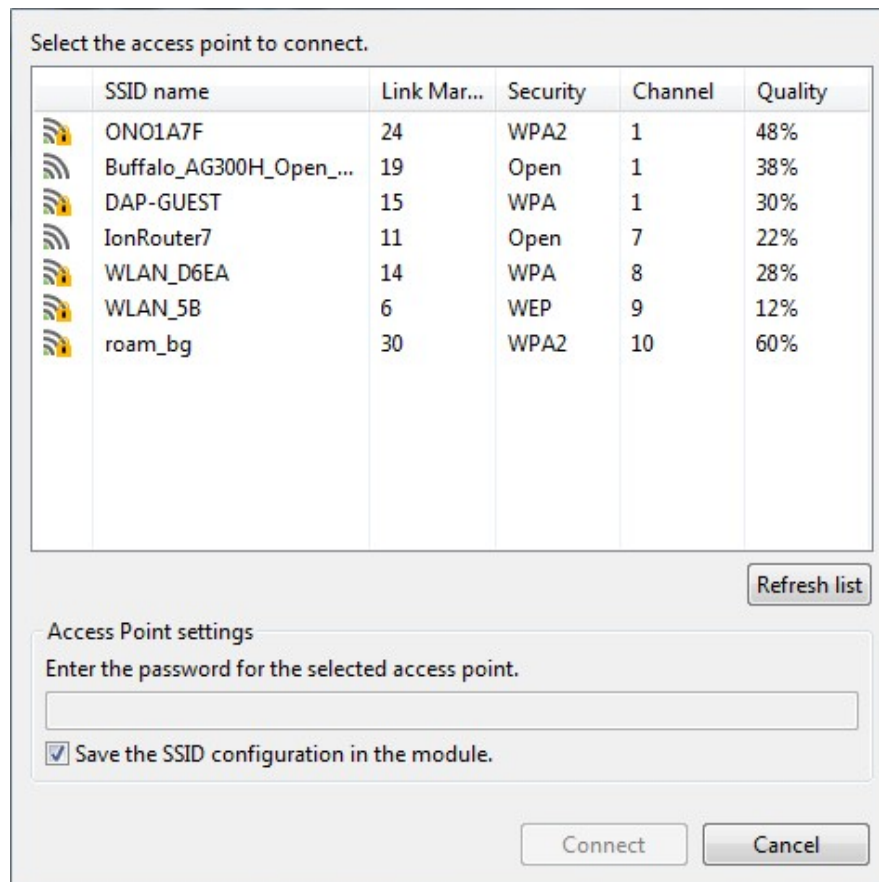


- If the module already has an SSID configured, click **Yes** to clear the configuration and perform a new SSID discovery.



- If the SSID configuration is empty, XCTU displays all nearby access points as well as

their security protocols and signal quality.



- Select the SSID you want the Wi-Fi module to connect to.  
For S6 Wi-Fi modules, the table displays the following fields:

Field	Description
SSID name	Name of the access point
RSSI (dBm)	RSSI of the access point (negated hex value)
Security	Security type of the access point
Quality	Link quality (based on the RSSI) with the access point

For S6B Wi-Fi modules, the table displays the following fields

Field	Description
SSID name	Name of the access point



Field	Description
Link margin (dBm)	Signal strength in dBm above sensitivity
Security	Security type of the access point
Channel	Channel number in use by the access point
Quality	Link quality (based on the link margin) of the access point


- If necessary, enter the password of the access point.
- If you would like to retain the SSID configuration for future use, check **Save the SSID configuration in the module**.
- Click **Connect** to connect the Wi-Fi module to that access point and refresh the settings of the radio module.

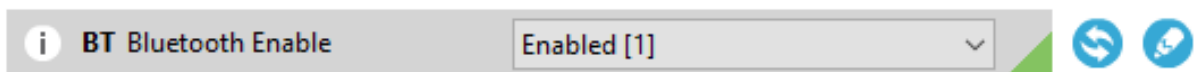
## Enable and configure Bluetooth


Some of the latest XBee3 modules support Bluetooth Low Energy (BLE) as an extra interface for configuration. If you want to use this feature, you have to enable BLE. You must also enable security by setting a password on the XBee device in order to connect, configure, or send data over BLE.

Before you begin, you should determine the password you want to use for BLE on the XBee device and store it in a secure place. Digi recommends a secure password of at least 8 characters and a random combination of letters, numbers, and special characters. Digi recommends using a security management tool such as LastPass or Keepass for generating and storing passwords for many devices.

The salt and verifier values are calculated when you set your password.

- Switch to Configuration working mode .
- Select a BLE compatible radio module from the device list.
- Select **Enabled[1]** from the **BT Bluetooth Enable** command drop-down.



- Click the **Write setting** button . The **Bluetooth authentication not set** dialog appears.
- Click **Configure** in the dialog. The **Configure Bluetooth Authentication** dialog appears.
- Type a password or a salt/verifier combination (as specified by the Secure Remote Password protocol—only for advanced users). When you connect the XBee device via BLE, it asks for this password (or the one that generated the verifier).

Basic configuration

**i** Password:

Advanced configuration

**i** Salt:

**i** Verifier:

Note that you must know the password which originated this verifier. Connecting to the XBee via BLE requires the known password.

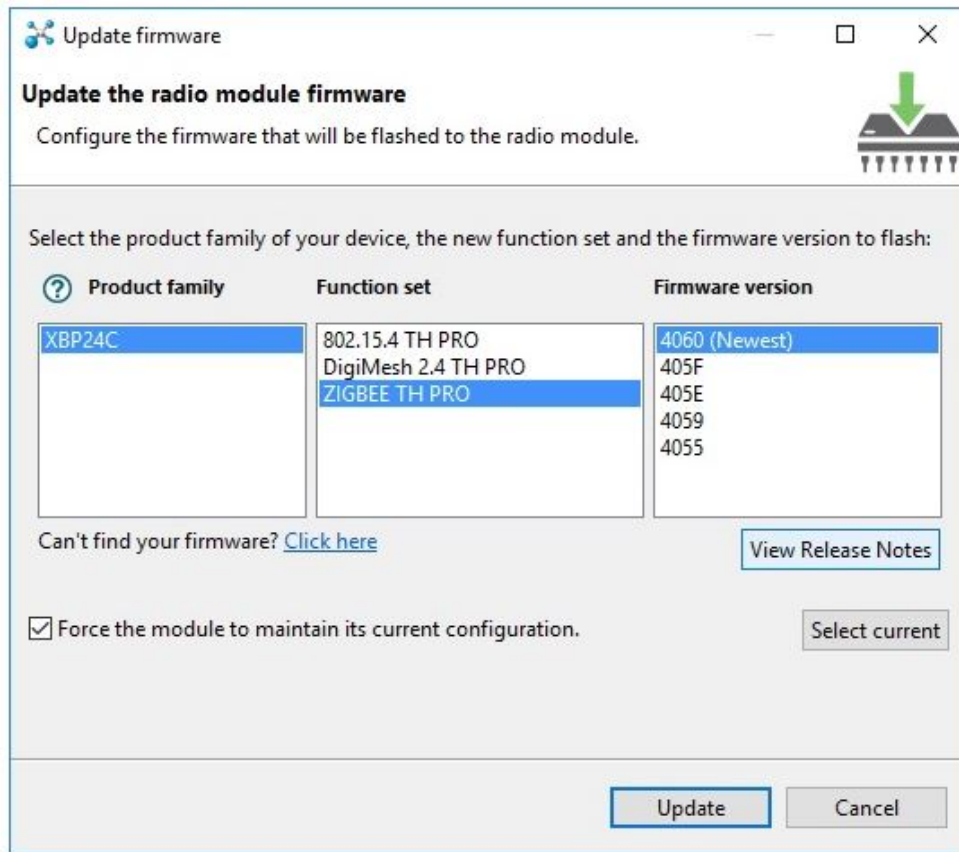
7. In the **Password** field, type the password for the device. As you type, the **Salt** and **Verifier** fields are automatically calculated and populated in the dialog as shown above. This password is used when you connect to this XBee device via BLE using the XBee Configurator app.

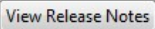
For more information about Bluetooth, see the user guide for your XBee device.

## View firmware release notes

XCTU allows you to review the release notes of some of the firmware it hosts.

1. Switch to Configuration working mode .
2. Select a radio module from the device list.
3. Click the **Update firmware** button . The **Update firmware** dialog appears, displaying the available compatible firmware for that module.



4. Select the firmware family, the firmware function, and the firmware version. If the selected firmware has release notes, the **View Release Notes** button  below the firmware list is enabled.
5. Click **View Release Notes**. A new window displays the release notes for the selected firmware.

---

**Note** You can also access the release notes of firmware via the [XBee recovery tool](#) and the [Firmware explorer tool](#).

---

## Communicate with your modules

---

This section describes how to use Consoles working mode to communicate with your modules once you have added a radio module or modules to your list of devices.

Consoles working mode .....	77
Connect and disconnect the console .....	79
Record a console session .....	80
Attach and detach the console .....	80
Communicate with modules running in API or API escaped mode .....	81
Communicate with modules running in AT mode .....	96
Console log files .....	104

## Consoles working mode



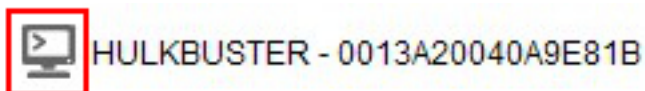
The Consoles working mode allows you to communicate with radio modules in the device list. When you click the **Consoles working mode** button on the toolbar, XCTU displays a tabbed list of consoles with one entry for each module of the devices list. Each tab is labeled with the name of the radio module and its physical address (MAC).

If you select a module from the device list, the associated console activates and moves to the front of the display. Conversely, selecting a console from the working area highlights the associated module in the device list.

The console type depends on the operating mode of the selected radio module.

- The API Console is used for radio modules working in API or API escaped operating mode. See [API console](#).
- The AT Console is used for radio modules working in AT (transparent) operating mode. See [AT console](#).

### Console status



The icon in the left corner of the console tab displays the following states:







Icon	Status
	The console is disconnected.
	The console is connected.
	The console is receiving data.
	The console is sending data.

The text of each tab contains the name of the radio module and its physical address (MAC). This makes it easy to identify the console corresponding to each radio module.

## Consoles toolbar

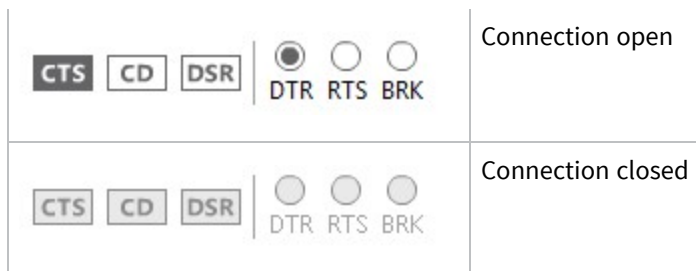
All consoles have a common toolbar that allows you to connect or disconnect the console and to attach or detach it from the working area.



Button	Name	Description
	Open	Establishes communication with the radio module corresponding to the console. When the console is connected, the background color of the button changes to green and the text changes to "Close" See <a href="#">Connect and disconnect the console</a> .
	Close	The console is connected. Clicking this button disconnects the module from the console. See <a href="#">Connect and disconnect the console</a> .
	Start recording	Records all incoming and outgoing console data as it is sent or received and saves to a console log file. The console must be connected to start or stop recording. See <a href="#">Record a console session</a> .
	Stop recording	Stops the process of saving any sent or received data. The console must be connected to start or stop recording. See <a href="#">Record a console session</a> .
	Detach	Detaches the console from the tabbed working area and displays it in a new floating dialog.
	Attach	Reattaches the console to the tabbed working area.

## Line status indicator

The line status indicator displays the status of the RS-232 hardware flow control lines. Dark gray indicates that the line is asserted, while white indicates that it is de-asserted. You must open the console connection to display line status and enable line status control.



You can view and manage the following lines from the line status indicator:

CTS	Clear to send	Indicates that the connected device is ready to accept data.
CD	Carrier Detect	Detects the presence of connection.
DSR	Data Set Ready	Indicates that the connected device is ready for communication.
DTR	Data Terminal Ready	Indicates that the terminal is ready for communication.
RTS	Ready to send	Requests that the connected device prepare to receive data.
BRK	Break	Engages the serial line break. Asserting this line places the DI line high, preventing data from being sent to the radio.

### Console overview

The console overview panel, located on the right side of the toolbar, displays the number of sent and received frames or bytes.

**Tx Frames:** 0


**Rx Frames:** 0

## Connect and disconnect the console


The first time you open a console, it is disconnected by default.

1. Switch to Consoles working mode.



2. Select a radio module from the list.
3. Click the **Open** button  to establish communication with the radio module corresponding to the console. The background color of the **Open** button changes to green and its text changes to **Close**.

When the console is connected, all the data traffic of the radio module is captured by the console and displayed in the corresponding controls.

4. To disconnect the console from the module, click the **Close** button . The background color of the button changes to white and its text changes to **Open**.





## Record a console session

In the Consoles working mode, you can record all incoming and outgoing console data as it is sent or received. XCTU saves this data in a Console log file. The console must be connected to start or stop recording.

---

**Note** You can also use the Serial console tool to record console sessions. See [Serial console tool](#).

---

1. Switch to Consoles working mode .
2. Select a radio module from the list.
3. Click the **Open** button  to establish communication. The background color of the button changes to green.
4. Click the **Start Recording** button . A **Save file** dialog appears.
5. Specify a destination for the console log file. Data is periodically written to this log file in comma separated values (CSV) format. See [Console log files](#).
6. Click the **Stop Recording** button  to stop the process of saving sent or received data.


---

**Note** You can use the Load console session tool to load a saved console session at a later time. For details, see [Load console session tool](#).

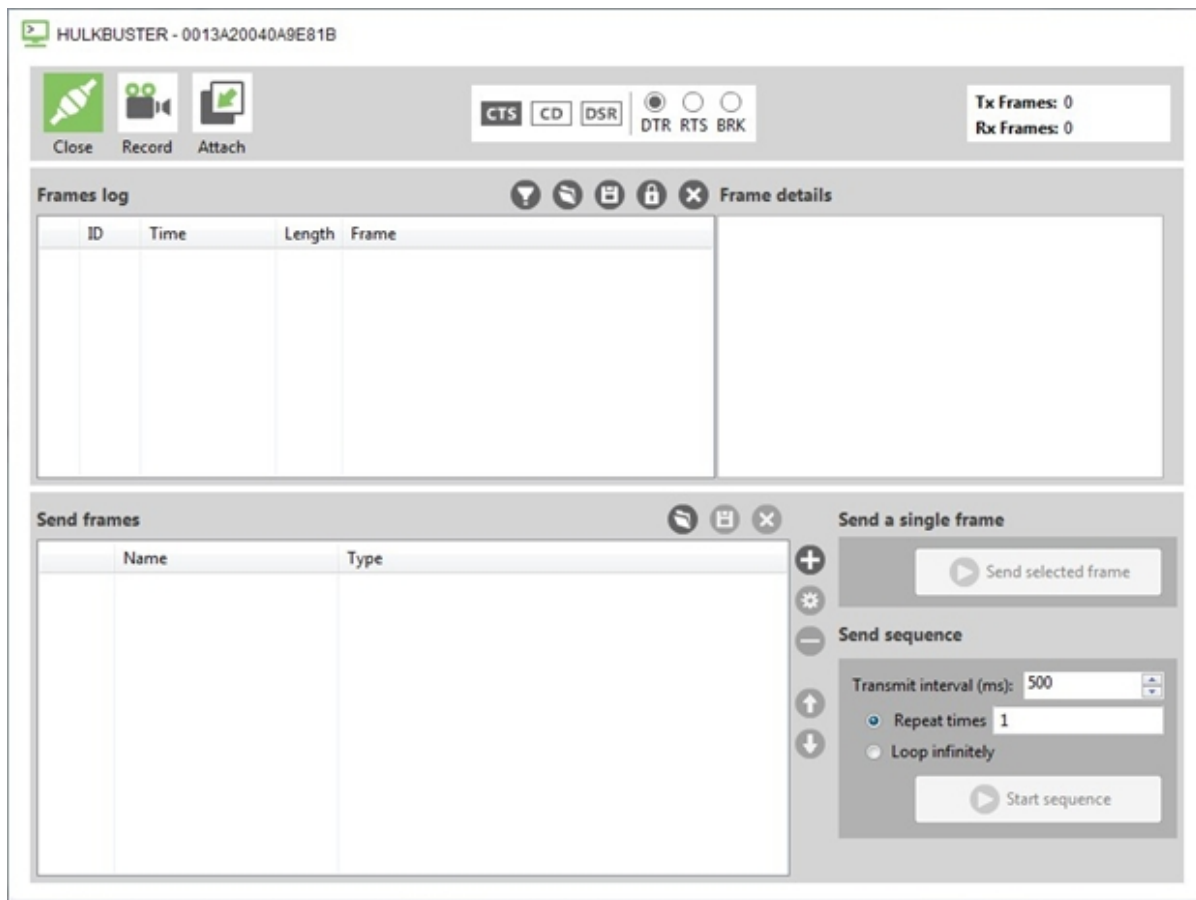
---

## Attach and detach the console

You can attach or detach the console from the working area by clicking the Attach and Detach buttons on the Console toolbar. By default, all the consoles are attached and sorted in tabs.

1. To detach a console from the working area, click the **Detach** button . The console is displayed in a floating dialog.






You can detach all the consoles to display them in multiple dialogs. This is useful if you need to see the traffic for different radio modules simultaneously.

2. To reattach the console view, click the **Attach** button  or close the floating dialog containing that console.

## Communicate with modules running in API or API escaped mode

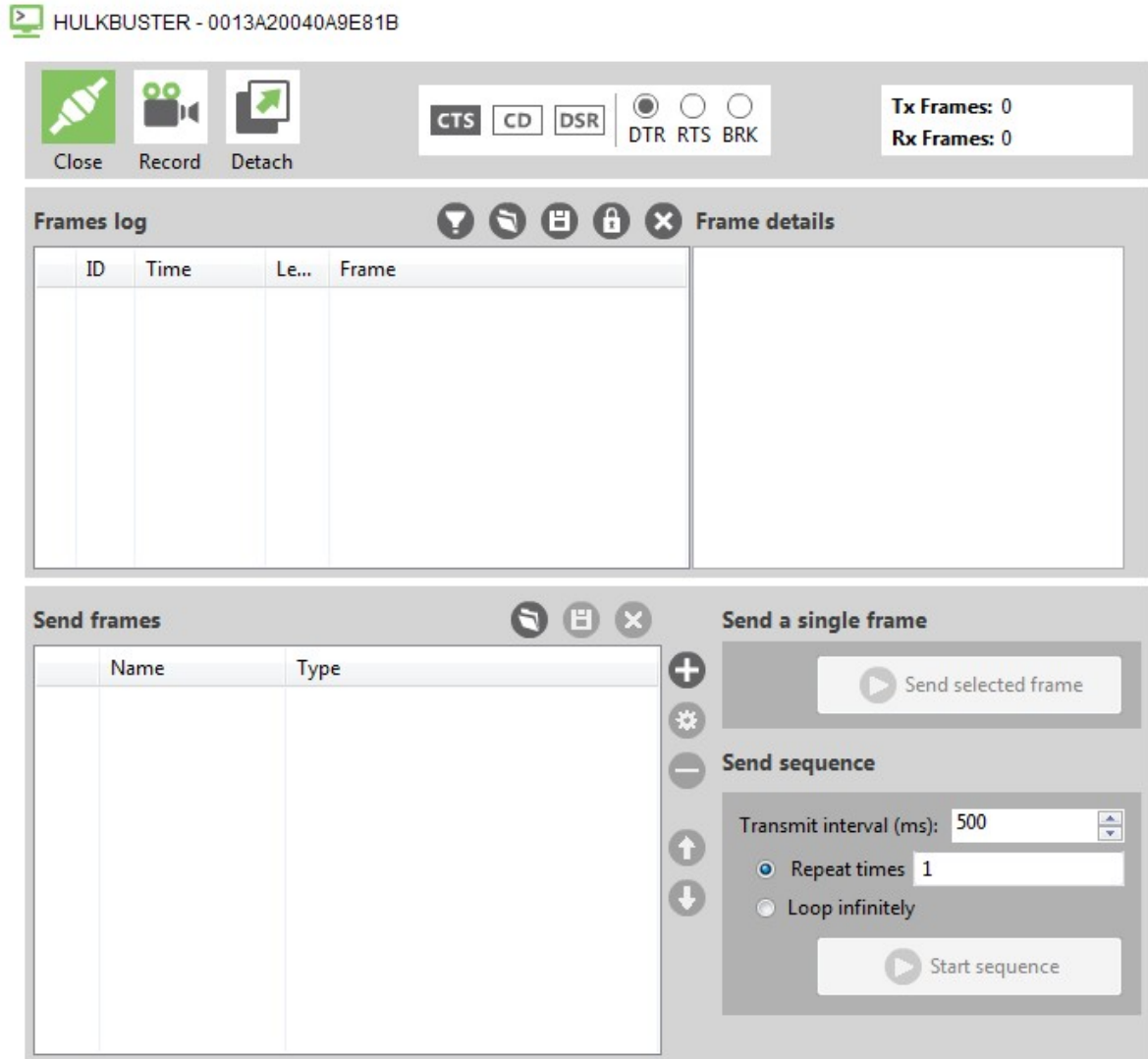
XCTU uses API frames to communicate with radio modules running in API or API escaped mode. You can use XCTU's API console to communicate with radio modules running in API or API escaped operating mode. XCTU automatically creates an API console for these modules when you switch to Consoles working mode.

### API console

The API console allows you to communicate with radio modules running in API or API escaped operating mode. To display the API console, switch to Consoles working mode  and select a radio module configured with either API or API escaped operating mode. For more information, see [API operating mode](#) and [API escaped operating mode](#).

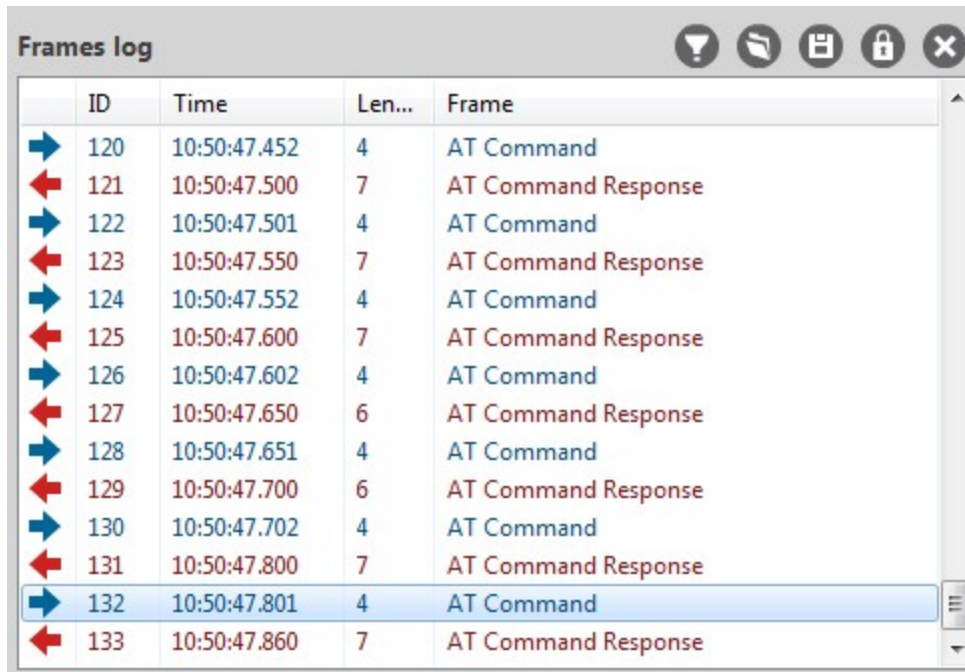
XCTU uses API frames to communicate with radio modules running in API mode. An API frame is an array of bytes with a specific structure defined by the API Frame Specifications. For more information, see [API frames](#).

### API console view



### Frames log

The **Frames log** is XCTU's API frames traffic monitor. When API frames are sent or received by the module, they are added to the **Frames log**. Depending on whether the frame is sent or received, the color of the frame fields is blue or red, respectively.



Button	Name	Description
	Configure filters	Opens the <b>Frames filtering configuration</b> dialog.
	Load console session	Opens the embedded <b>Load console session</b> tool. For details, see <a href="#">Load console session tool</a> .
	Save console session	Saves the active console session in a <a href="#">Console log files</a> .
	Lock scroll	Toggles the scroll behavior of the data box. If <b>Lock scroll</b> is enabled, the Frames log will not automatically scroll when a new frame is sent or received.
	Clear data	Clears the list of sent and received API frames.

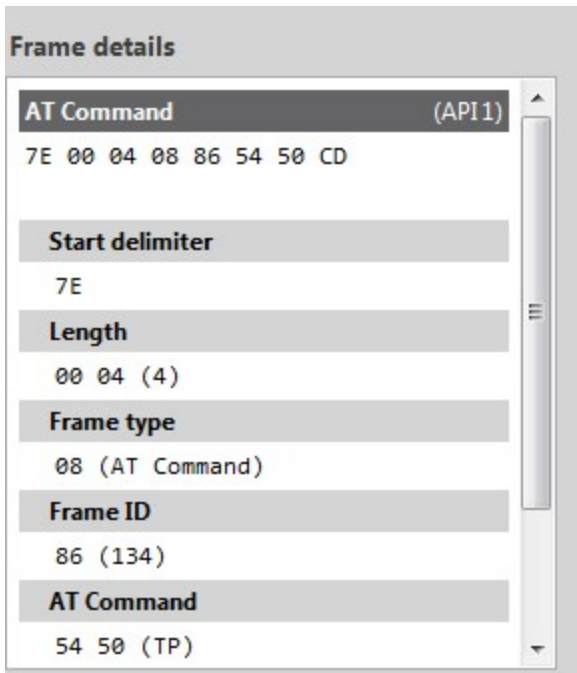
The **Frames log** displays the following properties of the transmitted and received API frames:

Field	Description
Icon	Displays as a right blue arrow if the API frame is sent, and as a left red arrow if it is received.
ID	Lists a numeric value given by XCTU to identify the API frame. When the list of frames is

Field	Description
	cleared, this value is reset to 0.
Time	Displays the time at which the API frame was sent or received.
Length	Contains the number of bytes of the API frame.
Frame	Displays the API frame type.



### Frame details







Frame details appear next to the API frames table and display the decoded contents of the selected API frame.



### Send frames

The **Send frames** section of the **API Console** allows you to add, manage, and send an API frame or sequence of frames.

Button	Name	Description
	Add new frame	Adds a new API frame to the list of frames to send
	Edit selected frame	Changes the name or contents of a frame

Button	Name	Description
	Remove selected frame	Removes a frame from the list
 	Move up / move down	Changes the order of frames in the list
	Save frames list	Saves the list of frames to be used in future sessions or on another computer
	Load frames list	Loads a saved list of frames
	Clear the list of frames	Clears the entire frames list


The **Send sequence** dialog contains the following fields:


Field	Description
Transmit interval	Amount of time, in milliseconds, to pause between frames. The minimum value is 0 ms and the maximum value is 60000 ms (1 minute).
Repeat times	Number of times the sequence should be repeated or sent. By default, this value is 1.
Loop infinitely	Sends the sequence of frames in an infinite loop.

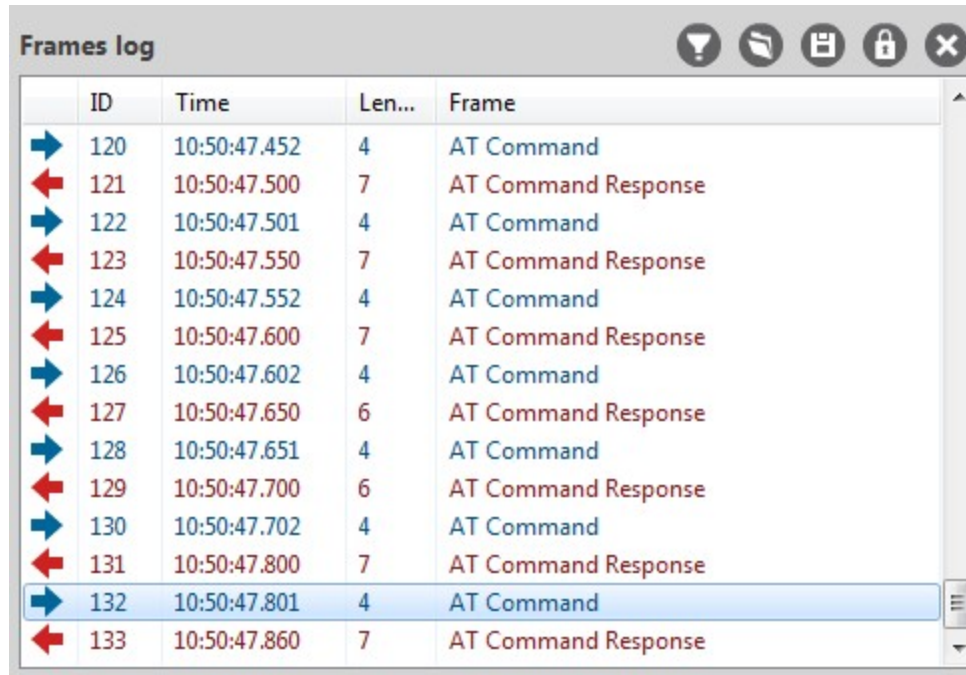
**Note** You can send a single frame (**Send selected frame**) or the configured sequence of frames (**Start sequence**).

### Filter sent and received frames

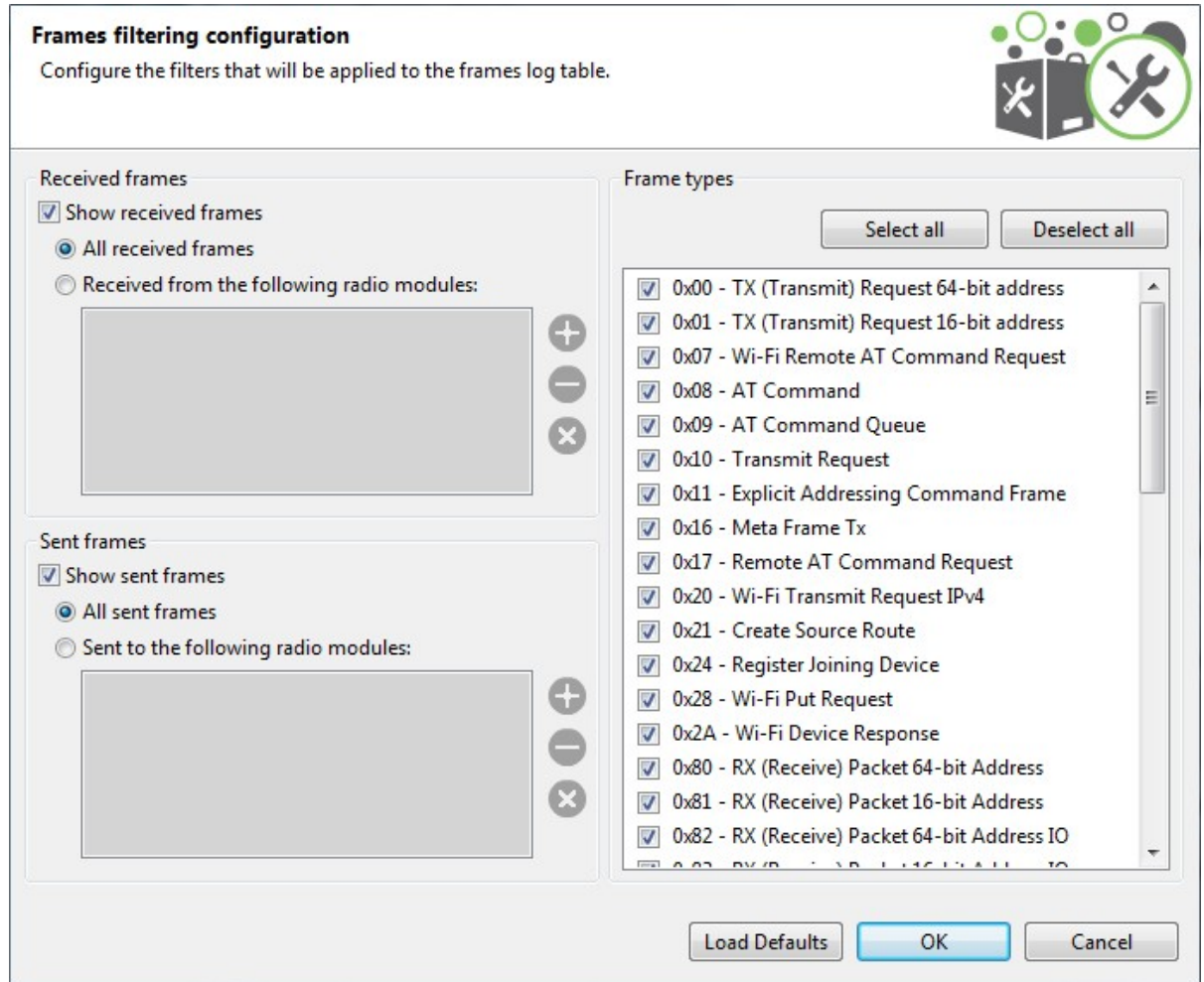
You can filter by sent and received frames in the API console to focus only on specific frames. For both sent and received frames, you can show any, show all, filter by sender MAC, and filter by frame type.

1. Switch to Consoles working mode .
2. Select a radio module configured in API or API escaped operating mode.

3. Click the **Configure filters** button  in the **Frames log** area.



The **Frames filtering configuration** dialog appears.



4. Configure the filtering options for one or more of the following: received frames, sent frames, and frame types.

### Received frames

When filtering the received frames, you have the following options:

Option	Description
Show received frames	Shows/hides received frames
All received frames	Shows all the received frames
Received from the following radio modules	Shows the received frames by source address and lists the source MAC addresses that can be filtered. You can add and remove addresses from this list at any time.

To add a new MAC address:

1. Select the **Received from the following radio modules** option.
2. Click the **Add new item** button. A new dialog appears.
3. If any remote devices have been discovered for your local device, you can select the specific device from the combo box:

If no remote devices are discovered for the local device, you can enter the MAC address manually:

4. Click the **Add** button to add the MAC address to the list:

### Sent frames

When filtering the sent frames, you have the following options:

Option	Description
Show sent frames	Shows/hides sent frames
All sent frames	Shows all the sent frames



Option	Description
Sent to the following radio modules	Shows the sent frames by destination addresses and lists the source MAC addresses that can be filtered.

**Note** You can add a MAC address to the list by following the steps in [Received frames](#).

### Frame types



When filtering frames by type, you can select or deselect the desired frame types. By default all frame types but the Invalid frame are selected. You can also use the following buttons:

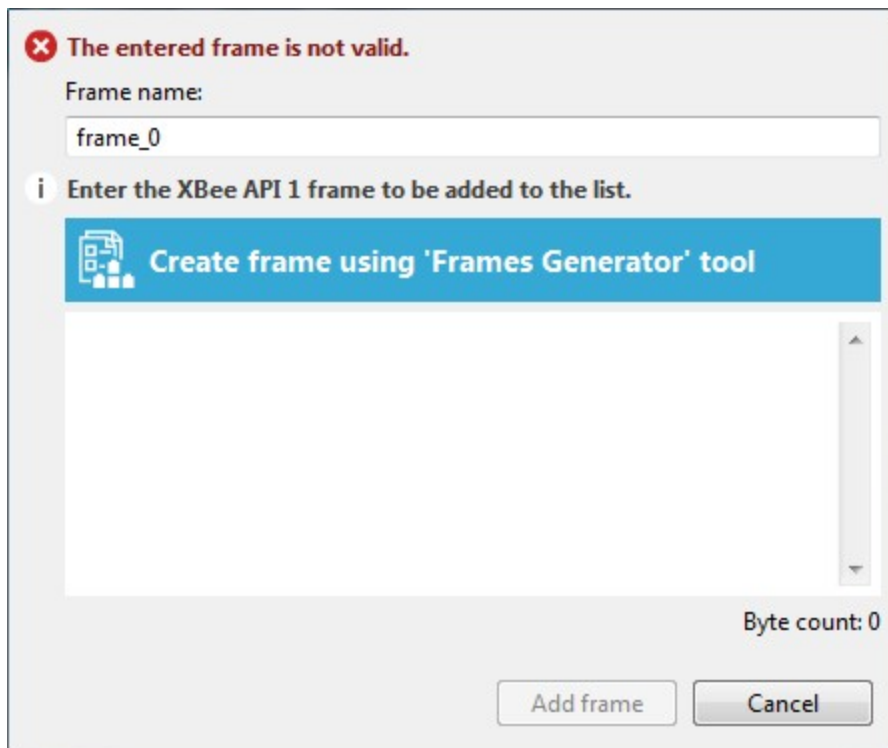
Option	Description
Select all	Checks all the frame types
Deselect all	Unchecks all the frame types

### Add an API frame

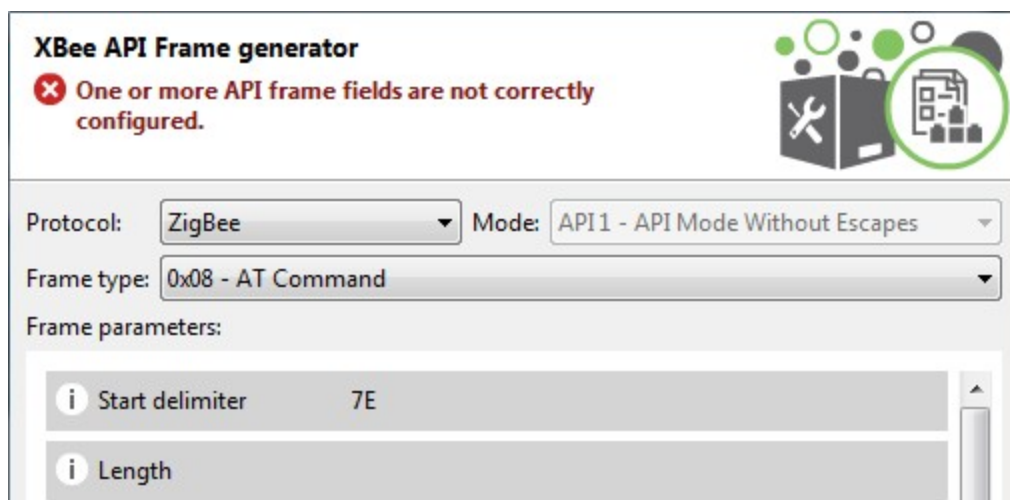
An API frame is the structured data sent and received through the serial interface of the radio module when it is configured in API or API escaped operating modes. API frames are used to communicate with the module or with other modules in the network. For more information, see [API frame](#).

If you want to send an API frame either individually or as part of a series, you first need to create the API frame:

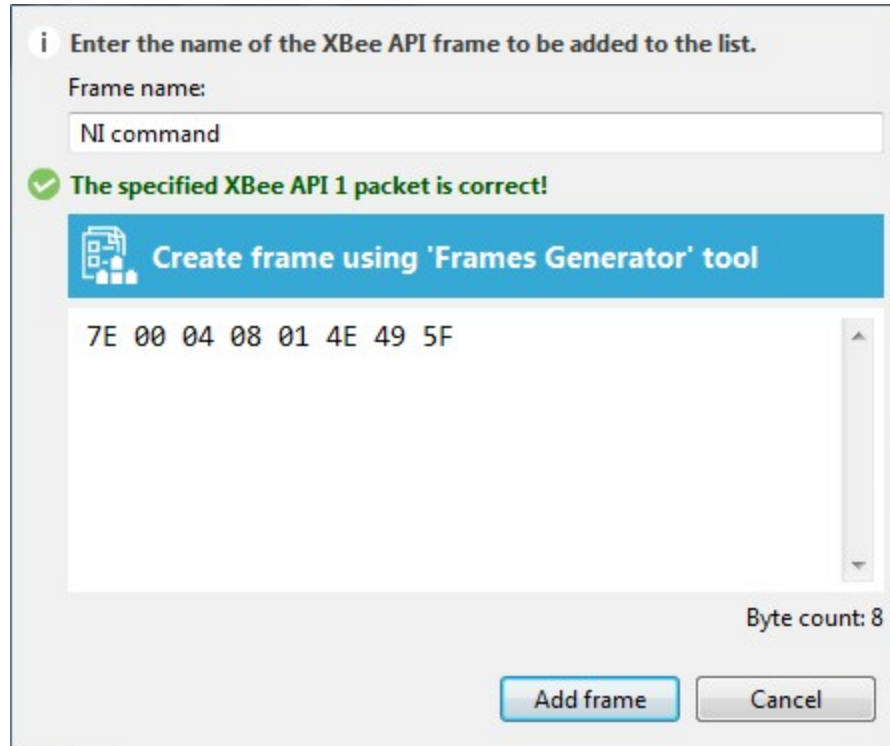
1. Switch to Consoles working mode .
2. Select a radio module from the list.
3. In the **Send frames** area, click the **Add new frame to the list** button . The **Add API frame to the list** dialog appears.



4. Enter a name for the frame.
5. Create the frame by doing one of the following:
  - Enter the byte array of the API frame.
  - Click the **Create frame using 'Frames Generator' tool** button to open the Frames Generator tool. Then, configure the frame and click **OK**. For more information, see [Frames generator tool](#).



Your generated frame appears in the API frame.



6. Click the **Add frame** button to add the API frame to the list of frames to send.

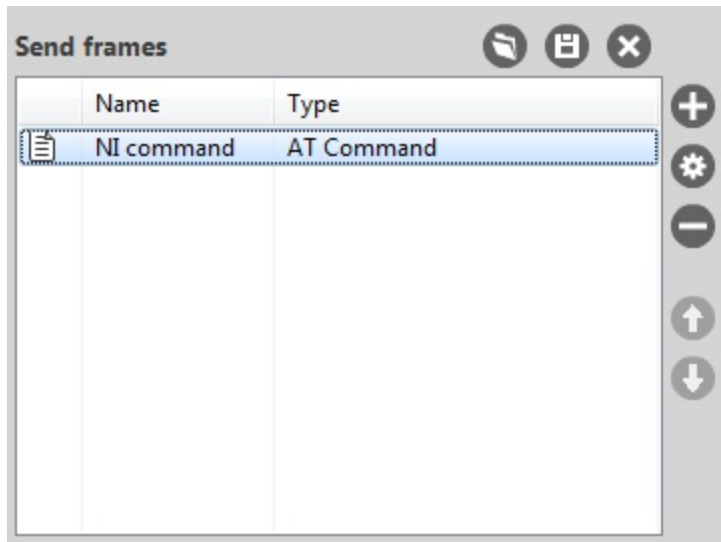
	Name	Type
	NI command	AT Command

7. Repeat the steps to add additional API frames.

When finished, you can [Send a single API frame](#) and [Send a sequence of API frames](#).

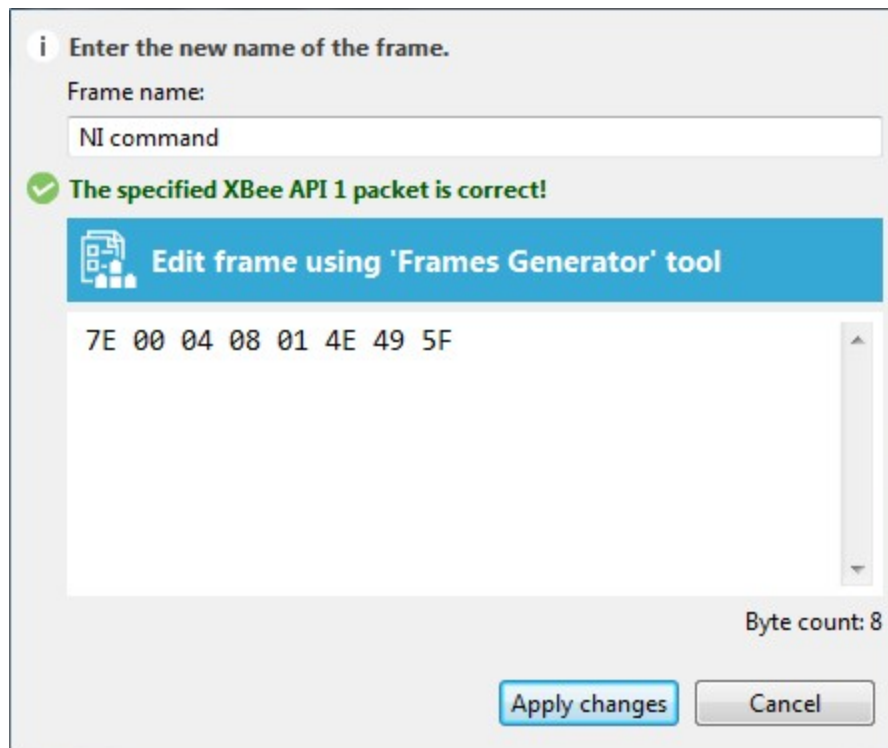
### Manage API frames

Once you add an API frame to the list, you can perform the following actions. For instructions on adding a frame, see [Add an API frame](#).



### Edit a frame


1. Select the frame and click the **Edit selected frame** button .





2. Do one of the following:
  - Manually edit the frame name and content.
  - Click **Edit frame using 'Frames Generator' tool** to edit the content of the frame within the tool.

3. Click **Apply changes**.

### **Remove a frame**


To remove a frame from the frames list, select the frame and click the **Remove selected frame** button .

### **Change the order of the frames list**

If your list of API frames contains more than one frame, you can change the order of the frames. Select an API frame and then click **Move up**  or **Move down**  to change the position of the frame in the list.

### **Save a list of frames**

You can save a list of frames for use in future XCTU sessions.

1. Click the **Save frames list** button . A **Save file** dialog box appears.
2. Specify a name and path for the API frames list XML file, and click **Save**.

---

**Note** XCTU also saves the sending configuration.

---

### **Load a saved list of frames**

You can load a previously saved list of API frames.


1. Click the **Load frames list** button . A **Load file** dialog box appears.
2. Select the saved API frame list XML file and click **Open**.

---

**Note** XCTU also loads the sending configuration.

---

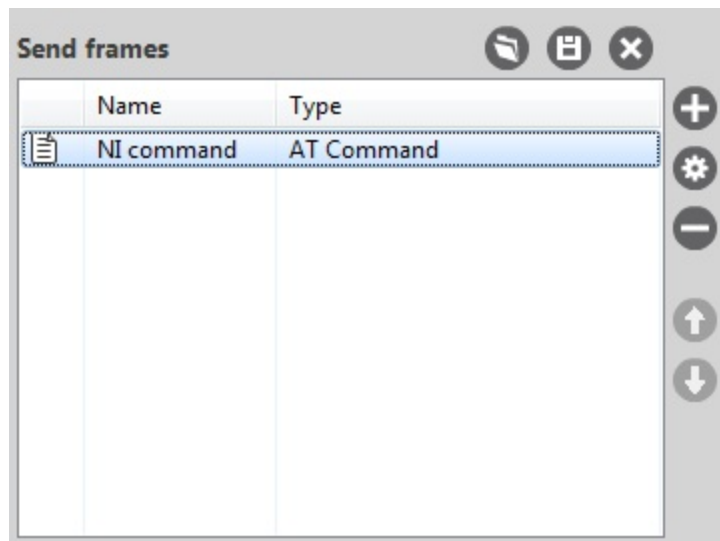
### **Clear the frames list**

To clear the list of API frames, click the **Clear list** button .

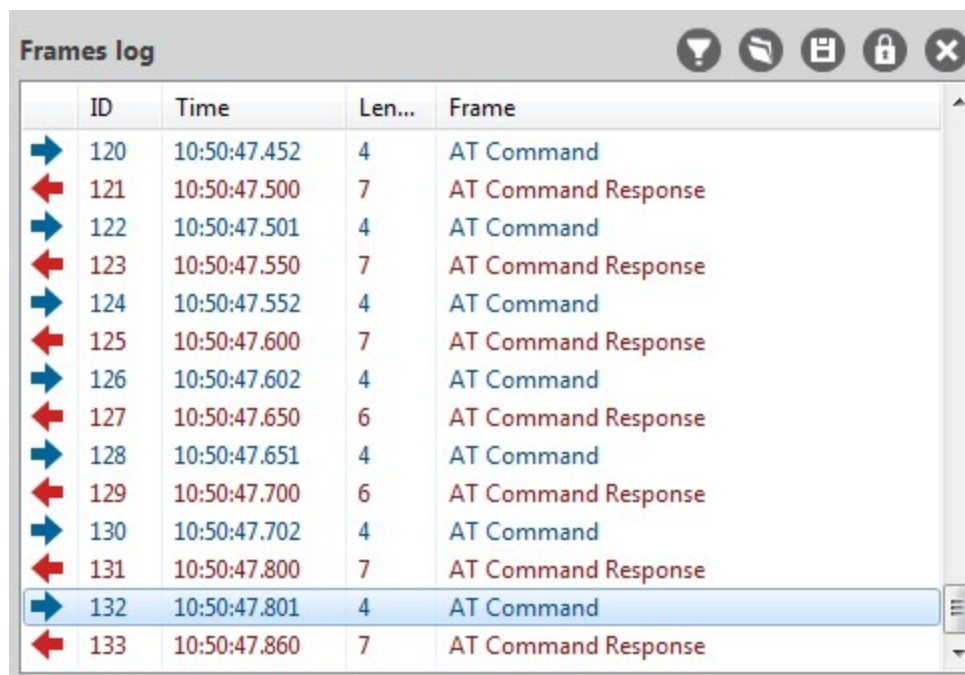
### **Send a single API frame**

After you have added at least one frame to the Send frames area, you can send a single API frame to the console's corresponding radio module. For instructions on adding a frame, see [Add an API frame](#).

1. In the API Console, select a frame in the **Send frames** area.



2. Click **Send selected frame**. The sent frame appears in the **Frames log** area.



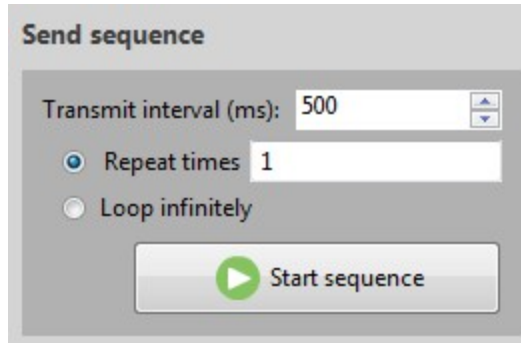
For more information, see [Send a sequence of API frames](#).

## Send a sequence of API frames

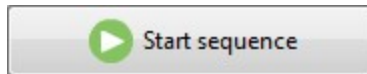
After you have added at least two API frames to the Send frames area, you can send a sequence of API frames to a radio module running in API mode. For instructions on adding a frame, see [Add an API frame](#).

An API frame sequence contains the list of frames to send as well as configured information about the way the frames will be sent.

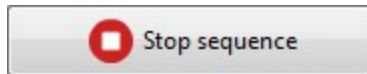
- In the **Send sequence** area of the API console, configure the following options:
  - **Transmit interval:** Specify the amount of time, in milliseconds, to pause between frames. The minimum value is 0ms and the maximum value is 60000ms (1 minute).
  - **Repeat times:** Select this option to specify the number of times the sequence should be repeated or sent. By default, this value is 1.
  - **Loop infinitely:** Select this option to send the sequence of frames in an infinite loop.



- Click **Start sequence**. XCTU sends all frames in the **Send frames** list. Sent and received frames appear in the **Frame logs** area.




- The process could end by itself, but you can click **Stop sequence** at any time to halt transmission.



For more information, see [Send a single API frame](#).

## Save an API console session

You can save the API console session, containing the list of sent and received API frames, by clicking the Save console session button.

- Switch to Consoles working mode .
- Select a radio module running in API or API escaped operating mode from the list.
- Connect the console.
- Generate some traffic.
- Click the **Save console session** button in the **Frames log** view. The **Save file** dialog appears.
- Designate a filename and path for the console session file.
- Click **Save**.

---

**Note** You can load an API console session using the Load console session tool. See [Load console session tool](#).

---

## Communicate with modules running in AT mode

In AT (Application Transparent) or transparent operating mode, all serial data received by the radio module is queued up for RF transmission. When RF data is received by the module, the data is sent out through the serial interface.

You can use XCTU's AT console to communicate with radio modules running in AT operating mode. XCTU automatically creates an AT console for these modules when you switch to Consoles working mode.

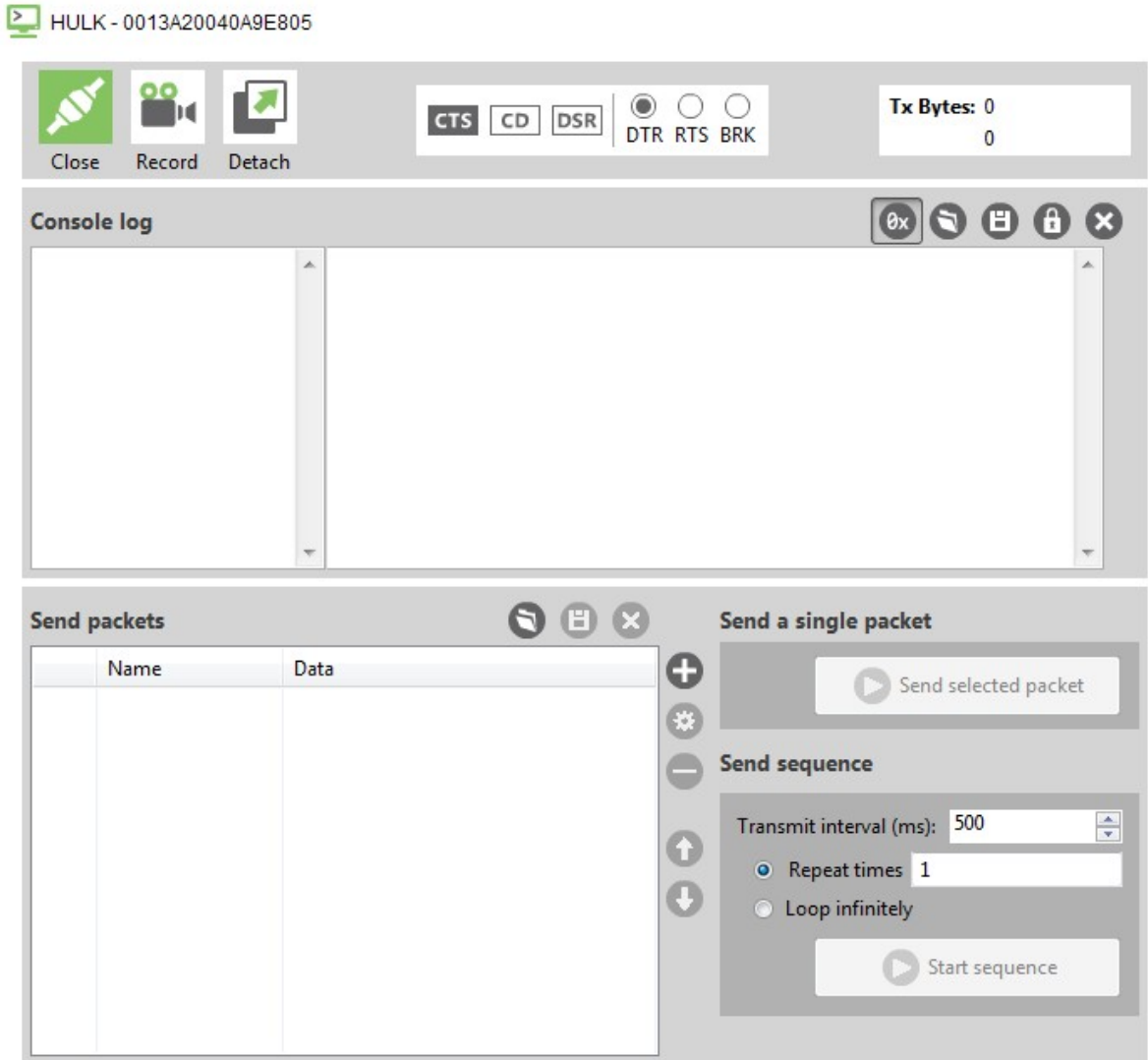
### AT console

The AT console allows you to communicate with radio modules running in AT (transparent) operating mode. To display the AT console, switch to Consoles working mode and select a radio module configured with AT operating mode.

In the AT console, communication with the device is direct. All the data you send through the serial interface is queued for transmission by the module, and all data received by the module is sent through the serial interface.

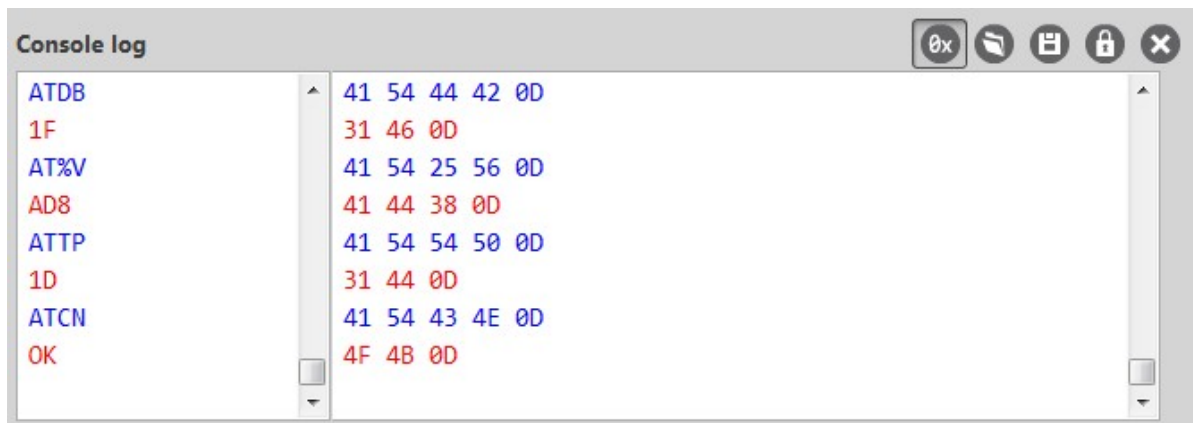
AT console view





### Console log

The Console log operates as a data traffic monitor. It displays all sent (blue) and received (red) data characters. Sent and received data is appended at the bottom of the data box. The right-hand side of the Console log displays corresponding hexadecimal values for all sent and received data characters.










Button	Name	Description
	Show/hide hexadecimal	Toggles hexadecimal view.
	Load console session	Opens the embedded Load console sessions tool. See <a href="#">Load console session tool</a> .
	Save console session	Saves the active console session in a <a href="#">Console log files</a> .
	Lock scroll	Toggles the scroll behavior of the data box. If Lock scroll is enabled, the data box will not automatically scroll when a new data character is sent or received.
	Clear data	Clears all data characters, as well as their hexadecimal representation, from the data box.

When operating in AT operating mode, XCTU does not require that you use structured data to communicate with a radio module. Communication with the device is direct. Type directly in the data box and the characters you enter are automatically sent to the radio module.

**Note** To insert CR/LF, CR or LF characters, right click on the ASCII panel of the Console log and select the appropriate option from the context menu. It is also possible to configure the default behavior when you press the ENTER key. For more information about configuring default behavior, see [Set console preferences](#).

### Send packets

The **Send packets** section of the **AT Console** allows you to add and manage data packets. You can send a data packet—one or more groups of characters—to a radio module in your device list. When you type in the data box, XCTU sends the data as individual characters. Sending a data packet sends all of the characters in a single operation.

Button	Name	Description
	Add new packet	Adds a packet to the list of packets to send
	Edit selected packet	Changes the name or contents of a data packet
	Remove selected packet	Removes a data packet from the list
	Move up / move down	Changes the order of packets in the list
	Save packets list	Saves the list of packets to be used in future sessions or on another computer
	Load packets list	Loads a saved list of data packets
	Clear the list of packets	Clears the entire packet list

The **Send sequence** dialog contains the following fields:

Field	Description
Transmit interval	Amount of time, in milliseconds, to pause between data packets. The minimum value is 0ms and the maximum value is 60000ms (1 minute).
Repeat times	Number of times the sequence should be repeated or sent. By default, this value is 1.
Loop infinitely	Sends the sequence of data packets in an infinite loop.

**Note** You can send a single packet (**Send selected packet**) or the configured sequence of packets (**Start sequence**).

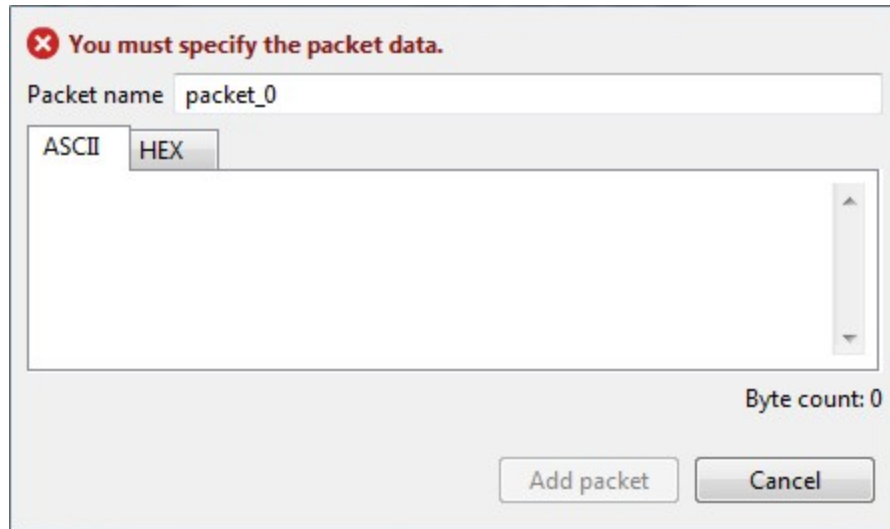
For more information, see [Consoles working mode](#) and [AT operating mode](#).

## Add a data packet

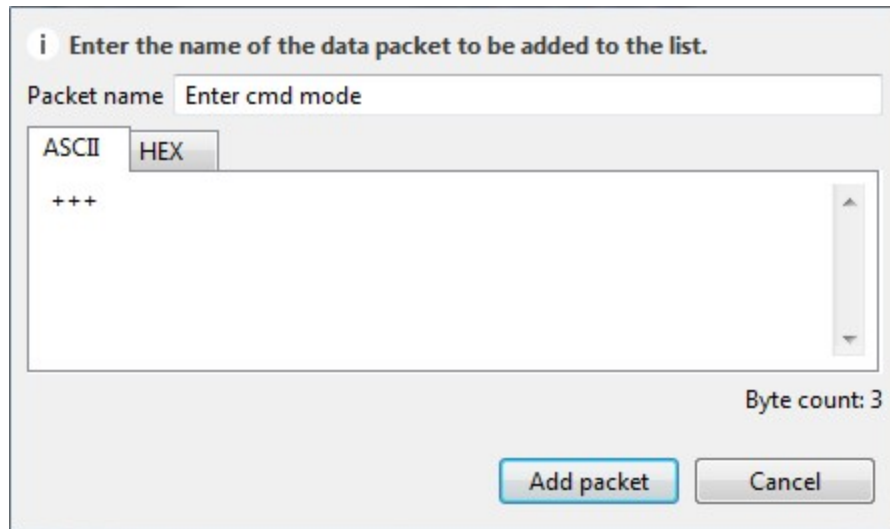
When a module is operating in AT mode, you can send data byte by byte, or you can send a set of bytes—a data packet. To send a data packet, you must first create one. By default, the data packets

list is empty.

1. From the AT console, click the **Add new packet**  button in the **Send packets** dialog. The **Add new packet** dialog appears.




2. Enter the name of the data packet to be added to the list.
3. Type in your data. You can use the tabs to toggle between ASCII and HEX input views.



4. Click **Add packet**. The data packet appears in the list of packets. Repeat the operation to add

additional packets to the list.


	Name	Data
	Enter cmd mode	+++

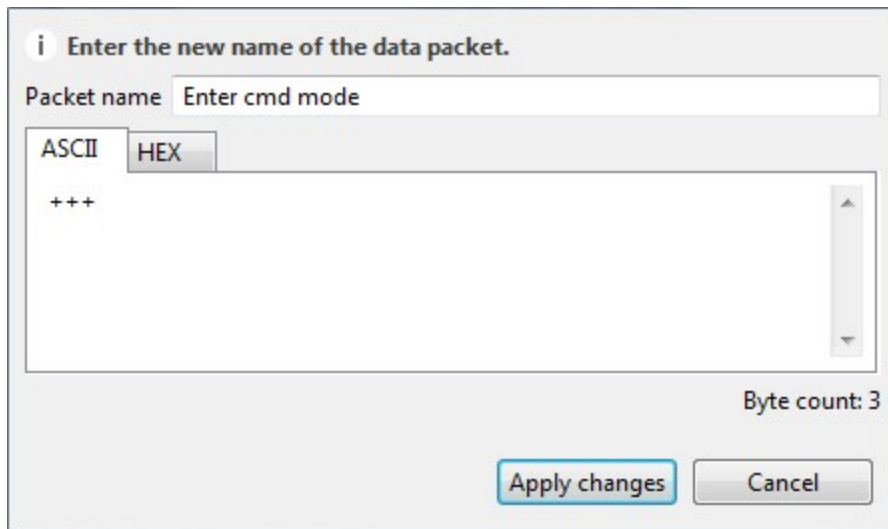
When finished, you can send a single packet or a sequence of packets. See [Send a single packet](#) and [Send a sequence of packets](#).

### Manage data packets

Once you create a data packet, you can perform the following actions. For instructions on adding a packet, see [Add a data packet](#).


#### ***Edit a data packet***

1. Select the packet and click the **Edit selected packet** button .
2. Manually edit the packet name and content.





3. Click **Apply changes**.

### **Remove a data packet**


To remove a data packet from the list, select the packet and click the **Remove selected packet** button .

### **Change the order of the data packets list**

If your list of data packets contains more than one packet, you can select a packet and then click **Move up**  or **Move down**  to change the position of the packet in the list.

### **Save a list of packets**

You can save a list of packets for use in future XCTU sessions.

1. Click the **Save packets list** button . A Save file dialog box appears.
2. Specify a name and path for the XML file and click **Save**.

---

**Note** XCTU also saves the sending configuration.

---

### **Load a saved list of packets**

You can load a previously saved list of data packets.


1. Click the **Load packets list** button . The Load file dialog box appears.
2. Select the saved XML file and click **Open**.

---

**Note** XCTU also loads the sending configuration.

---


### **Clear the packet list**

To clear the list of data packets, click the **Clear list** button .

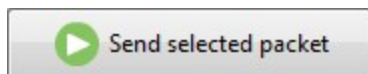
### **Send a single packet**

After you have added at least one data packet to the Send packets area, you can send a single packet to the console's corresponding radio module. For instructions on adding a packet, see [Add a data packet](#).

1. In the Consoles working mode, select a packet in the **Send packets** area.

	Name	Data
	Enter cmd mode	+++

2. Click **Send selected packet**.



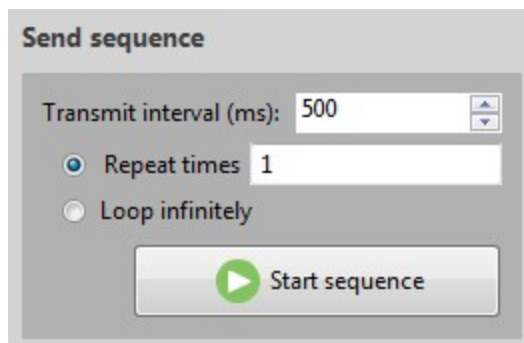
The sent data appears in the Data traffic section.

For more information, see [Send a sequence of packets](#).

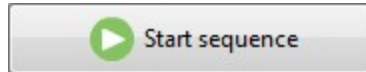
## Send a sequence of packets

After you have added at least two data packets to the Send packets area, you can send a sequence of data packets to a radio module running in AT mode. The sequence is defined by the list of packets to send and the send sequence options located next to the list. For instructions on adding packets, see [Add a data packet](#).

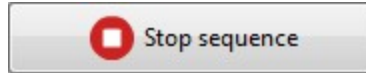
1. In the **Send sequence** dialog of the AT Console, configure the following options:
  - **Transmit interval:** Specify the amount of time, in milliseconds, to pause between frames. The minimum value is 0ms and the maximum value is 60000ms (1 minute).
  - **Repeat times:** Select this option to specify the number of times the sequence should be repeated or sent. By default, this value is 1.
  - **Loop infinitely:** Select this option to send the sequence of frames in an infinite loop.



2. Click **Start sequence**. The sent and received data appear in the **Data traffic monitor**.




3. The process can end by itself. You can also click **Stop sequence** at any time to halt transmission.



For more information, see [Send a single packet](#).

## Save an AT console session

A saved AT console session contains sent and received data. You can use the AT console and Serial console tool to save a console session. XCTU saves this data in a Console log file.

1. Switch to Consoles working mode  .
2. Select a radio module running in AT (transparent) operating mode.
3. Connect the console.
4. Generate some traffic. (The save feature doesn't work if there is no data to save.)
5. Click the **Save console session** button. The **Save file** dialog appears.
6. Designate a filename and path for the console session file.
7. Click **Save**.

---

**Note** You can load an AT console session using the Load console session tool. For more information, see [Load console session tool](#).

---

## Console log files

A console log file is the file generated when saving or recording a console session. It consists of a record containing console session information followed by multiple data records with sent and received data. XCTU writes data in a console log file using Comma Separated Value (CSV) format.

### Console session records

A console session record contains the following fields:

Field	Description
Date	Date and time when the session record started
ID	Not used in this record; this field contains a "-" character
Type	API, API2, AT, or Serial
Description	Comma-separated console session information such as the module node identifier, MAC address, function set, firmware version, and COM port



Console session records look like this:

---

```
02-05-2015 11:44:56.017,-,API,"A,0013A2004031A8D7,ZigBee Router
API,23A7,COM4 - 9600/8/N/1/N,0"
```

---

## Data records

Data records contain the following fields:

Field	Description
Date	Date and time when the data was sent or received
ID	ID number of the packet or API frame
Type	SENT or RECV
Data	Data sent or received in Hexadecimal string format

Data records look like this:

---

```
02-05-2015 11:44:58.857,0,SENT,7E000408764944F4
```

---

## View your radio network

---

This section describes how to use Network working mode to view your radio network once you have added a radio module or modules to your list of devices.

---



Modules in AT (transparent) operating mode do not support the network discovery process.

---

Network working mode .....	107
Scan the network for available modules .....	114
Search for network nodes .....	114
Change network perspective .....	114
Set network layout .....	115
Filter network connections by quality .....	115
Take a screenshot of the network .....	116
Set zoom level in Graph view .....	117
Export a network table .....	117

## Network working mode












The Network working mode allows you to discover and see the topology of your network.





The Network perspective only works in API operating mode. Radio modules in AT (transparent) mode do not support the network discovery process.


### Network toolbar

The Network toolbar presents the network actions you can perform.

Button	Name	Description
	Start scan	Scans the radio module's network and displays a list of available modules.
	Stop scan	When scans specified in scan preferences are finished, the discovery process stops automatically. You can also click the <b>Stop scan</b> button at any time.
	Graph view	Displays the modules as nodes in a graph.
	Table view	Displays the modules as rows in a table.
	Screenshot	Saves an image of the network to your computer.
	Export table	Exports network information in a tabular file format.
	Settings	Configures network-related parameters.
	Layout	Changes the network layout to one of six types: composite, spring, vertical tree, horizontal tree, grid (default), and radial.
	Filter links	Hides or displays connections between RF modules based on their quality. Enables/disables quality ranges via radio buttons. All filters are enabled by default so all connections are visible. You can establish connection quality colors and ranges in <b>Preferences &gt; Network &gt; Appearance</b> or by clicking <b>Configure quality ranges</b> link in the <b>Filter links</b> dialog.

Button	Name	Description
		See <a href="#">Set network discovery preferences</a> .
	Zoom	Changes the zoom level of the network.
	Find box	Searches for a particular module in the network.

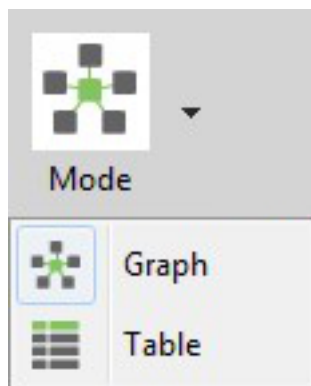
## Network scan status

When you scan  a radio module's network to find available modules, XCTU displays the status of the scan along the bottom edge of the Network working mode display.

17 nodes [PAN ID: BAB05A] [CH: 11] <Scanning> Scan 2 (Remaining: 00:00:21 | Total: 00:00:39)

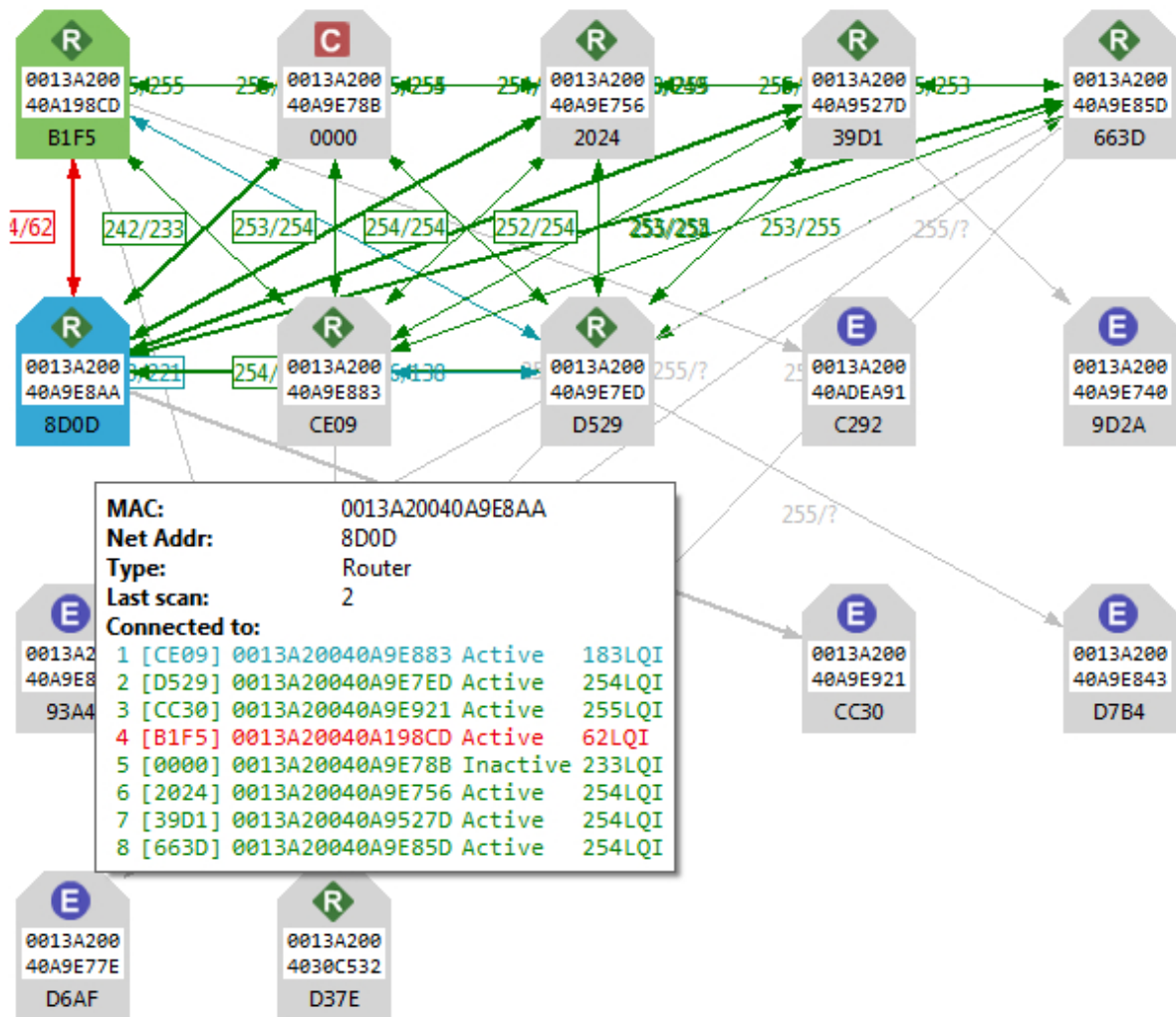
Field	Description
17 nodes	Number of nodes in the network
[PAN ID: BAB05A]	PAN ID
[CH: 11]	Channel
<Scanning>	Status: Getting information, Initializing scan, Scanning, Waiting for next scan, or Stopped
Scan 2	Current scan number
Remaining: 00:00:21	Estimated time remaining to finish current scan
Total: 00:00:39	Total elapsed time of scan

## Graph view



In Graph view, XCTU displays the modules as nodes in a graph. A green background denotes the local radio module and a blue background denotes the selected module. Each module is labeled with its

role (coordinator, router, or end device) and MAC address as well as the 16-bit network address for ZigBee networks and node identifier for other protocols. You can click any node to inspect its connections.








- Each node is connected to its neighbors with solid lines (active connections) or dotted lines (undiscovered connections), with arrows indicating the direction of communication. Selected nodes are blue, and connections are black. Nodes discovered in previous scans that have not been found in the current scan are delineated via lighter-colored lines; they will "turn on" as they are discovered.
- When a radio module leaves the network, some devices continue to store information about their relationship. When you perform a new discovery, the module appears because of the preserved information but the device is unreachable if you attempt to verify the connection. These modules are shown in red.
- When you double click a node, XCTU switches to Configuration mode and displays the node's settings. If the node was not previously discovered, XCTU will add it to the device list as a remote node before displaying its settings.

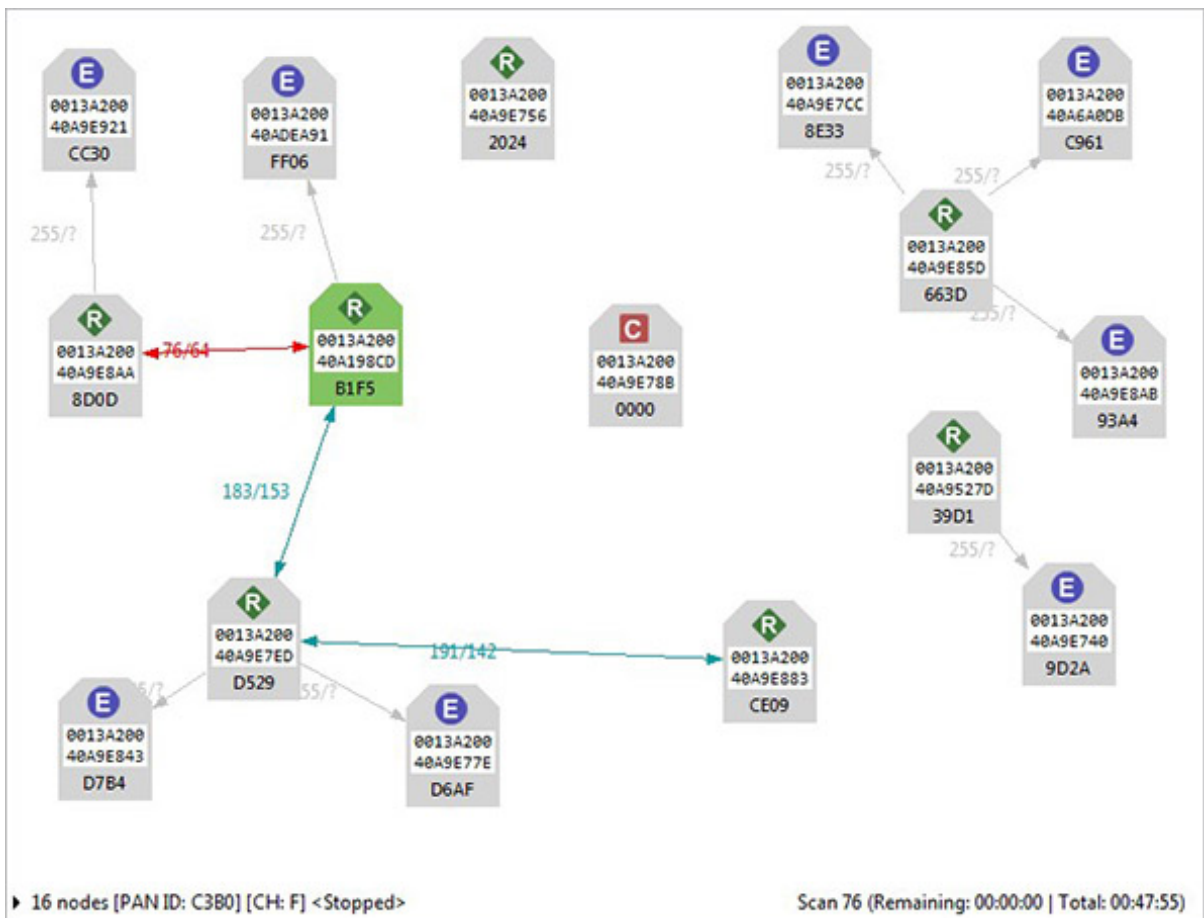
### Connection quality filters

You can filter connections by quality in Graph view to show/hide connections between modules according to configurable quality ranges.

Show connections based on their quality:

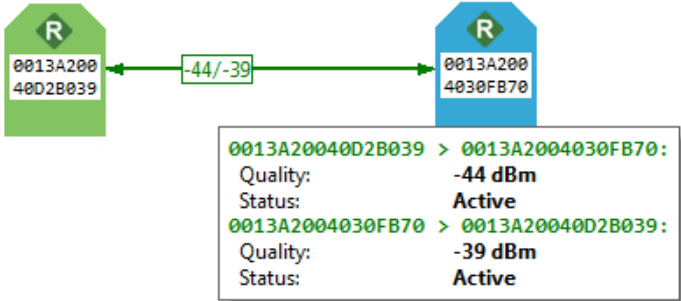
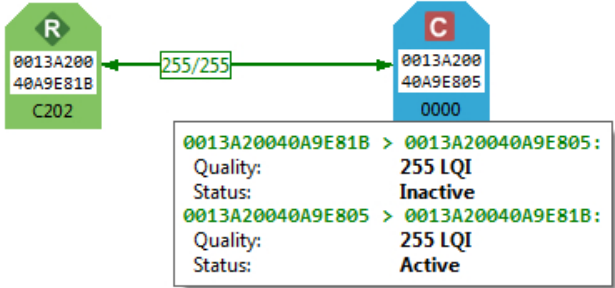
<input type="checkbox"/> Very strong	
<input checked="" type="checkbox"/> Strong	
<input checked="" type="checkbox"/> Moderate	
<input checked="" type="checkbox"/> Weak	
<input checked="" type="checkbox"/> Unknown	

[Configure quality ranges](#)

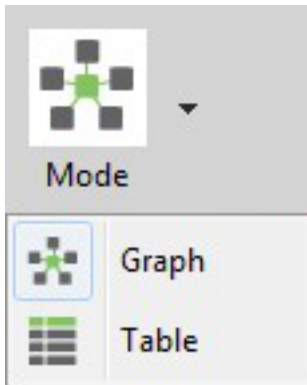


### Connections in 802.15.4, DigiMesh, and ZigBee

When viewing 802.15.4, DigiMesh, and ZigBee modules in Graph view, consider the following differences in node display:

Protocol	Description and Node Display
802.15.4	XCTU does not display the quality of the connection between two nodes. The nodes displayed are in the RF range of the local radio module.
DigiMesh	<p>The quality of the connection between two nodes is displayed next to the line that connects them. Hover over the line joining the nodes to see bidirectional quality (in dB) and status of the connection, respectively.</p> 
ZigBee	<p>The quality of the connection between two nodes is displayed next to the line that connects them. Hover over the line joining the nodes to see bidirectional quality and status of the connection, respectively. Link quality is represented by Link Quality Indication, or LQI, a number between 0 and 255 where 0 is the weakest and 255 is the strongest.</p> 

## Table view



In Table view, network radio modules are displayed as rows in a table. ZigBee devices display end devices as children of their coordinators or routers. Other protocols display all nodes at the same level.

Nodes discovered in previous scans that have not been found in the current scan are delineated via italicized grey text. Connections not yet discovered are also displayed with grey text. A green background denotes the local radio module and a blue background denotes the selected module. When a radio module leaves the network, some devices continue to store information about their relationship. When you perform a new discovery, the module appears because of the preserved information but the device is unreachable if you attempt to verify the connection. These modules are shown in red.


Role	MAC	Network Address	Last scan	Connections																																													
Router (1)	0013A20040A198CD	B1F5	76	Show connections																																													
End device	0013A20040ADEA91	FF06	76	Show connections																																													
Coordinator	0013A20040A9E78B	0000	76	Show connections																																													
Router	0013A20040A9E756	2024	76	Show connections																																													
Router (1)	0013A20040A9527D	39D1	76	Show connections																																													
End device	0013A20040A9E740	9D2A	76	Show connections																																													
Router (3)	0013A20040A9E85D	663D	76	Show connections																																													
End device	0013A20040A9E8AB	93A4	76	Show connections																																													
End device	0013A20040A9E7CC	8E33	76	Show connections																																													
End device	0013A20040A6A0DB	C961	76	Show connections																																													
Router (1)	0013A20040A9E8AA	8D0D	76	<table border="1"> <thead> <tr> <th>Role</th> <th>Net Addr</th> <th>MAC</th> <th>LQI</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>1 Router</td> <td>[CE09]</td> <td>0013A20040A9E883</td> <td>212</td> <td>Active</td> </tr> <tr> <td>2 Router</td> <td>[D529]</td> <td>0013A20040A9E7ED</td> <td>252</td> <td>Active</td> </tr> <tr> <td>3 End device</td> <td>[CC30]</td> <td>0013A20040A9E921</td> <td>255</td> <td>Active</td> </tr> <tr> <td>4 Router</td> <td>[B1F5]</td> <td>0013A20040A198CD</td> <td>64</td> <td>Active</td> </tr> <tr> <td>5 Coordinator</td> <td>[0000]</td> <td>0013A20040A9E78B</td> <td>204</td> <td>Inactive</td> </tr> <tr> <td>6 Router</td> <td>[2024]</td> <td>0013A20040A9E756</td> <td>254</td> <td>Active</td> </tr> <tr> <td>7 Router</td> <td>[39D1]</td> <td>0013A20040A9527D</td> <td>247</td> <td>Active</td> </tr> <tr> <td>8 Router</td> <td>[663D]</td> <td>0013A20040A9E85D</td> <td>255</td> <td>Active</td> </tr> </tbody> </table>	Role	Net Addr	MAC	LQI	Status	1 Router	[CE09]	0013A20040A9E883	212	Active	2 Router	[D529]	0013A20040A9E7ED	252	Active	3 End device	[CC30]	0013A20040A9E921	255	Active	4 Router	[B1F5]	0013A20040A198CD	64	Active	5 Coordinator	[0000]	0013A20040A9E78B	204	Inactive	6 Router	[2024]	0013A20040A9E756	254	Active	7 Router	[39D1]	0013A20040A9527D	247	Active	8 Router	[663D]	0013A20040A9E85D	255	Active
Role	Net Addr	MAC	LQI	Status																																													
1 Router	[CE09]	0013A20040A9E883	212	Active																																													
2 Router	[D529]	0013A20040A9E7ED	252	Active																																													
3 End device	[CC30]	0013A20040A9E921	255	Active																																													
4 Router	[B1F5]	0013A20040A198CD	64	Active																																													
5 Coordinator	[0000]	0013A20040A9E78B	204	Inactive																																													
6 Router	[2024]	0013A20040A9E756	254	Active																																													
7 Router	[39D1]	0013A20040A9527D	247	Active																																													
8 Router	[663D]	0013A20040A9E85D	255	Active																																													
End device	0013A20040A9E921	CC30	76																																														
Router	0013A20040A9E883	CE09	76																																														
Router (2)	0013A20040A9E7ED	D529	76																																														
End device	0013A20040A9E843	D7B4	76																																														
End device	0013A20040A9E77E	D6AF	76																																														
<p>▶ 16 nodes [PAN ID: C3B0] [CH: F] &lt;Stopped&gt; <span style="float: right;">Scan 76 (Remaining: 00:00:00   Total: 00:47:55)</span></p>																																																	



Field	Description
Role	Coordinator, router, or end device
MAC	64-bit address of the module
Network address	Network address (ZigBee) / Node identifier (other protocols)
Last scan	Scan number when the device was last discovered
Connections	Description of current module connections. Click <b>Show connections</b> to view a table with additional information about the module's connections.

## Scan the network for available modules



You can scan a radio module's network to find available modules. Note that the search may take some time. To scan the network:

1. Switch to the Network mode .
2. Select a radio module running in API mode.

---

**Note** Modules in AT (transparent) operating mode do not support the network discovery process.

---

3. Click the **Scan the radio module network** button  to begin the network discovery process. Radio modules are dynamically added to the working area as they are found, showing connections and link quality.
4. The scan process automatically stops according to the network preferences configured in the **Network Preferences** dialog. For details, see [Set network discovery preferences](#). You can also click the **Stop scanning the network** button  at any time.

XCTU displays the network scan status along the bottom edge of the **Network working mode** display. For a description of the status fields, see [Network working mode](#).

17 nodes [PAN ID: BAB05A] [CH: 11] <Scanning>

Scan 2 (Remaining: 00:00:21 | Total: 00:00:39)

---

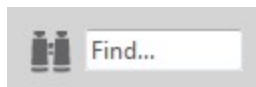
**Note** You may need to perform multiple scans to discover the entire network if, for example, modules are sleeping or the network is very large.

---

## Search for network nodes

If a network scan is active, you can use search expressions to find particular nodes in the network:


1. In the search box, type in a search expression to search for particular nodes in the network. For more information on available search prefixes and parameters, see [Find radio modules](#).



2. The background color of the search box changes based on search status: yellow indicates matches found, and red indicates no matches found.

## Change network perspective

The Network working mode has two different view perspectives: graph view and table view. The Network perspective button allows you to switch between them.

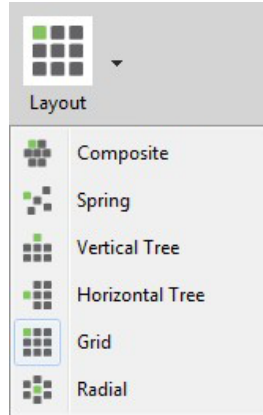
1. Switch to Network working mode .
2. Select a radio module running in API mode from the list.
3. Click the **Network perspective** button and select either **Graph view** or **Table view**.

For a description of each view, see [Network working mode](#).

## Set network layout

If a network scan is active, you can change the network layout:


1. Change the network perspective to Graph view. For details, see [Change network perspective](#).
2. Click the **Set layout** drop-down menu.



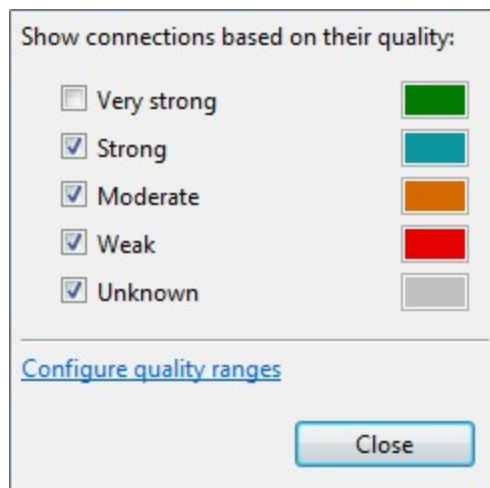
3. Select one of the six layout options.

## Filter network connections by quality

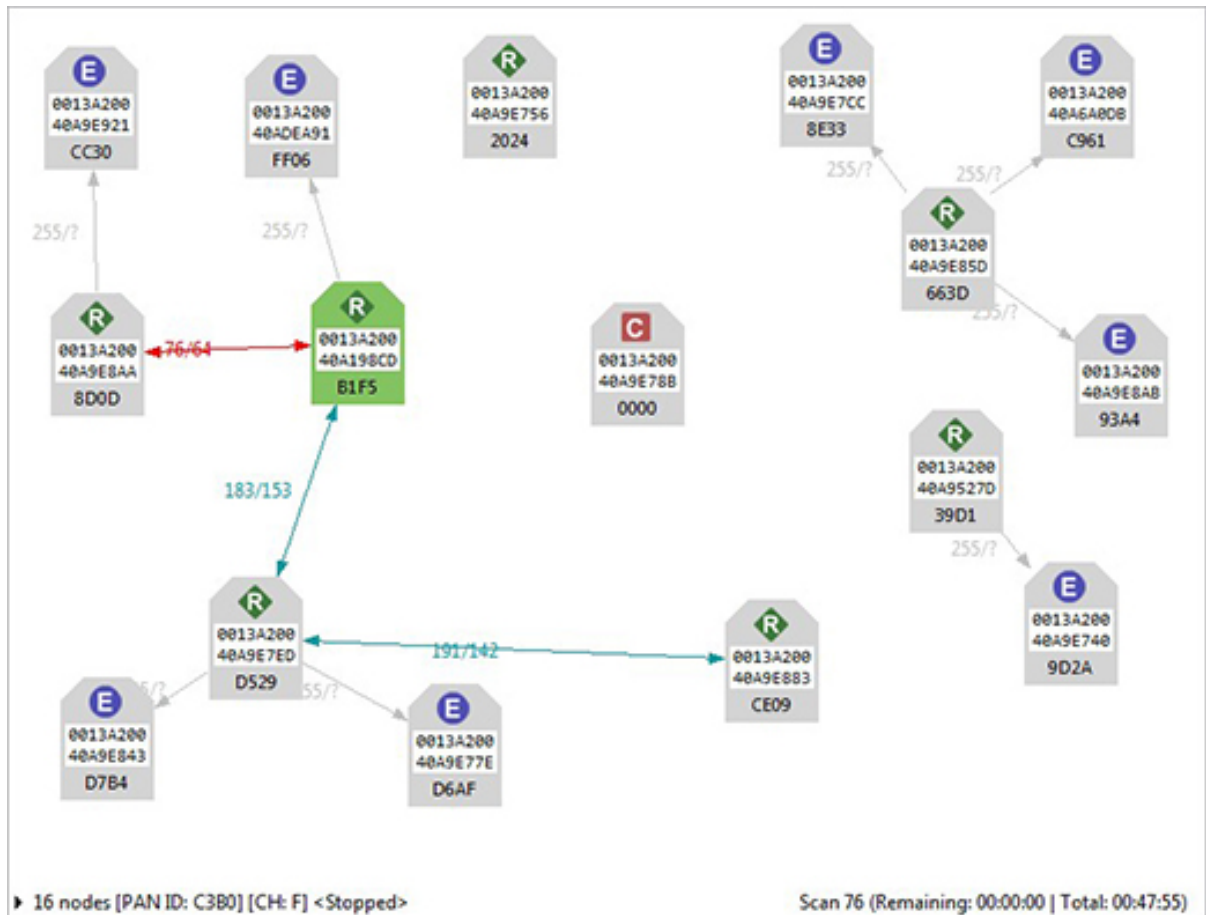
If a network scan is active, you can hide and show connections between RF modules depending on their quality. All filters are enabled by default, so all connections are initially visible.

1. Click the **Filter connections** button . A dialog appears with all of the filters enabled.
2. Check or uncheck quality connection filters. Unchecking a box hides all connections associated with that quality filter, and checking a box displays all connections associated with that quality filter.

In this example, we unchecked the **Very strong** filter.



3. Click **Close**. XCTU displays network connections based on the filters you selected. In this example, we can see all of the connections except for those that are very strong.

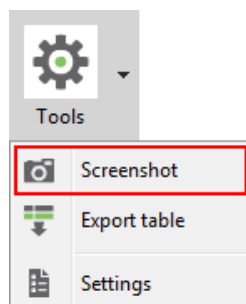


**Note** Click **Configure quality ranges** to configure quality range values and display colors. You can also access quality range configuration options from **Tools > Settings** on the Network view toolbar.

## Take a screenshot of the network

If a network scan is active, you can save a Graph view or Table view screenshot.

1. Click the **Options** drop-down menu and select **Screenshot**. A **Save file** dialog appears.



2. Specify the name and location of the image and click **Save**. A screenshot is saved to your clipboard.

## Set zoom level in Graph view

If a network scan is active, you can use the zoom control in Graph view to change the zoom level of the view:

1. Change the network perspective to Graph view. For details, see [Change network perspective](#).
2. In the Zoom level dialog, use the up/down arrows to change the zoom level or enter a percentage between 10 and 400.

---

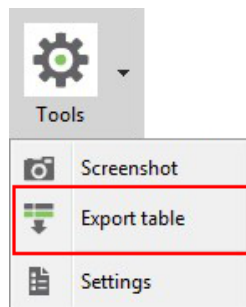
**Note** You can also zoom in and out by pressing the **Control** key while scrolling the mouse wheel.

---

## Export a network table

If a network scan is active, you can save network information from either graph view in CSV (comma separated value) file format. To export a network table when a network scan is active:

1. Click the **Options** drop-down menu and select **Export table**.



A **Save file** dialog appears.

2. Designate a filename and location and click **Save**.

## Configure XCTU

---

This section describes how to configure several XCTU settings such as firmware updates, display settings, and network connectivity. To open the Preferences dialog box, click the Preferences button on the XCTU toolbar.




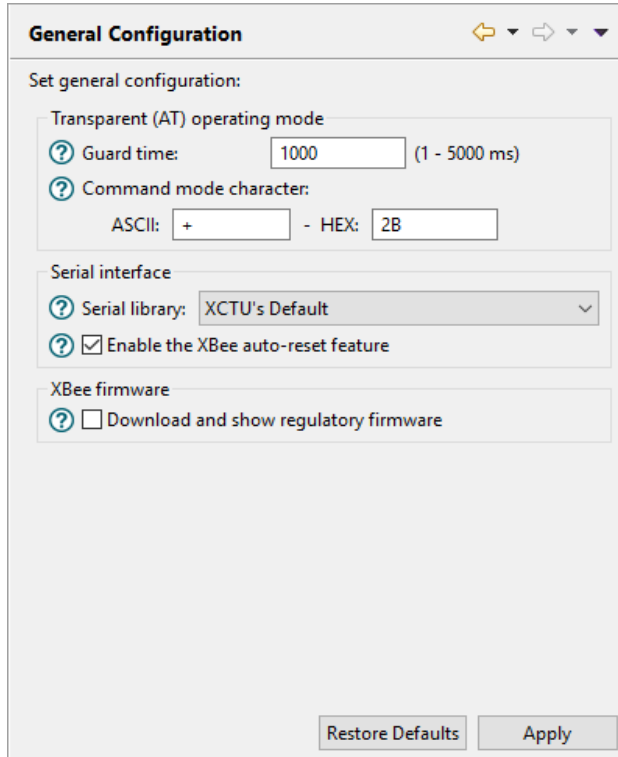
The preference categories are listed on the left side of the Preferences dialog box. You can configure settings for the following categories:

Set general preferences .....	119
Set appearance preferences .....	120
Set automatic software update preferences .....	121
Set console preferences .....	123
Set firmware update preferences .....	124
Set network discovery preferences .....	125
Set network appearance preferences .....	127
Set radio firmware library preferences .....	129
Set MicroPython Terminal preferences .....	130

## Set general preferences

You can configure some general settings related with the transparent operating mode.

1. On the XCTU toolbar, click the **Preferences** button . The Preferences dialog appears.
2. On the left side of the Preferences dialog, select **General Preferences**.




3. Configure your preferred settings and click **Apply**.

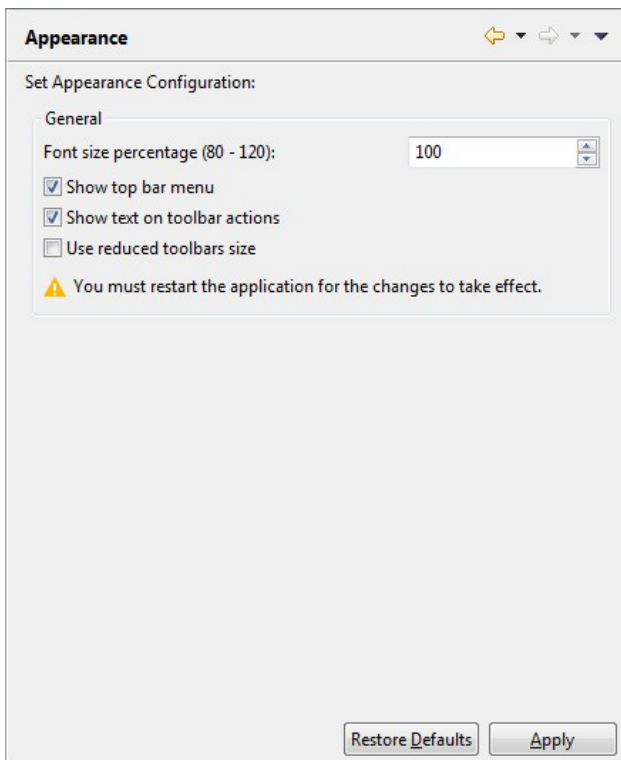
Settings	Description
Guard time	Sets the period of silence XCTU waits before and after entering command mode when the radio module operates in Transparent (AT) mode. This value must match the <b>GT</b> value of the radio modules.
Command mode character	Sets the character value to be used to enter command mode when the radio module operates in transparent (AT) mode. The command mode character can be specified either in ASCII or hexadecimal format. This value must match the <b>CC</b> value of the radio modules.
Serial library	Choose the serial library to be used to communicate with the XBee devices through the serial port. Select the <b>Alternative</b> option only if you have troubles or crashes when trying to discover or communicate with XBee devices.
Enable the	Enables or disables the XCTU feature that tries to reset the target XBee

Settings	Description
auto-reset XBee devices functionality	device automatically instead of performing a manual reset. <hr/> <b>Note</b> This feature only works in <b>Windows</b> . <hr/>
Download and show regulatory firmware	Regulatory firmware is used for regulatory testing purposes and in most cases the available settings are limited in XCTU. Check this setting if you want XCTU to <b>download and install</b> regulatory firmware from the FTP when executing the <b>Update the Radio Firmware Library</b> operation. It will also <b>show</b> regulatory firmware in the following dialogs so that you can inspect it or flash it in an XBee module: <ul style="list-style-type: none"> <li>▪ Update Firmware dialog</li> <li>▪ XBee Recovery tool</li> <li>▪ Firmware Explorer tool</li> </ul> Uncheck the setting if you do not want to download or see regulatory firmware.

## Set appearance preferences

You can configure some graphic aspects of the tool and how some elements are displayed.

1. On the XCTU toolbar, click the **Preferences** button . The **Preferences** dialog appears.
2. On the left side of the Preferences dialog, select **Appearance**.





3. Configure your preferred settings and click **Apply**.

Field	Description
Font size	Changes all the XCTU texts size in percentage, from 50 to 120%.
Show top bar menu	Displays an application top bar menu with texts.
Show text on toolbar actions	Displays the name of the action below each toolbar element for a better understanding of the meaning of each action.
Use reduced toolbars size	Changes the size of the application toolbars reducing them.


---

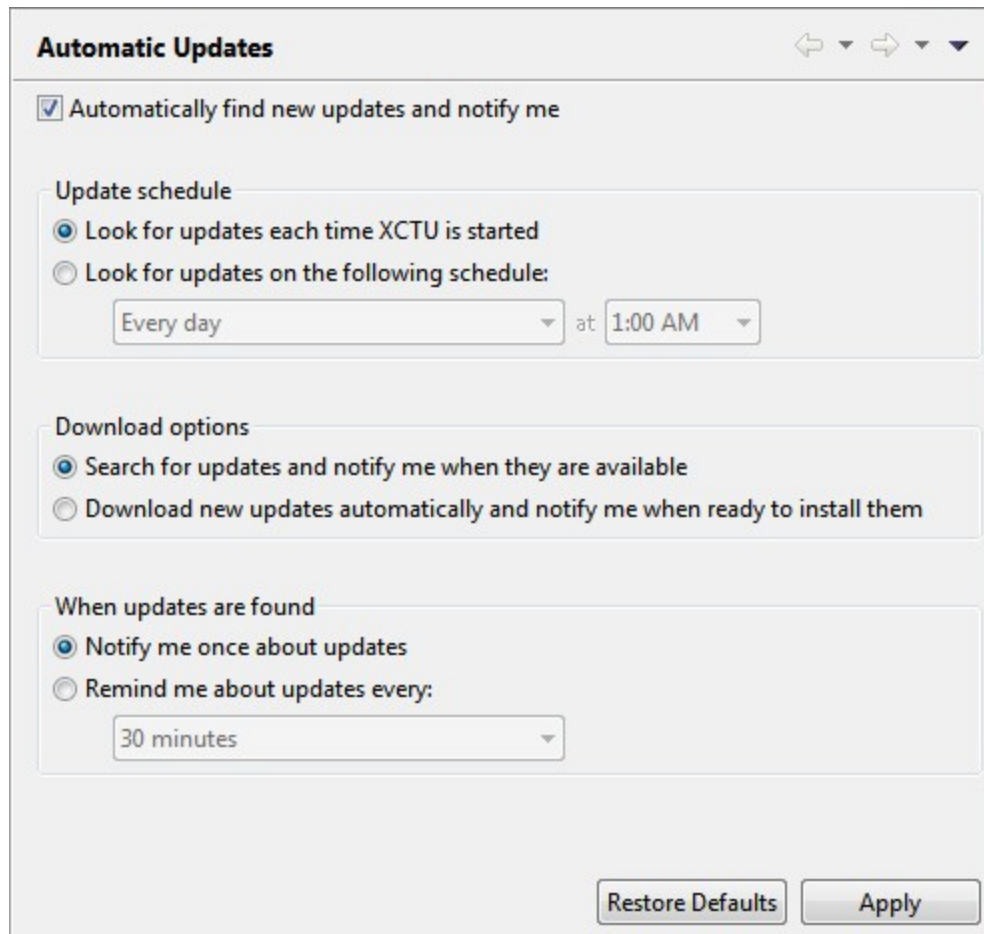
**Note** You must restart XCTU for the changes to take effect.

---

## Set automatic software update preferences

You can configure when and how new updates for XCTU are downloaded and installed.

1. On the XCTU toolbar, click the **Preferences** button . The configuration dialog appears.
2. On the left side of the configuration dialog, select **Automatic Updates**.




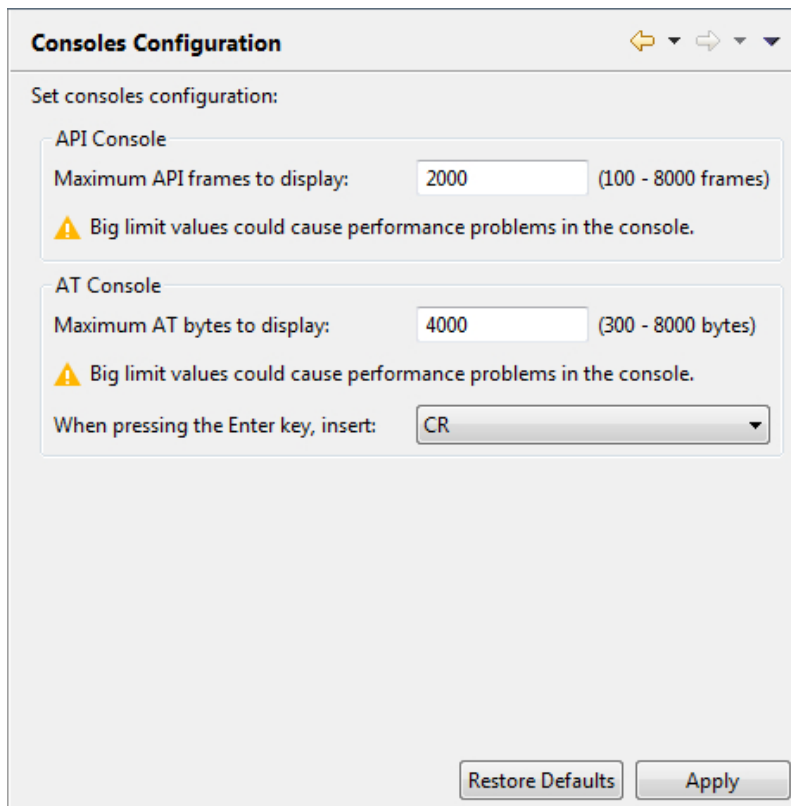
3. Check **Automatically find new updates and notify me** to enable the automatic updates feature.
4. Configure your preferred settings and click **Apply**.

Field	Description
Update schedule	Sets a schedule to search for updates or to update when XCTU is started. If you select <b>Look for updates on the following schedule</b> , you must also specify the search interval and hour.
Download options	Establishes when new updates should download and sets permissions for whether updates are automatically downloaded.
When updates are found	Sets the frequency of update notification.

## Set console preferences

You can configure the number of frames to display in an API console and/or the number of bytes to display in an AT console.

1. On the XCTU toolbar, click the **Preferences** button  to open the **Preferences** dialog.
2. On the left side of the **Preferences** dialog, select **Consoles preferences**.




3. Configure your preferred settings and click **Apply**.

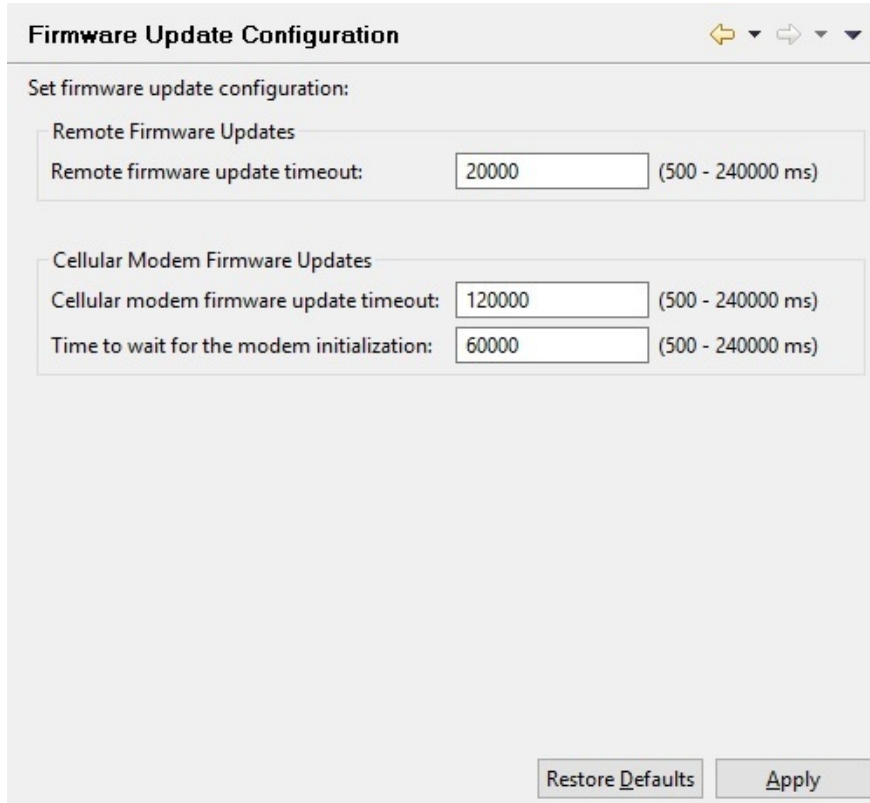
Settings	Description
API console	Configures the maximum number of API frames that can be displayed in the frames log during a session. When the maximum limit is reached, the session starts overwriting frames.
AT console	Configures the maximum number of bytes that can be displayed during a session. When the maximum limit is reached, the session starts overwriting bytes. It also establishes the default behavior when you press the ENTER key in the AT Console to insert CR/LF, CR or LF characters.

**Note** If you set high values for maximum API frames and/or AT bytes, you may notice performance issues in the consoles.

## Set firmware update preferences

You can configure some preferences related to the firmware update process.

1. On the XCTU toolbar, click the **Preferences** button  to open the **Preferences** dialog.
2. On the left side of the **Preferences** dialog, select **Firmware Update preferences**.



The screenshot shows the 'Firmware Update Configuration' dialog box. It has a title bar with the text 'Firmware Update Configuration' and navigation arrows. Below the title bar, the text 'Set firmware update configuration:' is displayed. There are two main sections: 'Remote Firmware Updates' and 'Cellular Modem Firmware Updates'. Each section contains a label and a text input field with a range in parentheses. At the bottom right, there are two buttons: 'Restore Defaults' and 'Apply'.


Section	Label	Value	Range
Remote Firmware Updates	Remote firmware update timeout:	20000	(500 - 240000 ms)
	Cellular modem firmware update timeout:	120000	(500 - 240000 ms)
Cellular Modem Firmware Updates	Time to wait for the modem initialization:	60000	(500 - 240000 ms)

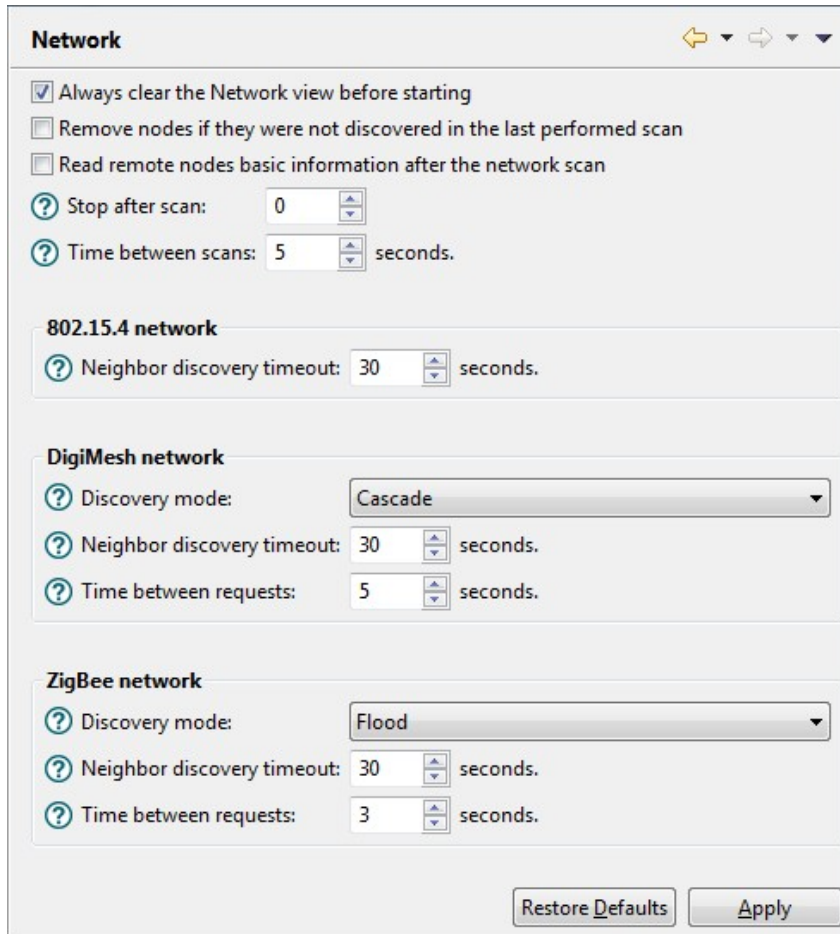
3. Configure your preferred settings and click **Apply**.

Field	Description
Remote firmware update timeout	Configures the remote firmware update timeout in milliseconds. This is the maximum time that XCTU waits for a response sent by the remote node during the remote firmware update before determining if there was an error during the process.
Cellular modem firmware update timeout	Configures the cellular modem firmware update timeout in milliseconds. This is the maximum time that XCTU waits for a response sent by the modem of a Cellular device during a local firmware update before determining if there was an error during the process.
Time to wait for the modem initialization	Configures the time to wait for the modem initialization in milliseconds. This is the maximum time that XCTU waits for the Cellular modem initialization. If this time expires and the modem version cannot be read, XCTU forces the modem firmware update.

## Set network discovery preferences

You can configure network discovery parameters in the **Network preferences** panel.

1. On the XCTU toolbar, click the **Preferences** button  to open the **Preferences** dialog.
2. On the left side of the **Preferences** dialog, select **Network**.



3. Configure your preferred settings and click **Apply**.  
The first set of options are common to all networks:

Settings	Description
Always clear the Network view before starting	Clears Network view before each new network scan.
Remove nodes if they were not discovered in the last performed scan	Removes any nodes not discovered in the last scan.
Read remote nodes basic information after the network scan	Reads the following information from the remote nodes after each network scan: Node identifier, hardware version, firmware version, network address and device type.
Stop after scan	Sets the number of scans to perform before stopping the

Settings	Description
	discovery process. A value of '0' means the process will not stop automatically.
Time between scans	Sets the duration of time XCTU waits before starting a new network scan. The value must be between 0 and 300 seconds.

The remainder of the options are specific to 802.15.4, DigiMesh, and ZigBee network types:

Settings	Description
Discovery mode	<p>Sets the method used by the network discovery process.</p> <ul style="list-style-type: none"> <li>■ Flood: The neighbor discovery process is performed for every node at the moment it is found. Several discovery processes may be running at the same time. This method may be faster, but it may also generate a lot of traffic and saturate the network.</li> <li>■ Cascade: The neighbor discovery process is performed for every node as soon as the discovery process finishes. Only one discovery process runs at a time. This method may be slower, but it is likely to generate less traffic.</li> </ul>
Neighbor discovery timeout	<p>Sets the maximum duration, in seconds, the discovery process should spend finding neighbors of a module. Value must be between 5 and 1800 seconds (30 minutes).</p> <p>This timeout is highly dependent on the nature of the network. For DigiMesh, the value should be greater than the highest NT (Node Discover Timeout) and include enough time to let the message propagate, depending on the sleep cycle of your devices.</p>
Time between requests	<p>Sets the wait time between node neighbor requests. The value must be between 0 and 300 seconds (5 minutes).</p> <p>For the Cascade method, this is the number of seconds to wait after completion of the neighbor discovery process of the previous node.</p> <p>For the Flood method, this is the minimum time to wait between each radio module's neighbor requests.</p>


---

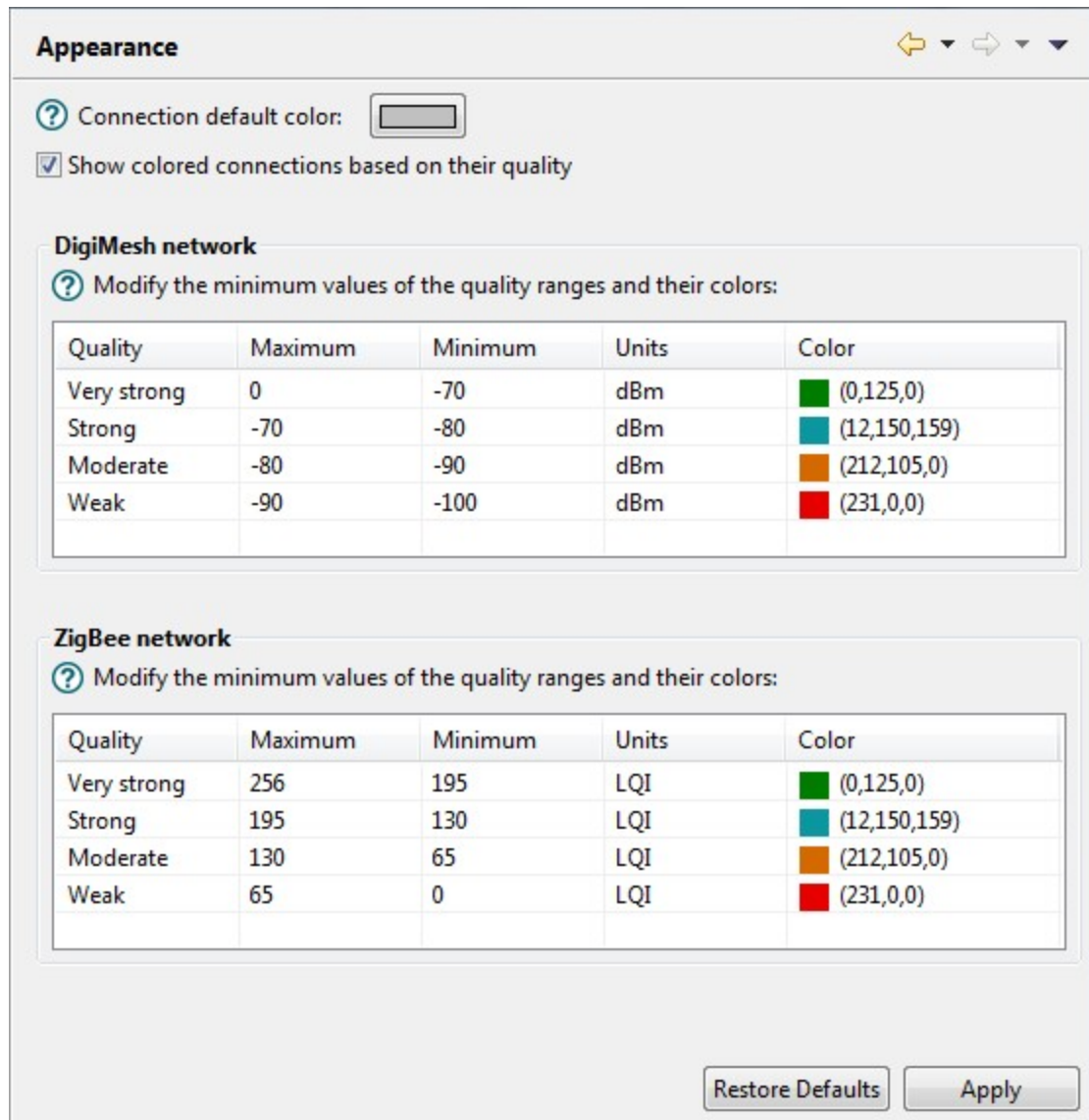
**Note** The Cascade discovery method is recommended for large networks.

---

## Set network appearance preferences

The Network preferences has a sub-category named Appearance where you can customize the appearance of node links in Network view.

1. On the XCTU toolbar, click the **Preferences** button  to open the **Preferences** dialog.
2. On the left side of the **Preferences** dialog, select **Network** and then select **Appearance**.



3. Configure your preferred settings and click **Apply**.


Settings	Description
Connection default color	Defines the default color of the node's connection lines.
Show colored connections based on their	Enables or disables the coloring of node connection lines based on their link quality.

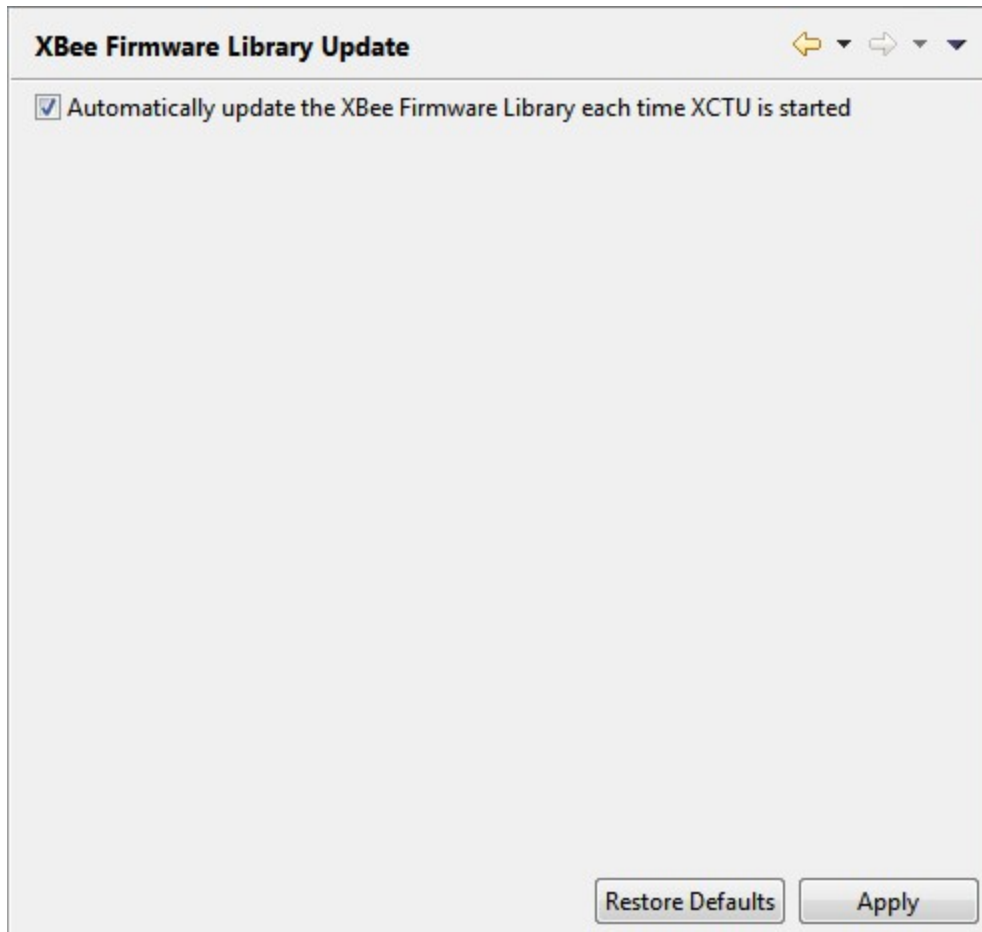


Settings	Description
quality	
DigiMesh / ZigBee network	<p>Enables you to modify the maximum and minimum values and RGB colors for each quality range. Click in the cell, type the value, and click <b>Enter</b> to change a value.</p> <p>Ranges include minimum values but not maximum values. When you change the minimum value of a quality range, the maximum value of the next range adopts a corresponding value.</p>

## Set radio firmware library preferences

You can configure XCTU to look for new radio firmware at startup.

1. On the XCTU toolbar, click the **Preferences** button . The **Preferences** dialog appears.
2. On the left side of the Preferences dialog, select **Radio Firmware Library Update**.



3. Check **Automatically update the Radio Firmware Library each time XCTU is started**. If you


do not select this option, you can only check for firmware updates manually.

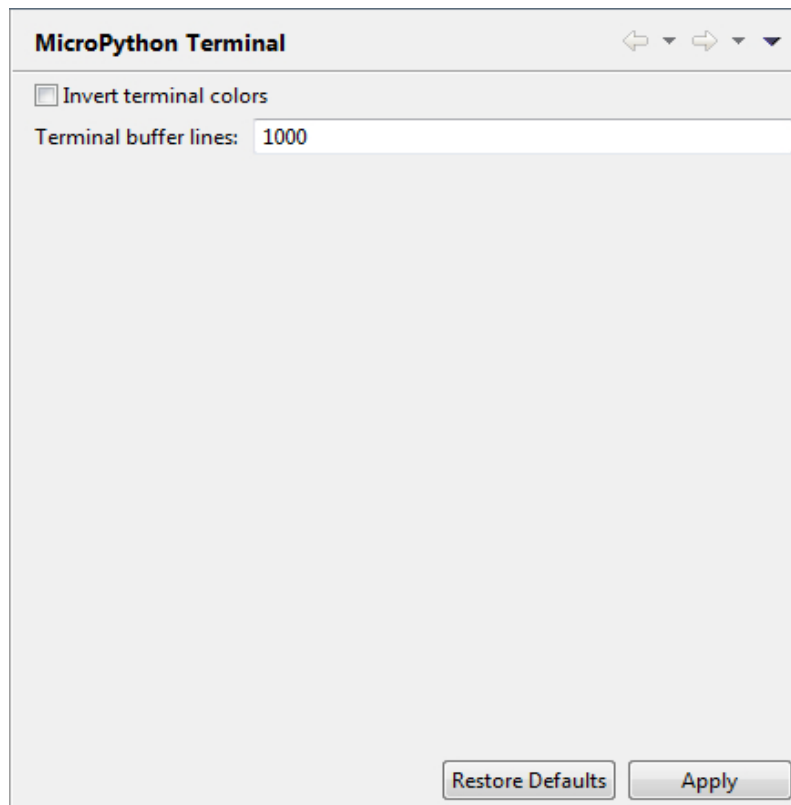
4. Click **Apply**.

For more information, see [Update radio firmware library](#).

## Set MicroPython Terminal preferences

You can configure some terminal related preferences of the MicroPython terminal tool.

1. On the XCTU toolbar, click the **Preferences** button . The Preferences dialog appears.
2. On the left side of the Preferences dialog, select **MicroPython Terminal**.



3. Configure your preferred settings and click **Apply**.

Field	Description
Invert terminal colors	Inverts the colors of the terminal. By default, this sets the background from white to black and the font color from black to white.
Terminal buffer lines	Sets the maximum number of lines that are displayed in the terminal.

## Update software

---

XCTU allows you to automatically update the radio firmware library and the XCTU application itself without downloading any extra files. These processes can be configured to execute automatically, but you can also execute them manually at any time. For more information about configuring automatic XCTU and radio firmware library updates, see [Configure XCTU](#).

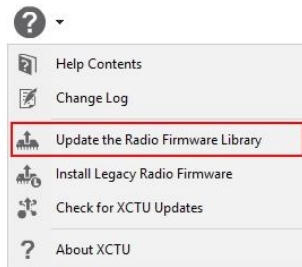
Update radio firmware library .....	132
Install legacy radio firmware .....	133
Install XCTU updates .....	134
Open change log .....	135

## Update radio firmware library

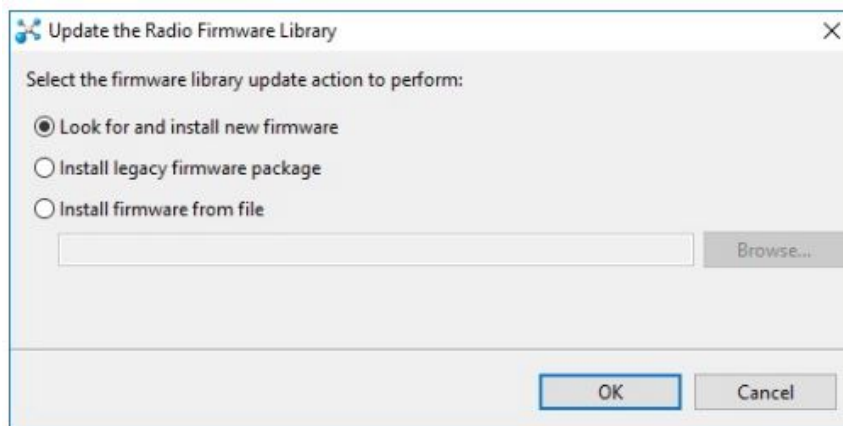
Digi periodically releases new versions of the radio firmware to fix issues, improve functionality, or add new features. These firmware files might not be included with XCTU and need to be added to the radio firmware library. Also, new RF products may be launched in the market that require new radio firmware to be configured with XCTU. For this reason, XCTU can update the radio firmware library from the application itself.

You can configure XCTU so the firmware is automatically updated. For more information, see [XCTU preferences](#). To manually update the radio firmware library:

1. Select **Help > Update Radio Firmware Library** option.



2. Do one of the following:
  - To look for new radio firmware inside Digi's update site, select **Look for and install new firmware**.
  - To install legacy radio firmware, select **Install legacy firmware package**.
  - To install a local radio firmware file, select **Install firmware from file** and specify the path where the file is located.



3. Click **OK** to start. A dialog displays the status of the download process.
4. You can click the **Run in background** button of the progress dialog to execute this process in the background. This allows you to keep working with XCTU while the new firmware is downloaded. The status bar displays the update process.

When the process finishes, a dialog displays the list of downloaded firmware.

---

**Note** Downloading the firmware does not automatically update attached modules.

---

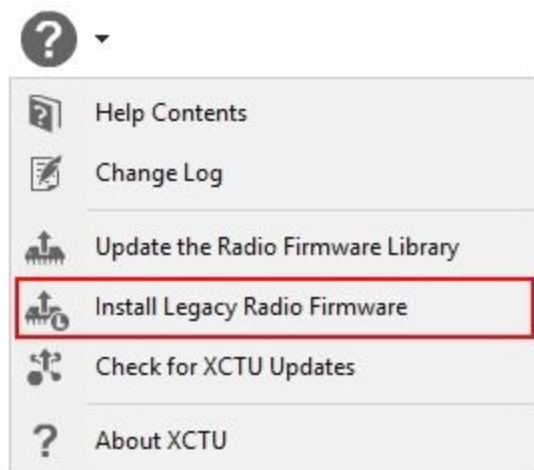
## Install legacy radio firmware

The latest versions of XCTU provide only the latest firmware versions of the currently supported products and families. In the past, the current version of XCTU installed all of the existing radio firmware versions.

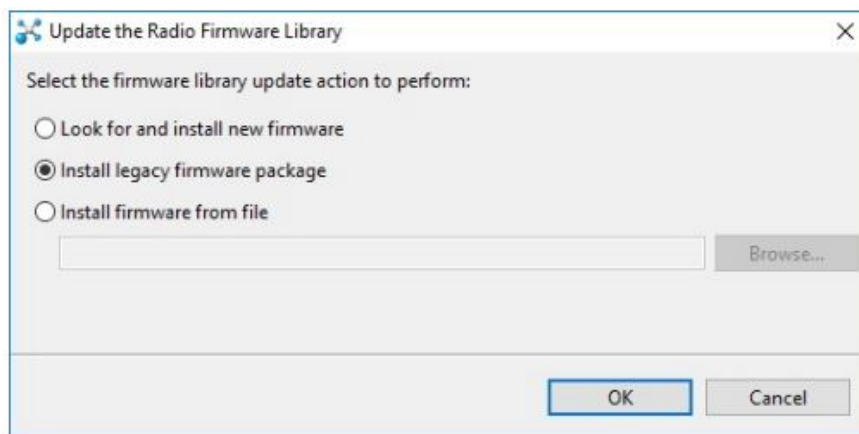
However, if you have a legacy product or an old firmware version that cannot be found in XCTU, you can install the legacy radio firmware package.

To install the legacy radio firmware package:

1. Select **Help > Install Legacy Radio Firmware** option. The **Update the Radio Firmware Library** dialog appears.



2. In the **Update the Radio Firmware Library** dialog, select the **Install legacy firmware package** option. This option should be selected by default.



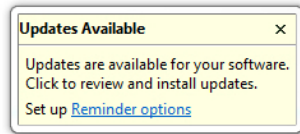
3. Click **OK** to start. A dialog displays the status of the download process.
4. Click the **Run in background** button in the progress dialog to run the download process in the background. This allows you to keep working with XCTU while the legacy firmware is downloaded. The status bar displays the update process.

When the process finishes, a dialog displays the list of installed firmware.

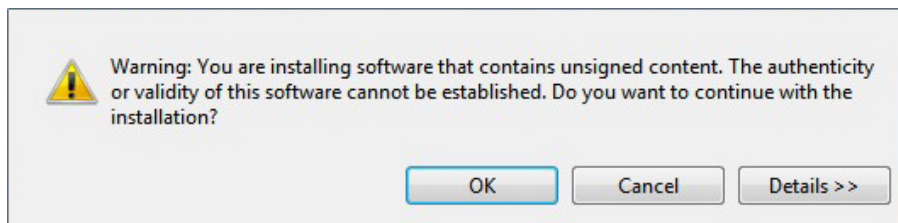
## Install XCTU updates

When you start XCTU, you may be notified about software updates. You should always run the latest version of XCTU.

1. When a new version is available, a pop-up window appears in the bottom-right corner of XCTU.



2. Click anywhere in that window and follow the prompts to proceed with the update.
3. During installation, you may be asked if you want to proceed with installing unsigned software.



Click **OK** to continue.

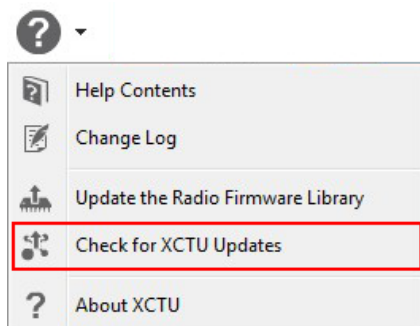
4. When the installation process is finished, you must reset XCTU so new changes can be applied. When prompted, click **Yes** to restart XCTU.

---

**Note** You can click the **Run in background** button of the progress dialog to execute this process in the background. The status bar displays the update process.

---

5. Once restarted, XCTU displays a change log. You can manually open the change log at any time by clicking **Help > Change Log**.
6. You can also check for updates and manually update the tool by clicking **Help > Check for XCTU Updates**.

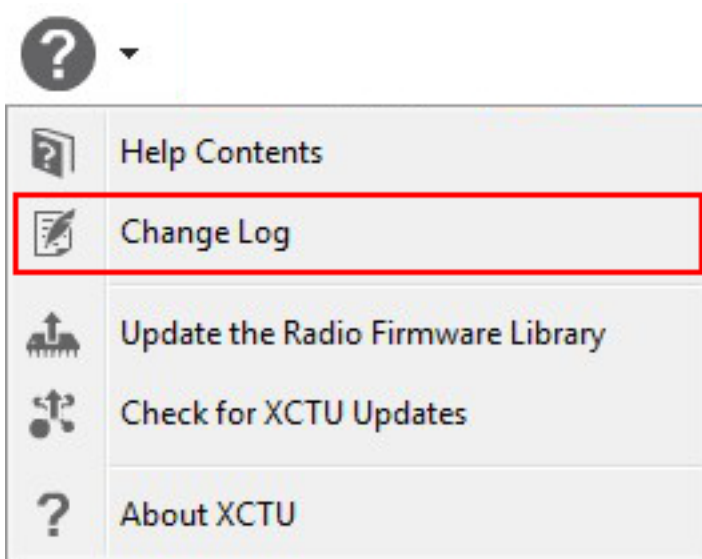


If an update for the application is found, a dialog box displays the update. Click **Finish** to start the update process.

## Open change log

XCTU includes a change log with historical changes and additions between versions. You can view the change log at any time. To open the change log:

Select **Help > Change Log**.



A new dialog opens listing all the XCTU versions with their corresponding dates and changes.

## Use the XCTU command line

---

This section describes how to automate tasks using the XCTU command line interface.

Understanding the XCTU command line interface .....	137
List ports via command line .....	137
Load profile via command line .....	138
Update firmware via command line .....	139



## Understanding the XCTU command line interface

The XCTU Command-Line Interface (CLI) enables you to manage and automate certain tasks without launching the graphic user interface.

The CLI uses a multipart structure on the command line:

---

```
XCTUcmd <command> <required parameters> [optional parameters]
```

---

The first portion of the syntax is the base call to `XCTUcmd` executable. The location of this file depends on your operating system:

OS	File location
Windows	<installation_directory>/XCTUcmd.exe
MacOS X	<installation_directory>/XCTU.app/Contents/MacOS/XCTUcmd
Linux	<installation_directory>/XCTUcmd

The next part specifies a top-level command, which often represents an XCTU management action supported in the CLI. You can specify additional parameters or options for each command in any order on the command line. If an exclusive parameter is specified multiple times, then only the last value applies.

### List all commands

You can run `XCTUcmd` by itself to display a list of all available commands. This is the complete list:

- [update\\_firmware](#) updates the firmware of the specified device.
- [load\\_profile](#) loads a profile to the specified device.
- [list\\_ports](#) lists all the available serial and USB ports on your computer.

### Program arguments

You can also run `XCTUcmd` with specific program arguments to display information about the tool:

#### *Display the tool usage information*

---

```
XCTUcmd --help
```

---

#### *Display the tool version*

---

```
XCTUcmd --version
```

---

#### *Print the tool error codes table*

---

```
XCTUcmd --errorlist
```

---

## List ports via command line

The `list_ports` command allows you to list all available serial and USB ports on your computer. To list the ports on your computer using the XCTU CLI:

1. Enter the following command syntax.

---

```
XCTUcmd list_ports
```

---

The location of the base call to XCTUcmd executable depends on your operating system:

- Windows: <installation\_directory>/XCTUcmd.exe
  - MacOS X: <installation\_directory>/XCTU.app/Contents/MacOS/XCTUcmd
  - Linux: <installation\_directory>/XCTUcmd
2. If the command is executed successfully, the command line prompt returns an error code 0. A return code different than 0 is represented by the error ID. Enter the following command for a list of possible errors:

---

```
XCTUcmd --errorlist
```

---

## Options

Some commands have multiple options. There are no options for the **list\_ports** command.

## Load profile via command line

The **load\_profile** command allows you to load a previously saved profile to the specified device. To load a profile using the XCTU CLI:

1. Enter the following command syntax. Both -f <file> and -p <port> are required parameters. The rest are optional. The file option must point to an existing XPRO profile. These files are generated from the XCTU graphic application and can be stored anywhere in your computer.

---

```
XCTUcmd load_profile -f <file> -p <port> [-b <baud_rate>] [-e] [-o] [-h]
[-F] [-R] [-P] [-v]
```

---

The location of the base call to XCTUcmd executable depends on your operating system:

- Windows: <installation\_directory>/XCTUcmd.exe
  - MacOS X: <installation\_directory>/XCTU.app/Contents/MacOS/XCTUcmd
  - Linux: <installation\_directory>/XCTUcmd
2. If the command is executed successfully, the command line prompt returns an error code 0. A return code different than 0 is represented by the error ID. Enter the following command for a list of possible errors:

---

```
XCTUcmd --errorlist
```

---

**Note** Legacy XCTU \*.PRO profile files are not supported by this command. If you have a legacy \*.pro file, you must first load the profile in the XCTU graphic application and save it in the new XPRO format.

---

## Options

Use the following options with the **load\_profile** command:

<b>-f &lt;file&gt;</b>	Required *	Specify the profile file location path.
<b>-p &lt;port&gt;</b>	Required	Specify the serial port name to be used.
<b>-b &lt;baud_rate&gt;</b>	Optional	Specify serial connection baud rate to use. [Default 9600]
<b>-e</b>	Optional	Set parity to even for the serial connection. [Default NONE]
<b>-o</b>	Optional	Set parity to odd for the serial connection. [Default NONE]
<b>-h</b>	Optional	Set hardware flow control for the serial connection. [Default NONE]
<b>-F</b>	Optional	Force firmware update if profile firmware does not match with firmware of device.
<b>-R</b>	Optional **	Force a device recovery if device cannot be discovered.
<b>-P</b>	Optional **	Indicate that the device is a programmable radio. These radios have a baud rate of 115200 by default.
<b>-v</b>	Optional	Show details about the process being executed.

\* File paths containing blank spaces must be quoted.

\*\* These options may trigger an action request event where users will be asked to reset the device.

## Examples

For additional guidance, see the following usage examples for the **load\_profile** command:

Load profile ZB1\_profile to the device connected to COM7 at 115200 bps, updating the firmware if required and displaying process details

---

```
XCTUcmd load_profile -f "C:\profiles\ZB1_profile.xpro" -p COM7 -b 115200 -F -v
```

---

Load profile ZB2\_profile to the programmable device connected to COM1 using parity even and displaying process details

---

```
XCTUcmd load_profile -f "C:\profiles\ZB2_profile.xpro" -p COM1 -e -P -v
```

---

Load profile ZB3\_profile to the device connected to COM4 and perform a device recovery if required

---

```
XCTUcmd load_profile -f"C:\profiles\ZB3_profile.xpro" -p COM4 -R
```

---

## Update firmware via command line

The **update\_firmware** command allows you to update the firmware of the specified device. To update firmware using the XCTU CLI:

1. Enter the following command syntax. Both **-f <file>** and **-p <port>** are required parameters. The rest are optional. The file option must point to an existing firmware description XML file. These files are stored within the XCTU installation path under a folder named `radio_firmwares`, sorted by protocols or hardware models.

---

```
XCTUcmd update_firmware -f <file> -p <port> [-b <baud_rate>] [-e] [-o] [-h] [-R] [-P] [-v]
```

---

The location of the base call to XCTUcmd executable depends on your operating system:

- Windows: <installation\_directory>/XCTUcmd.exe
  - MacOS X: <installation\_directory>/XCTU.app/Contents/MacOS/XCTUcmd
  - Linux: <installation\_directory>/XCTUcmd
2. If the command is executed successfully, the command line prompt returns an error code 0. A return code different than 0 is represented by the error ID. Enter the following command for a list of possible errors:

---

```
XCTUcmd --errorlist
```

---

## Set options

Use the following options with the **update\_firmware** command:

Option	Required/optional	Action
<b>-f &lt;file&gt;</b>	Required *	Specify the firmware file location path.
<b>-p &lt;port&gt;</b>	Required	Specify the serial port name to be used.
<b>-b &lt;baud_rate&gt;</b>	Optional	Specify serial connection baud rate to use. [Default 9600]
<b>-e</b>	Optional	Set parity to even for the serial connection. [Default NONE]
<b>-o</b>	Optional	Set parity to odd for the serial connection. [Default NONE]
<b>-h</b>	Optional	Set hardware flow control for the serial connection. [Default NONE]
<b>-R</b>	Optional **	Force a device recovery if device cannot be discovered.
<b>-P</b>	Optional **	Indicate that the device is a programmable radio. These radios have a baud rate of 115200 by default.
<b>-v</b>	Optional	Show details about the process being executed.

\* File paths containing blank spaces must be quoted.

\*\* These options may trigger an action request event where users will be asked to reset the device.

## Examples

For additional guidance, see the following usage examples for the **update\_firmware** command:

Flash XBP24-ZB\_23A7\_S2B firmware to the device connected to COM7 at 115200 bps and display process details:

---

```
XCTUcmd update_firmware -f "C:\Program Files (x86)\Digi\XCTU-NG\radio_firmwares\xbee_zb\XBP24-ZB_23A7_S2B.xml" -p COM7 -b 115200 -v
```

---

Flash XBP24-ZB\_23A7\_S2B firmware to the programmable device connected to COM1 using HW flow control and display process details:

---

```
XCTUcmd update_firmware -f "C:\Program Files (x86)\Digi\XCTU-NG\radio_
firmwares\xbee_zb\XBP24-ZB_23A7_S2B.xml" -p COM1 -h -P -v
```

---

Flash XBP24-ZB\_23A7\_S2B firmware to the device connected to COM4 and perform a recovery if required:

---

```
XCTUcmd update_firmware -f "C:\Program Files (x86)\Digi\XCTU-NG\radio_
firmwares\xbee_zb\XBP24-ZB_23A7_S2B.xml" -p COM4 -R
```

---






## XCTU tools








---

XCTU includes a set of embedded tools that you can execute at any time, regardless of the active working mode.

Access the following XCTU tools from the **Tools** drop-down menu of the main toolbar:



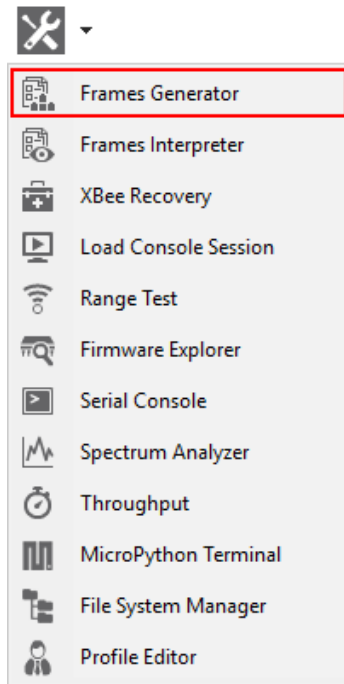
Icon	Tool	Description
	<a href="#">Frames generator tool</a>	Generate an API frame.
	<a href="#">Frames interpreter tool</a>	Decode any API frame and get the value of any field.
	<a href="#">XBee recovery tool</a>	Recover a radio module that is out of communication.
	<a href="#">Load console session tool</a>	Load and navigate previously saved console sessions.
	<a href="#">Range test tool</a>	Perform a range test between two radio modules in your network.

Icon	Tool	Description
	<a href="#">Firmware explorer tool</a>	Navigate and investigate any XCTU-hosted radio firmware.
	<a href="#">Serial console tool</a>	Open a serial communication with a radio module without adding it to your device list.
	<a href="#">Spectrum analyzer tool</a>	Test and measure the spectrum of a radio band.
	<a href="#">Throughput tool</a>	Measure the transfer ratio between two radio modules in the same network.
	<a href="#">MicroPython Terminal tool</a>	Communicate with the XBee Cellular Modem when it is in MicroPython mode.
	<a href="#">File system manager tool</a>	Access and interact with the file system of your XBee module.
	<a href="#">Profile Editor tool</a>	Create new profiles and edit existing ones.

## Frames generator tool

The Frames generator tool generates any type of API frame (output or input), automatically generating the byte array from a series of inputs in the tool. You do not have to add a radio module to your device list to use the Frames generator tool. Frame parameters are dependent on the protocol, mode, and frame type you choose from the drop-down boxes. For more information, see [API frames](#).

To open the **Frames generator** tool, select **Frames generator** from the **Tools** drop-down menu on the main XCTU toolbar.



---

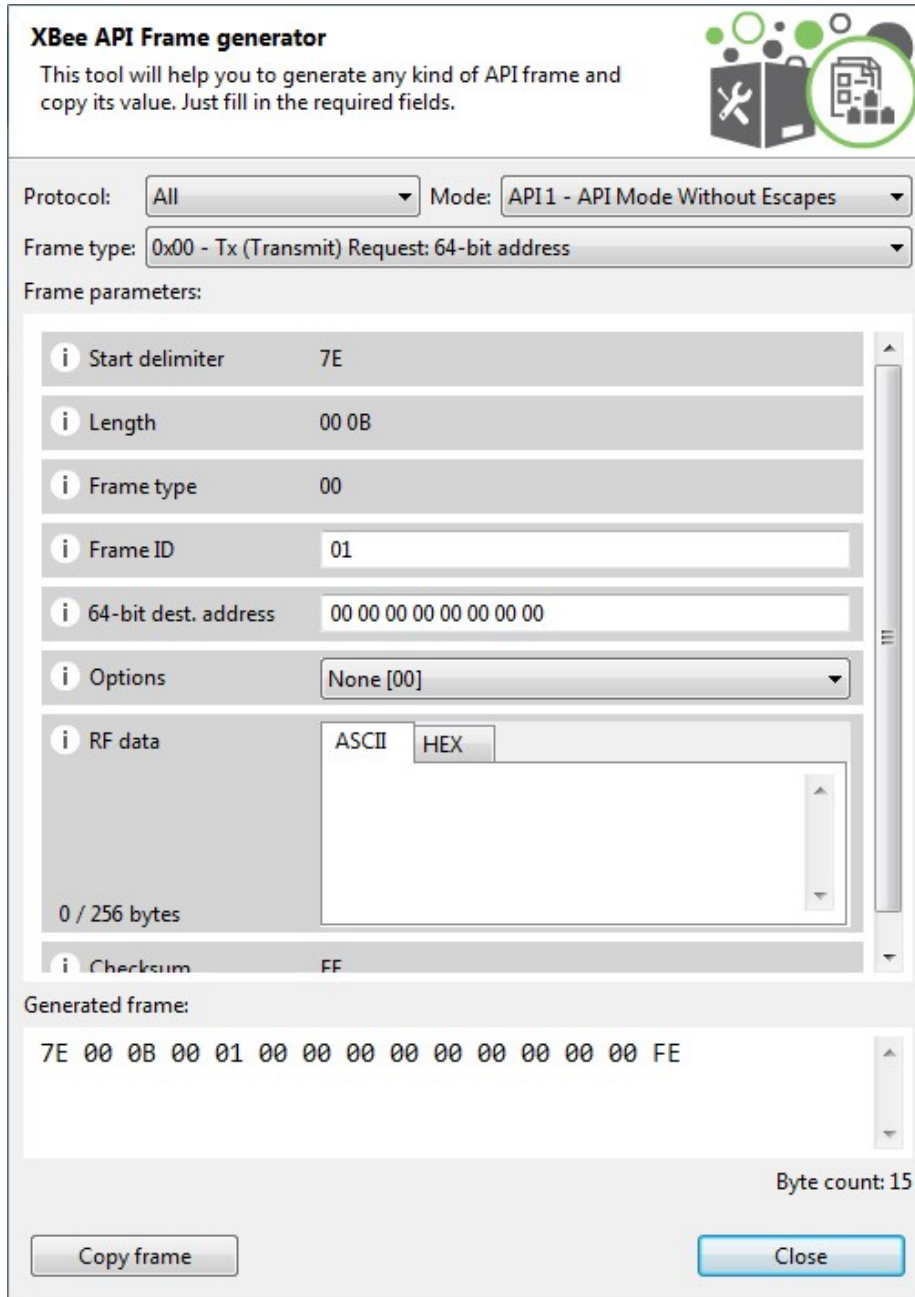
**Note** You can also access the Frames generator tool from the API console when you add a new API frame to the list of frames to send. Use the **OK** button to automatically copy the generated byte array to the content of the frame to be added.

---

## XBee API Frame generator dialog

Use the Frame generator dialog to generate an API frame. For instructions, see [Generate an API frame](#).





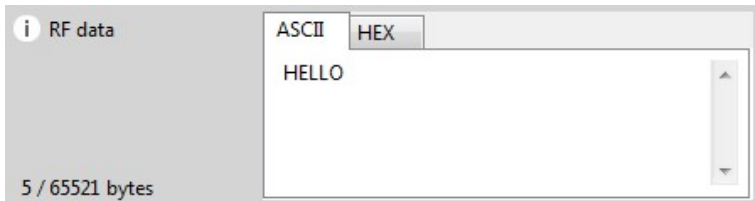
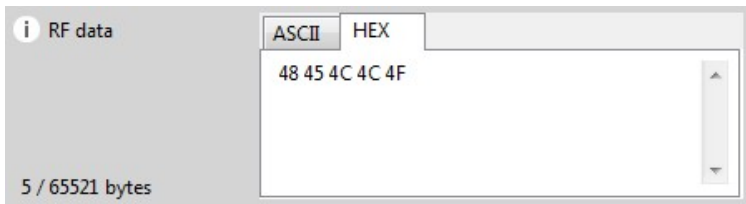
The following table provides descriptions of the fields in the XBee API Frame generator dialog.

Field	Description
Protocol	Radio protocol of the frame to be generated. Select <b>All</b> to display all frame types.
Mode	API mode (API or API escaped) of the frame to be generated.
Frame type	API frame type of the frame to be generated.
Frame	Options are dependent upon the selected API frame. Hover over the information icon

Field	Description
parameters	next to each setting to view a short description of the setting and its parameters.
Generated frame	Byte array result of API frame generation. You can select and copy the text or use the <b>Copy frame</b> button.


### ASCII/HEX conversion

Settings with text boxes have a tab control with two tabs, HEX and ASCII, that allow you to fill the setting with ASCII or HEX values. If you enter a value in HEX, the same value is represented in the ASCII tab, and vice versa.



### Generate an API frame

Follow these instructions to generate an API frame (output or input). For more information, see the [Frames generator tool](#).

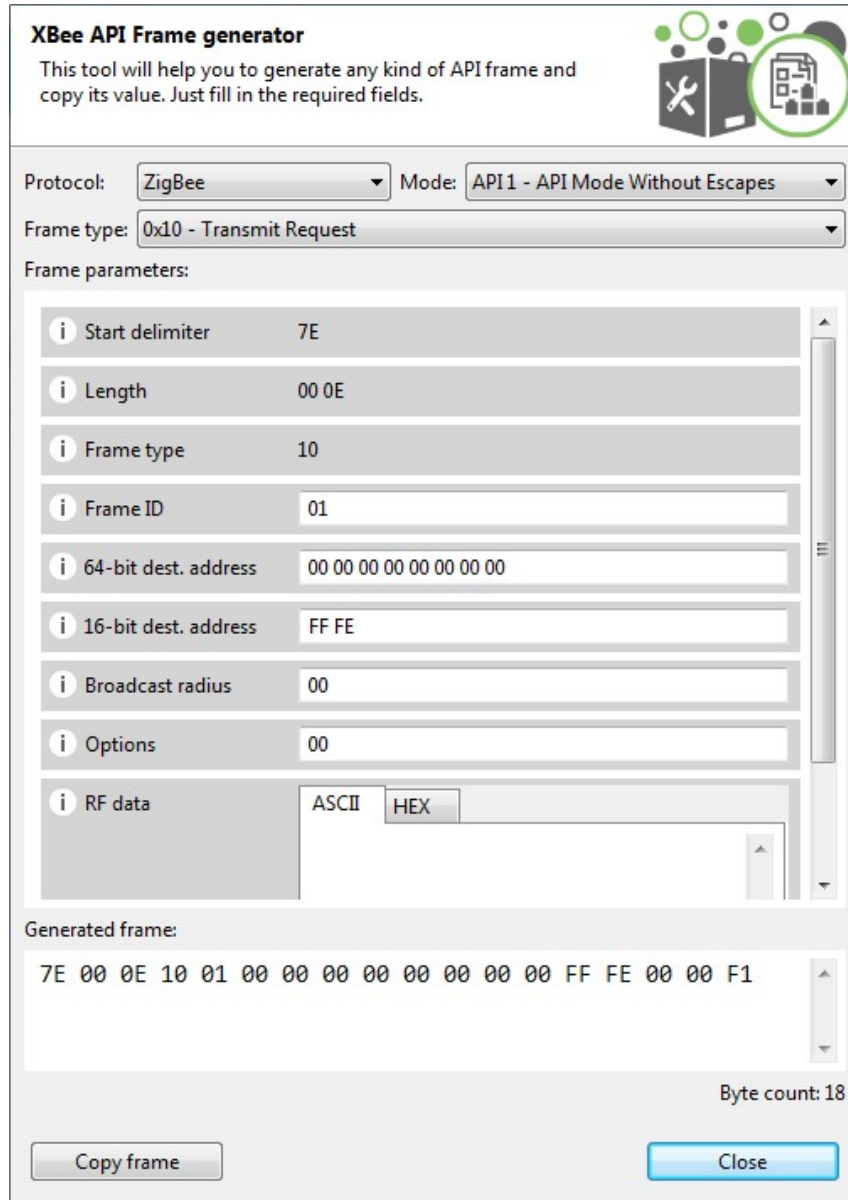
1. Launch the XBee API Frame generator tool by selecting **Frames generator** from the **Tools** drop-down menu  on the main toolbar. The **XBee API Frame generator** dialog appears.
2. For **Protocol**, select a radio protocol to display its corresponding API frames. If you select **All**, all the frame types are displayed.
3. For **Mode**, select the API mode (API or API Escaped) of the frame to generate.

---

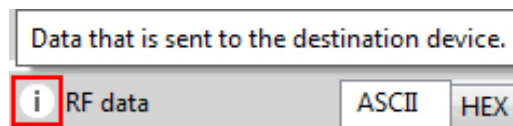
**Tip** To learn more about these modes, see [API operating mode](#) and [API escaped operating mode](#).

---

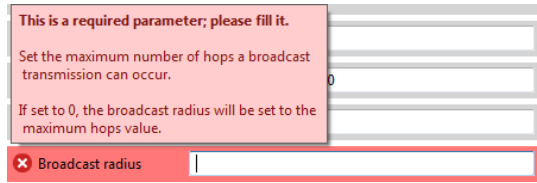
4. For **Frame type**, select the API frame type you want to generate.  
The selected frame's settings appear in the **Frame parameters** area. In this example, we selected the **ZigBee** protocol and the **ZigBee Transmit Request** API frame type.



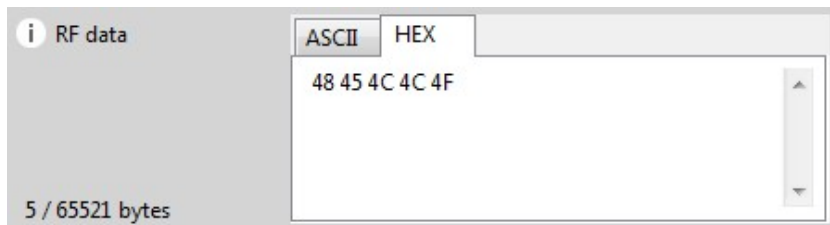
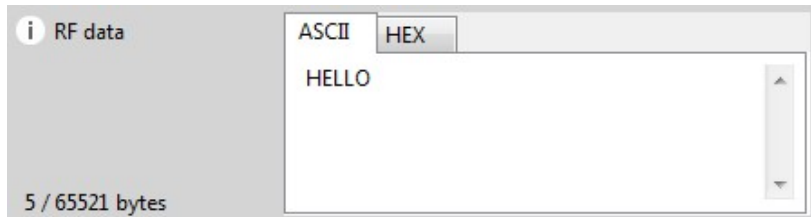
- 5. Hover over a setting's information icon to see a short description of the setting and its parameters.



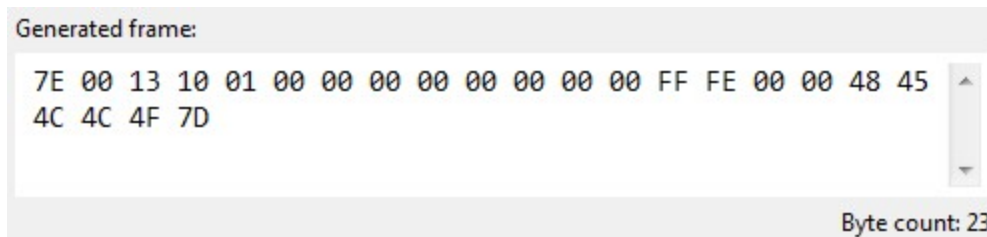
- 6. If any setting is not configured correctly, its background color changes to red and the information icon is replaced by a red cross. Hover over the icon to display the error message.



- 7. For **RF data**, you can enter a value in ASCII or HEX. If you enter a value in HEX, the same value is represented in the ASCII tab, and vice versa.



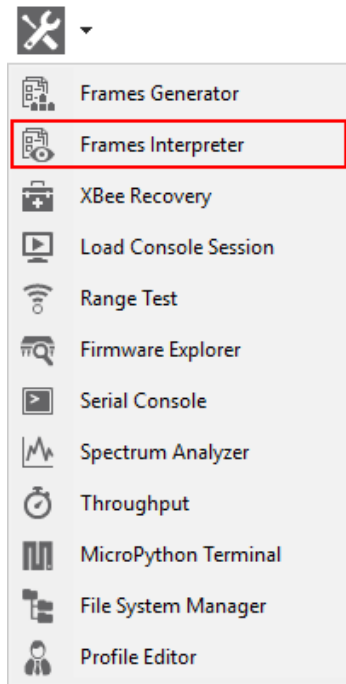
- 8. After you fill in all of the settings correctly, the frame is generated automatically and the byte array of the frame is displayed in the **Generated frame** box. You can copy it from here or click the **Copy frame** button.



## Frames interpreter tool

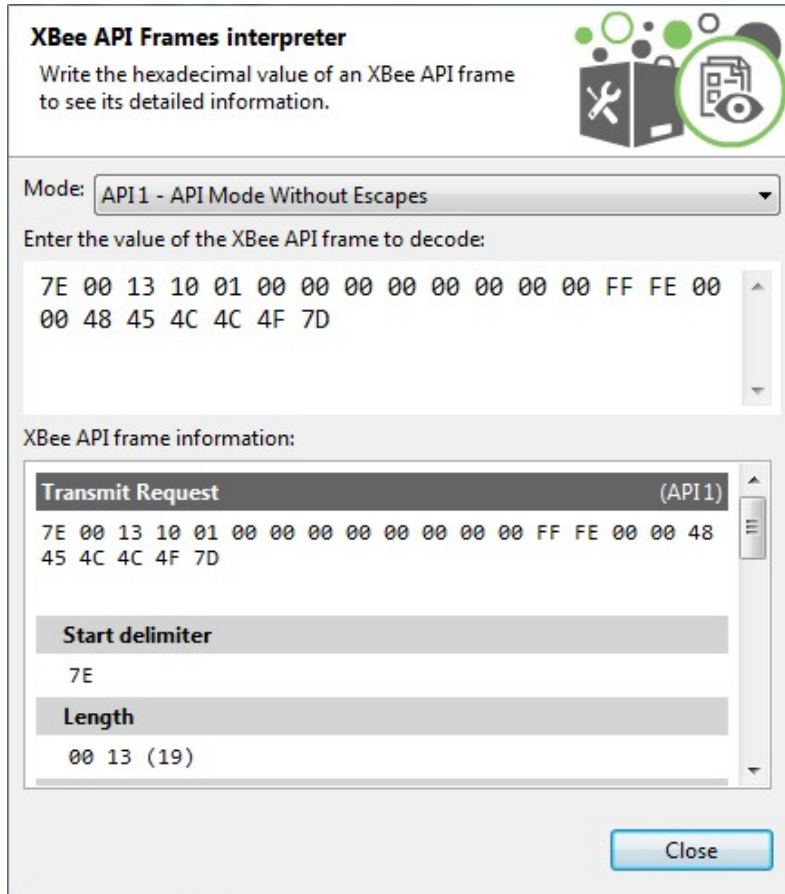
The **Frames interpreter** tool decodes the byte array of an API frame and displays it in a set of fields with corresponding values. The number of frame fields depends on the API frame type.

To open the **Frames interpreter** tool, select **Frames interpreter** from the **Tools** drop-down menu on the main XCTU toolbar.



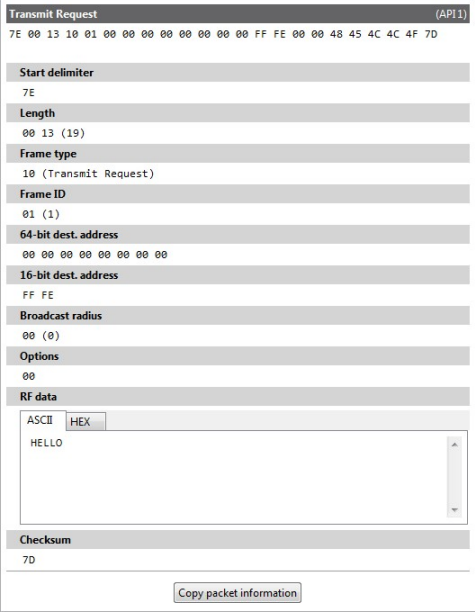
## XBee API Frames interpreter dialog

Use the Frames interpreter dialog to decode the byte array of an API frame.




The following table provides descriptions of the fields in the XBee API Frames interpreter dialog.

Field	Description
Mode	API mode or API escaped mode
API frame value	Byte array of the API frame to decode. The value of each byte must be hexadecimal without the "0x" prefix.
API frame information	Includes the type of API frame with all specific frame fields as well as the start delimiter, length, and checksum of the frame. Frames with an RF data field display the data in both Hexadecimal and ASCII formats.

Field	Description
	
Copy packet information	Copies the decoded API frame information to the clipboard in plain text.

## Decode a frame

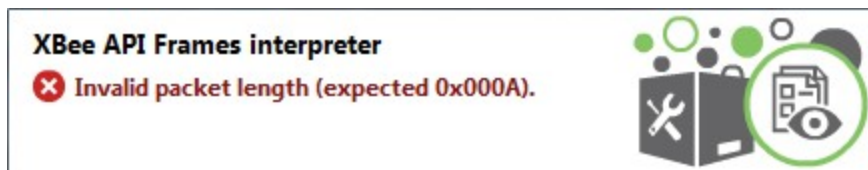
Follow these steps to decode a frame. For more information, see the [Frames interpreter tool](#).

1. Launch the Frames interpreter tool by selecting **Frames interpreter** from the **Tools** drop-down menu  on the main toolbar. The **XBee API Frame interpreter** dialog appears.
2. For **Mode**, select the API mode (API or API Escaped) of the frame to generate.

**Tip** To learn more about these modes, see [API operating mode](#) and [API escaped operating mode](#).

3. Below the API mode selection there are two text boxes. Enter the byte array of the API frame to decode in the top text box. The value of each byte must be hexadecimal and without the "0x" prefix.


If the byte array is not valid, the tool displays the corresponding error at the top.



If the byte array is valid, the bottom text box displays information about the API frame.

### **XBee API Frames interpreter**

Write the hexadecimal value of an XBee API frame to see its detailed information.



Mode: API 1 - API Mode Without Escapes

Enter the value of the XBee API frame to decode:

```
7E 00 13 10 01 00 00 00 00 00 00 00 00 00 FF FE 00
00 48 45 4C 4C 4F 7D
```

XBee API frame information:

<b>Transmit Request</b>	(API 1)
7E 00 13 10 01 00 00 00 00 00 00 00 00 00 FF FE 00 00 48 45 4C 4C 4F 7D	
<b>Start delimiter</b>	7E
<b>Length</b>	00 13 (19)

Close



**Transmit Request** (API1)

7E 00 13 10 01 00 00 00 00 00 00 00 FF FE 00 00 48 45 4C 4C 4F 7D

**Start delimiter**  
7E

**Length**  
00 13 (19)

**Frame type**  
10 (Transmit Request)

**Frame ID**  
01 (1)

**64-bit dest. address**  
00 00 00 00 00 00 00 00

**16-bit dest. address**  
FF FE

**Broadcast radius**  
00 (0)

**Options**  
00

**RF data**

ASCII  HEX

HELLO

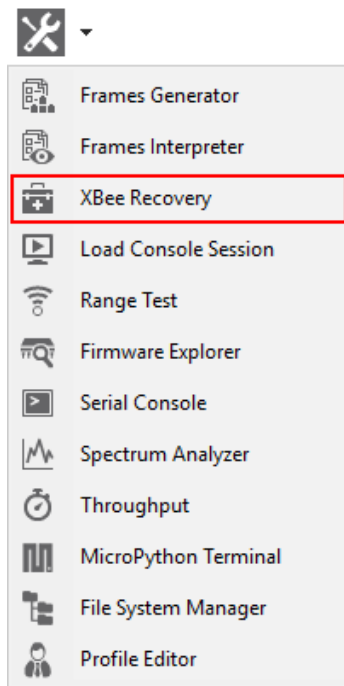
**Checksum**  
7D

Copy packet information

## XBee recovery tool

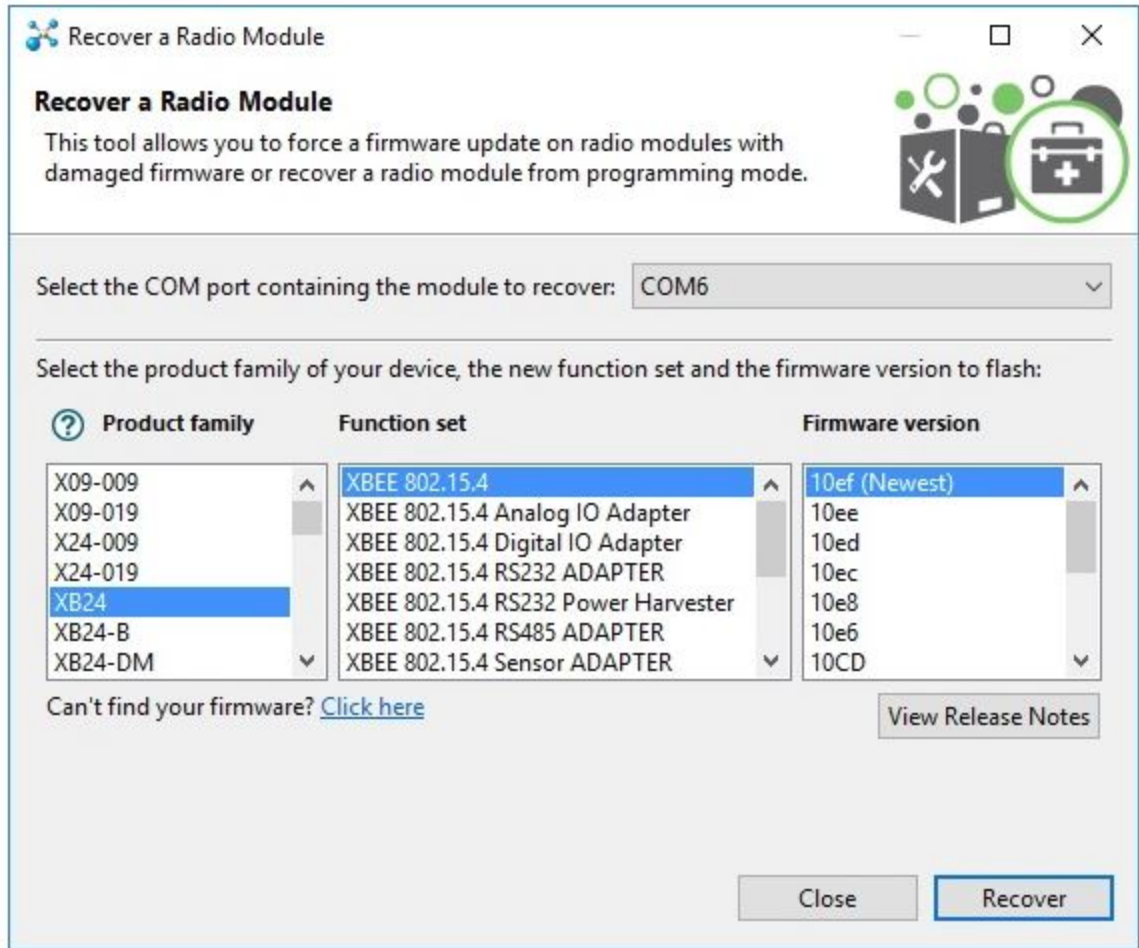
The **XBee recovery** tool is an embedded tool that you can use to force a firmware update on radio modules with damaged firmware. You can also use this tool to recover a radio module from programming mode.

To open the **XBee recovery** tool, select **XBee recovery** from the **Tools** drop-down menu on the main XCTU toolbar.




## Recover a radio module dialog

Use the Recover a radio module dialog to force a firmware update on radio modules or recover radio modules. For instructions, see [Recover a radio module](#).



The following table provides descriptions of the fields in the Recover a radio module dialog.

Field	Description
COM port	Serial port where the module to be recovered is connected
Product family	Product family of the module to be recovered. If you do not know the product family, check the label on the back of the device. 
Function set	Determines the available functionality such as mode of operation; end device, router, or coordinator designation; special sensor and adapter settings; etc. Available options depend on the module selected.
Firmware	Firmware version to flash to the module to be recovered. The most recent firmware

Field	Description
version	version appears at the top of the list.
View release notes	Enabled if the selected firmware has release notes available. Click to open a new dialog containing release notes for the selected firmware.




Programmable XBee radio modules and the XLR PRO cannot be recovered using the **XCTU recovery** tool.

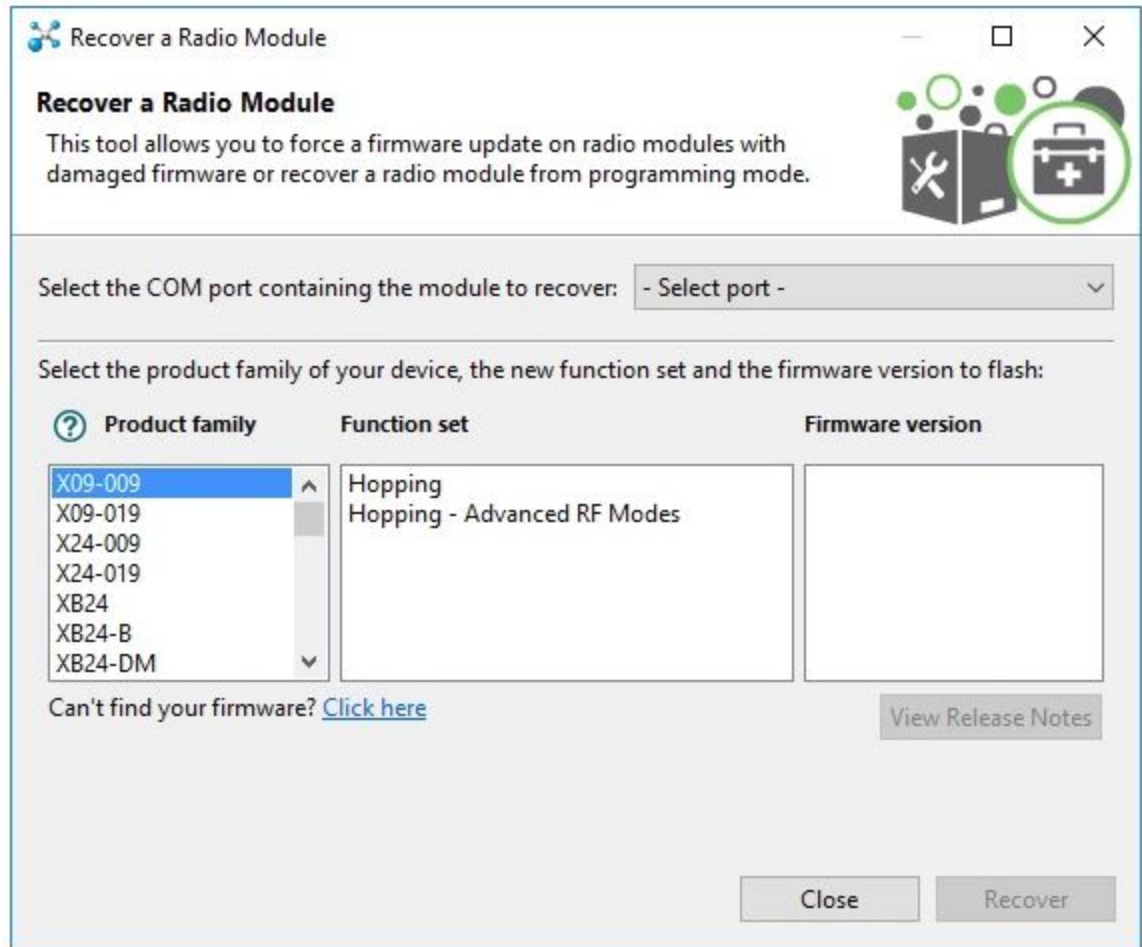
**Note** The recovery process may require you to reset the radio module. Reset the module and wait for the dialog to close; the recovery process will continue after the module is reset.

### Recover a radio module

Follow these steps to recover radio modules that have damaged firmware or are in programming mode. For more information, see the [XBee recovery tool](#).

**Note** You cannot use the recovery tool to recover programmable radio modules or the XLR-PRO.

1. Launch the XBee recovery tool by selecting **XBee recovery** from the **Tools** drop-down menu  on the main toolbar. The **Recover a radio module** dialog appears.



2. Select the serial port where the damaged radio module is attached.
3. Select the firmware you want to use for recovery. Specify the product family, function set, and version.
4. Click **Recover** to program the new firmware in the radio module. Click **Show Details** from the progress box to view a detailed progress log.
5. If XCTU displays a dialog asking you to reset your radio module, reset the module and wait for the dialog to close.
6. If the recovery is successful, click **OK** to close the validation message. Errors are displayed as separate error messages and are also logged in the progress log.

---

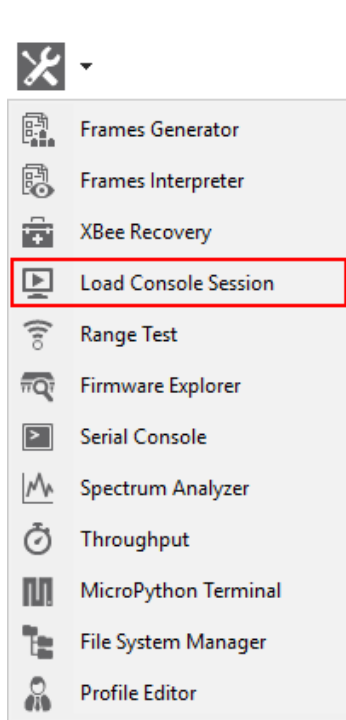
**Note** You can click **View Release Notes** if the selected firmware has available release notes.

---

## Load console session tool

The **Load console session** tool allows you to load a saved API or AT console session to review the recorded API frames or data. You can load a session saved directly from the frames/packets log of the console or a session generated using the recording feature.

To open the **Load console session** tool, select **Load console session** from the **Tools** drop-down menu on the main XCTU toolbar.



## Console session viewer dialog


The Console session viewer dialog is where you can load a saved API or AT console session. For instructions, see [Load a console session](#).

If the console session file successfully loads, the tool displays three new controls inside the Console session viewer:

- [Radio module information box](#)
- [Console session control](#)
- [Session data splitting](#)

**Console session viewer**

This tool allows you to load and read an XBee API or AT console sessions.



---

Load an API or AT console session file:



**Record date:** 11-04-2015 14:14:30.236

**Module ID:**

**Module address:** 0013A20040D2B039

**Firmware function:** XBee DigiMesh 2.4

**Firmware version:** 8073

**Port configuration:** COM1 - 9600/8/N/1/H

---

**Frames log**

ID	Time	Le...	Frame
0	14:14:02.1...	4	AT Command
1	14:14:02.2...	6	AT Command Response
2	14:14:02.2...	4	AT Command
3	14:14:02.5...	7	AT Command Response
4	14:14:02.5...	4	AT Command
5	14:14:02.6...	6	AT Command Response
6	14:14:02.6...	4	AT Command
7	14:14:02.6...	6	AT Command Response
8	14:14:02.6...	4	AT Command
9	14:14:02.7...	6	AT Command Response
10	14:14:02.7...	4	AT Command
11	14:14:02.8...	6	AT Command Response
12	14:14:02.8...	4	AT Command
13	14:14:02.9...	9	AT Command Response
14	14:14:02.9...	4	AT Command

**Frame details**

**AT Command** (API1)

7E 00 04 08 01 43 48 6B

**Start delimiter**

7E

**Length**

00 04 (4)

**Frame type**

08 (AT Command)

**Frame ID**

01 (1)

**AT Command**

43 48 (CH)

**Checksum**

Loaded frames from 0 to 157 of 157

**Radio module information box**

The radio module information box displays information about the radio module used in the session.



**Record date:** 11-04-2015 14:14:30.236

**Module ID:**

**Module address:** 0013A20040D2B039

**Firmware function:** XBee DigiMesh 2.4


**Firmware version:** 8073

**Port configuration:** COM1 - 9600/8/N/1/H

Field	Description
Icon	Represents the protocol of the radio module

Field	Description
Record date	The date the console session was saved
Module ID	The name or identifier of the radio module
Module address	The physical (MAC) address of the device
Firmware function	The function of the firmware running in the radio module
Firmware version	The version of the firmware running in the radio module
Port configuration	The radio module port statistics from the session

**Note** If the session you attempt to load corresponds to a Serial console session, the Radio module information box displays only the Record date and Port configuration fields.


**Record date:** 02-24-2015 11:17:55.233  
**Port configuration:** COM4 - 9600/8/N/1/N

### Console session control

The console session control varies depending on the type of console session. If the saved console session is an AT (transparent) or Serial session, the control is the same as the one in the Data traffic monitoring section of the AT or Serial consoles. In other words, it is a data text box with the hexadecimal representation of the data.

**Console log** 0x

+++OK		2B 2B 2B 4F 4B 0D
ATID		41 54 49 44 0D
D161		44 31 36 31 0D
ATSC		41 54 53 43 0D
7FFF		37 46 46 46 0D
ATSD		41 54 53 44 0D
3		33 0D
ATZS		41 54 5A 53 0D
0		30 0D
ATNJ		41 54 4E 4A 0D
FF		46 46 0D
ATOP		41 54 4F 50 0D
D161		44 31 36 31 0D
ATOI		41 54 4F 49 0D
---		---

If the saved console session is an API session, XCTU displays an API frames table with an API frame details view attached on the right side. This is the same control as the one in the API frames traffic monitoring section of the API Console.



ID	Time	Le...	Frame
0	14:14:02.1...	4	AT Command
1	14:14:02.2...	6	AT Command Response
2	14:14:02.2...	4	AT Command
3	14:14:02.5...	7	AT Command Response
4	14:14:02.5...	4	AT Command
5	14:14:02.6...	6	AT Command Response
6	14:14:02.6...	4	AT Command
7	14:14:02.6...	6	AT Command Response
8	14:14:02.6...	4	AT Command
9	14:14:02.7...	6	AT Command Response
10	14:14:02.7...	4	AT Command
11	14:14:02.8...	6	AT Command Response
12	14:14:02.8...	4	AT Command
13	14:14:02.9...	9	AT Command Response
14	14:14:02.9...	4	AT Command

**AT Command** (API1)

7E 00 04 08 01 43 48 6B

**Start delimiter**

7E

**Length**

00 04 (4)

**Frame type**

08 (AT Command)

**Frame ID**

01 (1)

**AT Command**

43 48 (CH)

**Checksum**

Regardless of the control displayed, you can only review the session data; any other functionality with the controls is disabled in this tool.

### Session data splitting

If the data to be loaded is too long for XCTU to display all at once, the data is split into blocks of bytes (when the session is an AT or Serial session) or blocks of frames (when the session is an API session). Depending on which console you used to save the session, XCTU displays a **Frames log viewer** (API) or a **Console log viewer** (AT and Serial Console).

Prev. bytes
Next bytes

Loaded bytes from 0 to 1499 of 5148

Prev. frames
Next frames

Loaded frames from 0 to 1200 of 2244

**Note** You can also use the console session scroll bar to load the next and previous blocks of data. The next block of data loads automatically when you scroll to the bottom of the control and scroll down again.

### Load a console session

Follow these steps to load a saved API or AT console session. For more information, see the [Load console session tool](#).

1. Launch the Load console session tool by selecting **Load console session** from the **Tools** drop-down menu on the main toolbar. The **Console session viewer** dialog opens.
2. Click **Browse**. An **Open file** dialog box asks you for a saved XML console session file.


3. Navigate to your saved console session file and click **Open**.

**Note** You can open log sessions in XML format that were generated with previous versions of XCTU. Although saving sessions in XML is no longer supported, XCTU maintains backward compatibility.

If the tool is not able to load the selected file, an error states the cause of the problem. The **Console session viewer** displays a loaded console session.

**Console session viewer**


This tool allows you to load and read an XBee API or AT console sessions.



---

Load an API or AT console session file:

C:\console\_sessions\api\_console\_session\_2015-11-04.log Browse...



**Record date:** 11-04-2015 14:14:30.236

**Module ID:**

**Module address:** 0013A20040D2B039

**Firmware function:** XBee DigiMesh 2.4

**Firmware version:** 8073

**Port configuration:** COM1 - 9600/8/N/1/H

**Frames log**

ID	Time	Le...	Frame
0	14:14:02.1...	4	AT Command
1	14:14:02.2...	6	AT Command Response
2	14:14:02.2...	4	AT Command
3	14:14:02.5...	7	AT Command Response
4	14:14:02.5...	4	AT Command
5	14:14:02.6...	6	AT Command Response
6	14:14:02.6...	4	AT Command
7	14:14:02.6...	6	AT Command Response
8	14:14:02.6...	4	AT Command
9	14:14:02.7...	6	AT Command Response
10	14:14:02.7...	4	AT Command
11	14:14:02.8...	6	AT Command Response
12	14:14:02.8...	4	AT Command
13	14:14:02.9...	9	AT Command Response
14	14:14:02.9...	4	AT Command

**Frame details**

**AT Command (API1)**

7E 00 04 08 01 43 48 6B

**Start delimiter**  
7E

**Length**  
00 04 (4)

**Frame type**  
08 (AT Command)

**Frame ID**  
01 (1)

**AT Command**  
43 48 (CH)

**Checksum**

Prev. frames
Loaded frames from 0 to 157 of 157
Next frames

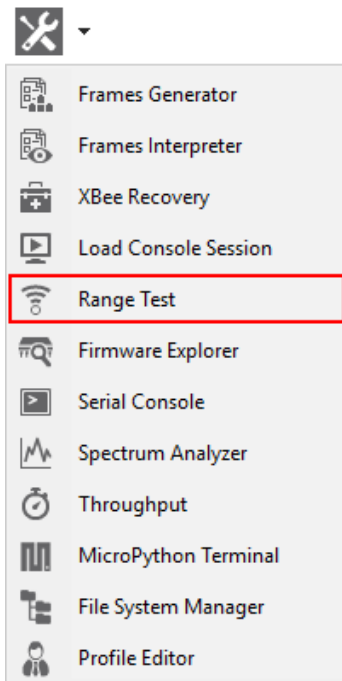
Close

## Range test tool

The Range test tool is an embedded tool that tests the real RF range and link quality between two radio modules in the same network. To perform a range test, you must have at least one local radio

module connected to your computer and added to XCTU, and a remote device in the same network configured as the local device. See [Range test](#).

To open the Range test tool, select **Range test** from the **Tools** drop-down menu on the main XCTU toolbar.




## Radio range test dialog

Use the Radio range test dialog to perform a range test. For instructions, see [Perform a range test](#).

### Radio Range Test

The Range Test utility is used to test the real RF range and link quality between two radio modules in the same network. Before starting the Range Test session you need to select a local device and a remote one or specify a remote destination address.



---

#### Device Selection

Select the local radio device:

	0013A20040A9E805	HULK	ZigBee	AT
	0013A20040A9E85B	ULTRON	ZigBee	API1
	0013A20040A9E81B	HULKBUST...	ZigBee	API1

Select the remote radio device:

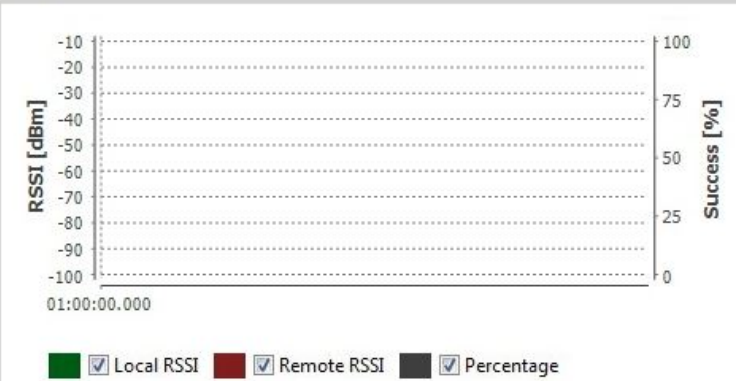
Discovered device: No devices discovered

Specify 64-bit address:

Specify 16-bit address:

---

#### Range Test



01:00:00.000

Local RSSI  
  Remote RSSI  
  Percentage

##### Configuration

Range Test type: Cluster ID 0x12

Packet payload: Configure Payload...

Rx timeout (ms): 1000

Tx interval (ms): 1000


Number of packets: 100

Loop infinitely


Skip RSSI measurements

Time window: Show all

**Local:** -110 dBm



**Remote:** -110 dBm



**Packets sent** 0%


0

Tx errors: 0

**Packets received** 0

0

Packets lost: 0



✖ Close
▶ Start Range Test

The following sections provide descriptions of the controls and fields in the Radio Range Test dialog.

### Device Selection


Use the **Device Selection** section to designate the local device that will perform the range test and the remote device the range test will be performed upon.

#### Select the local radio device:

Field/button	Description
Local device list	Populates with devices currently added to XCTU. The list provides the following information about each local device:

XCTU User Guide

164

Field/button	Description
	<ul style="list-style-type: none"> <li>▪ MAC address</li> <li>▪ Device name</li> <li>▪ Protocol</li> <li>▪ Operating mode</li> </ul>
	Discovers remote devices for the selected local device.

**Select the remote radio device:**

Field	Description
Discovered device	Displays a list of any remote devices discovered for the selected local device. Only devices whose protocols support node discovery list remote devices.
Specify 64-bit address	Allows a user to manually enter the 64-bit address of the destination device.
Specify 16-bit address	Allows a user to manually enter the 16-bit address of the destination device.

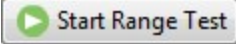
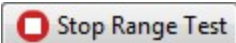
---

**Note** Not all protocols support 64- and 16-bit addressing.

---

## Range Test Configuration


You can configure the following range test options from within the Range test tool:

Field	Description
Range Test type	<ul style="list-style-type: none"> <li>▪ <b>Cluster ID 0x12:</b> Performed using explicit addressing frames/packets directed to the Cluster ID 0x12 on the data endpoint (0xE8), which returns the received data to the sender. Not all protocols and operating modes support the Cluster ID 0x12 range test. XCTU displays an error message for unsupported devices.</li> <li>▪ <b>Loopback:</b> Performed using the serial port/USB hardware loopback capabilities. Before initiating this range test, you must configure the remote module to work in transparent mode and close the loopback jumper. Works only with remote devices in AT operating mode.</li> </ul>
Packet payload	Allows you to configure the packet payload that will be sent to the remote device during range test. Opens a new window for ASCII/HEX input.
Rx Timeout (ms)	Sets the maximum time in milliseconds to wait for response from the remote device before considering a packet to be lost.
Tx interval (ms)	Sets the minimum time in milliseconds to wait before sending a new packet to the remote device.
Number of packets	Configures the number of packets to send in a single range test session.
Loop infinitely	When checked, configures the <b>Range test</b> tool to send packets infinitely until the session is stopped manually.
Skip RSSI measurements	When checked, the test skips the RSSI measurements. Note that you can only skip them when your local module works in <b>AT</b> (transparent) mode.
Time window	Configures the visible time window of the RSSI data measured by the range test.
	When you have configured all the options, click <b>Start Range Test</b> to start sending packets and measure the signal strength.
	You can stop the process any time by pressing the same button, now showing the text <b>Stop Range Test</b> .

## Data representation

XCTU represents range test data in three ways:

	Field	Description
	RSSI chart	Represents the RSSI values of the local and remote devices during the range test session. The line graph also contains the percentage of success for the total packets sent. You can hide/show chart data by checking or unchecking <b>Local RSSI</b> , <b>Remote RSSI</b> , and <b>Percentage</b> .
	Local RSSI value	Displays instant RSSI value of a local device, in dBm, for the last packet transmitted.
	Remote RSSI value	Displays instant RSSI value of a remote device in dBm, for the last packet transmitted. When the local module is working in

	Field	Description
		AT (transparent) mode, XCTU cannot read the remote device RSSI value.
 <p>The screenshot displays a summary of packet statistics. On the left, it shows 'Packets sent' as 87 with 'Tx errors: 0'. Below that, 'Packets received' is 86 with 'Packets lost: 0'. On the right, there is a blue progress bar labeled '100%' indicating the success rate.</p>	Packet summary	Displays total number of packets sent and received, transmission errors, and packets lost. It also displays the percentage of success sending and receiving packets during the range test session.

### Supported products

Range test is only supported in the following:

- SX
- XLR Pro Module
- ZigBee
- DigiMesh
- XTend
- XTend - DigiMesh
- XC/XSC
- 802.15.4
- DigiPoint

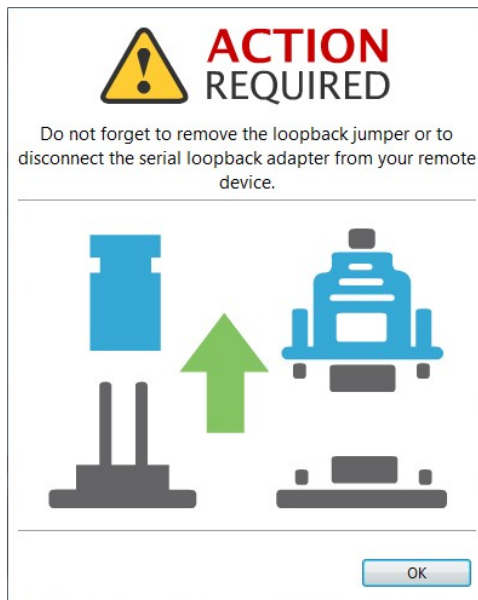
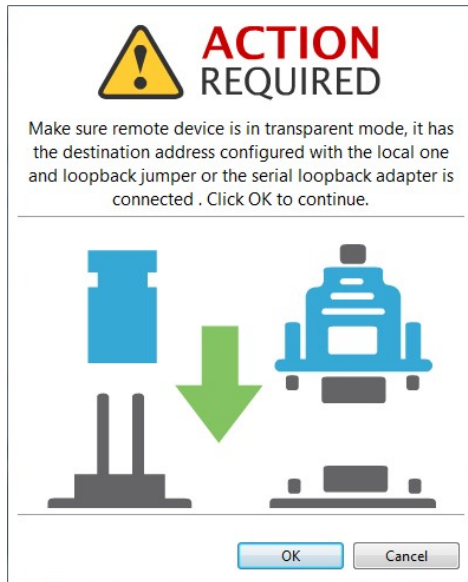
### Special considerations

There are some special considerations to be aware of while working with the Range test tool:

- When the local module is working in AT (transparent) mode, the tool automatically performs configuration steps in the module before starting the range test, and after finishing or stopping it. A progress bar shows the level of completion for this process.



- When the local module is working in AT (transparent) mode, it is not possible to read the remote device RSSI value.
- The Loopback range test type only works with remote devices in AT (transparent) operating mode.
- When performing a Loopback range test, you need to connect the loopback jumper or the loopback adapter in the remote device before starting and disconnect it after finishing. A dialog box notifies you of the action required:



## Range test

Since communication between XBee RF modules takes place over the air, the quality of the wireless signal can be affected by many factors: absorption, reflection of waves, line-of-sight issues, antenna style and location, etc. A range test demonstrates the real-world RF range and link quality between

two XBee modules in the same network. During a range test, XCTU send data packets from the local XBee module to the remote module and waits for the echo to be returned from the remote module to the local module. XCTU counts the number of packets sent and received by the local module and measures the signal strength of both sides as an RSSI (Received Signal Strength Indicator) value. Every sent packet from the local XBee should be received again as an echo by the same local XBee.

The following products support range testing:

- ZigBee
- DigiMesh
- XTend
- XTend - DigiMesh
- XC/XSC
- XLR PRO
- 802.15.4
- DigiPoint
- SX

---

**Note** When deploying an actual network, perform multiple range tests to analyze varying conditions in your application.

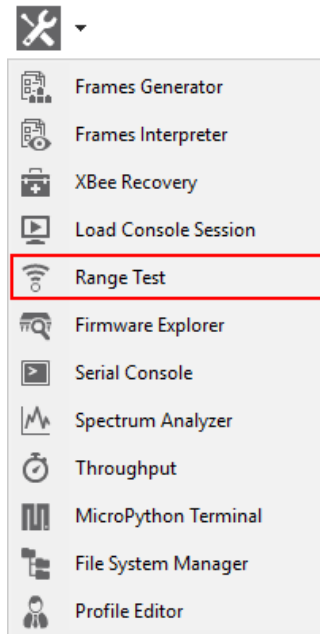
---

## Perform a range test

Follow these steps to perform a a range test. For more information, see the [Range test tool](#). When the local module is working in AT (transparent) mode, the Range test tool automatically pre-configures the module before starting the range test, as well as after the range test is completed or manually stopped. A progress bar displays the level of completion for this process.

To perform a range test, you must have at least one local radio module connected to your computer and added to XCTU, and a remote device in the same network as the local device.

1. Launch the Range test tool by selecting **Range test** from the **Tools** drop-down menu  on the main toolbar.



2. Select the local device that will perform the range test and the remote device against which the range test will be performed.
3. Configure the following range test parameters:
  - Range Test type
  - Packet payload
  - Rx timeout
  - Tx interval
  - Number of packets
  - Loop infinitely
  - Skip RSSI measurements
  - Time window
4. Click **Start range test** to begin sending packets and measuring signal strength. You can stop the range test at any time by clicking the **Stop range test** button.

---

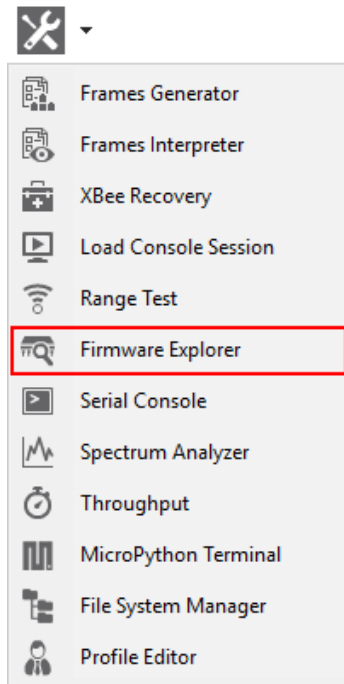
**Note** If the Loopback range test type is configured, you must connect the loopback jumper in the remote module before starting the range test and remove it when the range test has finished.

---

## Firmware explorer tool

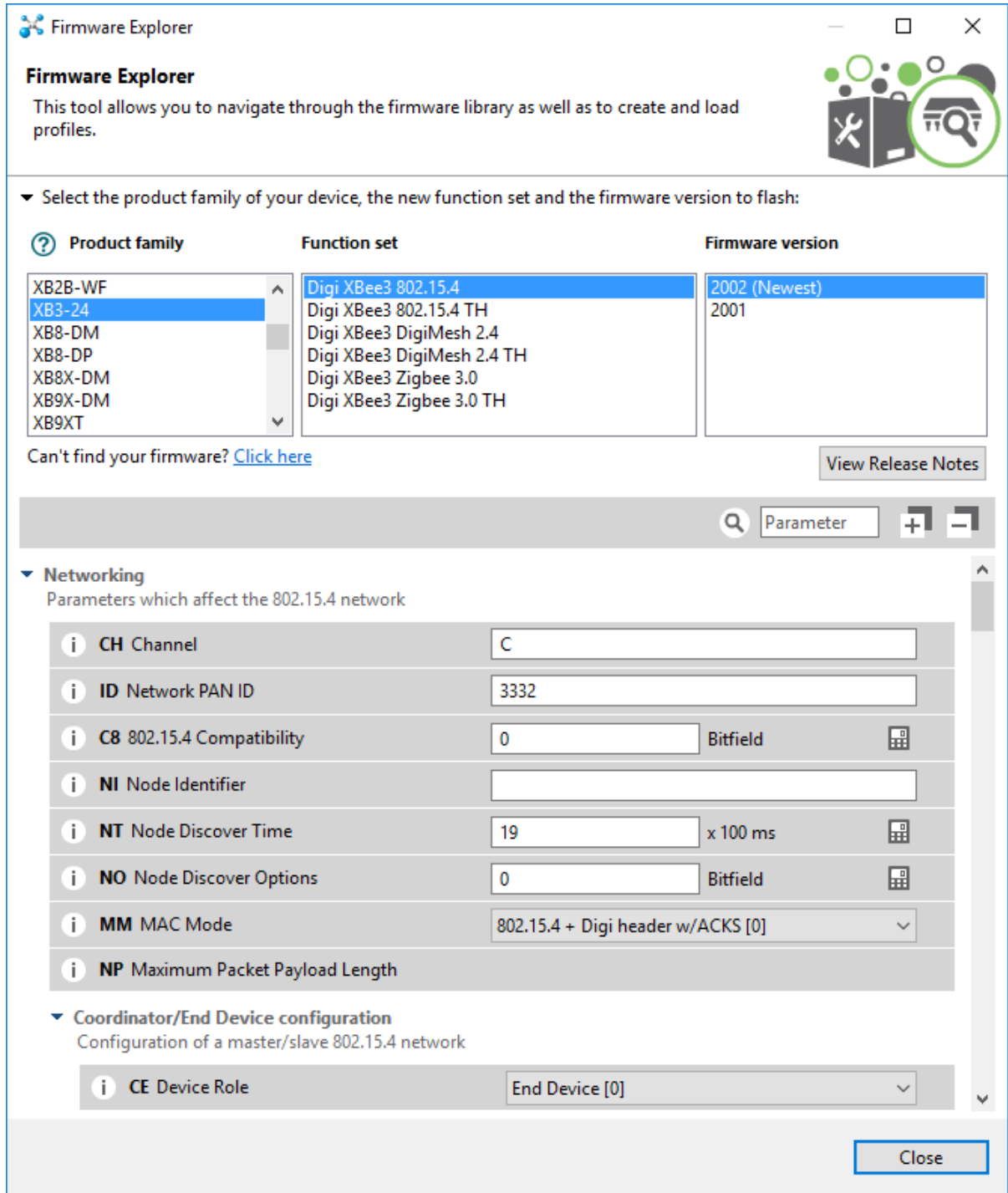
The **Firmware explorer** tool allows you to navigate through XCTU's firmware library to review available firmware versions and their settings, without having a radio module attached to your computer. It also lets you save and load firmware profiles.

To open the **Firmware explorer** tool, select **Firmware explorer** from the **Tools** drop-down menu on the main XCTU toolbar.



## Firmware explorer dialog

Use the Firmware explorer dialog to review available firmware versions and their settings. For instructions, see [Inspect a firmware version](#).



## Firmware selection panel



The firmware selection panel allows you to specify the firmware to be loaded. To display the settings of a specific radio firmware, specify the firmware family, function, and version from the corresponding lists.

**Note** The firmware selection panel does not flash the firmware on a module. It loads the firmware in the dialog so you can view its settings.

Field	Description
Product family	Product family of the firmware to load. If you do not know the product family, check the label on the back.
Function set	Determines the available functionality such as mode of operation; end device, router, or coordinator designation; special sensor and adapter settings; etc. Available options depend on the module selected.
Firmware version	Firmware version to be loaded. The most recent firmware version appears at the top of the list.
View release notes	Displays the release notes of the selected firmware, if available.

## Firmware toolbar

The toolbar displays all the actions you can perform on the selected firmware:

Field	Description
 Search	Enables you to search firmware settings by AT parameter.
 Expand/collapse settings	Expands or collapses all firmware settings sections.


## Firmware settings panel

The firmware settings panel is located below the firmware selection panel and contains all the settings that make up the firmware, in categories.

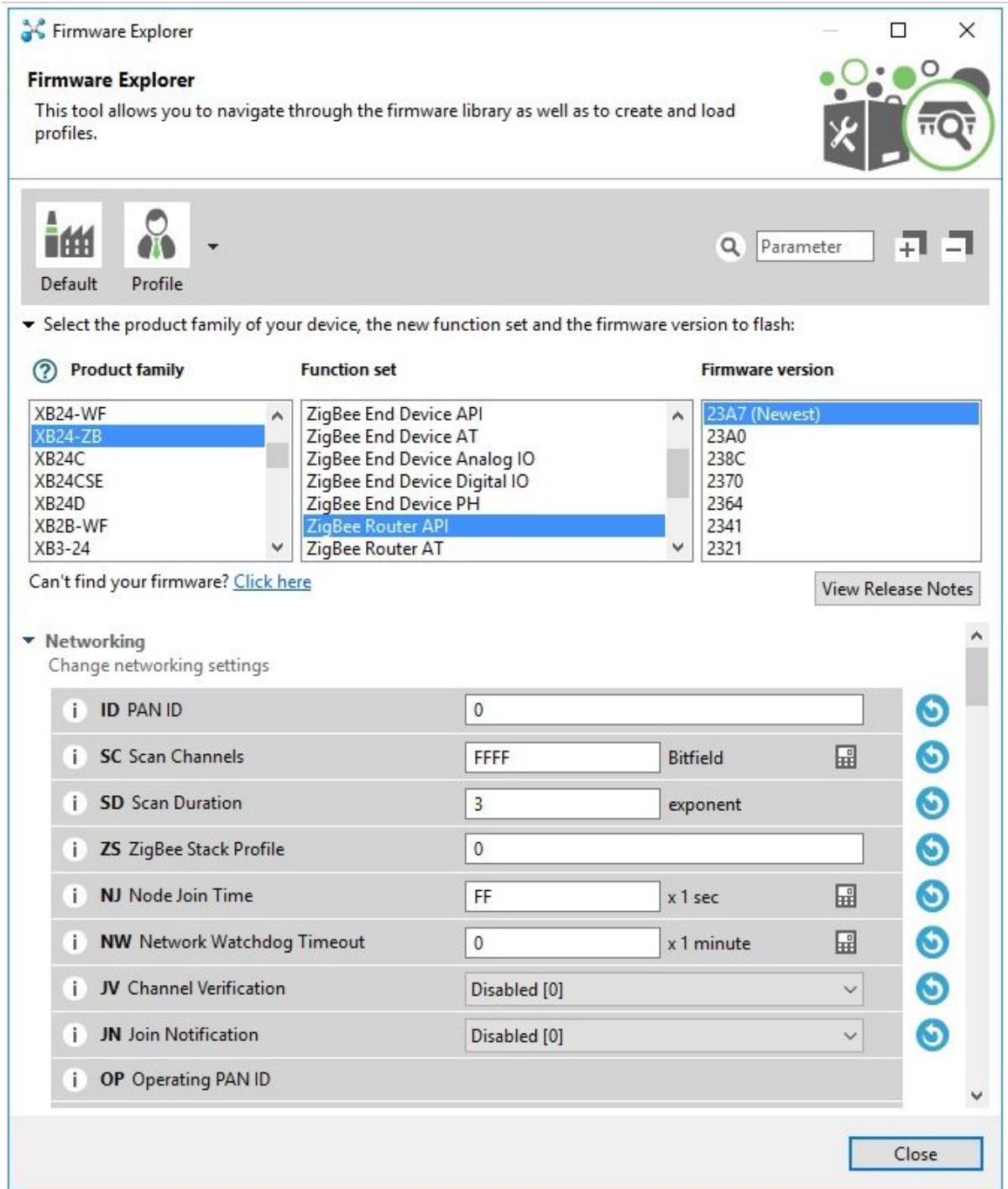
The settings are displayed and managed the same way as they are in the Radio configuration section. For more information, see [Configure your modules](#).

## Inspect a firmware version

Follow these steps to inspect a firmware version. For more information, see the [Firmware explorer tool](#).

1. Launch the Firmware explorer tool by selecting **Firmware explorer** from the **Tools** drop-down menu  on the main toolbar. The **Firmware Explorer** dialog displays all the available firmware hosted in XCTU.
2. Select the product family of the firmware, the function set, and the desired firmware version.

3. You can inspect available settings for the selected firmware version.

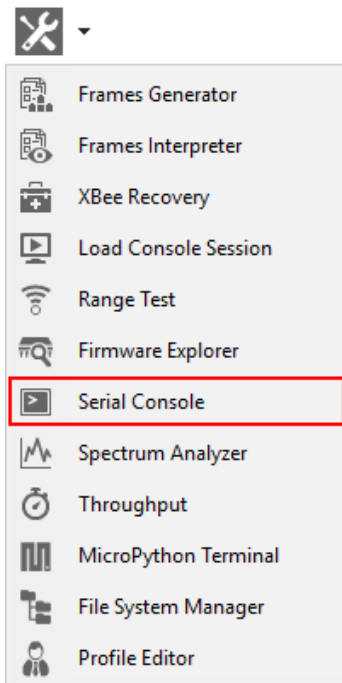


## Serial console tool

The **Serial console** tool allows you to interact with your radio modules without having to first discover them and add them to the list of radio modules. It is essentially an AT console, but instead of

associating directly with a module it communicates with the port attached to the module you want to communicate with.

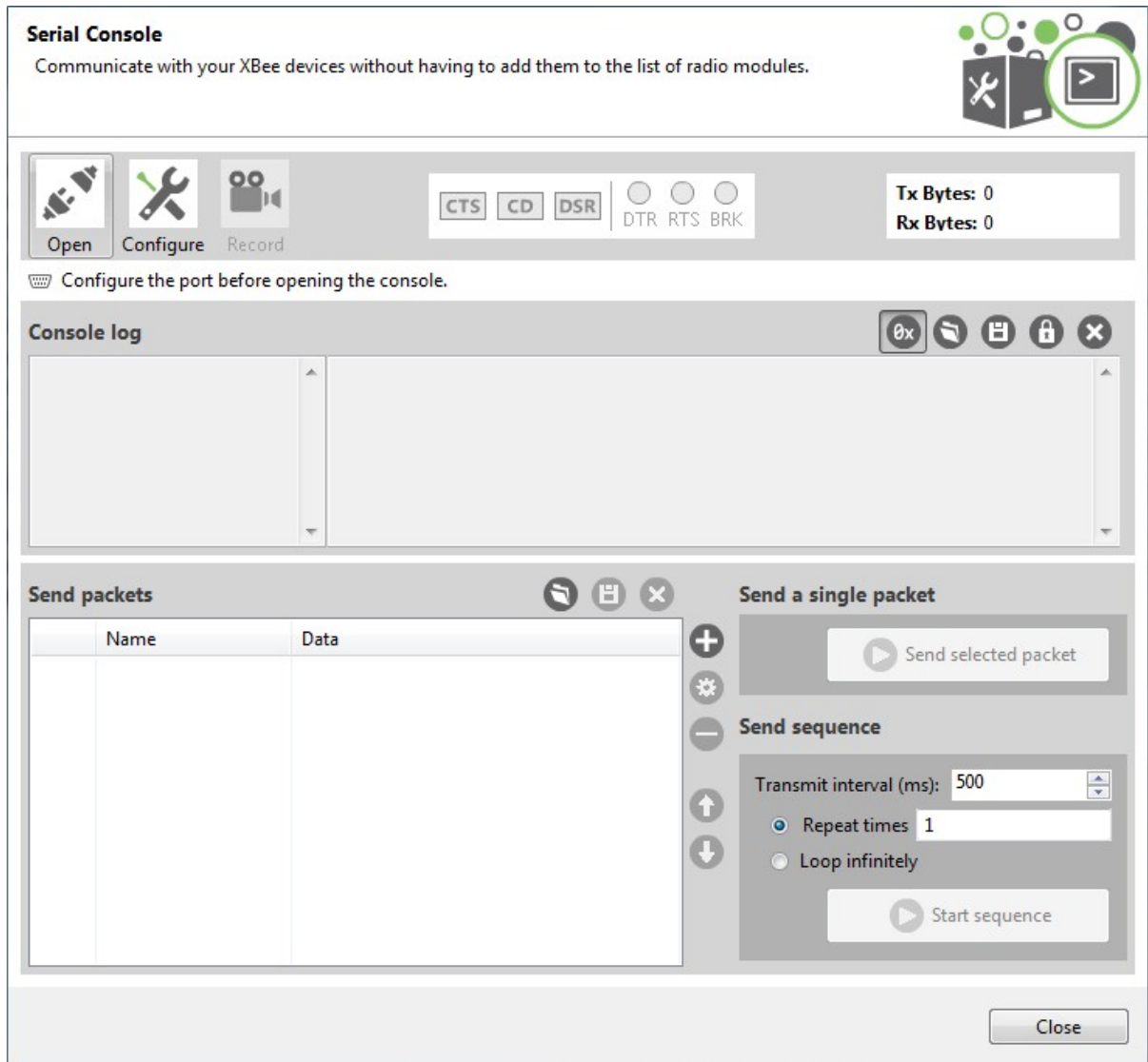
To open the **Serial console** tool, select **Serial console** from the **Tools** drop-down menu on the main XCTU toolbar.



## Serial console dialog

The workflow and components of the serial console are similar to those of the AT console. However, since the serial console is not associated with a radio module, you must configure the serial port the console will open a connection with. For instructions, see [Configure the serial port settings](#).







For more information on the console toolbar, see [Consoles working mode](#). For a description of **Console log** and **Send packets** dialogs, see [AT console](#).



## Open a serial console session

The first time you open a console, it is disconnected by default. This is indicated by the gray background of the toolbar and the status text reading "Closed."

Buttons	Description
	<p>Click the <b>Open</b> button to establish communication with the radio module corresponding to the console. If it is the first time you connect the console, you are prompted to configure the serial connection.</p> <p>The background color of the Open button changes to green and its text changes to Close. When the console is connected, all the data traffic of the radio module is captured by the console and displayed in the corresponding controls.</p>
	<p>Click the <b>Close</b> button if you want to disconnect the console from the module.</p>

## Configure the serial port settings

Follow these steps to configure the serial port settings of an instance of the Serial console tool. For more information, see [Serial console tool](#).


1. Click the **Configure** button  on the Serial Console toolbar. If the console is not yet connected, click the **Open** button .



The **Serial Port Configuration** dialog list all of the serial connection parameters to be configured.

### Serial Port Configuration

Select the COM port from the list and configure its settings.



Select the Serial/USB port:

COM1	Communications Port
COM3	Intel(R) Active Management Technology - SOL
COM6	USB Serial Port
COM13	USB Serial Port
COM17	USB Serial Port

Provide a port name manually:

Baud Rate:

Data Bits:

Parity:

Stop Bits:

Flow Control:

2. Select the device's serial port, or provide it manually. The rest of the parameters are loaded with their default values.
3. Change the configuration if needed. The most common serial configuration is:

**Baud rate:** 9600 or 115200

**Data bits:** 8

**Stop bits:** 1

**Parity:** None

**Flow control:** None

---

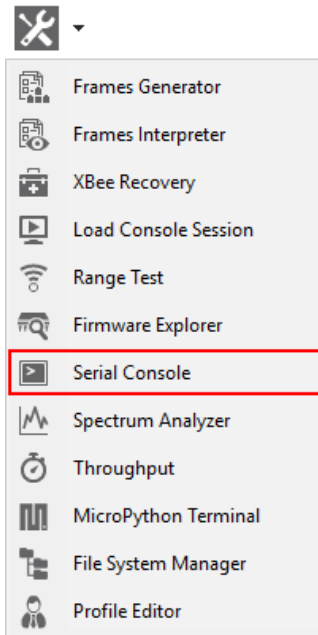
**Note** Custom baud rates can only be typed under Windows OS.

---

## Open a serial console session

The first time you open a console, it is disconnected by default. To open a serial console session:

1. Select **Serial console** from the **Tools** drop-down menu on the main XCTU toolbar. The **Serial console** dialog appears.



2. Click the **Configure** button from the toolbar and configure the serial port where the radio module is attached. See [Configure the serial port settings](#) for details.
3. Click the **Open** button to establish communication with the radio module.

---

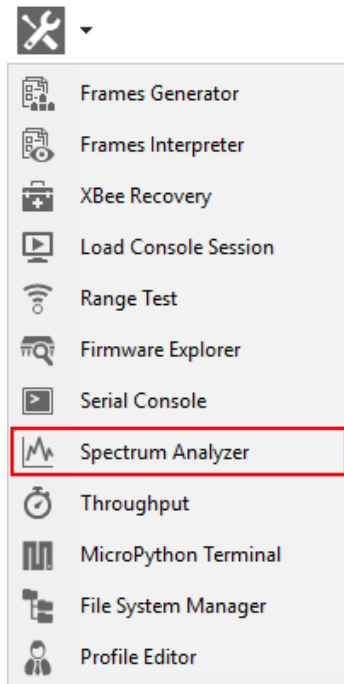
**Note** The serial console tool operates just like an AT console. For more information, see [Consoles working mode](#), [AT console](#), and [Serial console tool](#).

---

## Spectrum analyzer tool

The Spectrum analyzer tool allows you to test and measure the spectrum of the selected radio band. The tool reports the noise level of each channel indicating its best, worst and average measures.

To open the Spectrum Analyzer tool, select **Spectrum analyzer** from the **Tools** drop-down menu on the main XCTU toolbar.




## Spectrum analyzer dialog

Use the Spectrum analyzer dialog to analyze the spectrum of a radio band. For instructions, see [Analyze the spectrum of a radio band](#).


### Spectrum Analyzer

The Spectrum Analyzer utility is used to test and measure the spectrum of the selected radio band. The analysis reports the noise level of each channel indicating its best, worst and average measure. Before starting the Spectrum Analysis session you need to select a local device



#### Device Selection

Select the local radio device:

	0013A20040DBC61F	PETER PAN	DigiMesh	AT
---	------------------	-----------	----------	----

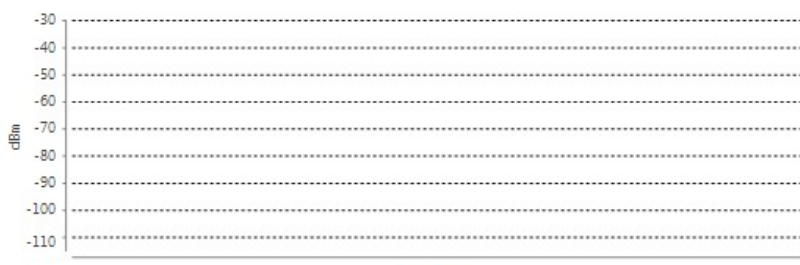
#### Configuration

Sampling interval (ms):

Number of samples:

Loop infinitely

#### Spectrum Analysis




Channels

Current value
 Max value
 Min value
 Average value

#### Channels

--	--

#### Channel values



- Current:
- Maximum:
- Minimum:
- Average:

#### Samples Read

0

### Device selection

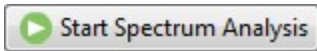
The Device selection area lists the devices that you have added to XCTU. Select the radio module you want to use to perform the analysis.

### Analysis configuration

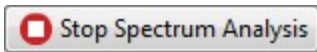
The Configuration area allows you to configure the spectrum analysis process.

Field	Description
Sampling interval (ms)	Determines the time to wait in milliseconds before reading a new noise level sample of the RF channels.
Number of	Select this option to configure the number of samples to read in the spectrum

Field	Description
samples	analysis session.
Loop infinitely	Select this option to read samples infinitely until the spectrum analysis session is stopped manually.



When you have configured all the options, click the **Start Spectrum Analysis** button to start reading samples and measure the noise level of each RF channel.

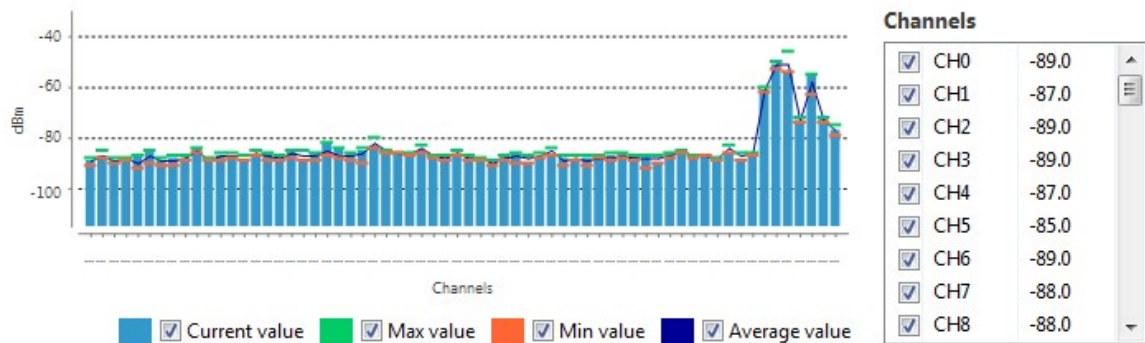


You can manually stop the analysis at any time by clicking the same button, which changes to **Stop Spectrum Analysis**.

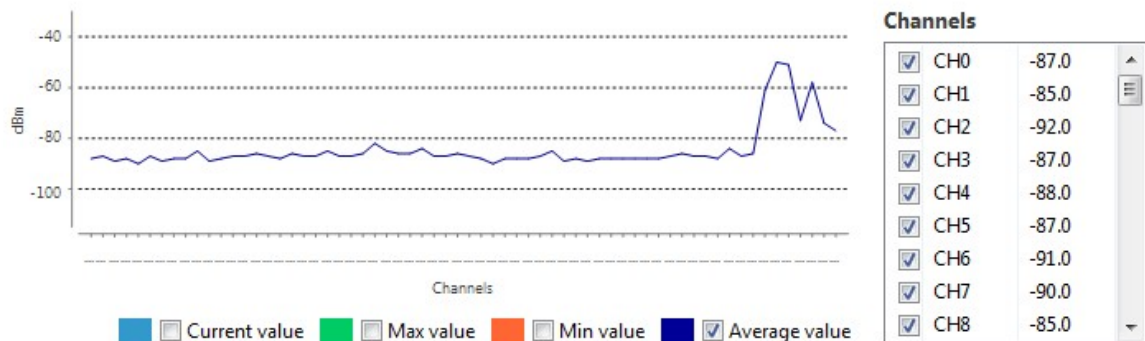
### Channels Chart

The Channels chart represents the noise level of all the RF channels. Each channel displays one bar with the current noise level and two little marks representing the highest noise level (green) and the lowest noise level (red). A blue line indicates the average noise level of all the channels. The spectrum analysis refreshes the noise levels of each channel continuously until the analysis ends or is stopped.

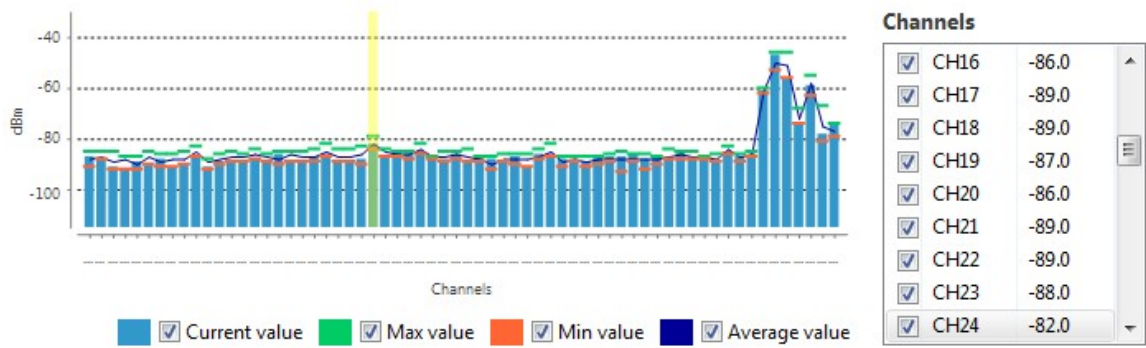
**Note** The list of supported channels may vary depending on the device type and device region.



You can use chart filters to hide or display the bars, the best and worst noise level values and the average noise level line.



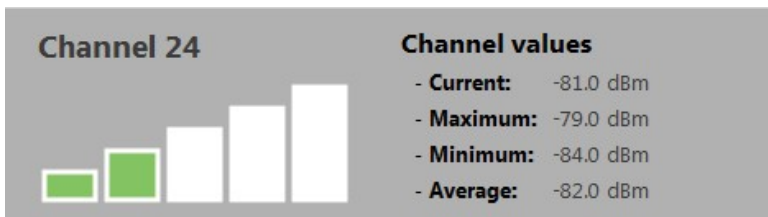
If you click a channel from the list of channels, it is highlighted in the chart. Likewise, you can select a channel in the chart to highlight it in the list of channels.



Selecting or clearing the check box of each channel within the list of channels shows and displays it in the chart.

### Channel summary values

The Channel area displays the instant noise level of a channel as well as its average, best and worst noise level. To display channel values, select a channel in the chart or highlight a channel in the list of channels on the right.






## Number of samples

The Samples Read area displays the number of noise level samples read by the selected radio module. Note that you can specify the number of samples to read in the Configuration area.

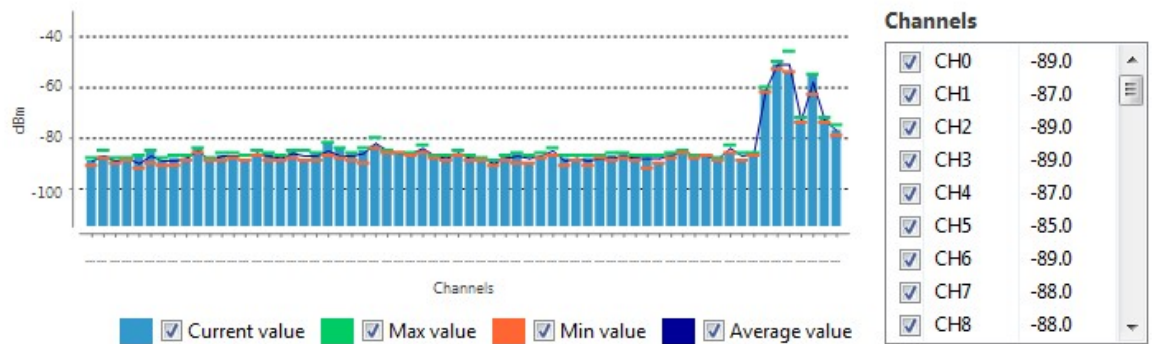


## Analyze the spectrum of a radio band

Follow these steps to analyze the spectrum of a radio band. For more information, see [Spectrum analyzer tool](#).

1. Launch the Spectrum analyzer tool by selecting **Spectrum analyzer** from the **Tools** drop-down menu  on the main toolbar. The **Spectrum Analyzer** dialog appears.
2. Under **Device Selection**, select the local radio device you want to analyze.
3. Under **Configuration**, specify the sampling interval and a configuration method.
4. Click the **Start Spectrum Analysis** button to start reading samples and measure the noise level of each RF channel. You can manually stop the analysis at any time by clicking the **Stop Spectrum Analysis** button.

When an analysis begins, the chart and channels list display the RF channels supported by the selected device.

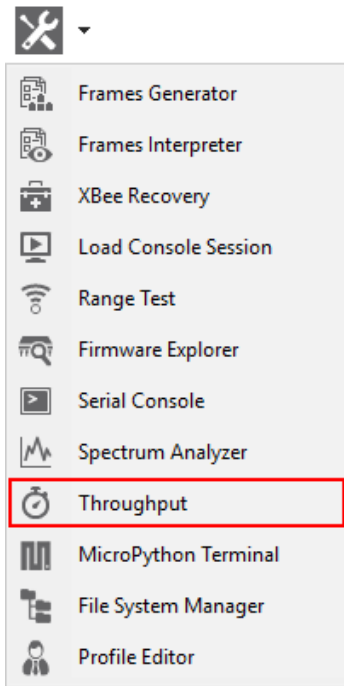


5. Use chart filters to hide or display channels, the best and worst noise level values and the average noise level line. You can also display channel values and samples read. For details, see [Spectrum analyzer tool](#).

## Throughput tool

The Throughput tool allows you to measure the transfer ratio between two radio modules in the same network. To perform a throughput measure, you need to have a local radio module added to XCTU and a remote device in the same network as the local one.

To open the Throughput tool, select **Throughput** from the **Tools** drop-down menu on the main XCTU toolbar.




## Throughput dialog

Use the Throughput dialog to measure the transfer ratio between two radio modules in the same network. For instructions, see [Measure the transfer ratio between two radio modules](#).

### Throughput

This tool allows you to test the transfer ratio from a radio module to another module in the same network. Before starting the Throughput session you need to specify the 2 devices that will be used in the process.



#### Device selection

Select the local radio device:

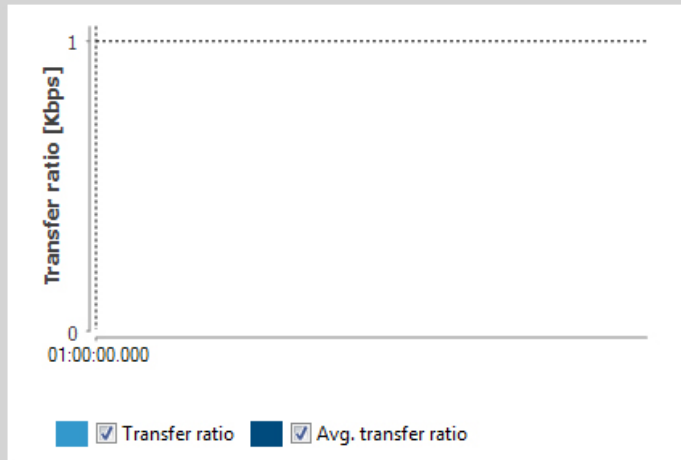
	0013A2004030FB70	BAYMAX	DigiMesh	AT
	0013A20040A9E81B	HULKBUST...	ZigBee	API 1
	0013A20040D2B039	HIRO	DigiMesh	AT

Select the remote radio device:

Remote selection: Discovered device

No devices discovered

#### Throughput



Transfer ratio [Kbps]

0 1

01:00:00.000

Transfer ratio  Avg. transfer ratio

#### Configuration

Throughput type: Unidirectional

Packet payload: Configure payload...

Duration:

Time (s): 10

Packets: 1

Loop infinitely

Time window: Show all

Instant transfer ratio: **0.0 Kbps**

Average transfer ratio: **0.0 Kbps**

Tx packets: 0

Tx bytes: 0

Elapsed time: 00:00:00

Close Start Throughput

## Device selection

Use the **Device selection** section to designate the local device that will perform the throughput and the remote device that receives the data:

Select the local radio device:


	0013A2004030FB70	BAYMAX	DigiMesh	AT
	0013A20040A9E81B	HULKBUST...	ZigBee	API 1
	0013A20040D2B039	HIRO	DigiMesh	AT

Select the remote radio device:

Remote selection: Discovered device

No devices discovered

### Select the local radio device

Field/button	Description
Local device list	Populates with devices currently added to XCTU. Select the device you want to use in the throughput session. The list provides the following information about each local device: <ul style="list-style-type: none"> <li>▪ MAC address</li> <li>▪ Device name</li> <li>▪ Protocol</li> <li>▪ Operating mode</li> </ul>
	Discovers remote devices for the selected local device.

### Select the remote radio device

The remote device selection can be performed in four ways:

Field	Description
Discovered device	Displays a list of any remote devices discovered for the selected local device. Only devices whose protocols support node discovery list remote devices.
Specify 64-bit address	Allows a user to manually enter the 64-bit address of the destination device.
Specify 16-bit address	Allows a user to manually enter the 16-bit address of the destination device.
Specify IPv6 address	Allows a user to manually enter the IPv6 address of the destination device. Only supported by Thread modules.

**Note** Not all protocols support node discovery. Only devices that support node discovery list remote devices.

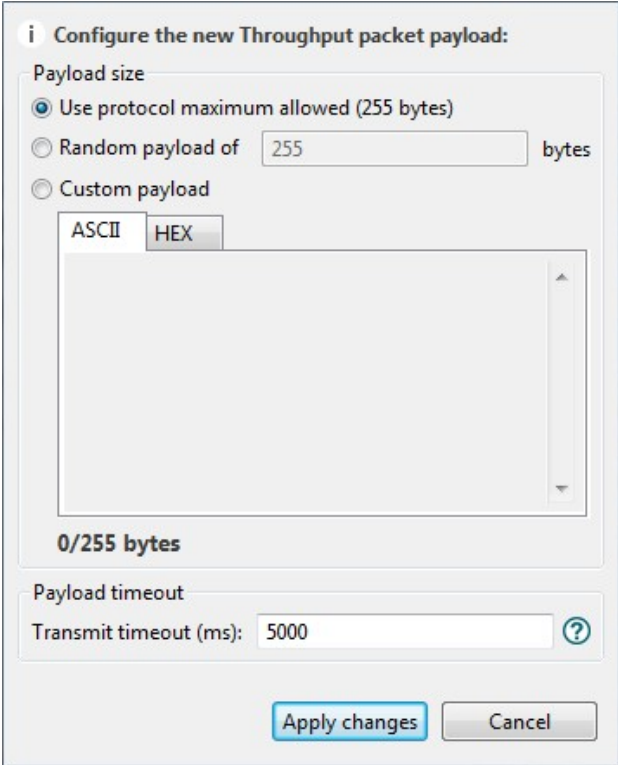
**Note** Not all protocols support IPv6, 64 and 16-bit addressing. Devices that do not support any of these mechanisms display an error at the top of the page.

### Throughput session configuration

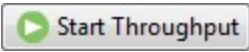
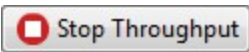
You can configure the following throughput session options from within the Throughput tool:

Field	Description
Throughput type	Determines the type of throughput process to perform. Available options include:

Field	Description
	<ul style="list-style-type: none"> <li>▪ <b>Unidirectional:</b> Sends data from the local device to the remote device. Before sending the next packet of data, it waits for the transmission status of the previous one.</li> <li>▪ <b>Unidirectional (UDP):</b> Sends data from the local device to the remote device using UDP protocol. Before sending the next packet of data, it waits for the transmission status of the previous one. Only supported by Thread modules.</li> <li>▪ <b>Unidirectional CoAP:</b> Sends data from the local device to the remote device using CoAP protocol. Before sending the next packet of data, it waits for the CoAP transmission status of the previous one. Only supported by Thread modules.</li> <li>▪ <b>Bidi - Cluster ID 0x12:</b> Sends data from the local device to the remote one and waits for the data receipt from the remote device. The transmission is performed using explicit addressing frames/packets directed to the Cluster ID 0x12 which returns the sent message.</li> </ul> <hr/> <p><b>Note</b> Not all protocols and operating modes support <b>Bidi - Cluster ID 0x12 Throughput</b> type. Devices that do not support it display an error at the top of the page.</p> <hr/> <ul style="list-style-type: none"> <li>▪ <b>Bidi - Loopback:</b> Same as previous bidirectional mode, this one also waits for the data back from the remote device before sending the next block of data. The data back is performed using the serial port/USB hardware loopback capabilities. Specific user actions are required to use this mechanism. See <a href="#">Special considerations</a>. This method is valid for local modules working in both AT and API operating modes.</li> <li>▪ <b>Bidi - Loopback (UDP):</b> Bidirectional mode. Sends data from the local device to the remote one using UDP protocol and waits for the data receipt from the remote device before sending the next block of data. The data back is performed using the serial port/USB hardware loopback capabilities. Specific user actions are required to use this mechanism. See <a href="#">Special considerations</a>. This method is valid only for Thread modules working either in AT or API operating modes.</li> <li>▪ <b>Bidi - Loopback (CoAP):</b> Same as Bidi - Loopback (UDP) but using CoAP protocol. Specific user actions are required to use this mechanism. See <a href="#">Special considerations</a>. This method is valid only for Thread modules working either in AT or API operating modes.</li> </ul>

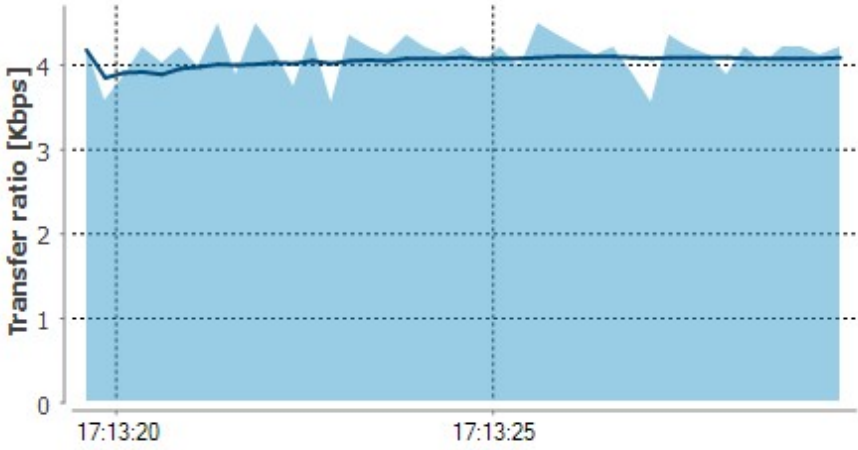
Field	Description
	<p><b>Note</b> Loopback throughput sessions require the remote device to be in AT (transparent) mode.</p>
Packet payload	<p>Allows you to configure the packet payload that will be sent to the remote device. Opens a new window where you can configure different payload values by operating mode of the local radio module (AT or API):</p> <p>Local module working in API mode:</p> <div data-bbox="659 554 1273 1314"></div> <p>Local module working in AT mode:</p>

Field	Description
	<div data-bbox="659 262 1268 978" style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <p><b>i</b> Configure the new Throughput packet payload:</p> <p>Payload size</p> <p><input checked="" type="radio"/> Random payload of <input type="text" value="1000"/> bytes</p> <p><input type="radio"/> Custom payload</p> <p>ASCII <input checked="" type="checkbox"/> HEX</p> <div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div> <p><b>0 bytes</b></p> <p>Payload timeout</p> <p>Transmit timeout (ms): <input type="text" value="5000"/> <span style="float: right;">?</span></p> <p style="text-align: right;"> <input type="button" value="Apply changes"/> <input type="button" value="Cancel"/> </p> </div> <hr/> <p><b>Note</b> In API mode the payload length is limited by the protocol. In AT mode the payload length is unlimited.</p> <ul style="list-style-type: none"> <li>▪ <b>Payload timeout:</b> This is the time that the Throughput session waits to receive the transmit status packet (when the throughput type is unidirectional) or the data back from the module (when the throughput type is bidirectional). The bigger the payload is the greater this time should be.</li> </ul>
Duration	<p>Determines the duration of the Throughput session. Available options include:</p> <ul style="list-style-type: none"> <li>▪ <b>Time (s):</b> Establishes the minimum duration of the session in seconds. The process sends data packets to the remote device until this time is reached.</li> <li>▪ <b>Packets:</b> The duration of the session is established by the number of packets configured in this field. Bear in mind that depending on the packet payload, a packet can take several seconds to be sent.</li> <li>▪ <b>Loop infinitely:</b> The process sends packets infinitely until the throughput session is stopped manually.</li> </ul>

Field	Description
Time window	Configures the visible time window of the transfer ration measured by the throughput session.
	When you have configured all the options, click the <b>Start Throughput</b> button to start measuring the transfer ratio of your radio module.
	You can stop the process at any time by pressing the same button, now showing the text <b>Stop Throughput</b> .

### Data representation

The chart in the tool represents the instant transfer ratio and the average transfer ratio.

Field	Description
Transfer ratio chart	<p>Displays the instant transfer ratio with a light blue area and the average transfer ratio as a dark blue line. The instant transfer ratio is calculated every 250ms or every time a packet is sent if sending takes longer than 250ms. You can hide and show chart data by checking or unchecking <b>Transfer ratio</b> and <b>Avg. transfer ratio</b>.</p> 
Transfer ratio values	<p>Displays the instant and average transfer ratio as packets are sent to the remote module during the session.</p> <div style="background-color: #cccccc; padding: 10px; text-align: center;"> <p>Instant transfer ratio: <b>4.2 Kbps</b>                      Average transfer ratio: <b>4.08 Kbps</b></p> </div>



Field	Description
Session statistics	<p>Displays session statistics such as the number of packets sent to the remote device, the number of transferred bytes, and the duration of the session.</p> <div style="background-color: #cccccc; padding: 5px; margin: 10px 0;"> <p style="text-align: center;">Tx packets: 61 Tx bytes: 5124 Elapsed time: 00:00:09</p> </div>

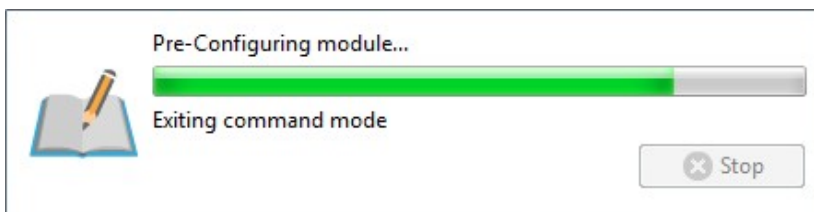
## Supported products

Throughput is only supported in these protocols:

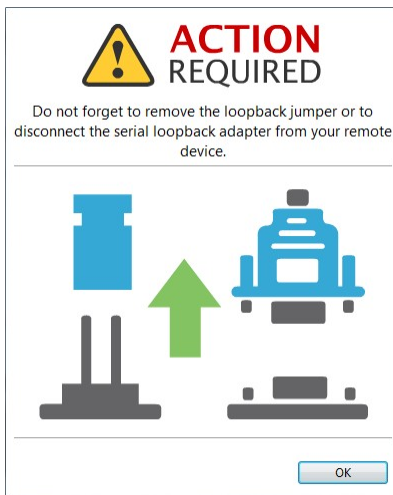
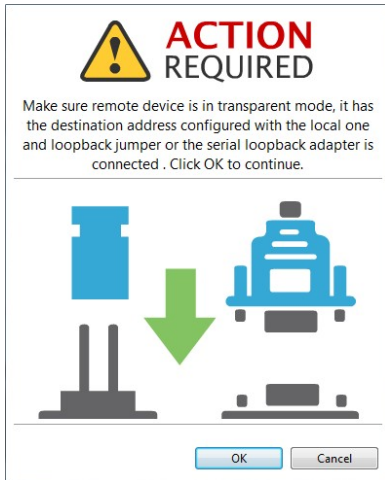
- SX
- XLR Module
- ZigBee
- DigiMesh
- XTend
- XTend - DigiMesh
- XC/XSC
- XLR PRO
- 802.15.4
- Digi Point
- Thread

## Special considerations

There are some special considerations to be aware of while working with the Throughput tool. When the local module is working in **AT (transparent) mode**, the tool automatically performs configuration steps in the module before starting the throughput session, and after finishing or stopping it. A progress bar shows the level of completion for this process:




- The **bidirectional loopback** throughput types only work with remote devices in AT (transparent) operating mode.
- When performing any **bidirectional loopback** throughput, you need to connect the loopback jumper or the loopback adapter in the remote device before starting and disconnect it after finishing. A dialog box notifies you of the action required:

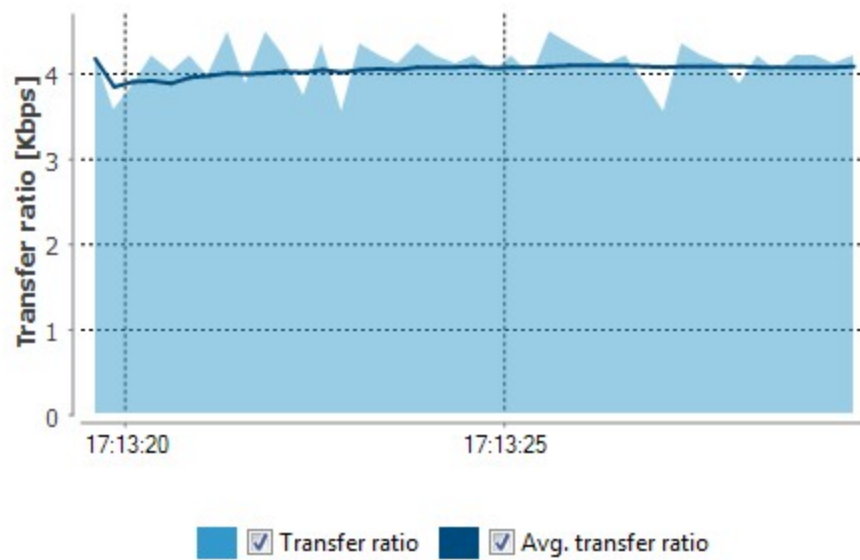


## Measure the transfer ratio between two radio modules

Follow these steps to measure the transfer ratio between two radio modules in the same network. For more information, see [Throughput tool](#).

1. Launch the Throughput tool by selecting **Throughput** from the **Tools** drop-down menu  on the main toolbar. The **Throughput** dialog appears.
2. Under **Device selection**, select a local device to perform the throughput and a remote device to receive the data.
3. Under **Configuration**, configure the throughput type, packet payload, duration, and time window.
4. Once you have configured all the options, click the **Start Throughput** button to start measuring the transfer ratio of your radio module. You can stop the process at any time by pressing the same button, which has changed to **Stop Throughput**.

The Throughput chart represents the instant transfer ratio and the average transfer ratio.



5. You can hide and show any of the chart data by selecting or clearing the transfer ratio check boxes.

---

**Note** There are special considerations to be aware of while working with the Throughput tool. For details, see [Throughput tool](#).

---

## MicroPython Terminal tool

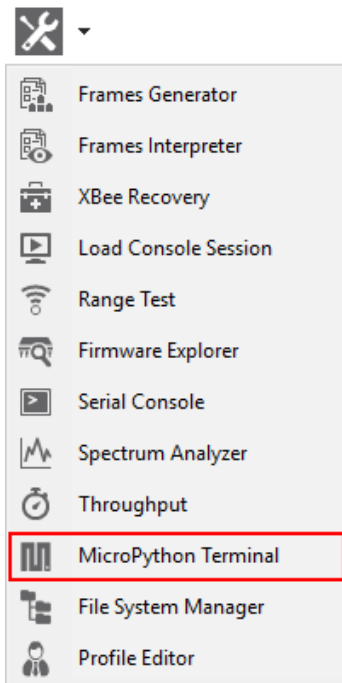
The MicroPython Terminal tool allows you to communicate with the MicroPython stack of your radio module through the serial interface.

---

**Note** Make sure your radio module supports MicroPython and is configured in that mode.

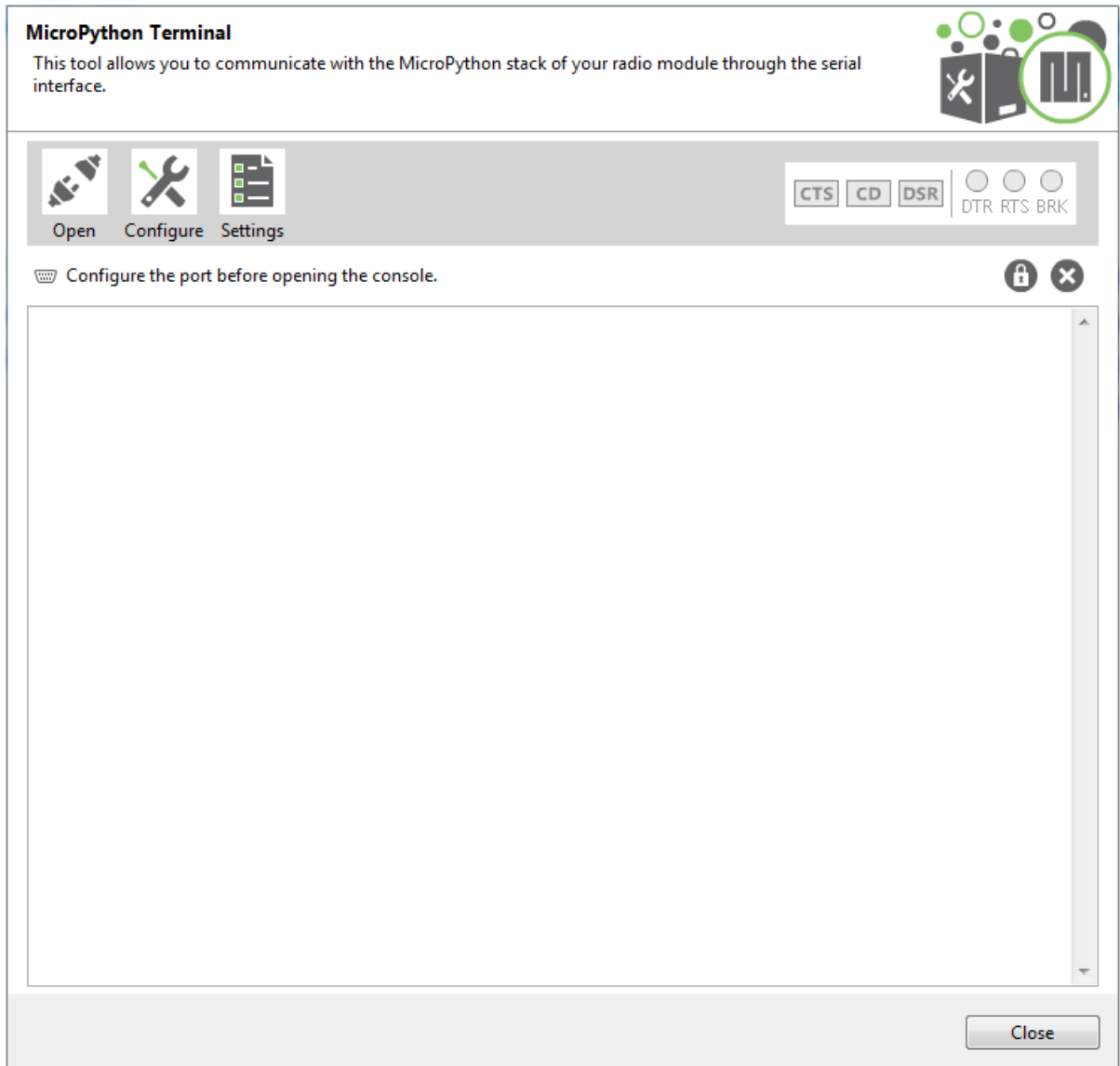
---

To open the MicroPython Terminal tool, select **MicroPython Terminal** from the **Tools** drop-down menu on the main XCTU toolbar.





## MicroPython Terminal dialog

The workflow and components of the MicroPython Terminal tool are similar to those of the Serial console one. Because the MicroPython Terminal is not associated with a radio module, you must configure the serial port the console will open a connection with. For more information, see [Configure the serial port settings - MicroPython Terminal tool](#).





### Open a terminal session

By default, the first time you open the terminal it is disconnected, as is indicated by the gray background of the toolbar.

Buttons	Description
	Click the <b>Open</b> button to establish communication with the radio module corresponding to the terminal. If it is the first time you are connecting the console, you are prompted to configure the serial connection. The background color of the Open button changes to green and its text changes to Close. When the terminal is connected, all data traffic of the radio module is captured by the console and displayed in the corresponding controls.
	Click the <b>Close</b> button to disconnect the terminal from the module.

### Configure the serial port settings - MicroPython Terminal tool

Follow these steps to configure the serial port settings of an instance of the MicroPython Terminal tool. For more information, see [MicroPython Terminal tool](#).


1. Click the **Configure** button  on the MicroPython Terminal. If the console is not yet connected, click the **Open** button .



The **Serial Port Configuration** dialog lists all of the serial connection parameters to be configured.

### Serial Port Configuration

Select the COM port from the list and configure its settings.



Select the Serial/USB port:

COM1	Communications Port
COM3	Intel(R) Active Management Technology - SOL
COM6	USB Serial Port
COM13	USB Serial Port
COM17	USB Serial Port

Provide a port name manually:

Baud Rate:

Data Bits:

Parity:

Stop Bits:

Flow Control:

2. Select the device's serial port, or provide it manually. The rest of the parameters are loaded with their default values.
3. Change the configuration if needed. The most common serial configuration is:

**Baud rate:** 9600 or 115200

**Data bits:** 8

**Stop bits:** 1

**Parity:** None

**Flow control:** None

---

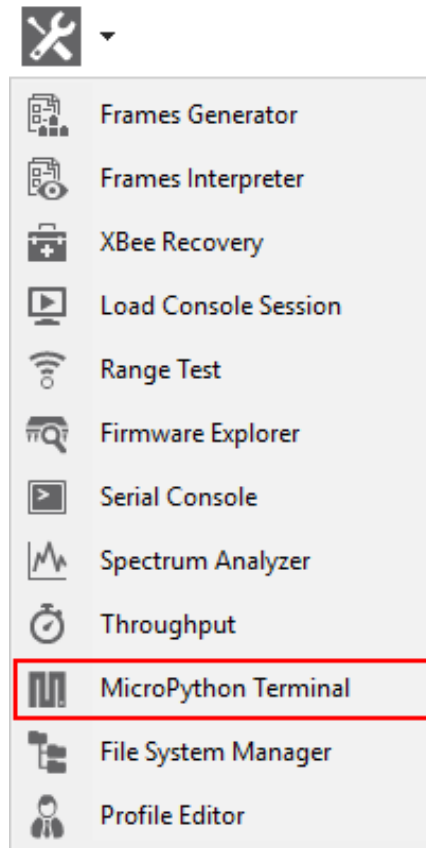
**Note** Custom baud rates can only be typed under Windows OS.

---

## Open a MicroPython Terminal session

The first time you open a console, it is disconnected by default. To open a MicroPython Terminal session:

1. Select **MicroPython Terminal** from the **Tools** drop-down menu on the main XCTU toolbar. The **MicroPython Terminal** dialog appears.



2. Click the **Configure** button from the toolbar and configure the serial port where the radio module is attached. See [Configure the serial port settings - MicroPython Terminal tool](#) for details.
3. Click the **Open** button to establish communication with the radio module.

---

**Note** The MicroPython Terminal tool operates the same as any other VT100 terminal emulator.

---

## File system manager tool

The **File system manager** tool allows you to access and interact with the file system of your XBee module.

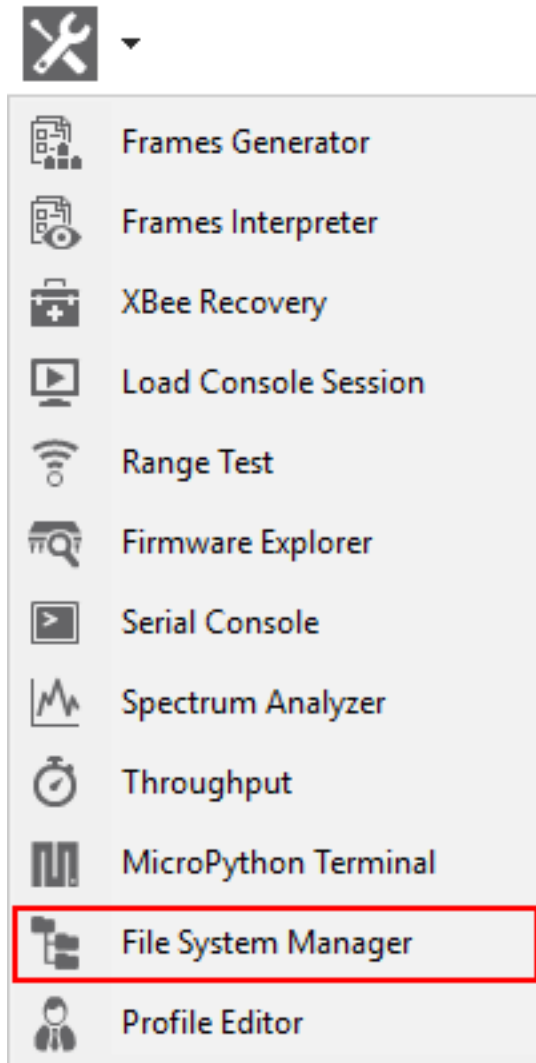
---

**Note** Make sure your radio module supports the file system feature.

---

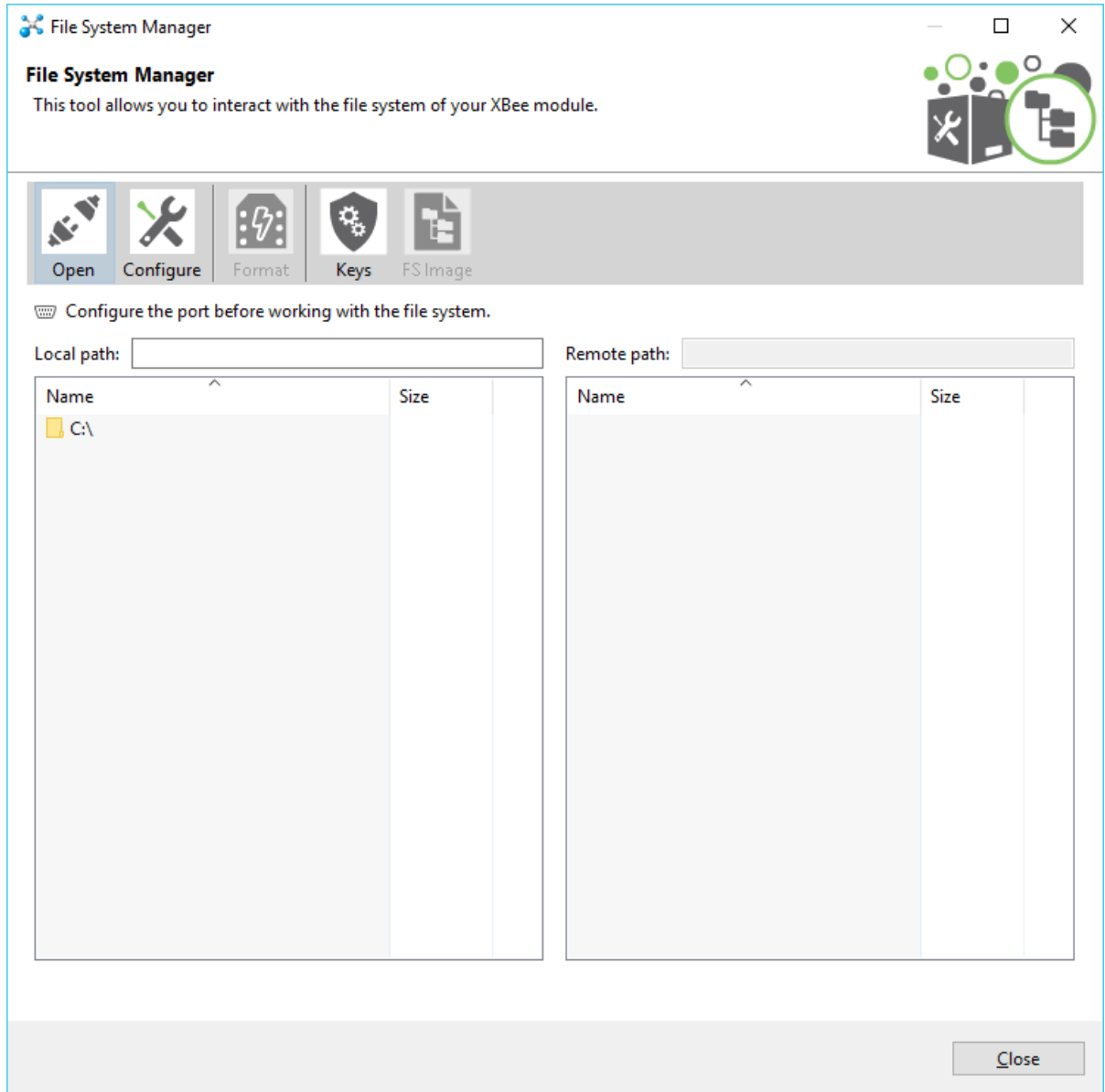
To open the **File System Manager** tool, select **File System Manager** from the **Tools** drop-down menu on the main XCTU toolbar.





## File system manager dialog

Use the **File System Manager** dialog to interact with the file system of your XBee module. For instructions, see [Interact with XBee file system](#).




## File System Manager sections

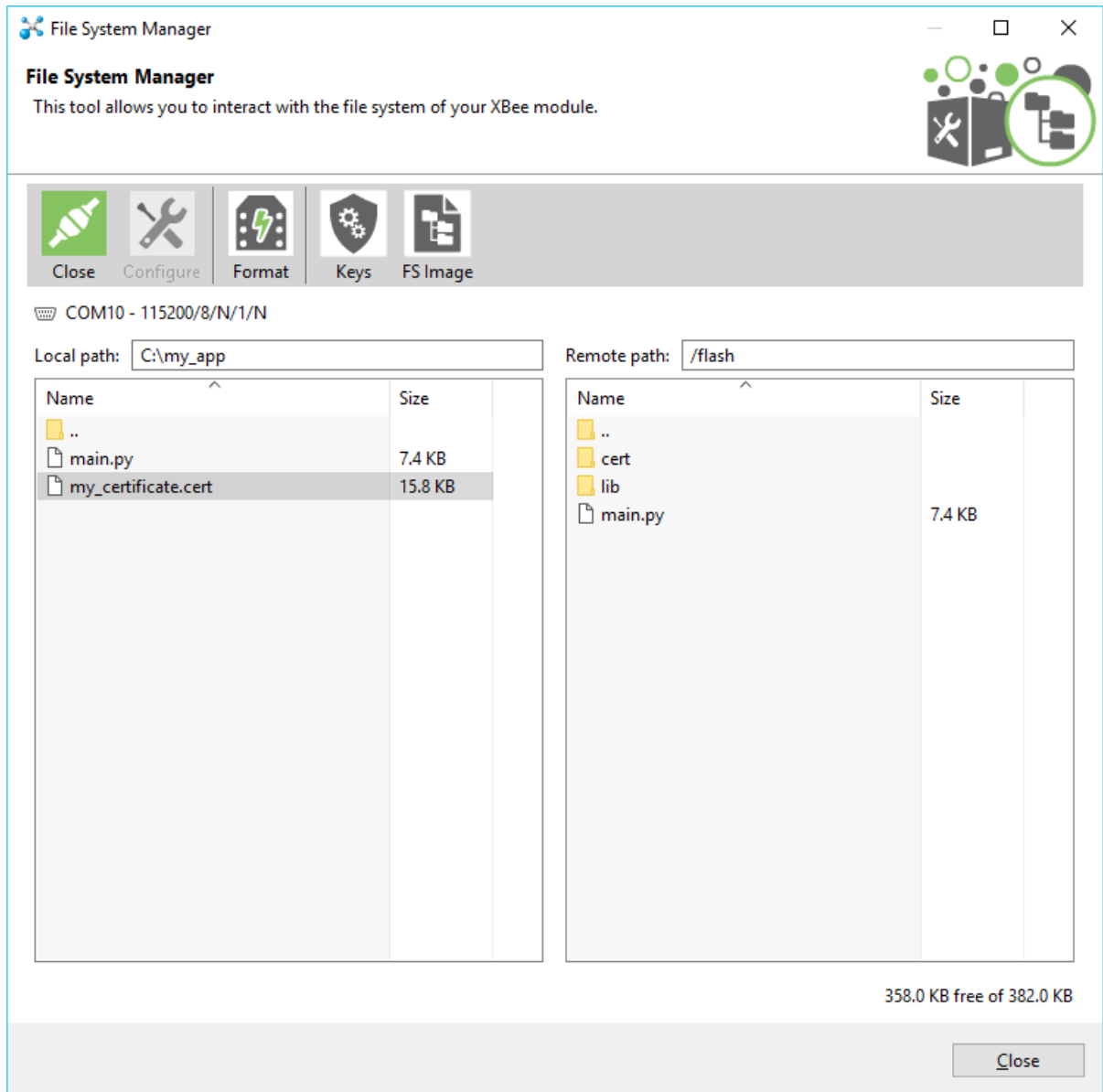
The **File System Manager** dialog is divided into three parts:

- **Toolbar:** Allows opening/closing the connection with the XBee module, configuring the serial port parameters, and formatting the XBee file system.
- **Local file system view:** Shows the current path of your computer and its contents.
- **Remote file system view:** Shows the current path of the XBee module and its contents.

## Interact with XBee file system


Follow these steps to interact with the file system of your XBee module. For more information, see [File system manager tool](#).

1. Launch the File System Manager tool by selecting **File System Manager** from the **Tools** drop-down menu  on the main toolbar. The **File System Manager** dialog appears.
2. Click **Open**.
3. If you did not add your XBee module to XCTU before, select its serial port configuration. Click **OK** to open the connection with the module.
4. If the serial configuration is correct and your XBee module supports file system, the contents of the default path are listed in the remote file system view.



### Operations with the file system

You can perform the following operations in the XBee file system:

Operation	Description
<b>Navigate</b>	Navigate through the different directories by double clicking on them. To go back, double click on the parent directory (..) or press the backspace key. You can also go directly to a specific path by typing it on the local/remote path text box and pressing enter.
<b>Upload</b>	Upload a file or directory to the XBee module. Drag-and-drop or right-click on a file or directory and then click <b>Upload</b> . If you want to upload several files or directories, select and drag them to the XBee file system view.
<b>Upload a secure file</b>	Upload special files, such as certificates or keys, as write-only. Once the files are uploaded to the XBee, they cannot be downloaded, renamed or moved; only deleted. These secure files are displayed in the XBee file system with a lock  and are grayed out.
<b>Download</b>	Download a file or directory from the XBee module. Drag-and-drop or right-click a file or directory and then click <b>Download</b> . If you want to download several files or directories, select and drag them to the local file system view.
<b>Create directory</b>	Create a directory. Right-click on either the local or remote file system view and click <b>Create directory</b> . You are asked to enter the name of the new directory.
<b>Delete</b>	Delete any file or directory. Right-click a file or directory and click <b>Delete</b> . You are asked to confirm the operation.
<b>Rename</b>	Rename any file or directory. Right-click a file or directory and click <b>Rename</b> .
<b>Get and compare hash</b>	Get the SHA256 hash of any secure file that is on the XBee file system and compare it with the hash of a local file. Since secure files cannot be downloaded, this operation allows you to verify that the secure file is actually equal to other located on your computer. Right-click on a secure file and click <b>Get hash</b> or <b>Compare hash</b> .
<b>Format</b>	Re-format the entire XBee file system by clicking the <b>Format</b> button located in the toolbar.  <b>Note</b> This process erases all your files on the XBee module and cannot be undone.

## Update the file system of remote devices

After an OTA firmware update of non-cellular devices, all file system data and bundled MicroPython code is erased. In order to continue running code, a new file system needs to be sent to the device after the firmware update is complete. This section contains information on how to update the file system of remote devices over the air.

**Note** This is only available for XBee3 devices running Zigbee, DigiMesh or 802.15.4 protocols. This does not apply to XBee Cellular devices.

### OTA file system update process

Since OTA file system updates are signed, remote radios must be configured so that they can validate incoming updates. The process to set up a network capable of doing OTA file system updates is:

1. Generate a public/private Elliptic Curve Digital Signature Algorithm (ECDSA) signing key pair.
2. Using the generated public key, set the public key locally on all radios that will receive OTA file system updates.
3. Prepare the file system on the local device. The file system of the local device will be downloaded and transferred to remote devices.
4. Download an image of the file system of your local device and sign it with the private ECDSA key.
5. Transfer the file system signed image to all the target remote devices.

---

**Note** The **FK** command cannot be set remotely. **FK** must either be set before the radio is deployed, or else serial access to the radio is needed to set it.

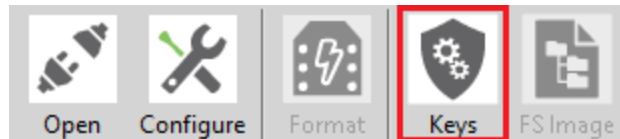
---

### 1. Generate ECDSA key pair

The security of the transfer process of file system images over the air relies on the ECDSA. As with other asymmetric cryptography algorithms, ECDSA requires a private key to sign the object and a public key to be able to read it.

To generate a valid key pair, open XCTU and follow these steps:

1. In the file system manager, click the **Keys** button of the toolbar.



2. The **Generate file system key pair** dialog displays. Click **Generate** to create a valid key pair.
3. Save each generated key to a different file—we recommend storing them in different locations, for security reasons. The keys are stored in PEM files, with the extension **.pem**.

### 2. Configure the public key on all the devices in the network

XBee3 devices running DigiMesh, 802.15.4 or Zigbee protocols have a new setting that allows the configuration of the public key generated in the previous step. These are the steps to configure the public key on a local XBee device:

1. In the **Configuration working mode** of XCTU, scroll to see the **File System Options** section. This section contains a setting called **File System Public Key**, which is configurable by a button.
2. Click the **Configure** button. The **Configure File System Public Key** dialog displays.
3. Click **Browse...** to search for the public key generated in the **Generate ECDSA key pair** section.
4. Once the key is added, the text area displays the value of the key in hexadecimal format.
5. Click **OK** to write the key into the module.
6. Do this for all the remote modules that should have the possibility of having their file system updated over the air.

---

**Note** Configuring the public key can only be done locally. All the nodes of the network must have the public key configured before deployment.

---

### 3. Prepare the local file system image

To update the file system on a remote device, you must generate the file system image with the local XBee device. This means that all the files that the file system of the remote devices should have must be replicated in the local device first.

### 4. Download a signed file system image

It is necessary to generate a file that contains an image of the file system to be transferred over the air to the remote devices. As stated previously, the image is generated with your local device, and is transferred to the remote devices afterwards.

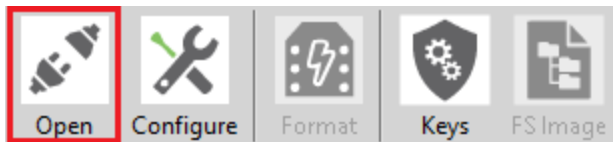
---

**Note** The local module used to create the file system image must have the same firmware version installed as the target module or the file system will be rejected. Use the **VR** command to check the version number on both, the local and remote, devices.

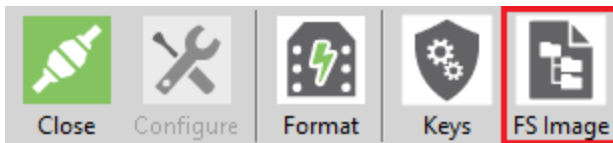
---

To generate the OTA file system image, follow these steps:

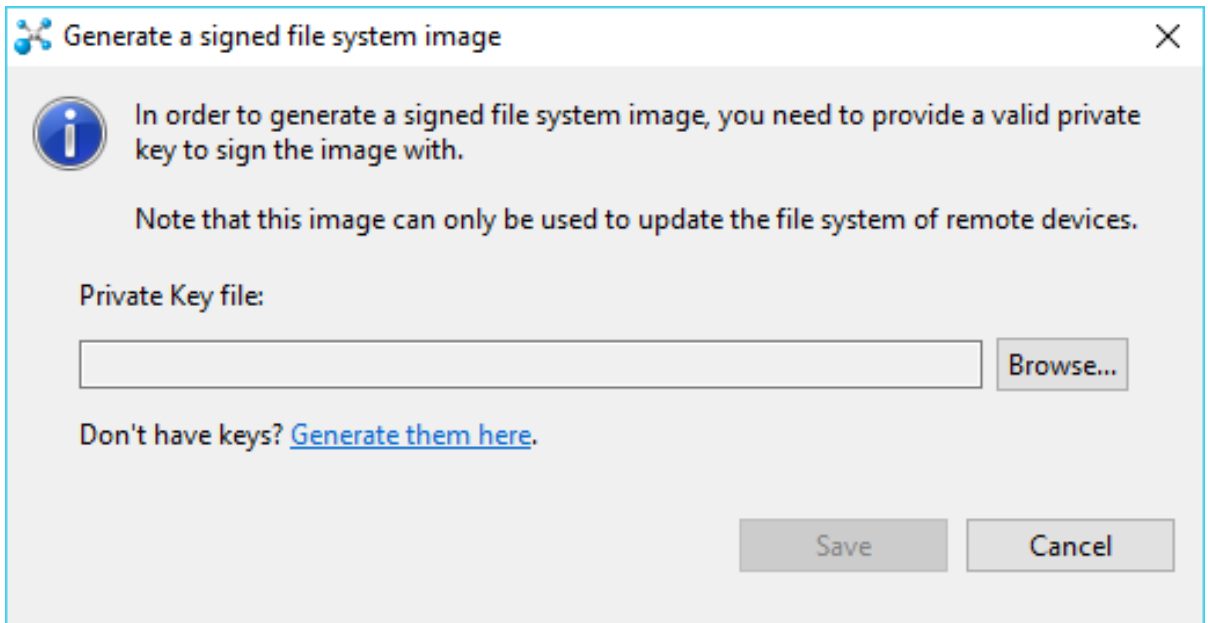
1. In the file system manager, **Open** the connection to your local device.



2. Once the connection is established, click **FS Image**.



3. The following dialog displays:

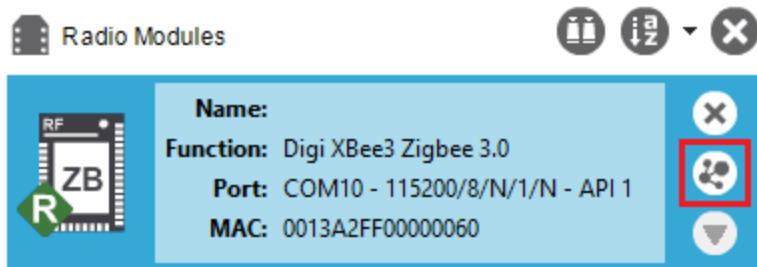


4. Click **Browse...** to find the private key generated in the **Generate ECDSA key pair** section. This key is used to sign the file system image once it is downloaded into your computer.
5. Click **Save**. A file dialog opens to let you decide where to store the signed file system image. The extension of the generated file is **.fs.ota**.

### 5. Transfer the image to a remote device

To send the generated signed file system image to a remote device over the air:

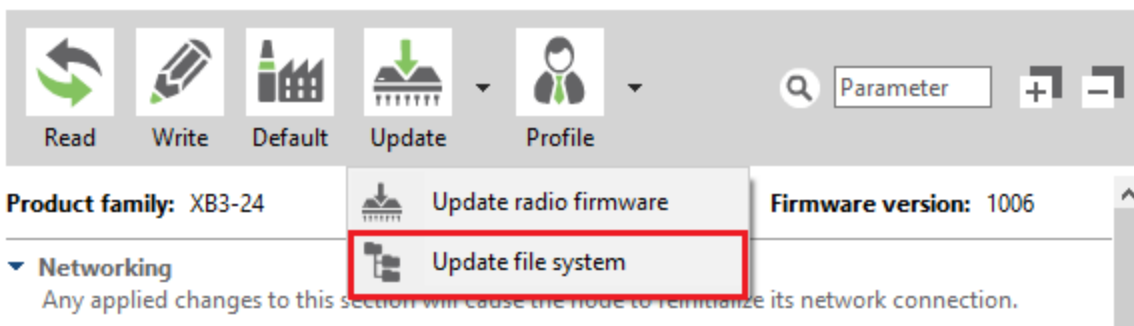
1. Once a local device is added to XCTU, discover the target remote device by clicking **Discover radio nodes in the same network** in the appropriate panel of the local XBee device.



2. Once the remote device that you want to update the file system of is found, add it to the list of remote devices.
3. Click the remote device to read its settings.
4. Click **Update** on the **Radio configuration working mode** toolbar.



5. Click **Update file system** on the context menu.

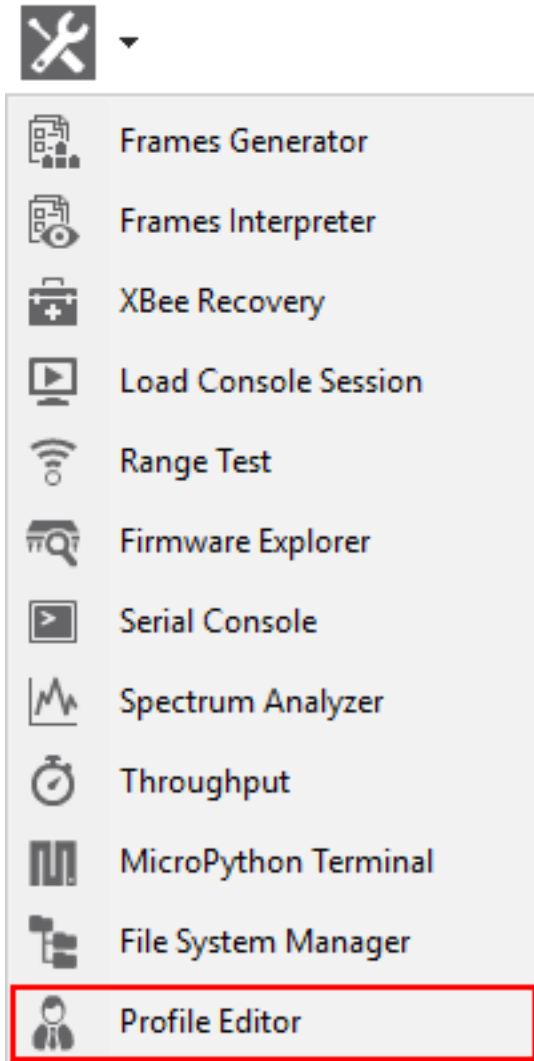


6. A file dialog displays to let you select the signed file system image generated in the previous section.
7. The transfer process starts automatically.

## Profile Editor tool

The **Profile Editor** tool allows you to create new profiles or load and edit existing ones.

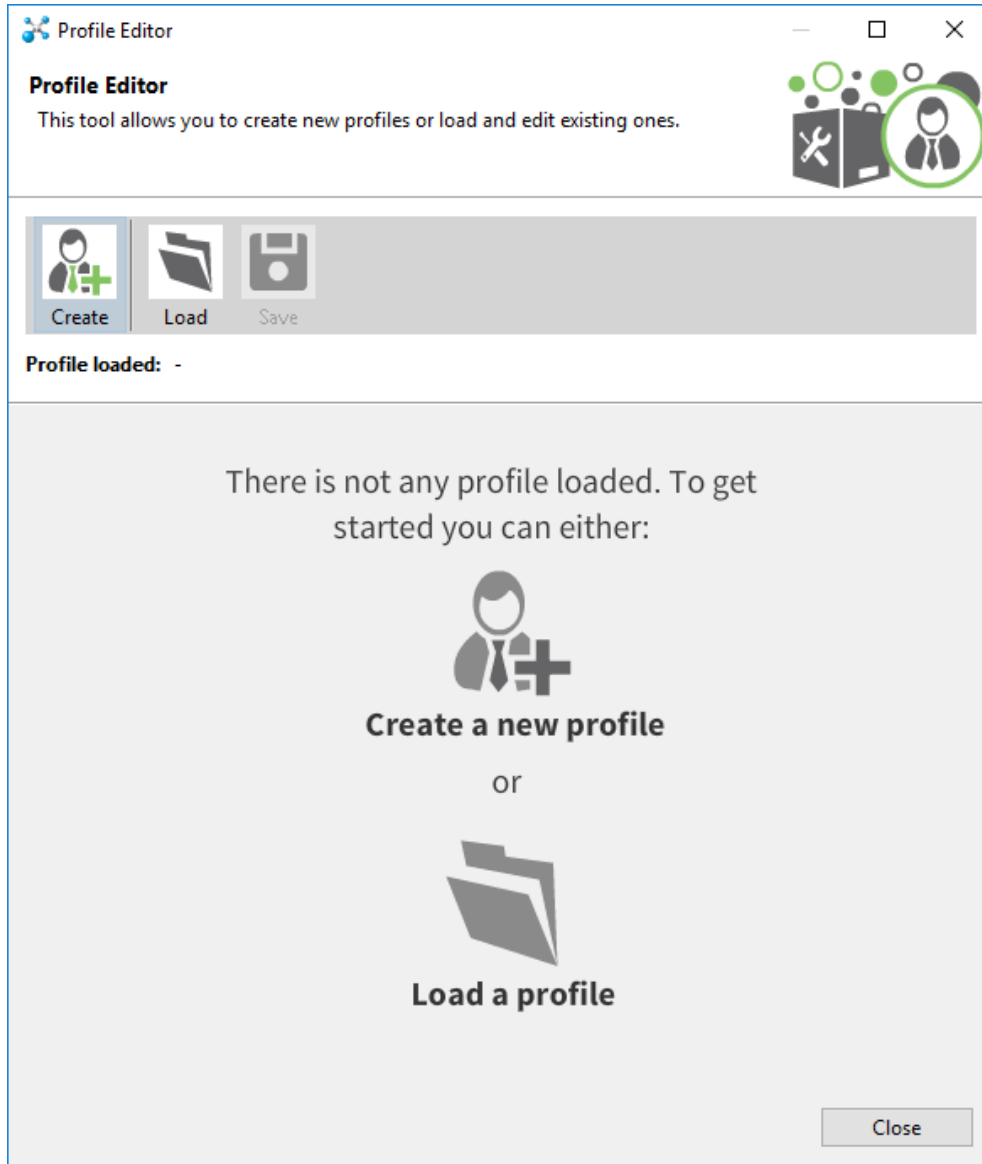
To open the **Profile Editor** tool, select **Profile Editor** from the **Tools** drop-down menu on the main XCTU toolbar.



### Profile Editor dialog


The first time you open the dialog:







### Profile Editor toolbar

The toolbar of the dialog displays all the actions you can perform in the tool.

Field	Description
 <p>Create a new profile</p>	<p>Opens the <b>Create a new profile</b> wizard to create an XBee configuration profile. See <a href="#">Create a configuration profile</a> for more information.</p>

Field	Description
 Load a profile	Loads an XBee configuration profile in the editor. See <a href="#">Load and edit a configuration profile</a> for more information.
 Save the profile	Saves the current profile. You can either overwrite the loaded profile or save it as a new one. <hr/> <b>Note</b> This action is disabled until you create or load a profile. <hr/>

## How-to articles and videos

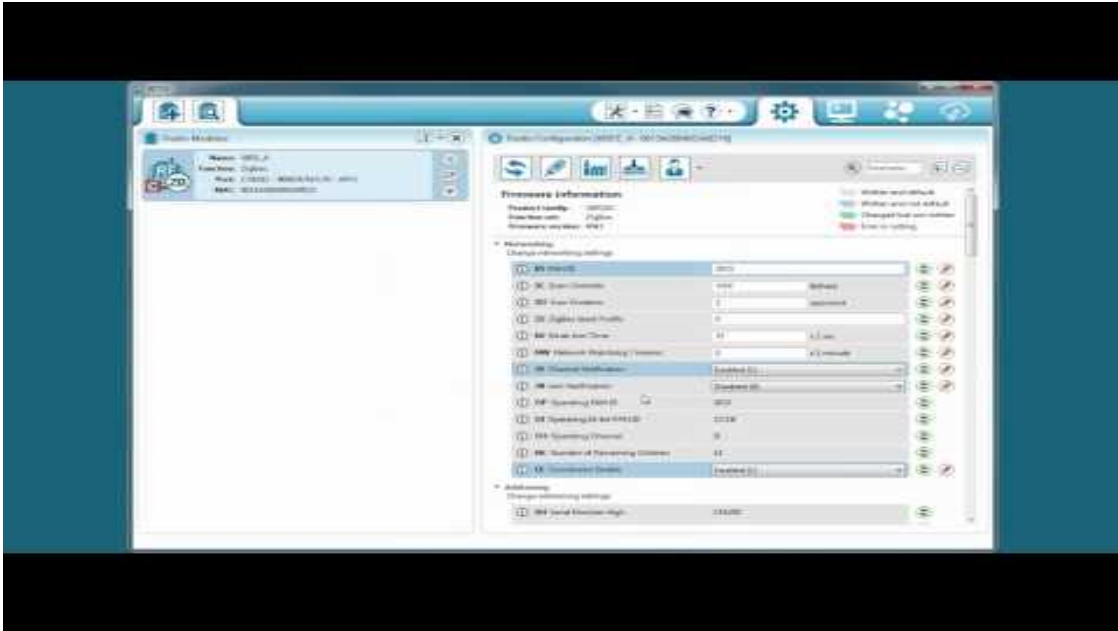
---

See how to get things done with XCTU. Check back for additional how-to articles describing the most common XCTU procedures.

How to update the firmware of your modules .....	212
How to visualize your network .....	214

## How to update the firmware of your modules

Watch the video to learn how to perform local and remote firmware updates.

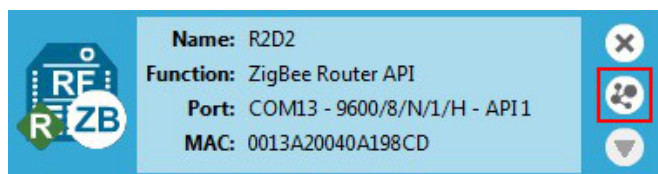


You can use XCTU to update firmware in both local and remote modules. For more information, see [Update firmware](#).

### Step 1: Add the module to XCTU

You must have a local module connected to your computer in order to perform firmware updates, either to update local firmware through the serial connection or to use the local device to remotely upgrade another module in the same network. With a local module properly attached to your computer, follow these steps:

1. Add the local module attached to your computer to XCTU so it is displayed in the radio modules list.
2. Add your remote module in the network to XCTU:
  - a. Configure the local module you have just added to work in API mode.
  - b. Click **Discover radio nodes in the same network** to start a search of the remote module.



- c. When a remote module is found, it is listed in the **Discovering remote devices** dialog. Select the device and click **Add selected devices**. The remote module is added to the radio modules list as a subordinate to the local module.

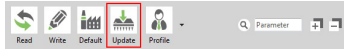
---

**Note** Once you add your module to the radio modules list in XCTU, the update process is exactly the same whether it is a local or remote module.

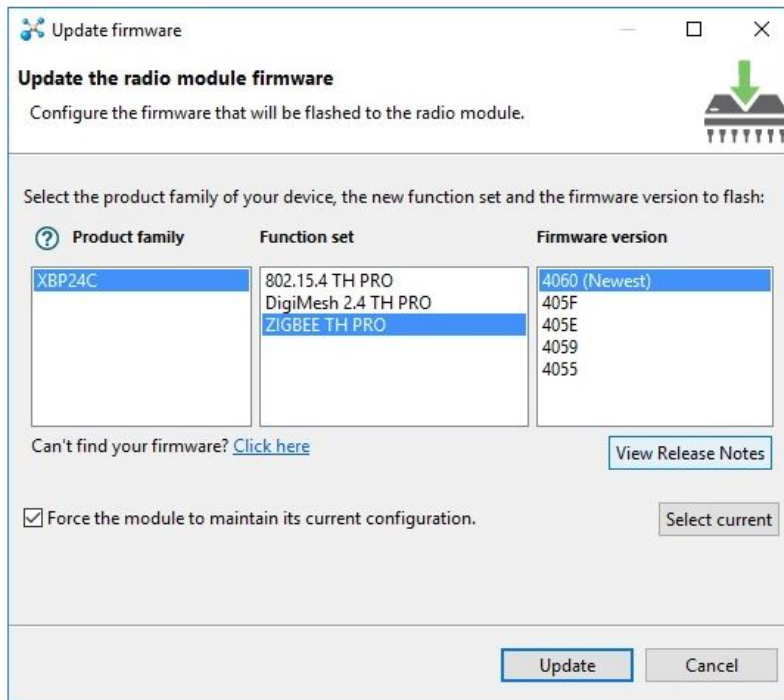
---

## Step 2: Update the firmware

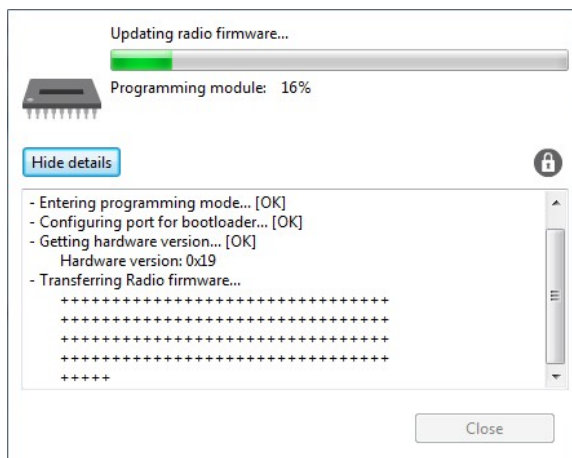
1. Select a local or remote module from the radio modules list. Click **Update firmware**. The **Update the radio module firmware** dialog is displayed.



2. Select the product family, function set, and firmware version.



3. Click **Update**. A progress dialog displays the status of the update. Click **Show details** to review the steps of the firmware update process.

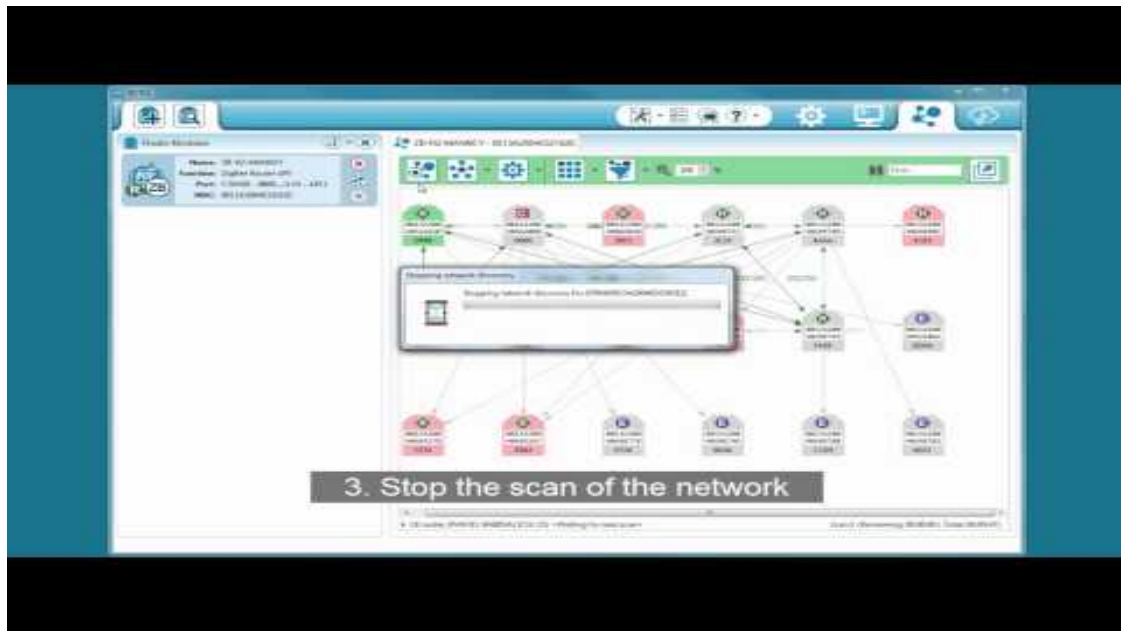


## Over-the-air firmware update considerations

1. To perform a firmware update of a remote XBee in the same network, you must configure the local module to work in API mode.
2. Remote firmware update functionality is limited to the following radio modules:
  - XBee/XBee PRO SX
  - XLR Pro Module
  - XBee/XBee PRO 802.15.4 (S2C module versions only)
  - XBee/XBee-PRO DigiMesh 2.4 (S2C module versions only)
  - XTend RF Module Family (SX module versions only)
  - XBee/XBee-PRO ZB and Programmable XBee-PRO ZB
  - XBee/XBee-PRO ZB SMT and Programmable XBee-PRO ZB SMT
  - XBee-PRO 900HP and Programmable XBee-PRO 900HP
  - XBee 865LP and Programmable XBee 865LP
  - XBee3 (Zigbee, DigiMesh 2.4, and 802.15.4)
3. If something goes wrong during the over-the-air firmware update of a remote node—for example, the communication is lost because the remote device is disconnected—you must perform a manual recovery. See [XBee recovery tool](#).

## How to visualize your network

Watch the video to learn how to explore your network with Network view.




Network working mode allows you to discover and visualize the topology and interconnections of your network. For more information, see [View your radio network](#).

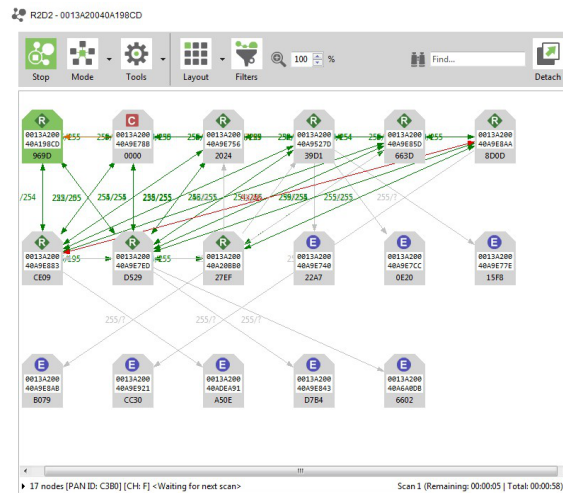
### Step 1: Scan the network


To discover the network, follow these steps:

1. Once you have added a module to XCTU, switch to Network working mode.



2. Click **Scan the radio module network**  to start the network discovery process. Modules are dynamically added as they are discovered, showing their connections and link quality.



3. The scan process automatically stops according to the network preferences configured in [Set network discovery preferences](#). You can also click **Stop scanning the network**  at any time.

---

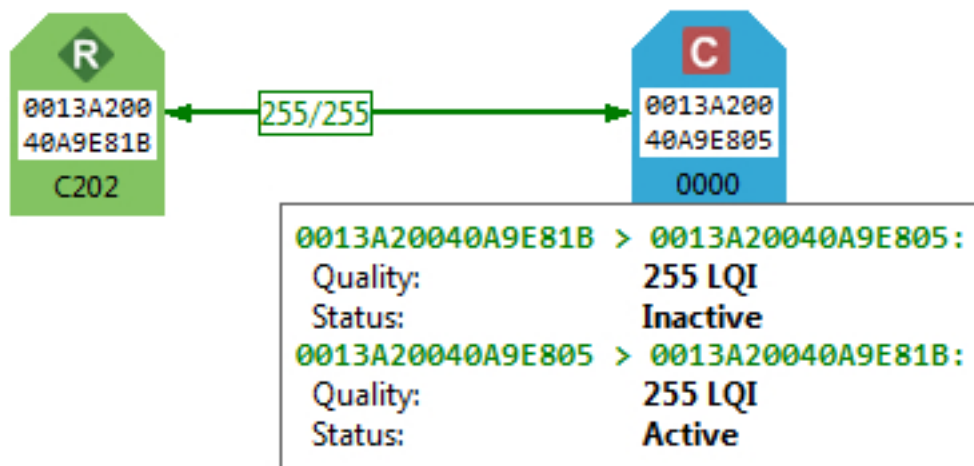
**Note** You may need to perform multiple scans to discover the entire network if, for example, modules are sleeping or the network is very large.

---

## Step 2: Explore the network

Each module displays information about itself such as its role or its address. Hover the cursor over a module to view more detail.




The modules are connected by lines indicating connections with neighboring modules. The arrows indicate the direction of communication. When you are using protocols such as ZigBee or DigiMesh, click on a connection to view additional information about link quality and status.



Network working mode features can help you examine your network:

Button	Name	Description
	Start scan	Scans the radio module's network and displays a list of available modules.
	Stop scan	When scans specified in scan preferences are finished, the discovery process stops automatically. You can also click the <b>Stop scan</b> button at any time.
	Graph view	Displays the modules as nodes in a graph.
	Table view	Displays the modules as rows in a table.
	Screenshot	Saves an image of the network to your computer.
	Export table	Exports network information in a tabular file format.
	Settings	Configures network-related parameters.
	Layout	Changes the network layout to one of six types: composite, spring, vertical tree, horizontal tree, grid (default), and radial.



Button	Name	Description
	Filter links	Hides or displays connections between RF modules based on their quality. Enables/disables quality ranges via radio buttons. All filters are enabled by default so all connections are visible. You can establish connection quality colors and ranges in <b>Preferences &gt; Network &gt; Appearance</b> or by clicking <b>Configure quality ranges</b> link in the <b>Filter links</b> dialog. See <a href="#">Set network discovery preferences</a> .
	Zoom	Changes the zoom level of the network.
	Find box	Searches for a particular module in the network.

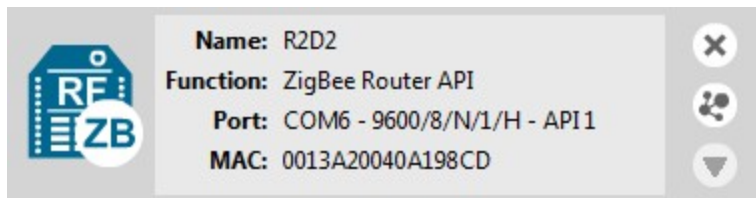
## Troubleshooting for XCTU

---

Check here for answers to common questions about working with XCTU.

### Troubleshooting: General

After adding a local device to the list of devices, the icon corresponding to the module does not display the role of the module within its network.



The local radio module might not be joined to any network. In some cases (mostly in ZigBee protocols), the device must be joined to a network in order to determine its configured role. After a local DigiMesh radio module executes a remote node discovery process, the device becomes unresponsive.

In the DigiMesh protocol, when you perform a Node Discovery (ND) or a Find Neighbors (FN) operation, the module will not process any AT command until its configured NT time expires. If you want to talk with the device immediately, you must restart it by pressing its reset button. I have an end device radio module configured to sleep. It works in API operation mode with an ST value less than one second, but XCTU is not able to find it or read its settings.

You may need to press the commissioning button of the device before performing any action with that module in XCTU. Pressing the commissioning button will wake the module for 30 seconds so XCTU can communicate with it.

I get a "Permission denied" error while trying to add or discover modules in Linux.

By default, access to the serial and USB ports in Linux is restricted to root and dialout group users. To access your devices and communicate with them using XCTU, it is mandatory that your Linux user belongs to this group. Follow these steps to add your Linux user to the dialout group:

1. Open a terminal console
2. Execute this command:

```
sudo usermod -a -G dialout <user>
```

Where <user> is the user you want to add to the dialout group.

3. Log out and log in again with that user in the system.

## Troubleshooting: Networking

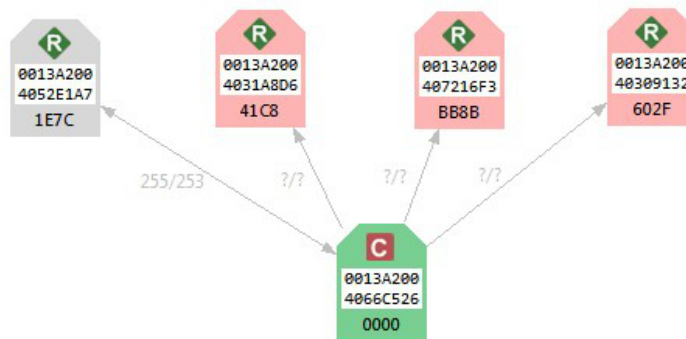
Not all remote devices are found after clicking the search button of a local radio module in a relatively big network.

In big networks, some devices may not answer the ND command in time. If not all devices are found after executing the remote discovery process, do the following:

1. Add those that have been found.
2. Click the search button of the local device again.
3. When asked whether to clear the list of remote modules, click **NO**.
4. Wait for new modules to be discovered.

Repeat this process until all the modules on your network are found.

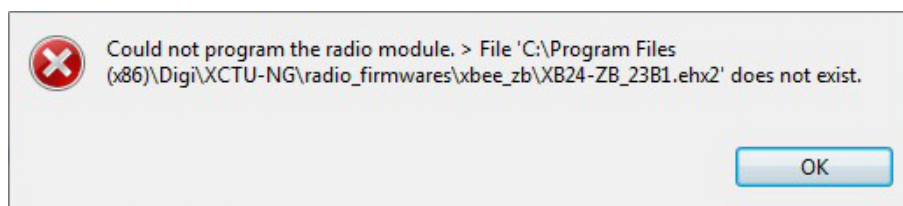
When discovering a network, some of the nodes are displayed in light red color. What does it mean?



Red radio modules represent devices that were in the network at any time in the past but are unreachable now. When a radio module leaves the network, some devices may still contain information about it. When a new discovery is performed, the module appears to be there but does not answer or is unreachable when XCTU tries to verify the connection.

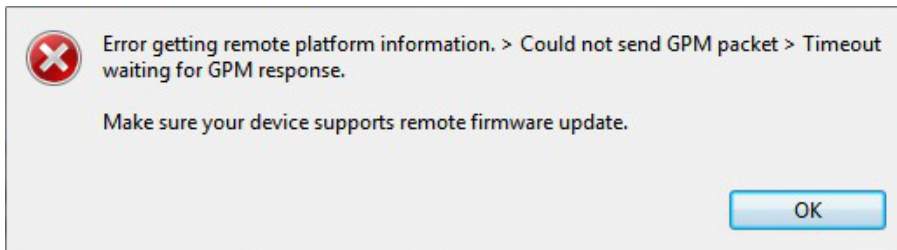
## Troubleshooting: Firmware update

When XCTU is attempting to update the firmware of a local radio module, an error appears indicating that the firmware binary does not exist.



You probably tried to install a firmware using an old firmware. Old firmware packets are not compatible with this new version of XCTU. To obtain an updated firmware version, contact [Digi support](#).

When XCTU is attempting to update the firmware of a remote radio module, an error appears indicating XCTU could not retrieve remote platform information.



Your remote radio module probably does not support remote firmware update. The following radio modules support the remote firmware update feature:

- XBee/XBee-PRO SX
- XLR Pro Module
- XLR PRO Radio Solution
- XBee/XBee-PRO 802.15.4 (S2C module versions only)
- XBee/XBee-PRO DigiMesh 2.4 (S2C module versions only)
- XTend RF Module Family (SX module versions only)
- XBee/XBee-PRO ZB and Programmable XBee-PRO ZB
- XBee/XBee-PRO ZB SMT and Programmable XBee-PRO ZB SMT
- XBee-PRO 900HP and Programmable XBee-PRO 900HP
- XBee 868LP and Programmable XBee 868LP
- XBee3 (Zigbee, DigiMesh 2.4, and 802.15.4)

## Troubleshooting: Add radio module

I can't find my module when trying to add it to the list of devices.

Investigate the following possible reasons that XCTU isn't finding your module:

- **Radio module not connected/Invalid settings.** The selected port or the serial port settings where the radio module is connected are not valid. Make sure you have selected the correct port and settings. The most common serial configuration is:
  - Baud rate: 9600 or 115200
  - Data bits: 8
  - Stop bits: 1
  - Parity: None
  - Flow control: None
- **Sleeping radio module.** The radio module may be a sleeping node. If the module is sleeping at the time XCTU tries to communicate with it, XCTU cannot add it to the device list. If you believe your module could be sleeping, try to wake it up by pressing the Commissioning button of the board the module is connected to. Immediately click **Retry** to attempt to add the radio module again.
- **MicroPython application running on radio module.** The device may have a MicroPython application running that could disrupt the serial communication. You can increment the XCTU guard time to mitigate this issue by clicking the **general preferences page** link. Another possibility is that your device is a legacy programmable radio module. In that case, add the

device manually and select the **programmable** checkbox. For more information, see [Add a programmable radio module](#).

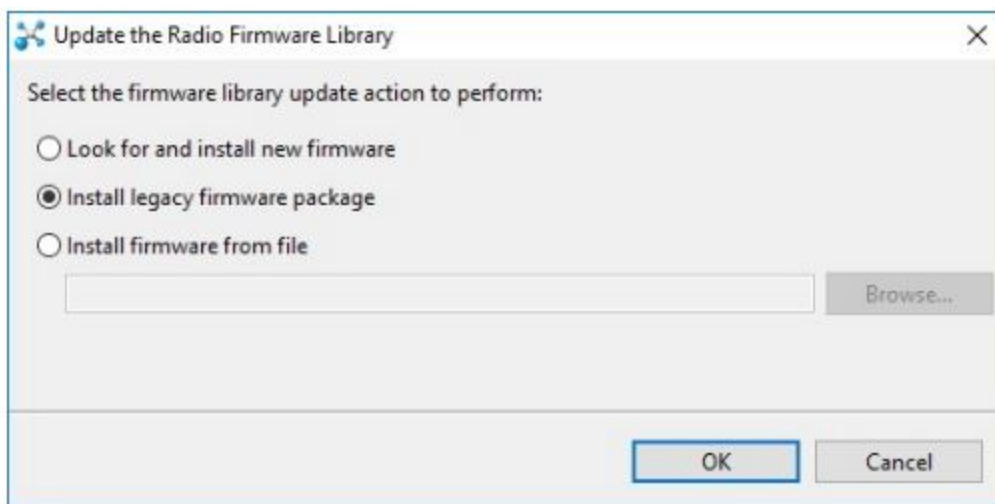
- **Damaged radio module.** The firmware of the module may be damaged or the module may be in programming mode. Click the **recovery tool** link from the dialog to open XCTU’s XBee recovery tool. For more information, see [XBee recovery tool](#).

After adding a device to the list of devices, the **Function** field states "XXX - Not found in XCTU". The firmware version of the radio module is not installed in XCTU. The radio module firmware may be legacy firmware, and you need to install the legacy radio firmware package on XCTU. Follow these steps to do so:

1. Select **Help > Install Legacy Radio Firmware** option.



2. The **Install legacy firmware package** option should be already selected in the **Update the Radio Firmware Library** dialog. If not, select it.



3. Click **OK** to start. A dialog displays the status of the download process.
  - If it does not fix the problem, try the same steps but choose the **Look for and install new firmware** option in the **Update the Radio Firmware Library** dialog.

- If the problem persists, contact with support so they can provide you the proper firmware for your module. Then, follow previous steps but the **Install firmware from file** option in the **Update the Radio Firmware Library** dialog and select the firmware file you were provided with.

## Known issues

---

XCTU currently has the following known issues and limitations:

- XCTU is not compatible with the following Digi RF devices:
  - XStream OEM RF modules family
  - XCite OEM RF modules family
- XCTU includes a mechanism to prevent the computer from going to sleep automatically in Windows and MacOS systems when it is performing long-duration tasks such as network discovery and range testing. Note that if you force the computer to sleep while XCTU is performing such tasks, the USB ports that XCTU was communicating with may become unresponsive after the system wakes from sleep. Furthermore, if you try to close the communication with them they could hang and they won't recover even if you close XCTU. If you experience this issue, you must unplug the USB cable and plug it in again to restore the communication with the port. You can do this even while XCTU is running.
- Zigbee local modules configured as end devices cannot perform network discovery in network working mode.
- The XLR PRO Radio Solution and programmable XBee radio modules do not support the XCTU recovery feature.
- UNIX-based operating systems allow you to open multiple instances of the same port. This scenario can produce unexpected behavior if you have the same port open simultaneously by different Linux apps.
- Cellular modem update over USB only works in Windows. In addition, it requires that the appropriate USB drivers are installed on the computer and the usage of either a Digi XBIB-CU-TH development board or own hardware that makes a USB port available to the PC.

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Adafruit:](#)

[247](#)