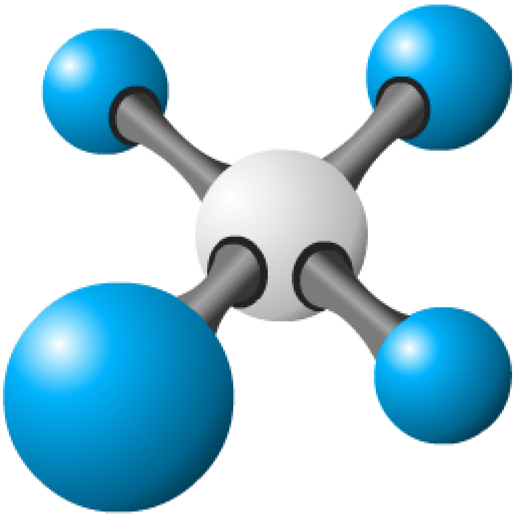


XCTU Documentation

XCTU 6.2.0 - 90001458_A

Table of Contents

- 1. XCTU Overview . . . 3
  - 1.1 XCTU Requirements . . . 4
  - 1.2 Download and install XCTU . . . 6
  - 1.3 Concepts and terminology . . . 9
  - 1.4 User Guide . . . 15
    - 1.4.1 XCTU layout . . . 16
    - 1.4.2 Add radio modules . . . 19
    - 1.4.3 Organize your modules . . . 29
    - 1.4.4 Configure your modules . . . 36
    - 1.4.5 Talk with your modules . . . 47
      - 1.4.5.1 API Console . . . 51
      - 1.4.5.2 AT Console . . . 62
    - 1.4.6 Manage your radio network . . . 70
    - 1.4.7 XCTU Configuration . . . 78
    - 1.4.8 Software Updates . . . 85
  - 1.5 XCTU tools . . . 90
    - 1.5.1 Frames generator tool . . . 91
    - 1.5.2 Frames interpreter tool . . . 95
    - 1.5.3 Recovery tool . . . 99
    - 1.5.4 Load console session tool . . . 106
    - 1.5.5 Range test tool . . . 111
    - 1.5.6 Firmware Explorer tool . . . 118
    - 1.5.7 Serial Console tool . . . 125
  - 1.6 Troubleshooting . . . 137
  - 1.7 Known issues and limitations . . . 141
  - 1.8 How-to articles . . . 142
    - 1.8.1 How-to: Update the firmware of your modules . . . 143
    - 1.8.2 How-to: Visualize your network . . . 146

XCTU Overview

XCTU is a multi-platform application that enables developers to interact with Digi radio frequency (RF) modules through a simple-to-use graphical interface. The application includes embedded tools that make it easy to set up, configure and test Digi RF modules.

XCTU now offers the following functionality:

- You can **manage and configure multiple RF devices**, even remotely (over-the-air) connected devices.
- The **firmware update** process **seamlessly** restores your module settings, automatically handling mode and baud rate changes.
- Two specific **API** and **AT consoles**, have been designed from scratch to communicate with your radio devices.
- You can now **save your console sessions** and load them in a different PC running XCTU.
- XCTU includes a set of embedded tools that can be executed without having any RF module connected:
  - **Frames generator**: Easily generate any kind of API frame to save its value.
  - **Frames interpreter**: Decode an API frame and see its specific frame values.
  - **Recovery**: Recover radio modules which have damaged firmware or are in programming mode.
  - **Load console session**: Load a console session saved in any PC running XCTU.
  - **Range test**: Perform a range test between 2 radio modules of the same network.
  - **Firmware explorer**: Navigate through XCTU's firmware library.
- An update process allows you to **automatically update the application itself and the radio firmware** library without needing to download any extra files.
- XCTU contains **complete and comprehensive documentation** which can be accessed at any time.

XCTU Requirements

This section describes the software and hardware requirements needed for the XCTU application.

Software

XCTU is compatible with the following operating systems:

- Windows Vista/7/8 (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)

Hardware

The following sections describe the hardware system requirements and the Digi RF modules supported by XCTU.

System requirements

These are the minimum hardware system requirements to properly run XCTU:

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

Supported RF modules

Although XCTU is compatible with the majority of RF modules from Digi, not all of them can be configured with this tool. Here is the complete list of the compatible RF modules:

- XBee®/XBee-PRO® RF Module Family
    - 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 868LP
    - Programmable XBee 868LP
- XTend® RF Module Family
- XLR PRO radio solution

All of these modules use a serial interface to communicate with the module and configure it. XCTU uses this serial link to interact with the radio module, providing an easy-to-use and intuitive graphical interface.

Download and install XCTU

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

1. Preparing your system

In some cases it is necessary to configure your system in order to work with XCTU. This chapter describes the required steps before using the tool.

1.1. 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 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 your user in the system.

1.2. OSX

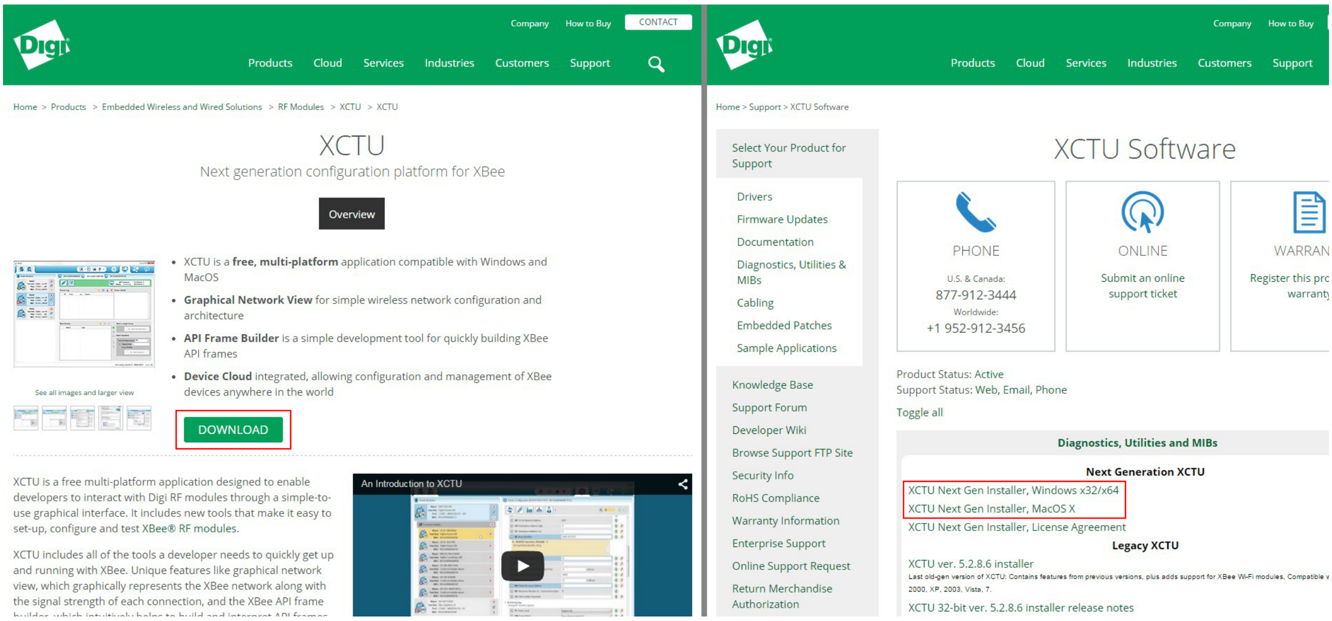
OSX version 10.8 (Mountain Lion) and greater will only allow you to install applications downloaded from the Apple Store. In order to install XCTU you will need to temporarily disable this setting. Follow these steps to do so:

1. Click on the **Apple** icon in the top left corner of your screen and choose **System Preferences**.
2. Next, click the **Security & Privacy** icon.
3. To edit the security settings you must click the **padlock icon** in the bottom left of the window.
4. Enter your Mac credentials and hit **Unlock**.
5. The next screen displays the options for the "Allow applications downloaded from" feature. Click the **Anywhere** radio button and, in the confirmation window, click **Allow From Anywhere**.

> We recommend that you set this option back to *Mac App Store* or *Mac App Store and identified developers* once you have finished installing the unsigned software.

2. Download XCTU

You can download XCTU from the following link:



XCTU

Next generation configuration platform for XBee

- XCTU is a **free, multi-platform** application compatible with Windows and MacOS
- **Graphical Network View** for simple wireless network configuration and architecture
- **API Frame Builder** is a simple development tool for quickly building XBee API frames
- **Device Cloud** integrated, allowing configuration and management of XBee devices anywhere in the world

XCTU is a free multi-platform application designed to enable developers to interact with Digi RF modules through a simple-to-use graphical interface. It includes new tools that make it easy to set-up, configure and test XBee® RF modules.

XCTU includes all of the tools a developer needs to quickly get up and running with XBee. Unique features like graphical network view, which graphically represents the XBee network along with the signal strength of each connection, and the XBee API frame builder, which intuitively helps to build and interpret API frames

XCTU Software

U.S. & Canada:
877-912-3444
Worldwide:
+1 952-912-3456

Submit an online support ticket

Register this pro
warranty

Product Status: Active
Support Status: Web, Email, Phone

Diagnostics, Utilities and MIBs

Next Generation XCTU

XCTU Next Gen Installer, Windows x32/x64
XCTU Next Gen Installer, MacOS X
XCTU Next Gen Installer, License Agreement

Legacy XCTU

XCTU ver. 5.2.8.6 installer
Last old-gen version of XCTU: Contains features from previous versions, plus adds support for XBee Wi-Fi modules, Compatible v
2000, XP, 2003, Vista, 7.
XCTU 32-bit ver. 5.2.8.6 installer release notes

> The installer of MacOS is distributed inside a zip file; you will need to uncompress it when downloaded.

3. Install XCTU

Once the download is complete, install the tool following the steps in the XCTU Setup Wizard.

> The installation process provided by the Setup Wizard is the same for all supported platforms (Windows, MacOS and Linux).

Once the tool is installed, you will be prompted with a "What's new" dialog where you can review the new features introduced in each version of XCTU. Close the dialog when you are done.

4. Install the updates

Once XCTU is loaded, you may be notified about XCTU software updates. It is recommended that you update XCTU to the latest available version.

A popup window will appear at the right bottom corner of XCTU when a new version is available. Click on that window to proceed with the update.

You can also update the tool manually from **Help > Check for XCTU Updates** menu option.

5. Install the drivers

> This step is **optional**. You don't need to install the USB drivers unless your operating system notifies you that it could not install them automatically.

Generally, when you connect the XBee board to your computer for the first time the drivers are installed automatically. However, there are times when this does not occur and the drivers have to be installed manually.

In that case, you can download and install the USB drivers from Digi Support Site. Choose your operating system, download the file and follow the steps of the installation wizard.

Concepts and terminology

The following are important concepts regarding the radio frequency modules and the application itself. It is important to understand these concepts to work effectively with XCTU.

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.

As described in the hardware requirements section, XCTU is compatible with Digi's **XBee** and **XTend** RF modules and **XLR PRO**.

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.

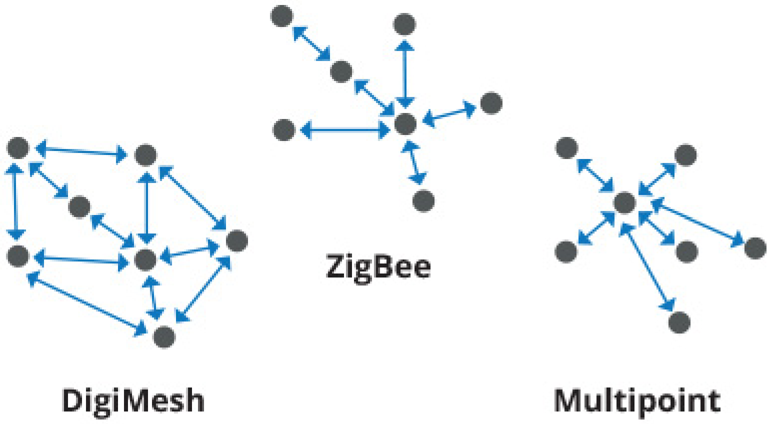
Radio firmware

Radio firmware is the program code stored in the radio module's persistent memory that provides the control program for the device. XCTU allows you to update or change the firmware of a module. This can be a common task if, for example, you want to change the role of the device, or you want to use the latest version of the firmware.

Radio communication protocols

A radio communication protocol is a set of rules for data exchange between radio devices. An XBee 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’s proprietary) - ZNet - IEEE 802.11 (Wi-Fi) - Point-to-multipoint (Digi’s proprietary) - XSC (XStream compatibility)	

Not all the XBee devices can run all these communication protocols. The combination of XBee hardware and radio firmware determines the protocol that an XBee device can execute. Refer to the XBee RF Family Comparison Matrix for more information about the available XBee RF modules and the protocols they support.

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, the 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).

Application Transparent (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, put it in command mode to send the configuration commands.

AT Command mode

When the radio module is working in AT operating mode, settings are configured using the command mode interface.

To enter AT command mode, you must send the 3-character command sequence (usually "+++") within one second. Once the 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.

The structure of an AT command follows:

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

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 issuing the CN AT 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 in API frames.

The API specifies how commands, command responses, and module status messages are sent and received from the module using the serial interface. API operation mode enables many operations, such as the following:

- 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) works similarly to API mode. The only difference is that when working in API escaped mode, some bytes of the API frame specific data must be escaped.

API escaped operating mode is used to add reliability to the RF transmission, which prevents conflicts with special characters such as the start-of-frame byte (0x7E). Since 0x7E can only appear at the start of an API packet, if 0x7E is received at any time, it can be assumed that a new packet has started regardless of length. In API escaped mode, those special bytes are escaped.

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:

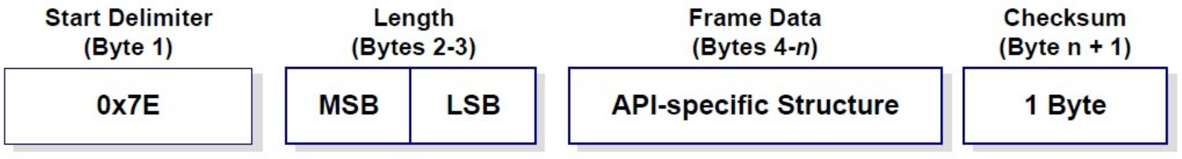
- **0x7E**: Frame delimiter
- **0x7D**: Escape
- **0x11**: XON
- **0x13**: XOFF

> XCTU is compatible with both API and API escaped operating modes, so you won't need to worry about escaping characters when interacting with API escaped radio modules. XCTU does this for you.

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:



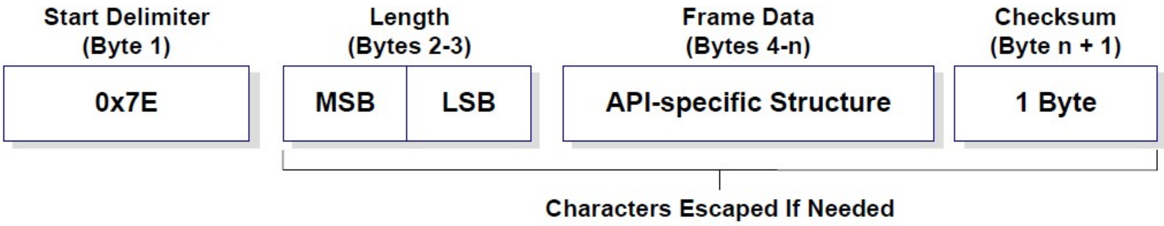
Where:

- **Start Delimiter**: This field is always 0x7E
- **Length**: The length field has a two-byte value that specifies the number of bytes contained in the frame data field. It does not include the checksum field.
- **Frame Data**: The content of this field is 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).

  > There are many API frame types or identifiers. If you want to know the specific data that should fill in a determined API frame, look at the API frames generator tool, which will help you to build and fill any type of API frame.

- **Checksum**: Byte containing the hash sum of the API frame bytes.

In API escaped mode, there may be some bytes in the Length, Frame Data and Checksum fields that need to be escaped.



> Remember that when you work with XCTU, you don't need to worry about escaping characters, as XCTU does this for you automatically.

AT settings or commands

The firmware running in the XBee 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 the XBee RF modules can be configured with AT commands.

All the firmware settings or commands are identified with two ASCII characters and some applications and documents refer to them as **AT settings** or **AT commands**.

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

- **AT operating mode**. In this mode, you must put the module in a special mode called command mode, so it can receive AT commands. For more information about configuring XBee RF modules working in AT operating mode, see the **Application Transparent (AT) operating mode** topic.
- **API operating mode**. To configure or execute AT commands when the XBee RF module operates in API 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 XBee RF module. For more information about API frames, see the **API frames** topic.

Local and remote radio modules

Depending on the way the radio modules are added to the devices list, they can be **local** modules or **remote** modules.

Local radio modules

A local radio module is any module added to the devices list using the **Add a radio module** or the **Discover radio modules** buttons. See the Adding radio modules manually and Discovering radio modules topics for more information.

They are called local because they are physically attached to the PC through a serial port or a USB port, and XCTU has direct communication with them.

These local radio modules have the ability to discover remote radio modules in the same network if their protocol is ZigBee or DigiMesh and, if remote modules are added, the list containing them can be expanded or collapsed.

Also, local radio modules are always configurable (if the Configuration working mode is active) and you can always communicate with these modules through their respective consoles when the Consoles working mode is active.

Remote radio modules

Remote radio modules are contained in a sub-list under a local radio module. They are connected to the same network as that local module and have been added using the local radio module search feature. See the Search button topic for more information.

Remote radio modules are not physically connected to the PC and communication with them is performed over-the-air through the corresponding local radio module.

If the local device containing remote modules is configured in AT (transparent) operating mode, it won't be possible to configure its remote radio modules due to a protocol limitation. If the local radio module is configured in API operating mode, its remote radio modules can be configured just like any local module. See the Radio module operating modes topic for more information about the operating modes of the radio modules.

As remote radio modules are not physically connected to the PC, they won't have a communication console when changing to Consoles working mode, neither the possibility to obtain its network topology in the Network working mode.

User Guide

This guide describes how to use the XCTU tool to configure and communicate with your radio devices.

- XCTU Layout

The main purpose of the application is to configure and interact with radio modules connected to your PC, so the next step you should perform just after starting the XCTU is to add one or more radio modules to the devices list.

- Add radio modules
- Organize your modules

Once you have one or more radio modules in the list, you can interact with them. The following list displays the different things you can do with your radio modules:

- Configure your modules
- Talk with your modules
- Manage your radio network

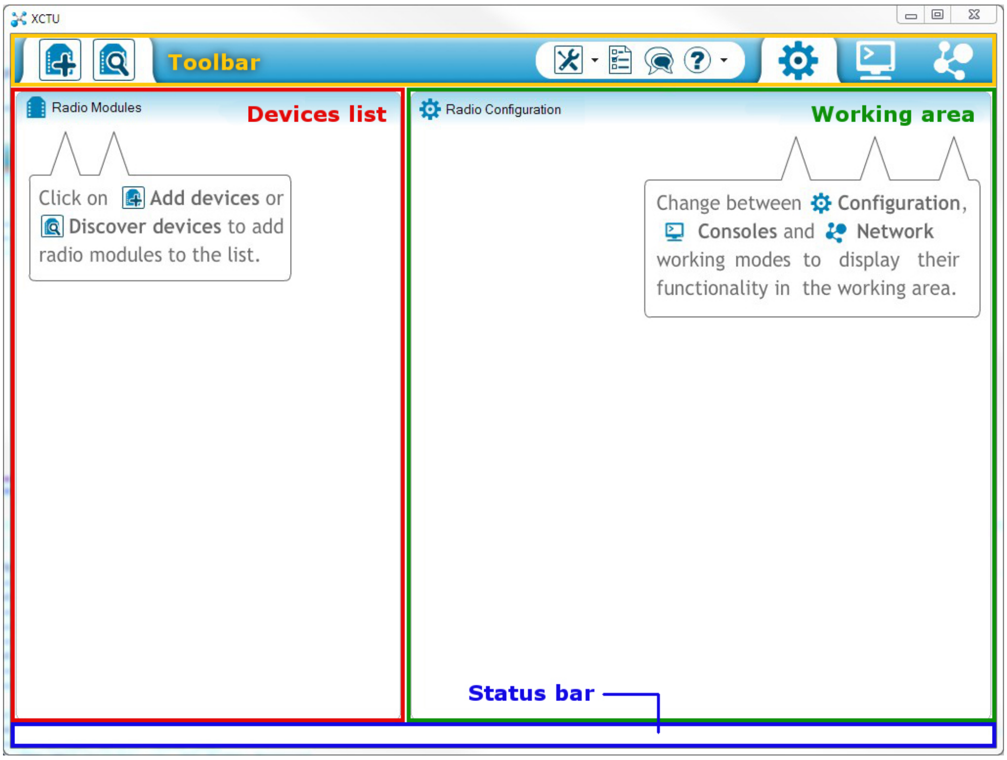
Here are some additional options for configuring XCTU and keeping it up to date.

- XCTU Configuration
- Software Updates

XCTU layout

The structure of the XCTU controls and working modes are as follows.

Application structure

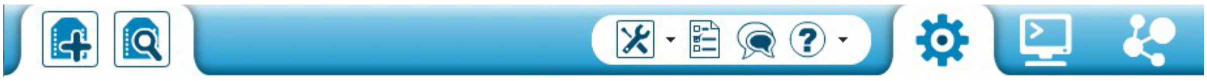


Click on **Add devices** or **Discover devices** to add radio modules to the list.

Change between **Configuration**, **Consoles** and **Network** working modes to display their functionality in the working area.

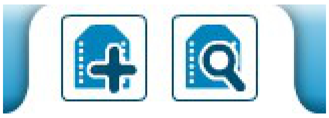
The tool is divided into four main sections:

Main Toolbar

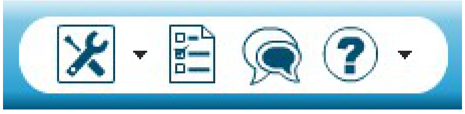


The **main toolbar** is located at the top and is divided into three different parts:

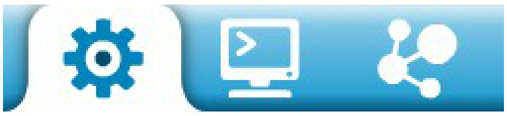
- The first part contains two icons used to add radio modules to the radio modules list.



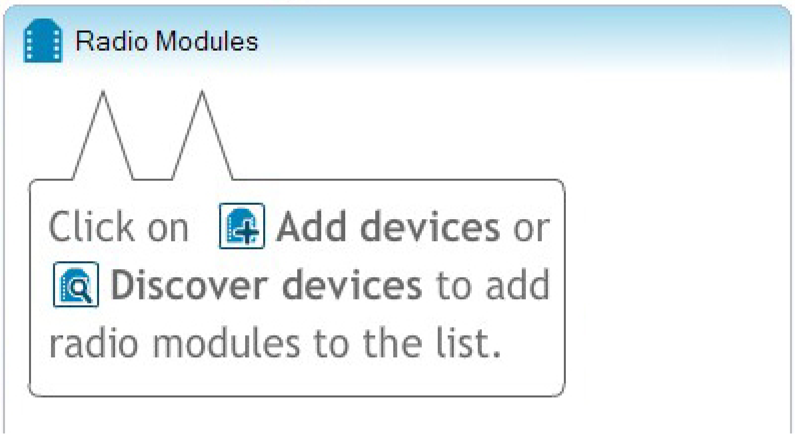
- The second part contains the XCTU static functionality. This is functionality that does not need a radio module to be executed. This part includes the XCTU tools, the XCTU configuration, the feedback form and the help and updates functions.



- The third part of the toolbar is a tabbed list that provides access to the different working modes of the tool and displays those modes in the working area. To use this functionality, you must have added one or more radio modules to the list.



Devices list



Radio Modules

Click on Add devices or Discover devices to add radio modules to the list.

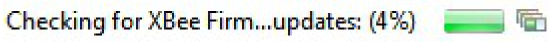
The **radio modules list** is located at the left side of the tool and displays the radio modules that are connected to your PC and which you can interact with. If you know the serial port configuration of a radio module, you can add it to the list directly. Otherwise you can use the discovery feature of XCTU to find radio modules connected to your PC and add them to the list.

Depending on the protocol of the local radio modules added, it is also possible to 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 them must be selected.

Status bar



Checking for XBee Firm...updates: (4%)

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

Application working modes

A **working mode** represents a layout which displays operations you can perform with a radio module. Usually, the working mode functionality is displayed in the working area. The tool has four working modes:

- **Configuration mode**: Allows you to configure the selected radio module from the list. For further information about this working mode see the Configure your modules topic.
- **Consoles mode**: Allows you to interact or communicate with the selected radio module. For further information about this working mode see the Talk with your modules topic.

- **Network mode**: Allows you to discover and see the network topology of 802.15.4, ZigBee and DigiMesh protocols. For further information about this working mode see the Manage your radio network topic.

Only one working mode can be selected at the same time and, by default, the Configuration mode is selected when you launch XCTU.

Add radio modules

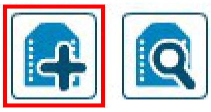
To interact with a radio module connected to your PC, you must add it to the list of devices. There are two ways to do that:

1. If you know the serial configuration of your radio module, you can add it **manually**.
2. If you don't know the serial configuration of your radio module, don't know the port it is connected to, or you simply want to add more than one module, you can use the device **discovery** utility of XCTU.

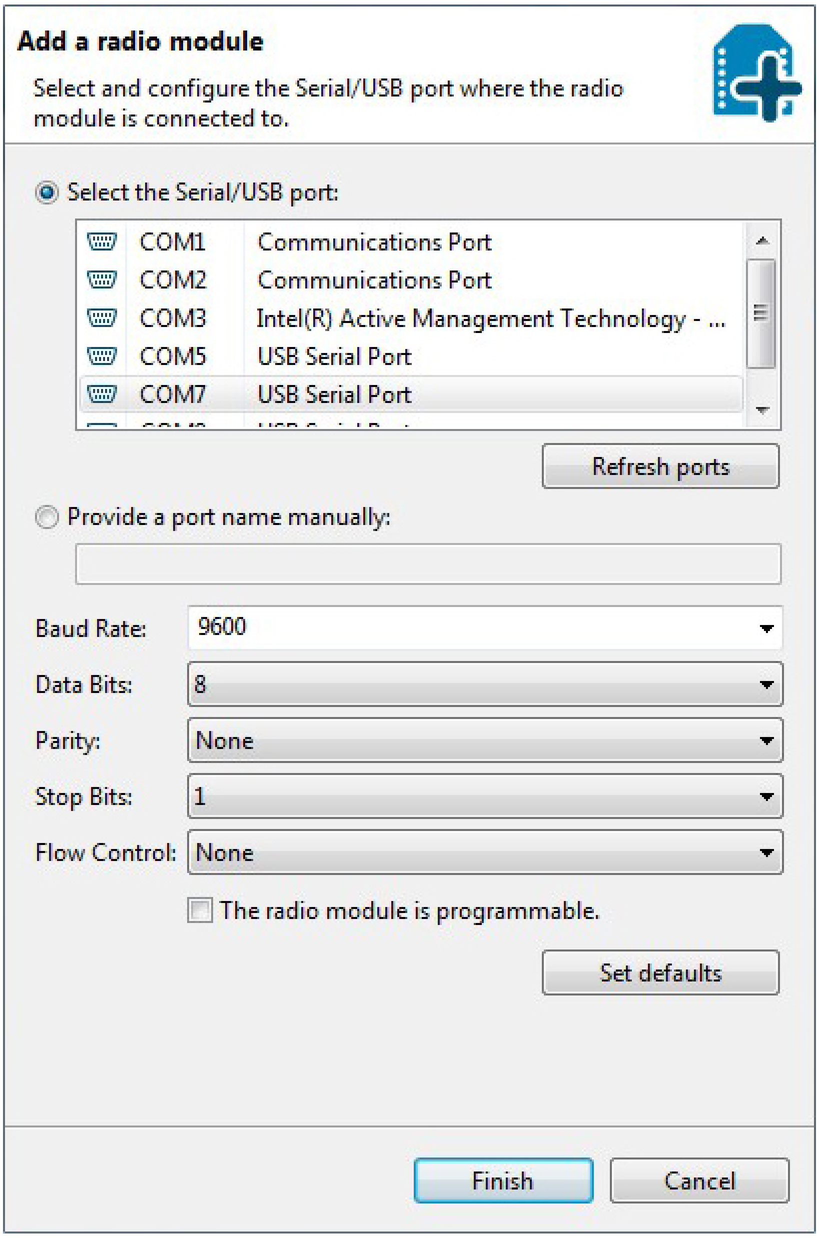
Adding radio modules manually

Follow these steps to add a module to the list of devices 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
COM2	Communications Port
COM3	Intel(R) Active Management Technology - ...
COM5	USB Serial Port
COM7	USB Serial Port

○ Provide a port name manually:

Baud Rate: 9600

Data Bits: 8

Parity: None

Stop Bits: 1

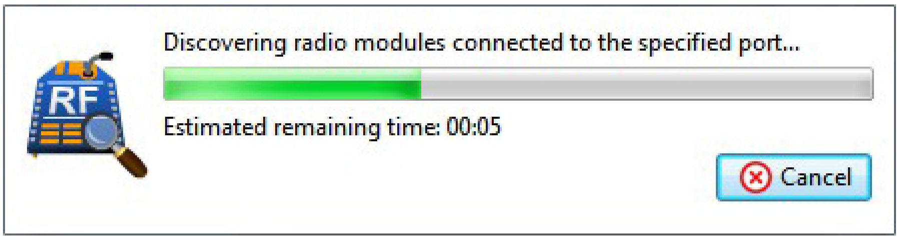
Flow Control: None

☐ The radio module is programmable.

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.

> Custom baud rates can only be typed under Windows OS.

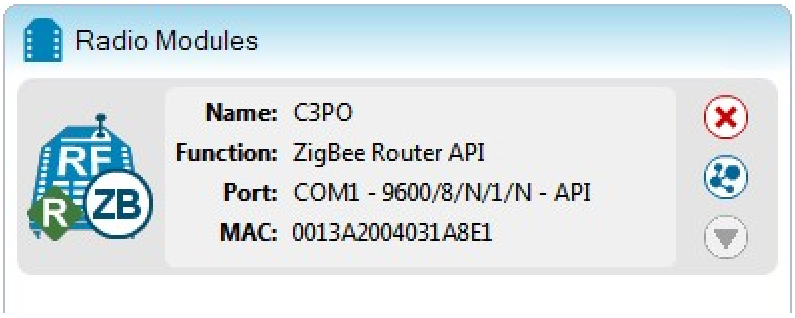
3. When ready, click **Finish** to add the radio module to the list of radio modules.



Discovering radio modules connected to the specified port...

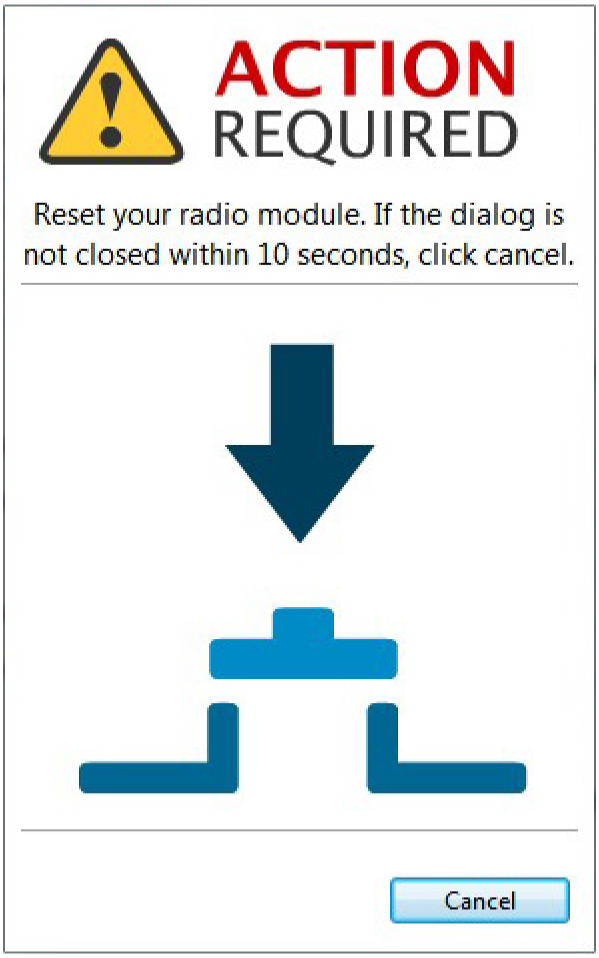
Estimated remaining time: 00:05

If the settings were correctly configured and the radio module was connected to the selected port, it should now be displayed in the list.



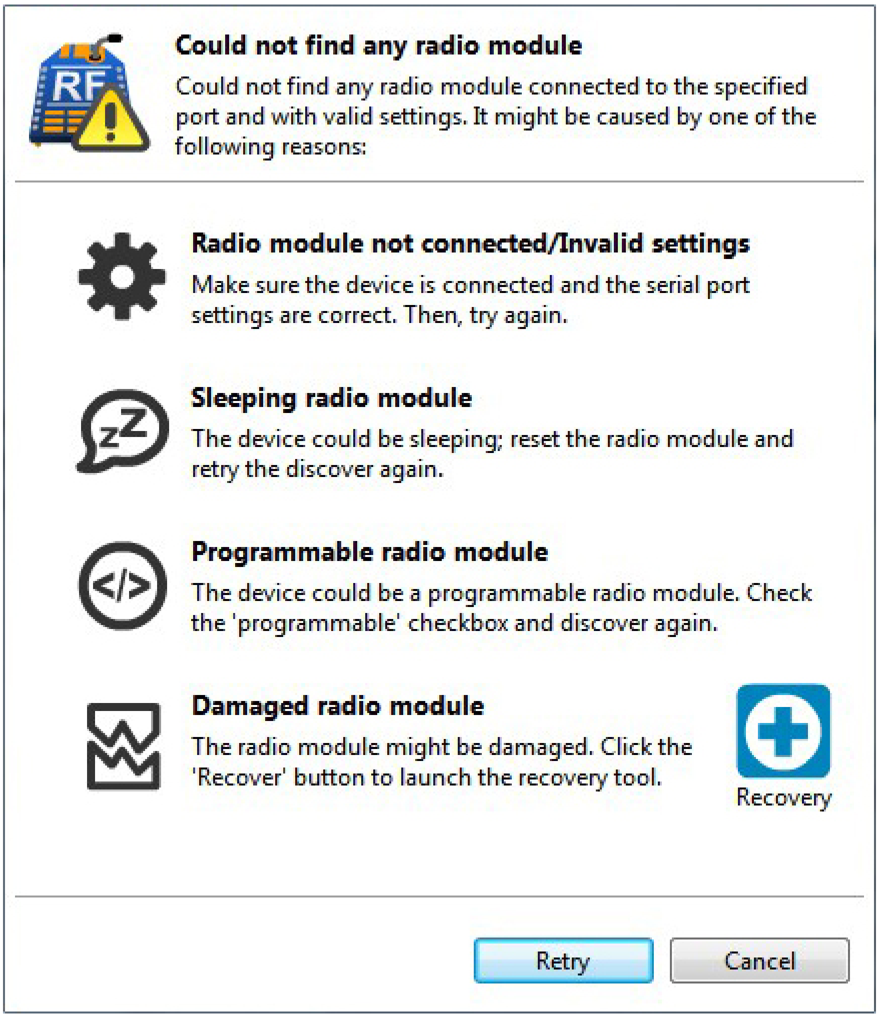
Name: C3PO
Function: ZigBee Router API
Port: COM1 - 9600/8/N/1/N - API
MAC: 0013A2004031A8E1

If not, an action required dialog asks you to reset the module:



Reset your radio module. If the dialog is not closed within 10 seconds, click cancel.

After resetting the module, the action required dialog should close and your module should be added to the list. If your module could not be found, an information dialog provides possible reasons why the module could not be added:



Could not find any radio module

Could not find any radio module connected to the specified port and with valid settings. It might be caused by one of the following reasons:

Radio module not connected/Invalid settings

Make sure the device is connected and the serial port settings are correct. Then, try again.

Sleeping radio module

The device could be sleeping; reset the radio module and retry the discover again.

Programmable radio module

The device could be a programmable radio module. Check the 'programmable' checkbox and discover again.

Damaged radio module

The radio module might be damaged. Click the 'Recover' button to launch the recovery tool.

- 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.

> Most common serial configuration is:
>
> - **Baud rate**: 9600 or 115200
> - **Data bits**: 8
> - **Stop bits**: 1
> - **Parity**: None
> - **Flow control**: None

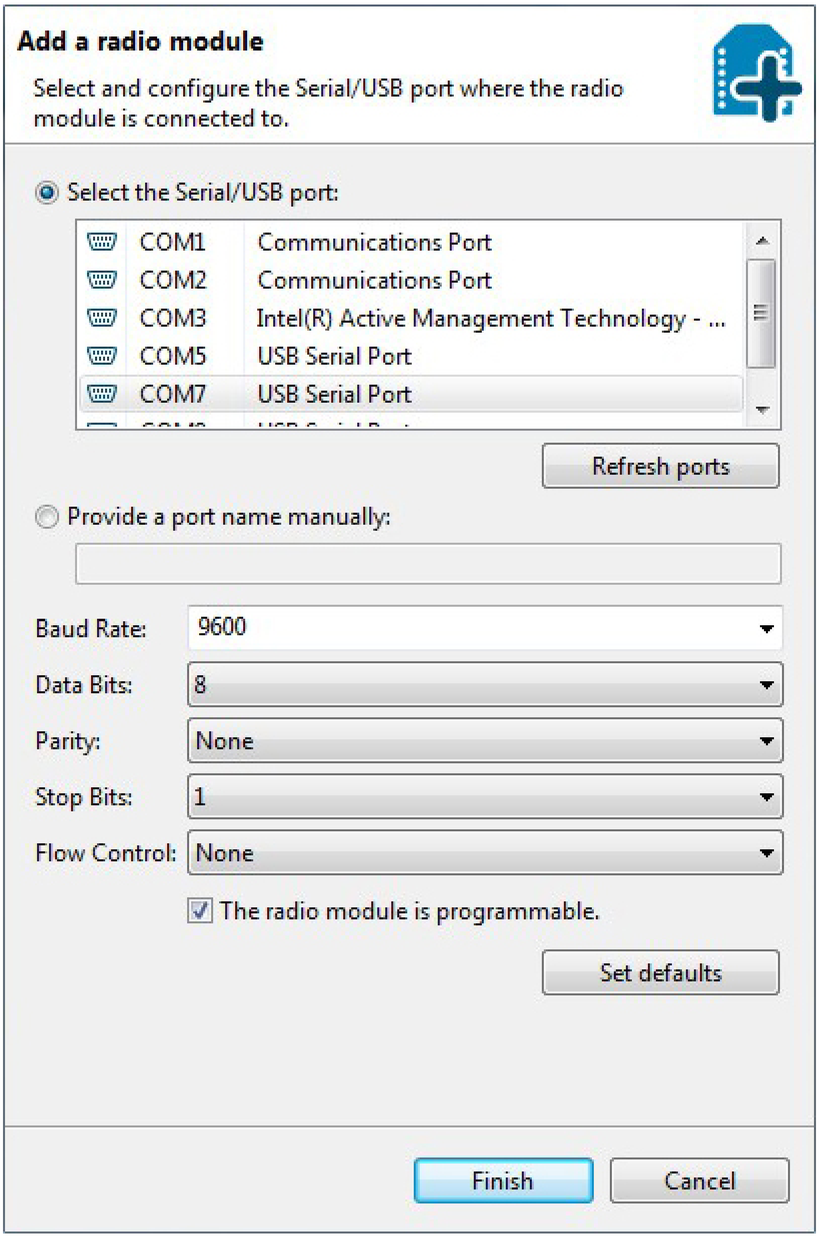
- The radio module to be added could be a sleeping node. Depending on the protocol, the module being added could be configured to sleep. If the module is sleeping at the time XCTU tries to talk with it, the module won't be added. If you believe your module could be sleeping, try to wake it up by pressing the Commissioning button of the board the module is connected to. Just after doing that, click **Retry** to try to add the radio module again.
- The radio module you are trying to add could be a programmable variant. See topic Adding a programmable radio module for further information.
- Finally, the firmware of the module could be damaged or the module could be in programming mode. XCTU provides an embedded tool that might recover your module if that is the case. Click the **Recovery** button of the dialog to open this tool. For more information about using the Recovery tool see topic Recovery tool.

Adding a programmable radio module

There are some radio module variants which are programmable and are able to run applications written in C. Normally, they are known as **Programmable XBee modules** and they can be identified by the part number labeled on the back. If they end with a **B**, they probably are programmable.

> People often confuse XBee-PRO with Programmable XBee. They are not the same thing. The -PRO suffix does not mean that the module is programmable.

Programmable modules must be added in a different way than the standard radio modules are added. You can add a programmable module through XCTU by checking the **My radio module is programmable** setting. XCTU performs the necessary actions automatically.



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
COM2	Communications Port
COM3	Intel(R) Active Management Technology - ...
COM5	USB Serial Port
COM7	USB Serial Port

○ Provide a port name manually:

Baud Rate: 9600

Data Bits: 8

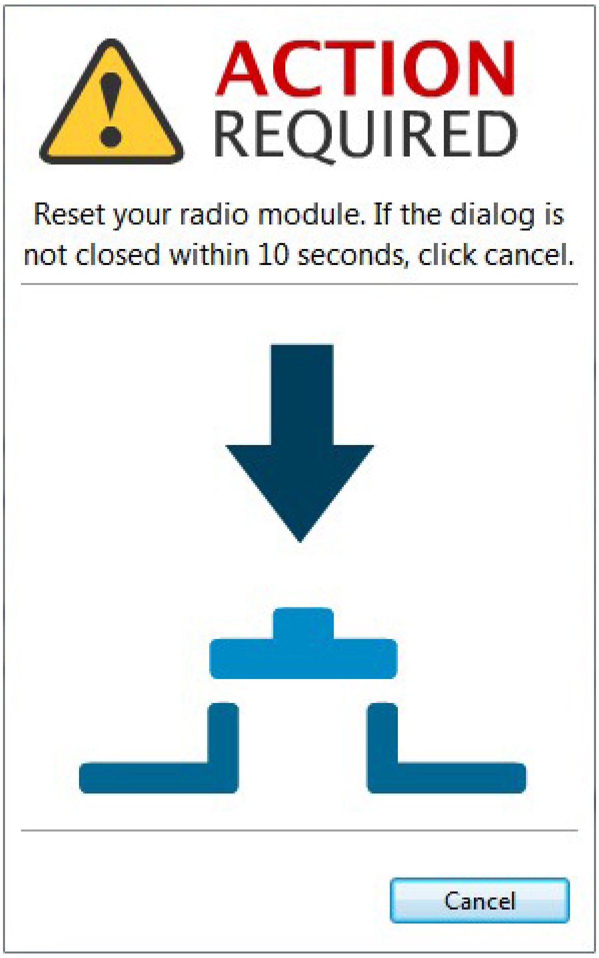
Parity: None

Stop Bits: 1

Flow Control: None

☑ The radio module is programmable.

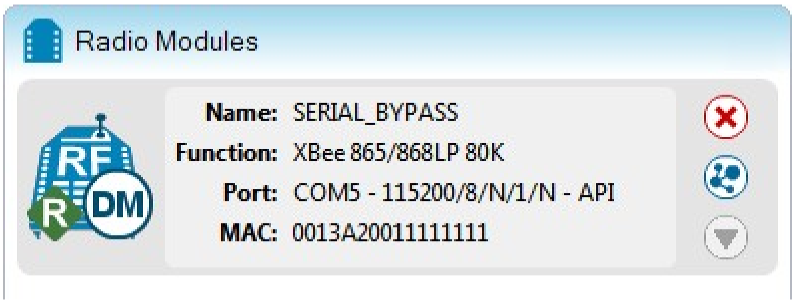
You need to reset the module to continue.



ACTION REQUIRED

Reset your radio module. If the dialog is not closed within 10 seconds, click cancel.

When the radio module has been added, it appears in the list.

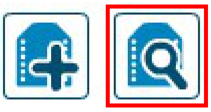


Name: SERIAL_BYPASS
Function: XBee 865/868LP 80K
Port: COM5 - 115200/8/N/1/N - API
MAC: 0013A20011111111

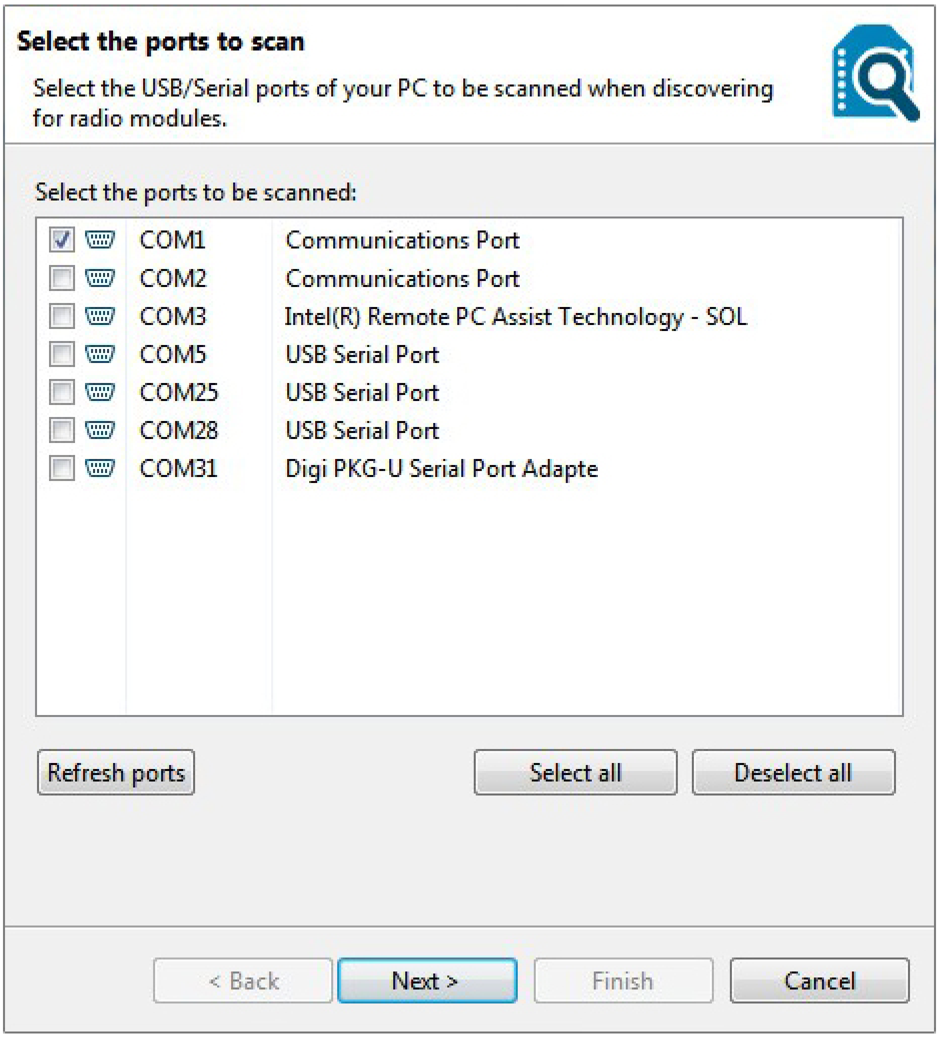
Discovering radio modules

Follow these steps to discover radio modules connected to your PC:

1. Click the **Discover radio modules** button in the toolbar.

   The Discover radio modules dialog box opens.



Select the ports to scan

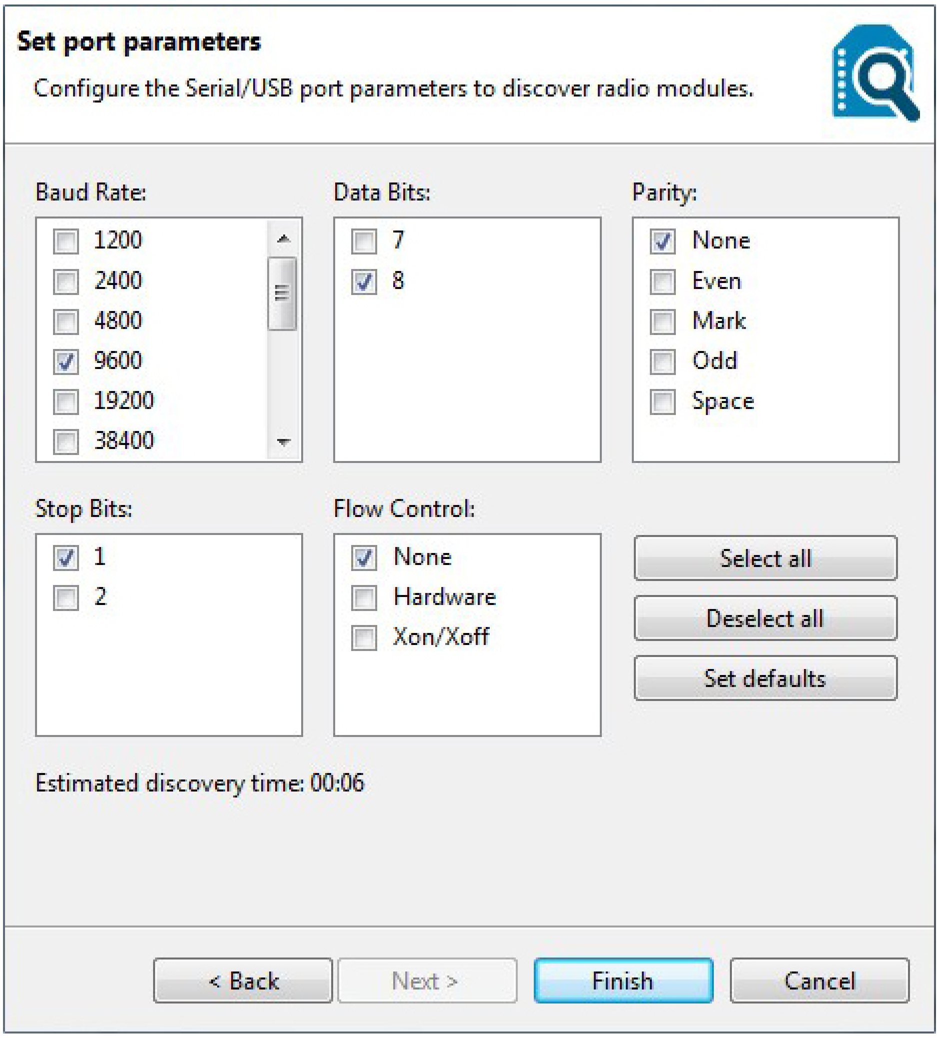
Select the USB/Serial ports of your PC to be scanned when discovering for radio modules.

Select the ports to be scanned:

☑	COM1	Communications Port
☐	COM2	Communications Port
☐	COM3	Intel(R) Remote PC Assist Technology - SOL
☐	COM5	USB Serial Port
☐	COM25	USB Serial Port
☐	COM28	USB Serial Port
☐	COM31	Digi PKG-U Serial Port Adapte

2. Select the serial ports where you want to find radio modules, then click **Next**.
3. Check all the settings you want to try in the discovery process to discover radio modules.

> The estimated discovery time is displayed in a line below the settings. The more settings added to the discovery process, the longer the discovery process.



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

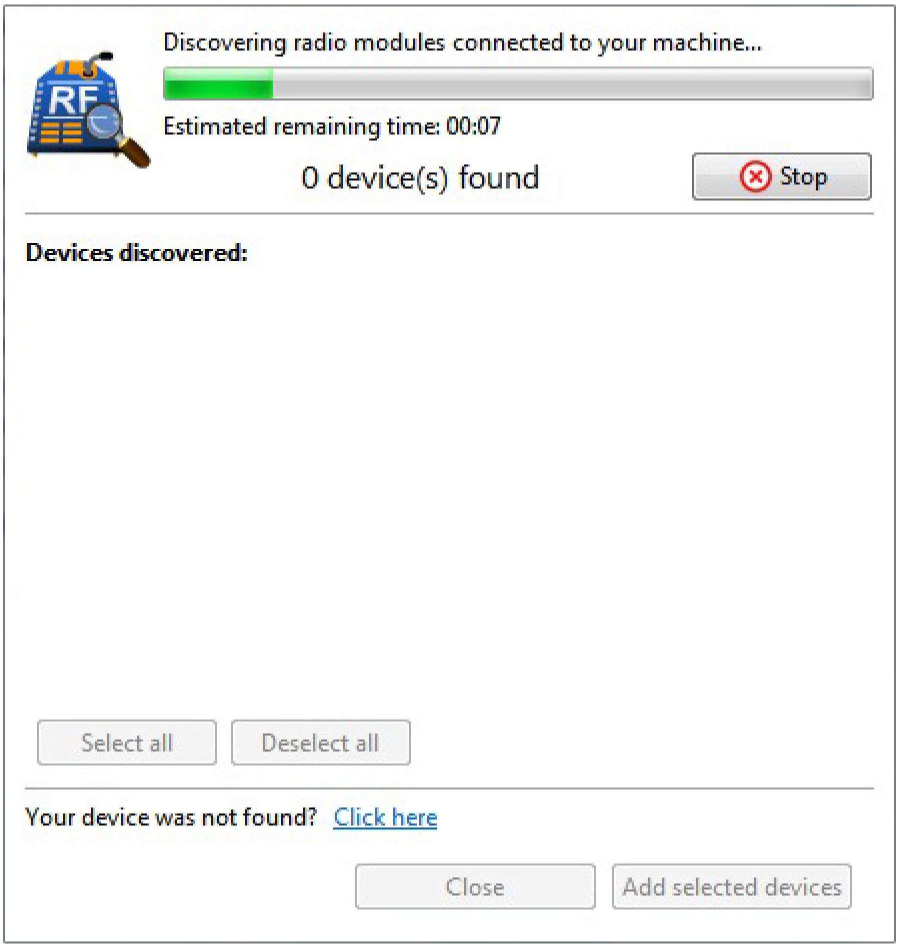
Parity:
☑ None
☐ Even
☐ Mark
☐ Odd
☐ Space

Stop Bits:
☑ 1
☐ 2

Flow Control:
☑ None
☐ Hardware
☐ Xon/Xoff

Estimated discovery time: 00:06

4. When ready click **Finish** to start the discovery process.



Discovering radio modules connected to your machine...

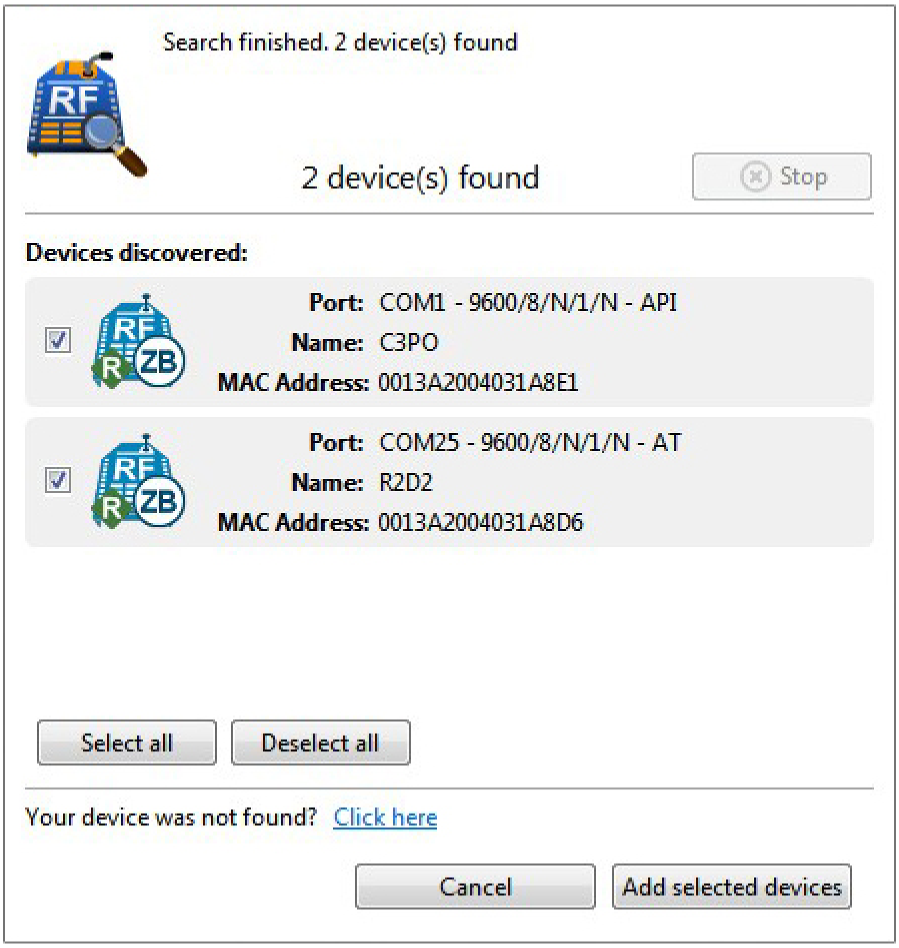
Estimated remaining time: 00:07

0 device(s) found

Devices discovered:

Your device was not found? Click here

5. As radio modules are found, they appear in the discovery process dialog box. You can stop the process at any time by clicking the **Stop** button. For example, you can stop the process if the module or modules you were looking for are already found.



Search finished. 2 device(s) found

2 device(s) found

Devices discovered:

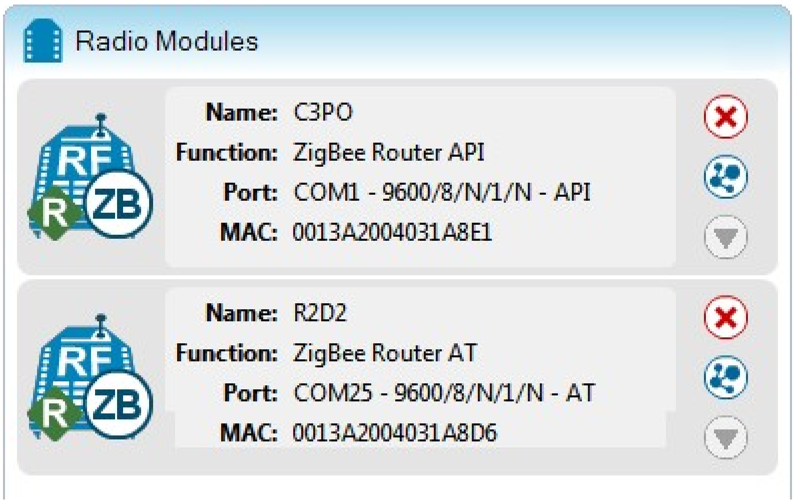
☑ Port: COM1 - 9600/8/N/1/N - API
Name: C3PO
MAC Address: 0013A2004031A8E1

☑ Port: COM25 - 9600/8/N/1/N - AT
Name: R2D2
MAC Address: 0013A2004031A8D6

Your device was not found? Click here

6. Finally, check the radio modules you want to add to the list and click the **Add selected devices** button.

The modules appear in the devices list.



Radio Modules

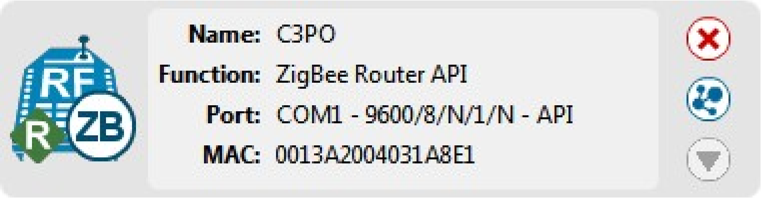
Name: C3PO
Function: ZigBee Router API
Port: COM1 - 9600/8/N/1/N - API
MAC: 0013A2004031A8E1

Name: R2D2
Function: ZigBee Router AT
Port: COM25 - 9600/8/N/1/N - AT
MAC: 0013A2004031A8D6

Organize your modules

When a device is added to the list of devices using the add or discover methods, it is always a local device. For more information about local and remote modules, see the Local and remote radio modules topic.

A local radio module appears as a big button that displays identifying information about the module.



Name: C3PO
Function: ZigBee Router API
Port: COM1 - 9600/8/N/1/N - API
MAC: 0013A2004031A8E1

- **Icon**: The icon displays the module type and protocol of the device. In some cases it contains a circle with two letters that indicate the protocol of the radio module.

RF ZB	**XBee** ZigBee protocol	RF WF	**XBee** Wi-Fi protocol
RF DM	**XBee** DigiMesh (Digi's proprietary protocol)	RF XC	**XBee** XStream Compatibility protocol
RF 802	**XBee** 802.15.4 protocol	XT	**XTend** XTend native protocol
RF DP	**XBee** Point-to-multipoint protocol	DM	**XTend** DigiMesh (Digi's proprietary protocol)
RF SE	**XBee** Smart Energy protocol	XLR	**XLR** XLR PRO native protocol
RF ZN	**XBee** ZNet protocol		

A small image at the bottom-left side indicates the role of the module within its network:

C	Coordinator
R	Router
E	End device

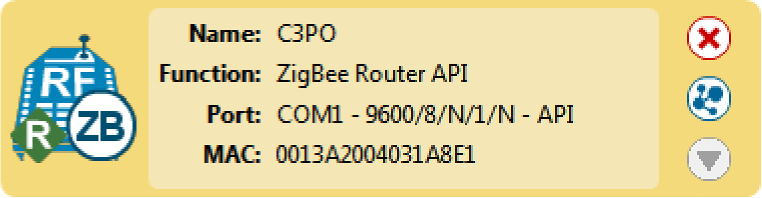
- Hovering the mouse pointer over the icon displays more information about the module.

- **Information box**: An information box next to the icon provides information about the radio module:
  - **Name**: Displays the radio module's Node Identifier (NI).
  - **Function**: Indicates the radio module's firmware.
  - **Port**: Displays the radio module's port as well as the operating mode - AT (transparent), API or API Escaped. See the Radio module operating modes topic for more information about operating modes.
  - **MAC**: Displays the 64-bit physical address of the radio module.
- **Management buttons**: The buttons on the right side of the radio module button perform actions on the module:

⊗	**Remove button**	Removes the radio module from the list. See the Remove button topic for further information.
●	**Search button**	Depending on the protocol of the radio module, this button lets you discover remote radio modules in the same network. See the Search button topic for further information.
▼	**Expand/Collapse button**	If a radio module has been discovered using the search button, this button is enabled, allowing you to expand or collapse the list. See the Expand/collapse button topic for further information.

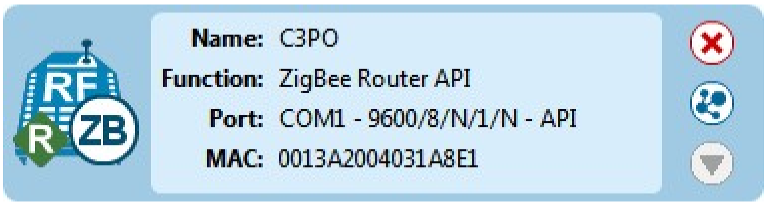
Selecting a module

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.



Name: C3PO
Function: ZigBee Router API
Port: COM1 - 9600/8/N/1/N - API
MAC: 0013A2004031A8E1

When you select it, the background changes to blue.

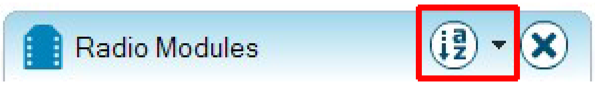


Name: C3PO
Function: ZigBee Router API
Port: COM1 - 9600/8/N/1/N - API
MAC: 0013A2004031A8E1

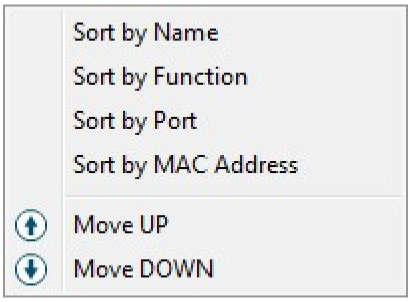
Selecting a radio module refreshes the contents of the working area, displaying the information or actions you can perform over the selected module. Remember that the contents of the working area depend on the active working mode.

Sorting radio modules

The radio modules are displayed in the list in the order they were added. You can also sort them by name, MAC address, function or serial port. To access this feature, select a device from the list and click the **sort** button of the devices list toolbar:



A popup menu lists the available sorting options:



- **Sort by Name:** Sorts the devices of the list alphabetically by name (Node Identifier).
- **Sort by Function:** Sorts the devices of the list alphabetically by function.
- **Sort by Port:** Sorts the devices of the list alphabetically by port name.
- **Sort by MAC Address:** Sorts the devices of the list by MAC address.

  Other sorting options:

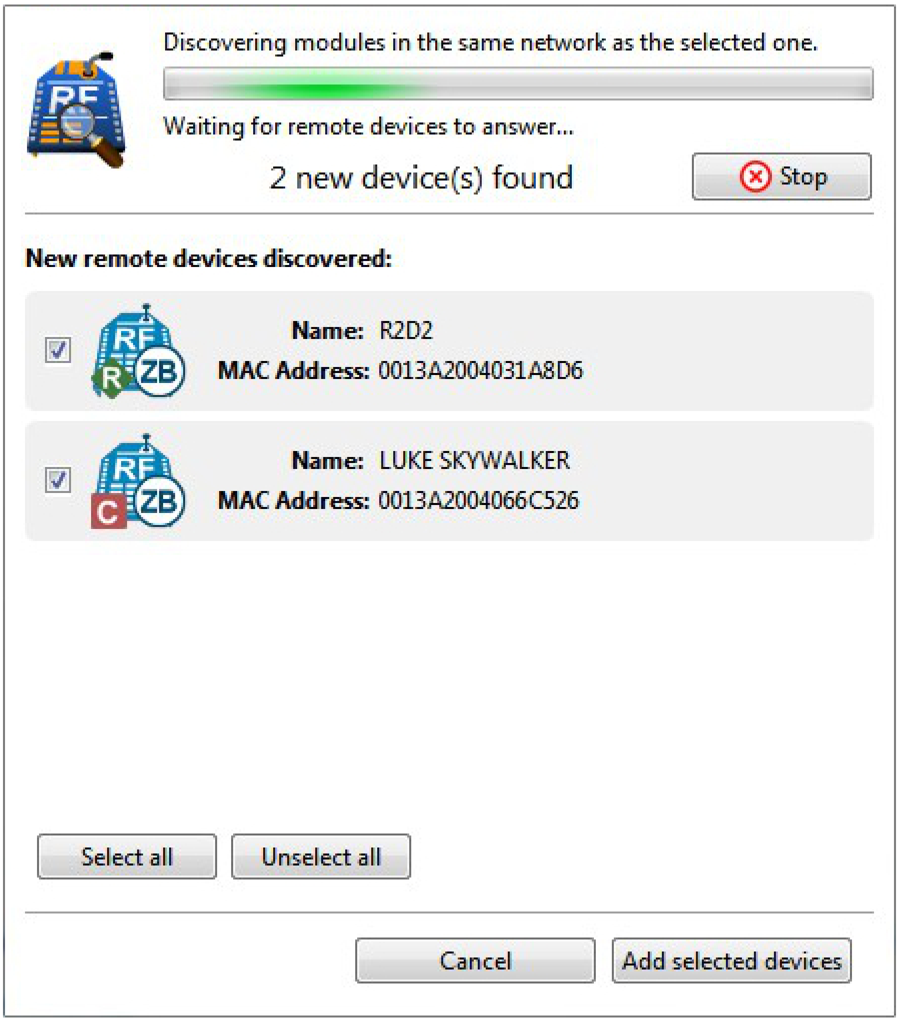
⬆	Moves the selected module one position up in the list.
⬇	Moves the selected module one position down in the list.

⊗	The **Clear radio modules list** button, next to the sorting menu, allows you to remove all modules from the list.

Search button

The action this button performs depends on the protocol of the radio module:

- **ZigBee or DigiMesh protocols**: The button executes a discovery process to find remote radio modules in the same network as the local module. When you click the search button, a new dialog indicates the progress of the discovery process. As new remote radio modules are found, they appear in the discovery process dialog box.



Discovering modules in the same network as the selected one.

Waiting for remote devices to answer...

2 new device(s) found

New remote devices discovered:

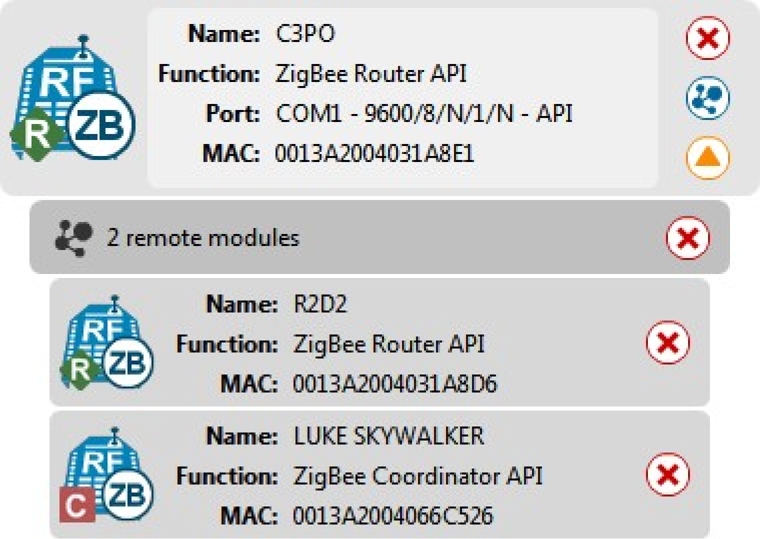
☑ **Name:** R2D2
MAC Address: 0013A2004031A8D6

☑ **Name:** LUKE SKYWALKER
MAC Address: 0013A2004066C526

This process continues listening for remote modules until you click the **Stop** button.

> You can also stop the discovery process by clicking the **Cancel** or **Add selected devices** buttons at any time.

Click the **Add selected devices** button to add the remote devices that have been found. They are added in a sub-list under the local module.



Name: C3PO
Function: ZigBee Router API
Port: COM1 - 9600/8/N/1/N - API
MAC: 0013A2004031A8E1

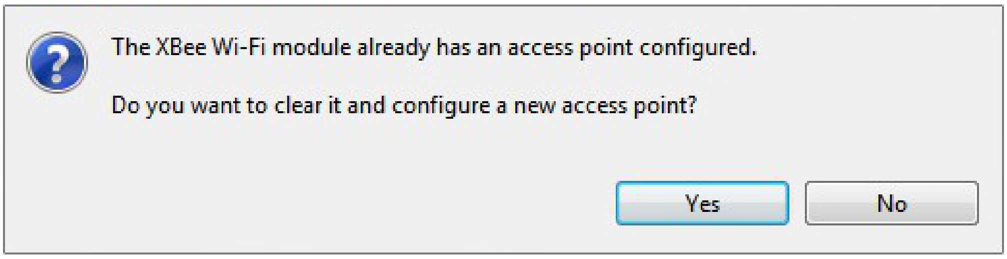
2 remote modules

Name: R2D2
Function: ZigBee Router API
MAC: 0013A2004031A8D6

Name: LUKE SKYWALKER
Function: ZigBee Coordinator API
MAC: 0013A2004066C526

If the mode and protocol are supported, you can select remote radio modules from the sub-list and perform the actions in the active working mode.

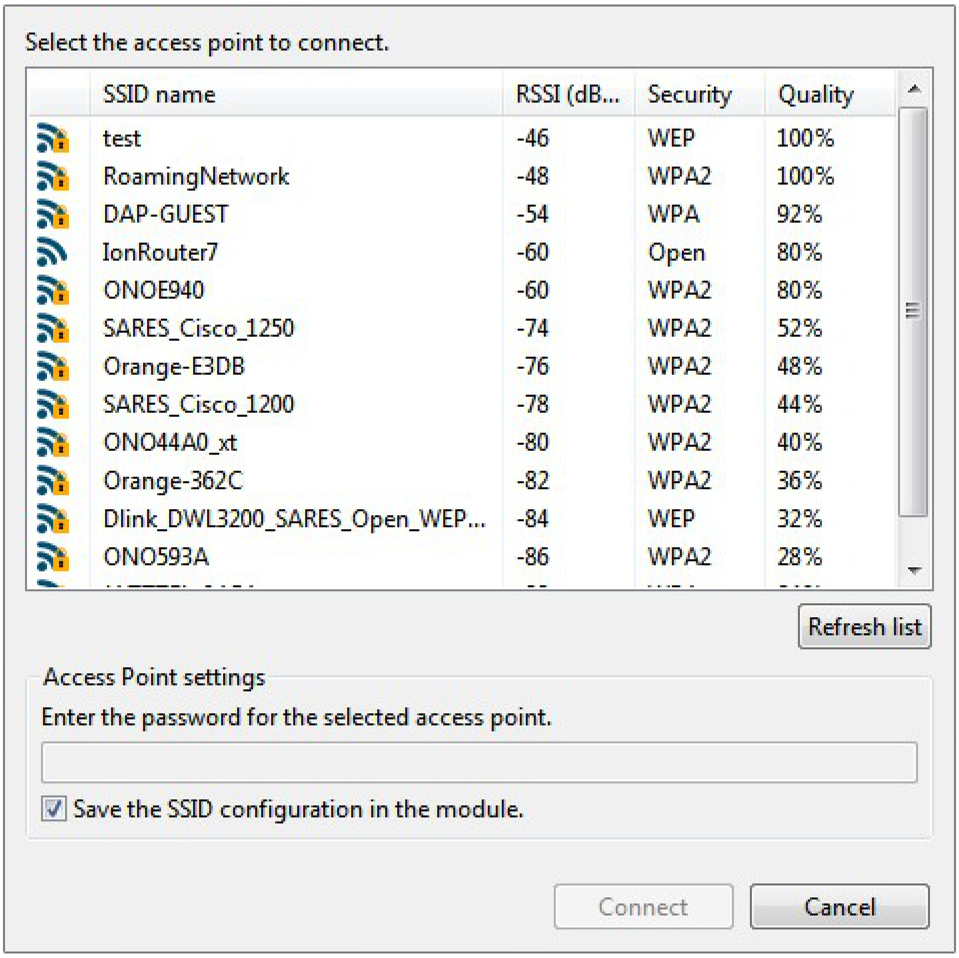
- **Wi-Fi protocol**: If the radio module's protocol is Wi-Fi, this button discovers and configures the Access Point for the module. When you click the button, XCTU reads the SSID configuration of the Wi-Fi module. If the module already has an SSID configured, you need to clear the configuration and perform a new SSID discovery.



The XBee Wi-Fi module already has an access point configured.

Do you want to clear it and configure a new access point?

If the SSID configuration is empty, a new dialog displays the nearby SSIDs. The information displayed depends on the Wi-Fi module version. For S6 Wi-Fi modules, the table displays the following fields:



Select the access point to connect.

SSID name	RSSI (dB...	Security	Quality
test	-46	WEP	100%
RoamingNetwork	-48	WPA2	100%
DAP-GUEST	-54	WPA	92%
IonRouter7	-60	Open	80%
ONOE940	-60	WPA2	80%
SARES_Cisco_1250	-74	WPA2	52%
Orange-E3DB	-76	WPA2	48%
SARES_Cisco_1200	-78	WPA2	44%
ONO44A0_xt	-80	WPA2	40%
Orange-362C	-82	WPA2	36%
Dlink_DWL3200_SARES_Open_WEP...	-84	WEP	32%
ONO593A	-86	WPA2	28%

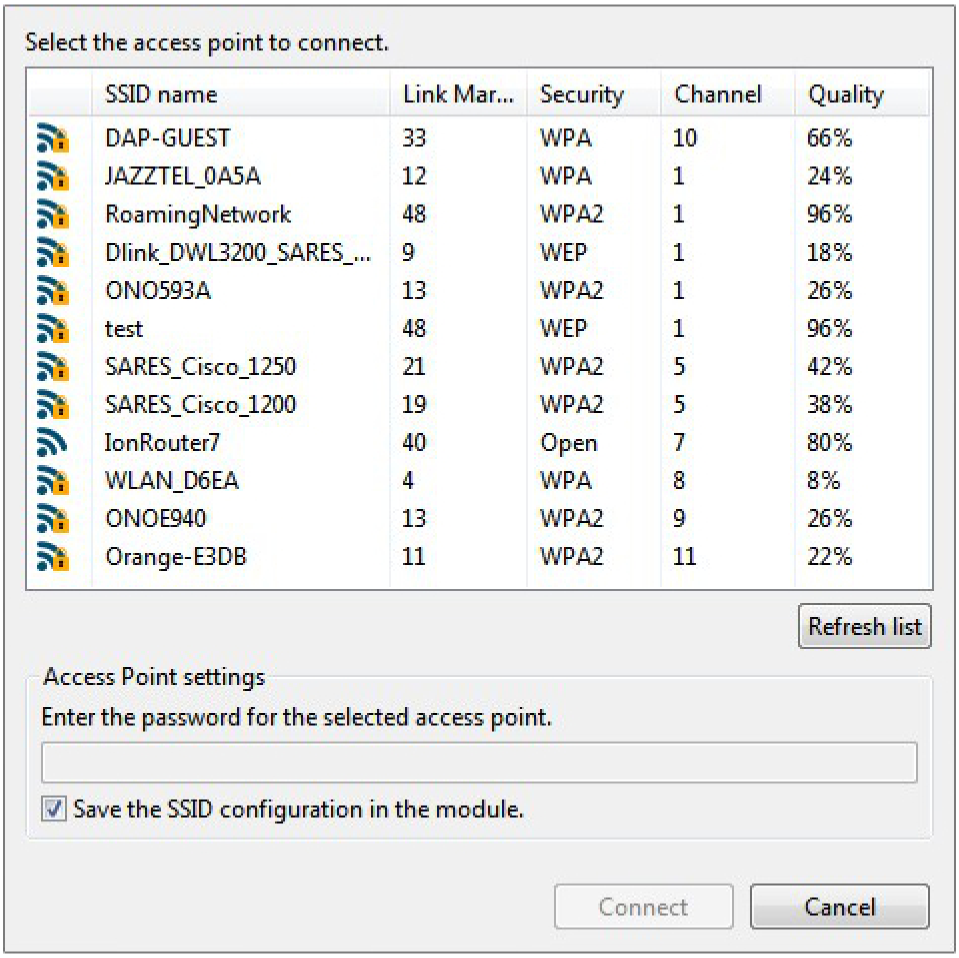
Access Point settings

Enter the password for the selected access point.

☑ Save the SSID configuration in the module.

- **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.

If the Wi-Fi module is S6B, the information displayed is as follows:



Select the access point to connect.

SSID name	Link Mar...	Security	Channel	Quality
DAP-GUEST	33	WPA	10	66%
JAZZTEL_0A5A	12	WPA	1	24%
RoamingNetwork	48	WPA2	1	96%
Dlink_DWL3200_SARES_...	9	WEP	1	18%
ONO593A	13	WPA2	1	26%
test	48	WEP	1	96%
SARES_Cisco_1250	21	WPA2	5	42%
SARES_Cisco_1200	19	WPA2	5	38%
IonRouter7	40	Open	7	80%
WLAN_D6EA	4	WPA	8	8%
ONOE940	13	WPA2	9	26%
Orange-E3DB	11	WPA2	11	22%

Access Point settings

Enter the password for the selected access point.

☑ Save the SSID configuration in the module.

- **SSID name**: Name of the access point.
- **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.

Regardless of the information displayed in the table, you can 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 checkbox 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.

When you click **Connect** the Wi-Fi module attempts to connect to that Access Point. After that, if the Wi-Fi module is selected and the Configuration mode is the active working mode, the settings of the radio module are refreshed.

- **Other protocols**: There is no specific functionality for this button in other protocols.

This button only appears in local radio modules. Remote radio modules do not have this functionality. See the Local and remote radio modules topic for more information.

Expand/collapse button

If the protocol of the radio module is **ZigBee** or **DigiMesh** and you have found remote modules in the same network, the expand/collapse button is enabled, allowing you to expand or collapse the list of remote modules that drops down from the local device.

Depending on whether the list is expanded or collapsed, the icon and description of this button change to represent the action associated with the button.

This button only appears in local radio modules. Remote radio modules do not have this functionality. See the Local and remote radio modules topic for more information.

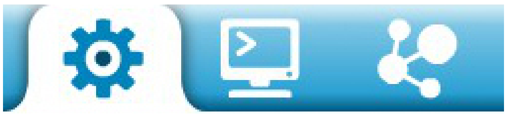
Remove button

The remove button removes the radio module from the list of devices. Removing a radio module from the list also removes the associated communication console and network view of the module.

If you click the remove button of a remote module, you are only removing that radio module from the sub-list of remote modules. However, if you click the remove button of a local module and that module has a sub-list of remote modules, the local module and all of its remote modules are removed from the list of devices.

Configure your modules

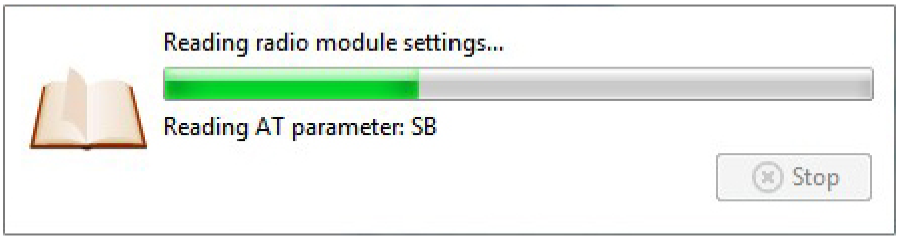
Once you have added a radio module to the list of devices, you can configure it using the Configuration working mode in the toolbar.



> This working mode is selected by default when you open the tool.

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 all the fields.

To configure a radio module, select it from the list of devices.

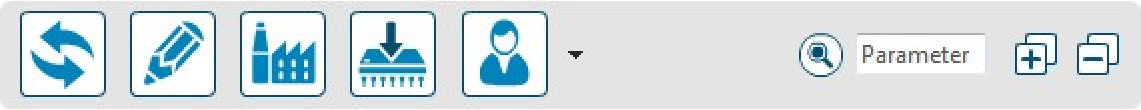


Reading radio module settings...

Reading AT parameter: SB

Configuration toolbar

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



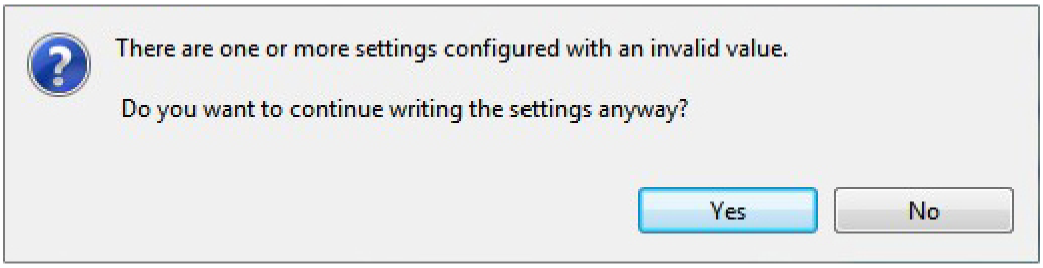
Reading module settings

	Click the **Read module settings** button to refresh the selected radio module's firmware settings.

Writing module settings

	If you have changed the value of any firmware setting, clicking the **Write module settings** button writes the new values to the radio module.

If a setting is configured with an invalid value, a dialog asks if you want to continue with the process; if you select **Yes**, it attempts to write all the values, including the invalid ones.



There are one or more settings configured with an invalid value.

Do you want to continue writing the settings anyway?

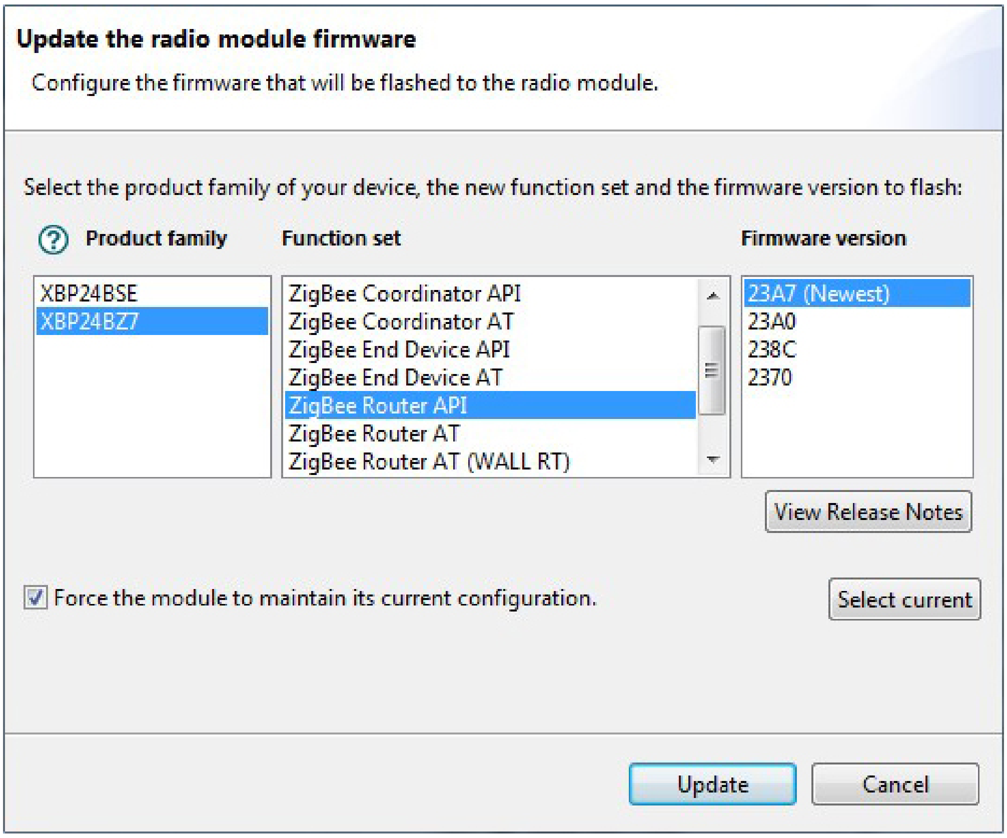
Loading default firmware settings

	The **Load default firmware settings** button allows you to load the default values established by the firmware.

Notice that this action does not write anything in the radio module. If you want to apply these default values, you need to click the **Write module settings** button after clicking Load default firmware settings.

Updating firmware

	This action allows you to update the firmware of the selected radio module. When you click the **Update firmware** button, the update firmware dialog box appears, displaying the available and compatible firmware for the module.



Update the radio module firmware

Configure the firmware that will be flashed to the radio module.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
XBP24BSE	ZigBee Coordinator API	23A7 (Newest)
XBP24BZ7	ZigBee Coordinator AT	23A0
	ZigBee End Device API	238C
	ZigBee End Device AT	2370
	ZigBee Router API	
	ZigBee Router AT	
	ZigBee Router AT (WALL RT)	

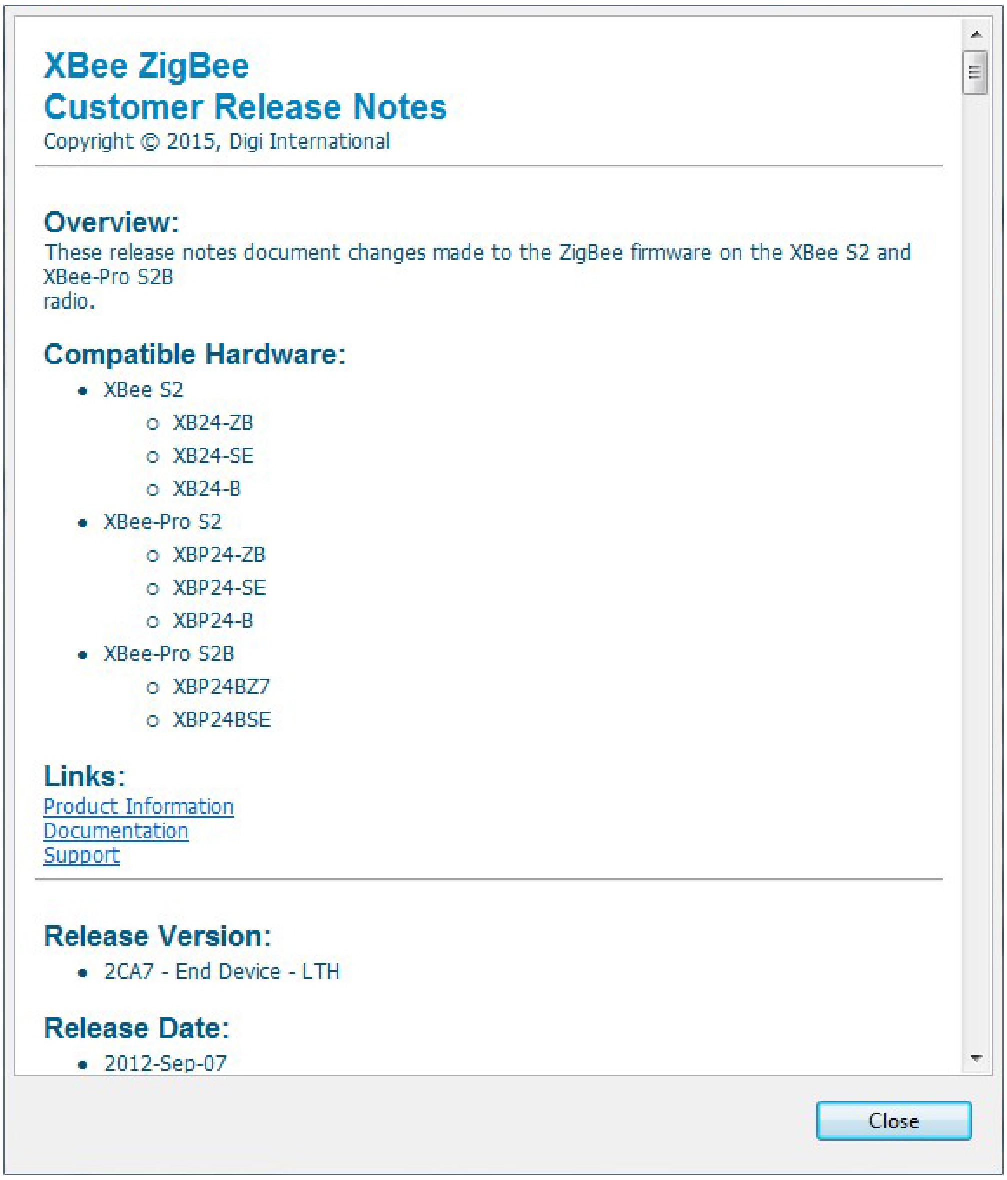
☑ Force the module to maintain its current configuration.

Firmware is defined by family, function and version; to select your firmware, choose the **firmware family**, the **firmware function** and the **firmware version**.

The button **Select current** selects the firmware family, function and version of the firmware that is currently installed in the radio module.

View Release Notes	If the selected firmware has any Release Notes available, the **View Release Notes** button below the firmware list is enabled.

To view the release notes for the selected firmware, click the View Release Notes button. A dialog box displays the release notes:



XBee ZigBee
Customer Release Notes
Copyright © 2015, Digi International

Overview:
These release notes document changes made to the ZigBee firmware on the XBee S2 and XBee-Pro S2B
radio.

Compatible Hardware:

- XBee S2
  - XB24-ZB
  - XB24-SE
  - XB24-B
- XBee-Pro S2
  - XBP24-ZB
  - XBP24-SE
  - XBP24-B
- XBee-Pro S2B
  - XBP24BZ7
  - XBP24BSE

Links:
Product Information
Documentation
Support

Release Version:

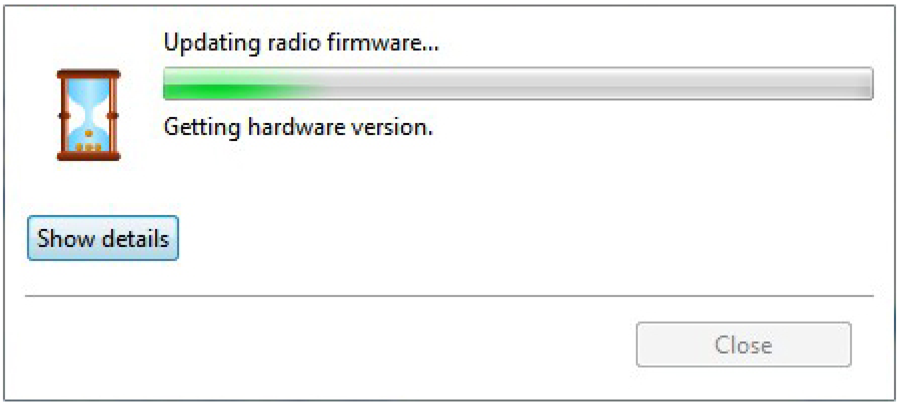
- 2CA7 - End Device - LTH

Release Date:

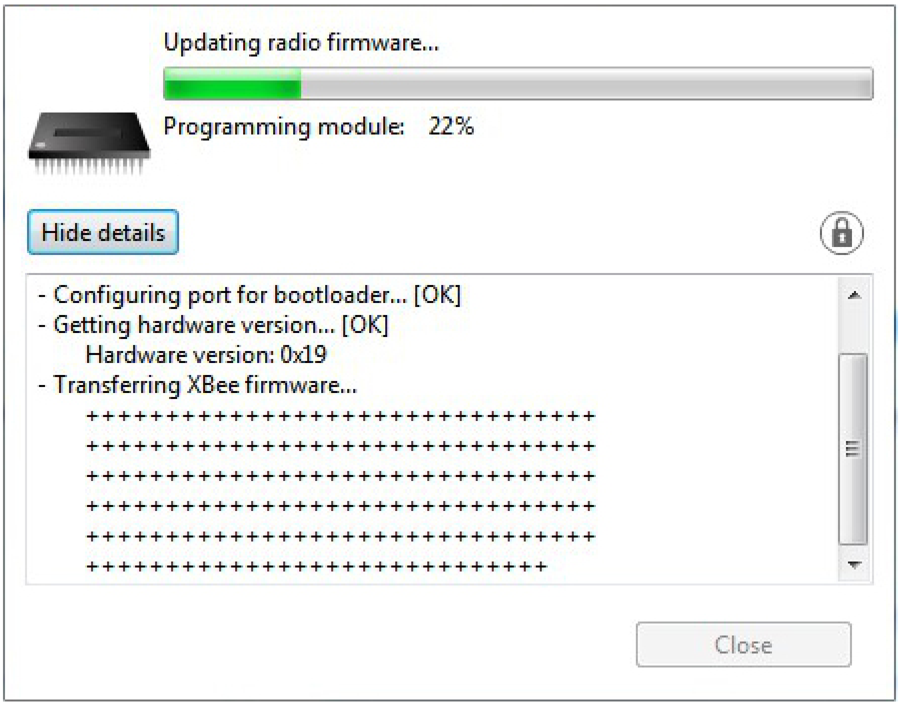
- 2012-Sep-07

The dialog box also contains a setting to maintain the current configuration in the new firmware. When the **Force the module to maintain its current configuration** setting is checked, XCTU attempts to reconfigure the module with the current setting values once the new firmware has been flashed.

Once you have chosen the firmware to be flashed, click the **Update** button to start the process. The update firmware progress dialog box displays the update progress.



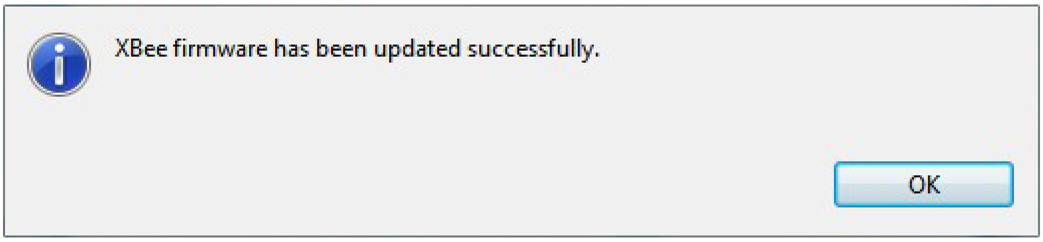
Clicking the **Show details button** displays the update firmware progress details.



```
- Configuring port for bootloader... [OK]
- Getting hardware version... [OK]
     Hardware version: 0x19
- Transferring XBee firmware...
     ++++++++++++++++++++++++++++++++
     ++++++++++++++++++++++++++++++++
     ++++++++++++++++++++++++++++++++
     ++++++++++++++++++++++++++++++++
     ++++++++++++++++++++++++++++++++
     +++++++++++++++++++++++++++++
```

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 was checked, XCTU writes the old configuration to the module and then reads the setting's values.

If there is an error in the process, an error message appears and is added to the details log. Otherwise, when the firmware update process finishes, a message indicates the firmware was updated successfully.



XBee firmware has been updated successfully.

You can update the firmware of a remote radio module the same way as for a local module. To perform a remote firmware update, you must configure the local radio module in API operation mode.

Remote firmware update functionality is limited to the following radio modules:

- XBee/XBee-PRO ZB

- Programmable XBee-PRO ZB
- XBee/XBee-PRO ZB SMT
- Programmable XBee-PRO ZB SMT
- XBee-PRO 900HP
- Programmable XBee-PRO 900HP
- XBee 865LP
- Programmable XBee 865LP
- XBee 868LP
- Programmable XBee 868LP
- XLR PRO Radio Solution

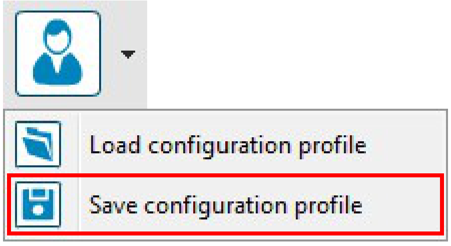
Configuration profiles

A configuration profile contains the information and settings values of a radio firmware. XCTU lets you save and write configuration profiles to the radio module. This feature is useful in a production environment when the same parameters need to be set on multiple radios.

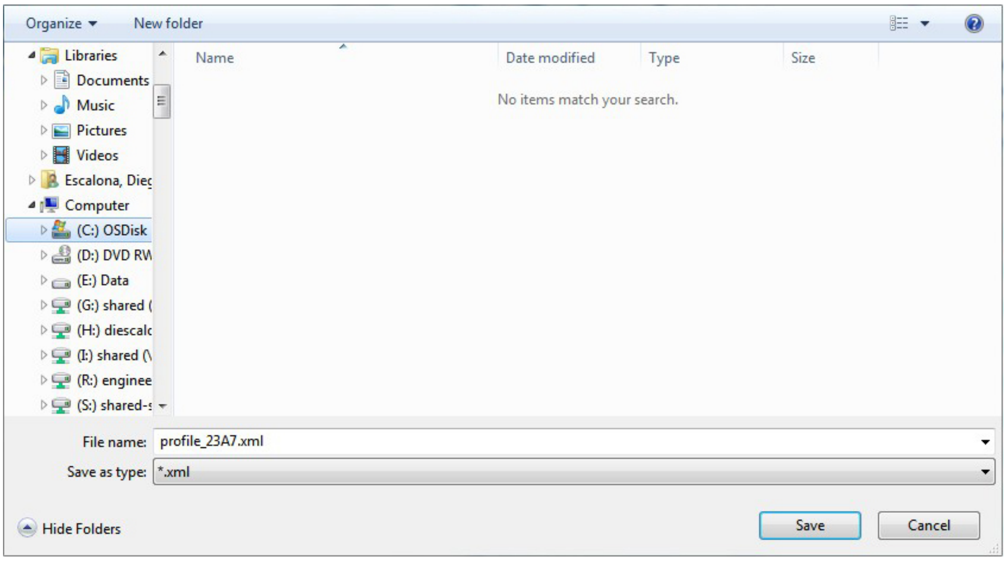
Saving a configuration profile

The first step to save a configuration profile is to configure all the settings of the module with your desired values. It is not necessary to write these settings to the module; you only need to change their values in the corresponding controls.

When finished, go to the drop-down menu and select **Save configuration profile**.



A new **Save file** dialog box requests the name and destination of the profile file. Choose a name and path and click **Save** to save your configuration profile.



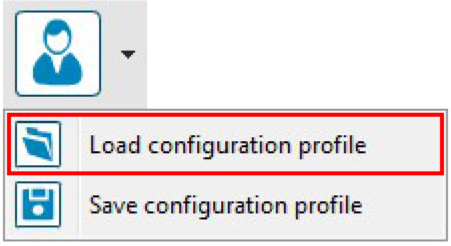
File name: profile_23A7.xml

Save as type: *.xml

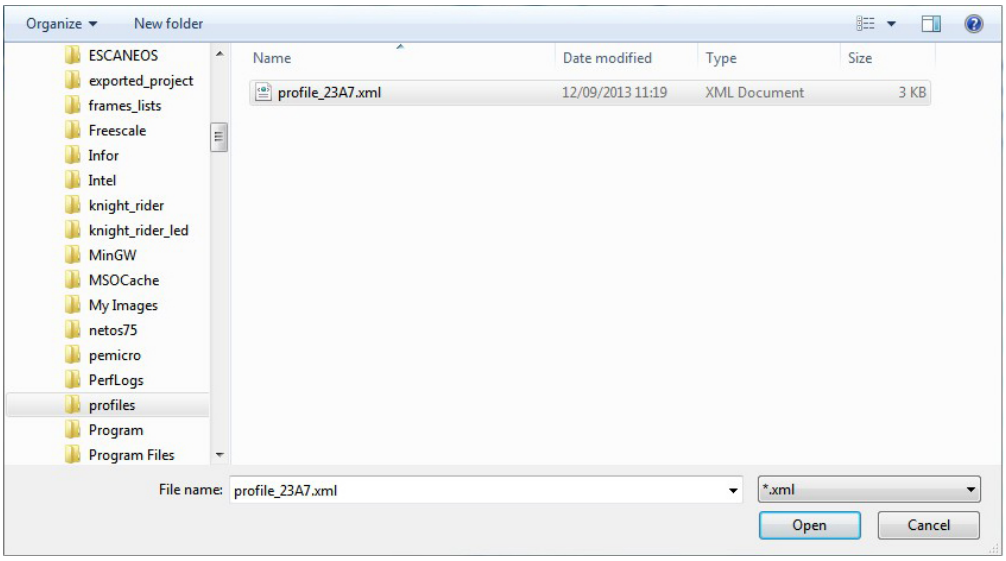
Loading a configuration profile

To load a configuration profile, go to the **Configuration profiles** drop-down menu and select **Load**

configuration profile.

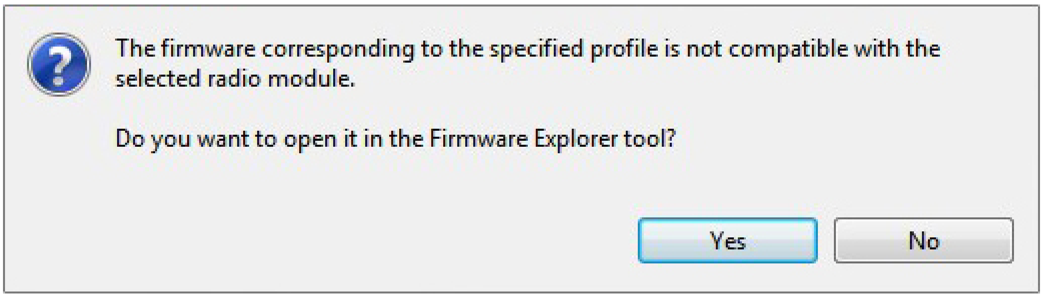


A new **Open file** dialog asks for the configuration profile file to load. Look for a previously saved configuration profile and click **Open**.



XCTU attempts to load the selected profile.

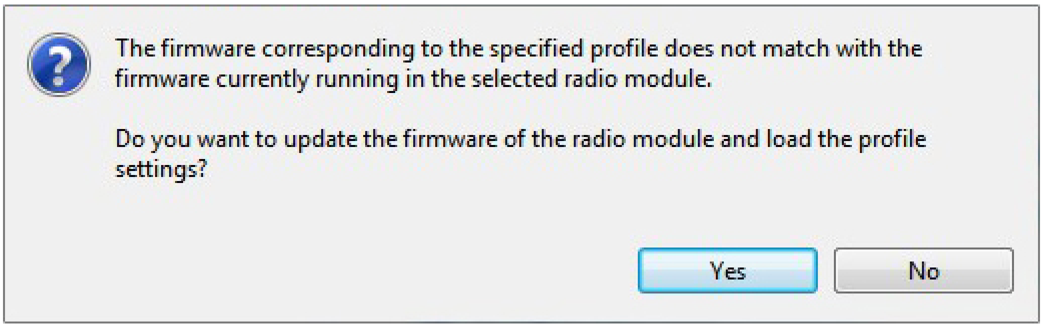
If the firmware of the profile you are loading is not compatible with the radio module, XCTU asks you to open it in the Firmware Explorer tool. See the Firmware Explorer tool topic for more information about this tool.



The firmware corresponding to the specified profile is not compatible with the selected radio module.

Do you want to open it in the Firmware Explorer tool?

If the firmware of the profile you are loading does not match the firmware running in the radio module, but it is compatible with the module, XCTU asks you to update the firmware of the module, so the profile can be loaded correctly.



The firmware corresponding to the specified profile does not match with the firmware currently running in the selected radio module.

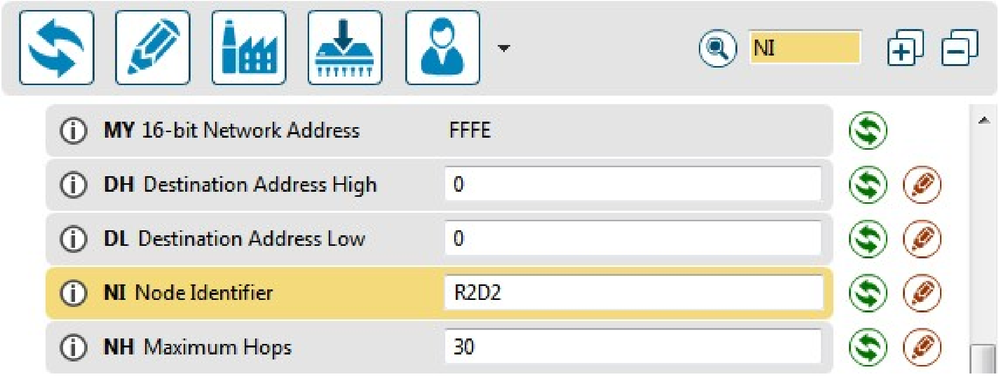
Do you want to update the firmware of the radio module and load the profile settings?

Note that if the firmware of the profile you are loading and the firmware running in the radio module match, the settings saved in the profile are loaded but not written to the radio module.

To save the profile settings in the module, click the **Write module settings** button of the configuration toolbar.

Searching a 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.



NI

MY 16-bit Network Address FFFE

DH Destination Address High 0

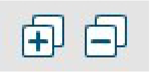
DL Destination Address Low 0

NI Node Identifier R2D2

NH Maximum Hops 30

Expanding and collapsing sections

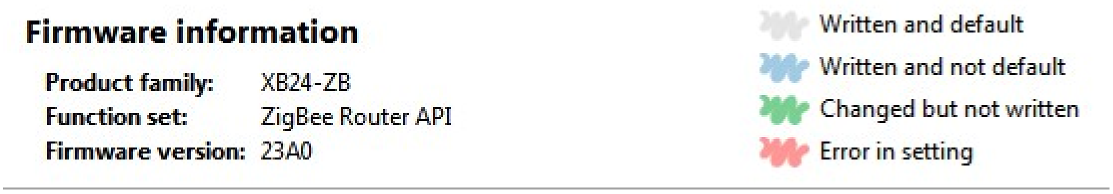
The controls on the far right side of the configuration toolbar correspond to the Expand and Collapse sections.



Click the **Expand sections** button to expand all the settings that were collapsed. Click the **Collapse sections** button to collapse all expanded settings.

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.



Firmware information

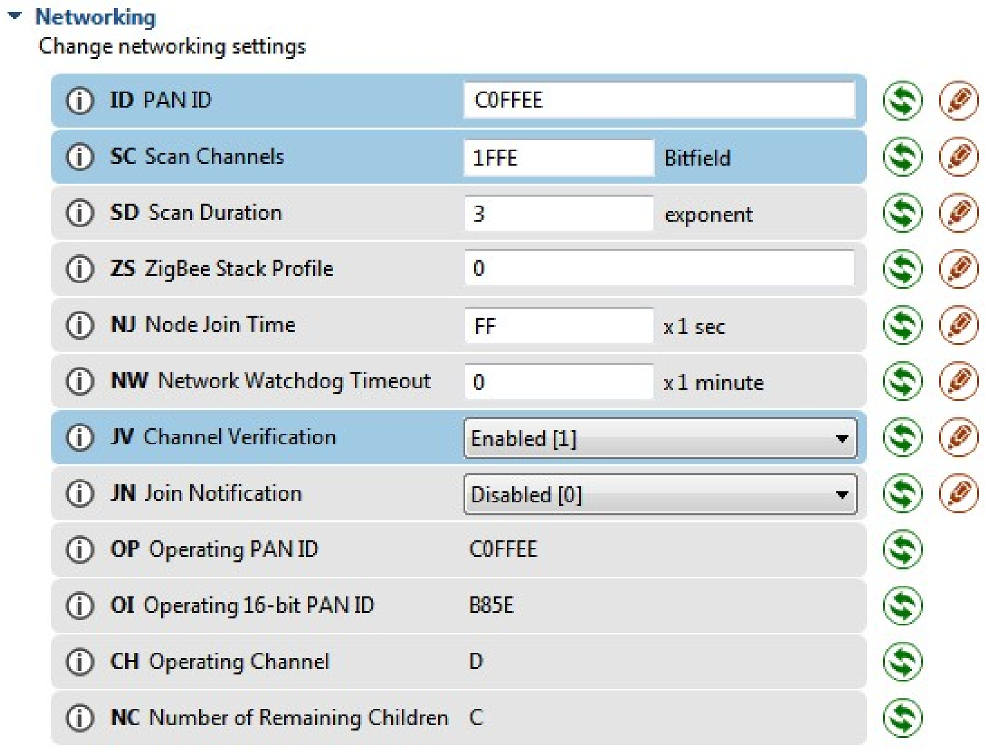
Product family: XB24-ZB
Function set: ZigBee Router API
Firmware version: 23A0

⬜ Written and default
🟦 Written and not default
🟩 Changed but not written
🟥 Error in setting

Next to the firmware information panel you can find the settings color legend, which indicates the status of a setting depending on its background color.

Firmware settings

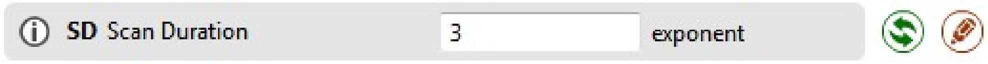
The firmware settings of the radio module are located below the firmware information panel. They are divided into sections or categories with a short description in each one.



Networking
Change networking settings

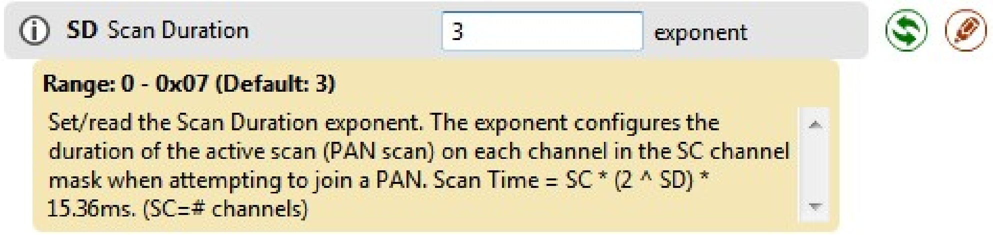
ID PAN ID	C0FFEE	
SC Scan Channels	1FFE	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x 1 sec
NW Network Watchdog Timeout	0	x 1 minute
JV Channel Verification	Enabled [1]	
JN Join Notification	Disabled [0]	
OP Operating PAN ID	C0FFEE	
OI Operating 16-bit PAN ID	B85E	
CH Operating Channel	D	
NC Number of Remaining Children	C	

The setting control contains several parts:



SD Scan Duration 3 exponent

- **Information button**: If you click this icon, a new panel appears just below the setting control displaying a short description of the setting, including the default value and the valid range, if the setting is numeric.

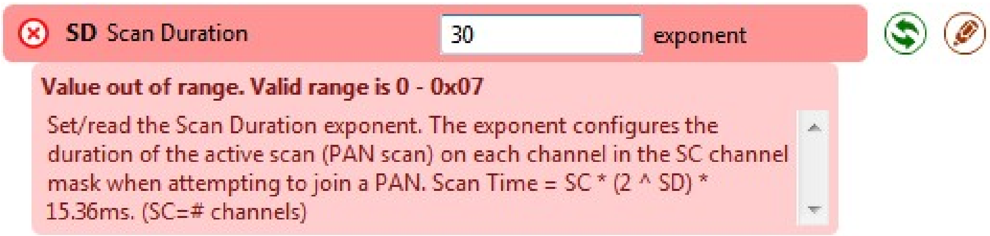


SD Scan Duration 3 exponent

Range: 0 - 0x07 (Default: 3)

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)

Additionally, if the setting has an invalid value, the information icon and background of the description panel change and the cause of the error is displayed.



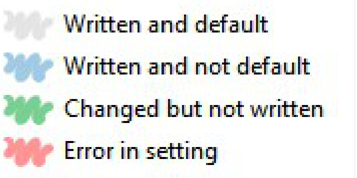
SD Scan Duration 30 exponent

Value out of range. Valid range is 0 - 0x07

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)

- **AT parameter**: Next to the information icon, the associated AT parameter of the setting is displayed in bold type. Some settings, such as the actions, may not have an associated AT parameter.
- **Setting name**: The next field, the setting name, is descriptive text describing how the setting should be configured.
- **Setting configuration control**: Depending on the kind of setting, this control changes (for example, text box or combo box), but in all cases this is the control where the setting value must be entered or configured.
- **Units label**: Some settings include a units label after the configuration control.
- **Refresh and write buttons**: The left side of the setting control is made up of two buttons that allow you to individually read or write the value of the setting. Some settings, such as the read-only settings, do not have a write button.

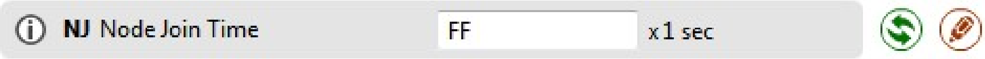
The background color of a setting changes depending on the status of its value. The color legend is displayed next to the firmware information panel and reads the following:



Written and default
Written and not default
Changed but not written
Error in setting

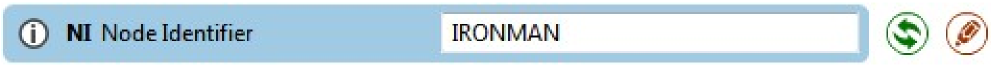
- **Gray**: The value of the setting is written in the radio module and matches the default value.
- **Blue**: The value of the setting is written in the radio module but is different from the default value.
- **Green**: The value of the setting has changed but it has not been written in the radio module yet.
- **Red**: The value of the setting is not valid.
- **Yellow**: Indicates that the setting is highlighted because it has been found using the Search parameter control of the configuration toolbar.

- **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:



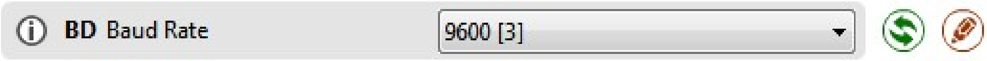
NJ Node Join Time FF x 1 sec

- **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.



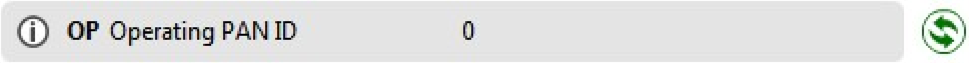
NI Node Identifier IRONMAN

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

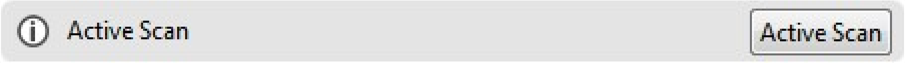


BD Baud Rate 9600 [3]

- **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.



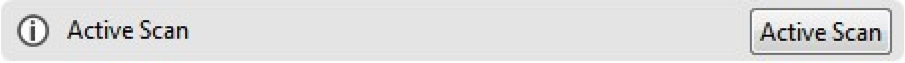
Special functions

There are some settings that 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**.

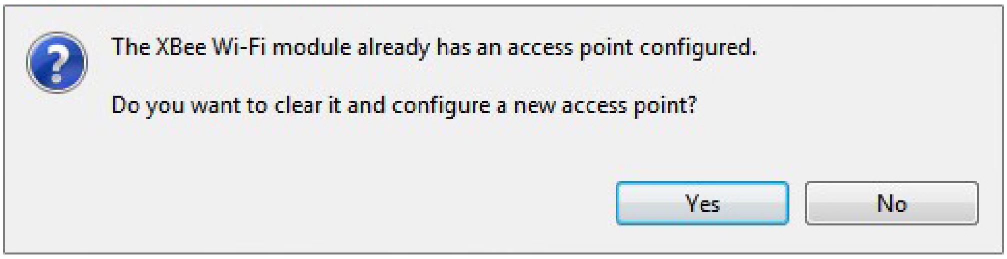
At this time, XCTU has only one special function: the **Active scan**.

Active Scan

The **Active Scan** special function is executed by an action setting of the Wi-Fi XBee radio modules. This function discovers and configures the Access Point for the XBee Wi-Fi module.

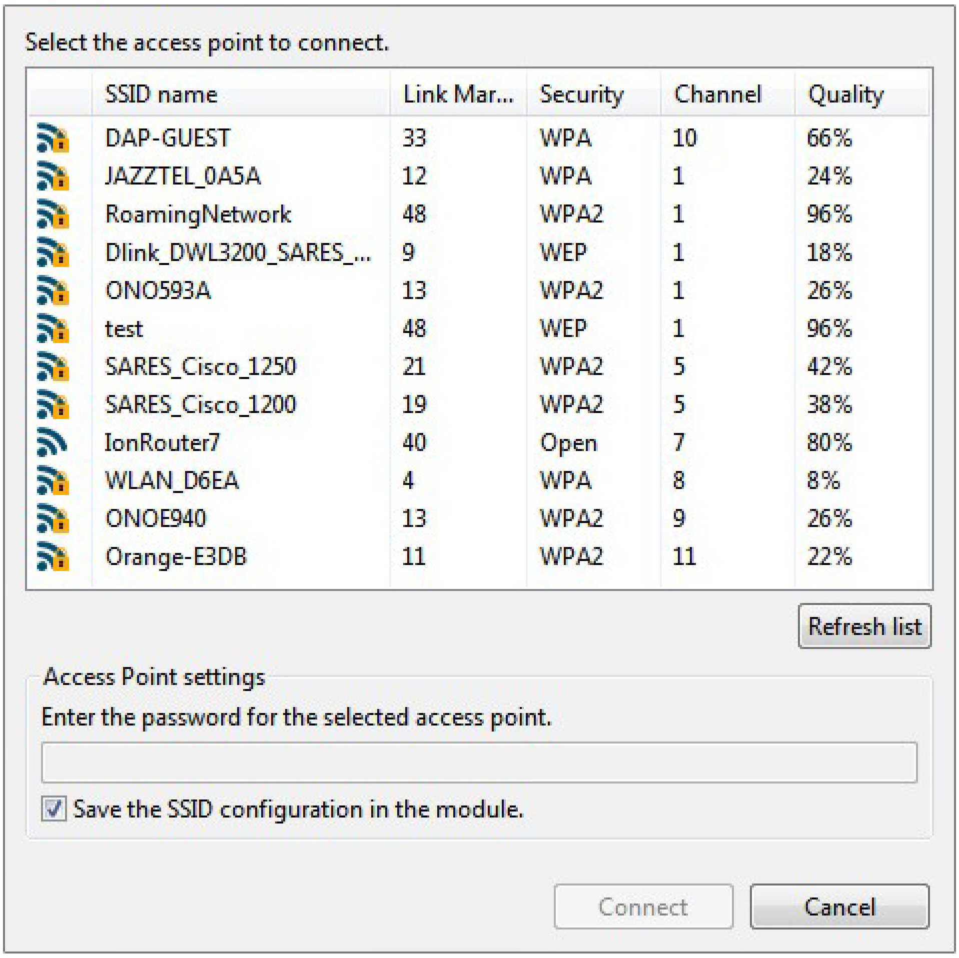


When you click the **Active scan** button, XCTU reads the SSID configuration of the Wi-Fi module. If the module has an SSID already configured, you need to clear the configuration and perform a new SSID discovery.


The XBee Wi-Fi module already has an access point configured.

Do you want to clear it and configure a new access point?

If the SSID configuration is empty, the nearby SSIDs are displayed in a new dialog.



Select the access point to connect.

SSID name	Link Mar...	Security	Channel	Quality
DAP-GUEST	33	WPA	10	66%
JAZZTEL_0A5A	12	WPA	1	24%
RoamingNetwork	48	WPA2	1	96%
Dlink_DWL3200_SARES_...	9	WEP	1	18%
ONO593A	13	WPA2	1	26%
test	48	WEP	1	96%
SARES_Cisco_1250	21	WPA2	5	42%
SARES_Cisco_1200	19	WPA2	5	38%
IonRouter7	40	Open	7	80%
WLAN_D6EA	4	WPA	8	8%
ONOE940	13	WPA2	9	26%
Orange-E3DB	11	WPA2	11	22%

Access Point settings

Enter the password for the selected access point.

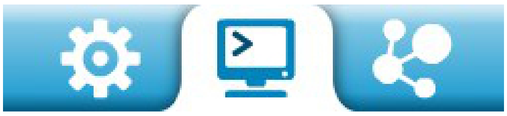
☑ Save the SSID configuration in the module.

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 checkbox 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.

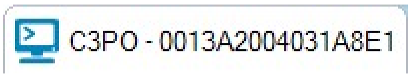
Click **Connect** to connect the Wi-Fi module to that Access Point and refresh the settings of the radio module.

Talk with your modules

The Consoles working mode of XCTU allows you to communicate with the radio modules added in the devices list. Select the Consoles working mode in the toolbar to begin.



XCTU loads a list of consoles in the working area, one for each module of the devices list, sorted in a tab control. 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.

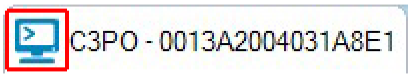

C3PO - 0013A2004031A8E1

Another way to identify the console of each module is by selecting the module. Its associated console activates and moves to the front.

> Selecting a console in the working area also selects the associated module of the devices list.

Console status

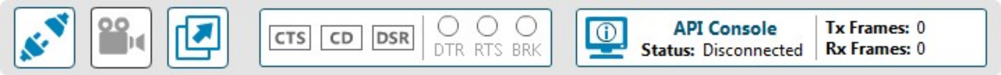
The console tab icon displays the following states:


C3PO - 0013A2004031A8E1

🖥	The console is disconnected
🖥	The console is connected
🖥	The console is receiving data
🖥	The console is sending data

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.



API Console
Status: Disconnected
Tx Frames: 0
Rx Frames: 0

The console toolbar provides the following functionality:

- Connecting and disconnecting the console
- Recording the console session

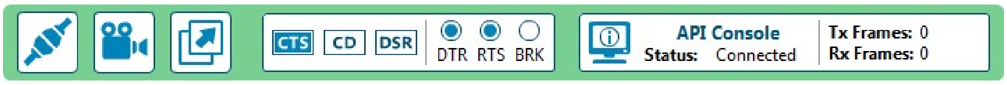
- Attaching and detaching the console
- Line status indicators control
- Console overview control

Connecting and disconnecting the console

The first time you open a console, it is disconnected by default. This is indicated by the the gray background of the toolbar and the status text reading "Disconnected".

Click the **Connect** button to establish communication with the radio module corresponding to the console.

The background color of the toolbar changes to green and the status text changes to "Connected." When the console is connected, all the data traffic of the radio module is captured by the console and displayed in the corresponding controls.



Clicking the **Disconnect** button disconnects the console from the module.

Recording the console session

The button with the video icon allows you to record all the upcoming and outgoing console data into a log file as it is sent or received. The console must be connected for you to be able to start or stop recording.

When you click the **Start Recording** button, a save file dialog asks for the destination file for the sent and received data.

Data is periodically written to a log file using the **Comma Separated Values** format (CSV), so that the file can be open and interpreted later by other consumer, business or scientific applications.

A console log file consists of a first record with the console session information and multiple data records with the data sent and received. The console session record has the following fields:

- **Date**: The date and time when the session record started.
- **ID**: Not used in this record; this field contains a "-" character.
- **Type**: One of **API**, **API2**, **AT** or **Serial**.
- **Description**: Other comma separated console session information such as the module Node Identifier, MAC address, function set, firmware version and COM port information.

Example

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

The data records have the following fields:

- **Date**: The date and time when the data was sent or received.
- **ID**: ID number of the packet or API frame.
- **Type**: One of **SENT** or **RECV**.
- **Data**: Data sent or received in Hexadecimal string format.

Example

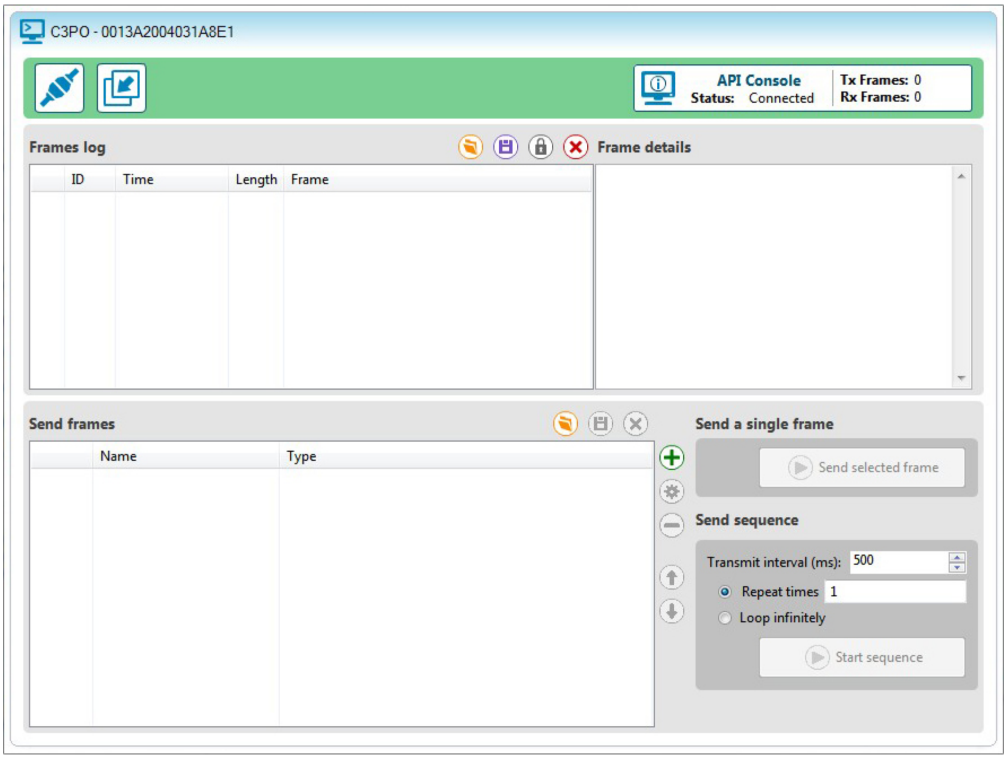
```
02-05-2015 11:44:58.857,0,SENT,7E000408764944F4
```

Clicking the **Stop Recording** button stops the process of saving any sent or received data.

Attaching and detaching the console

The button with the arrow icon allows you to attach or detach the console from the working area. By default, all the consoles are attached and sorted in tabs.

If you click the **Detach** button, the console is detached from the working area and displayed in a new floating dialog box.



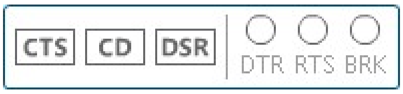
Status: Connected
Tx Frames: 0
Rx Frames: 0
Transmit interval (ms): 500
● Repeat times 1
○ Loop infinitely

It is possible to detach all the consoles to display them in multiple dialogs. This could be useful if you need to see the traffic of different radio modules simultaneously.

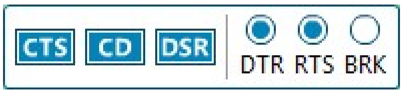
To reattach the console view, click the **Attach** button of the toolbar or close the floating dialog box containing that console.

Line status indicators control

The line status indicators control, located in the middle of the toolbar, displays the status of the RS-232 hardware flow control lines. Blue status inficates that the line is asserted, while white indicates that it is de-asserted. Status of the lines is not displayed until you open the connection of the console; meanwhile the control is disabled.



Connection closed



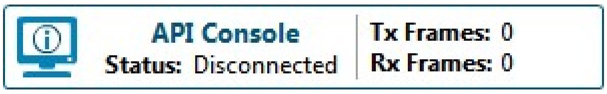
Connection open

You can view and manage the following lines from this control:

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

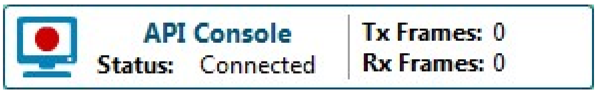
Console overview control

The console status panel. on the right side of the toolbar, displays the console type (AT or API), its status (Connected or Disconnected) and the number of sent and received frames or bytes.



API Console
Status: Disconnected
Tx Frames: 0
Rx Frames: 0

The console overview control icon changes to indicate when the console session is being recorded.



API Console
Status: Connected
Tx Frames: 0
Rx Frames: 0

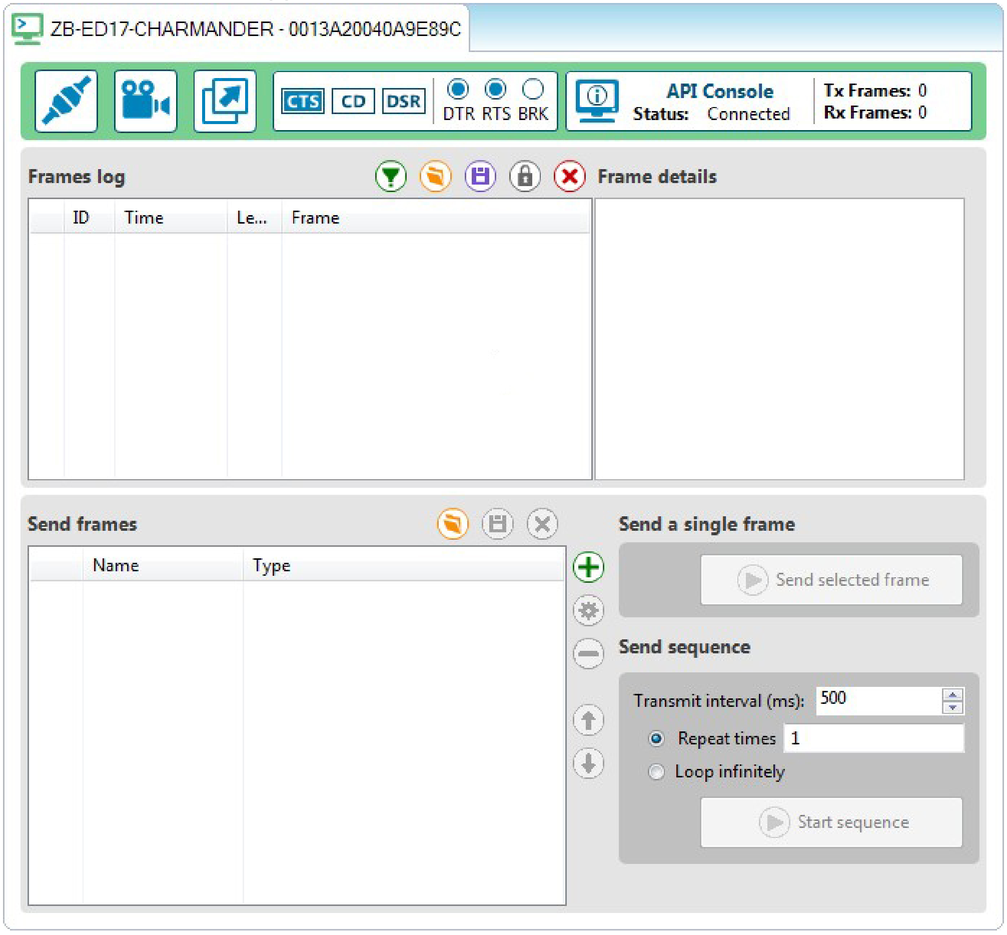
Console types

The console type depends on the operating mode of the radio module. (See topic Radio module operating modes for more information about the operating modes.) There are two kinds of consoles:

- API Console
- AT Console

API Console

The API Console communicates with radio modules running in API or API escaped operating modes. If XCTU is in Consoles mode, and you select a radio module configured in API or API escaped operating modes, this console appears:



ZB-ED17-CHARMANDER - 0013A20040A9E89C

Status: Connected

Tx Frames: 0
Rx Frames: 0

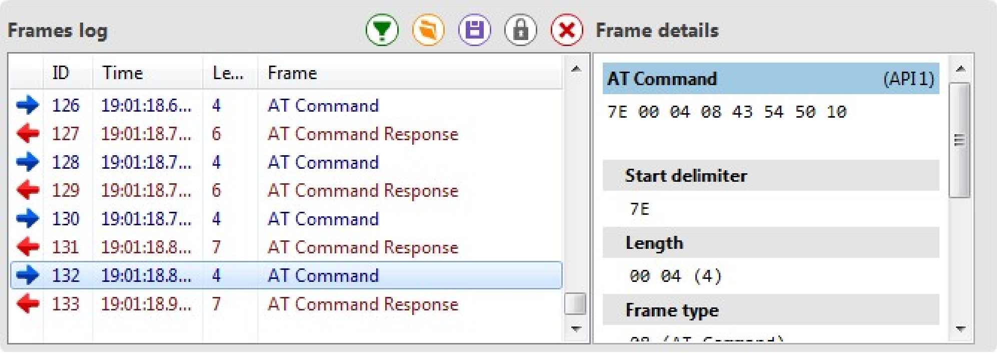
Transmit interval (ms): 500

● Repeat times 1

○ Loop infinitely

Communication with radio modules running in API mode uses API frames. An API frame is an array of bytes with a specific structure defined by the API Frame Specifications. For more information about the API frames see the API frames topic.

API frames traffic monitoring section



	ID	Time	Le...	Frame
→	126	19:01:18.6...	4	AT Command
←	127	19:01:18.7...	6	AT Command Response
→	128	19:01:18.7...	4	AT Command
←	129	19:01:18.7...	6	AT Command Response
→	130	19:01:18.7...	4	AT Command
←	131	19:01:18.8...	7	AT Command Response
→	132	19:01:18.8...	4	AT Command
←	133	19:01:18.9...	7	AT Command Response

AT Command (API 1)

```
7E 00 04 08 43 54 50 10
```

Start delimiter

7E

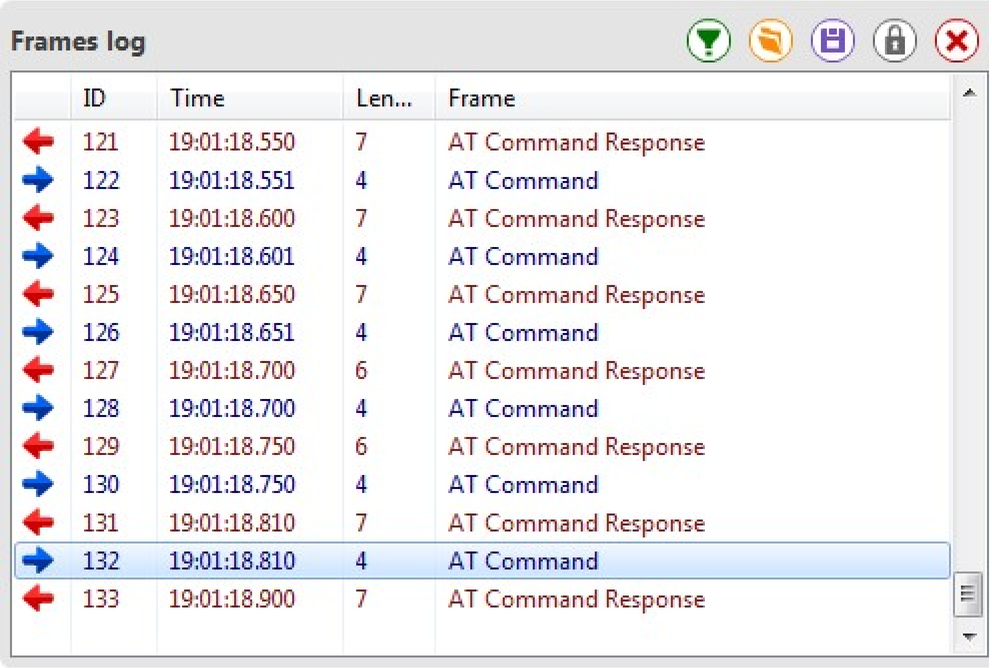
Length

00 04 (4)

Frame type

08 (AT Command)

The main control of the API frames traffic monitoring section is the API frames table. The table displays the following properties of the transmitted and received API frames:



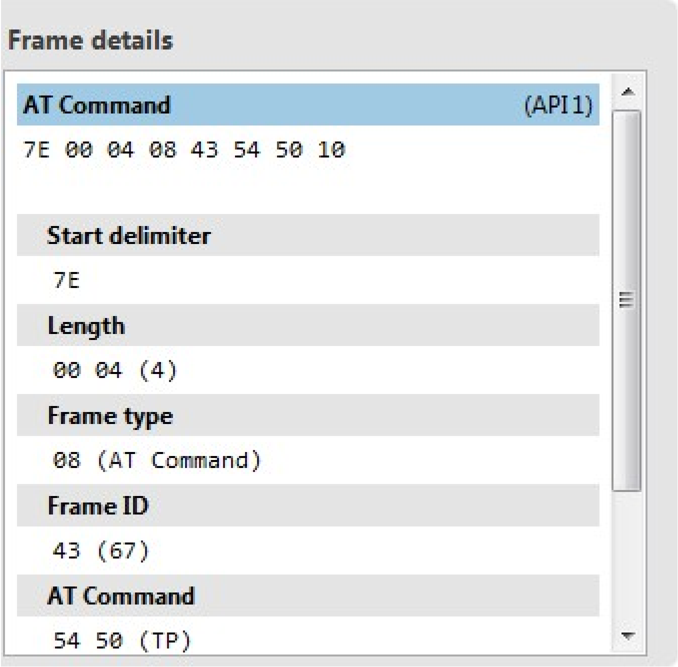
	ID	Time	Len...	Frame
←	121	19:01:18.550	7	AT Command Response
→	122	19:01:18.551	4	AT Command
←	123	19:01:18.600	7	AT Command Response
→	124	19:01:18.601	4	AT Command
←	125	19:01:18.650	7	AT Command Response
→	126	19:01:18.651	4	AT Command
←	127	19:01:18.700	6	AT Command Response
→	128	19:01:18.700	4	AT Command
←	129	19:01:18.750	6	AT Command Response
→	130	19:01:18.750	4	AT Command
←	131	19:01:18.810	7	AT Command Response
→	132	19:01:18.810	4	AT Command
←	133	19:01:18.900	7	AT Command Response

- **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 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.

When API frames are sent or received by the module, they are added to the API frames table. Depending on whether the frame is sent or received, the color of the frame fields becomes blue or red, respectively.

The frame details control appears next to the API frames table. This box displays the decoded contents of the selected API frame.

If you select an API frame from the sent and received frames table, the frame contents appear in detail in fields with their corresponding values. The number of fields depends on the API frame type.



AT Command (API 1)

7E 00 04 08 43 54 50 10

Start delimiter

7E

Length

00 04 (4)

Frame type

08 (AT Command)

Frame ID

43 (67)

AT Command

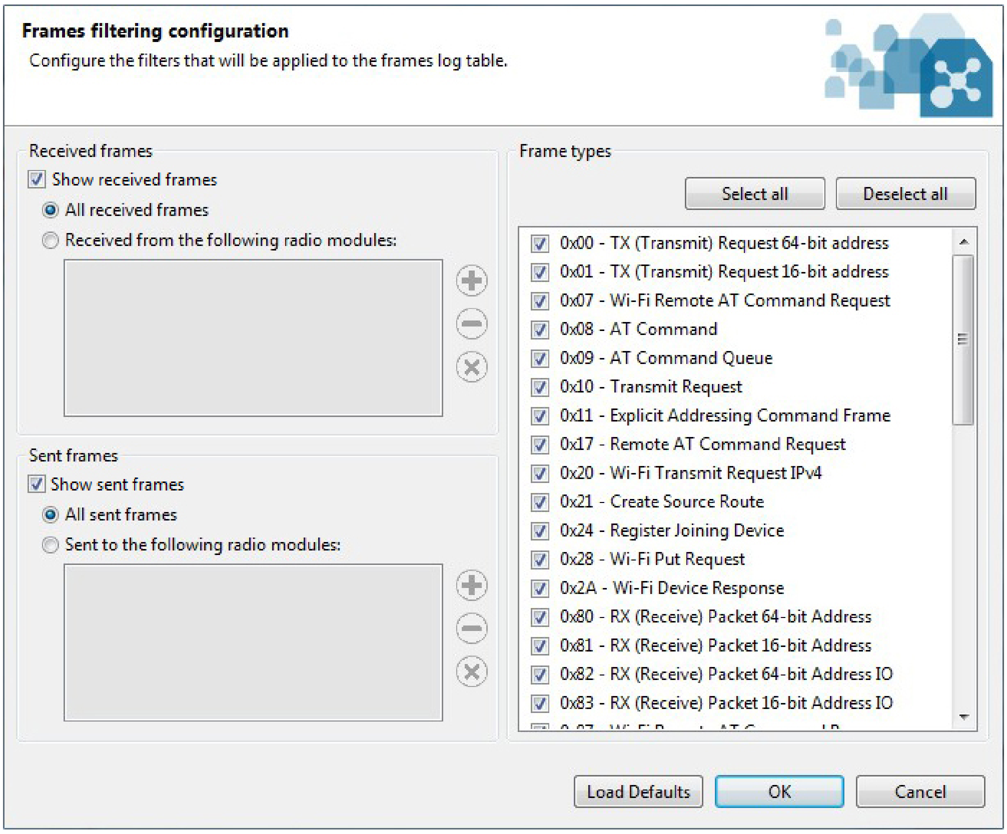
54 50 (TP)

The decoded information includes the type of the API frame with all the specific frame fields as well as the start delimiter, length and checksum of the frame. Frames with an RF data field display it in a tabbed control with the option to see the data in Hexadecimal and ASCII formats.

Filtering frames

You can filter the sent/received frames in the API console to narrow the list contents and focus only on some frames. Click the **Configure filters** button to begin.

A new dialog displays all the available filtering options:



Frames filtering configuration

Configure the filters that will be applied to the frames log table.

Received frames

☑ Show received frames

● All received frames

○ Received from the following radio modules:

Sent frames

☑ Show sent frames

● All sent frames

○ Sent to the following radio modules:

Frame types

- ☑ 0x00 - TX (Transmit) Request 64-bit address
- ☑ 0x01 - TX (Transmit) Request 16-bit address
- ☑ 0x07 - Wi-Fi Remote AT Command Request
- ☑ 0x08 - AT Command
- ☑ 0x09 - AT Command Queue
- ☑ 0x10 - Transmit Request
- ☑ 0x11 - Explicit Addressing Command Frame
- ☑ 0x17 - Remote AT Command Request
- ☑ 0x20 - Wi-Fi Transmit Request IPv4
- ☑ 0x21 - Create Source Route
- ☑ 0x24 - Register Joining Device
- ☑ 0x28 - Wi-Fi Put Request
- ☑ 0x2A - Wi-Fi Device Response
- ☑ 0x80 - RX (Receive) Packet 64-bit Address
- ☑ 0x81 - RX (Receive) Packet 16-bit Address
- ☑ 0x82 - RX (Receive) Packet 64-bit Address IO
- ☑ 0x83 - RX (Receive) Packet 16-bit Address IO

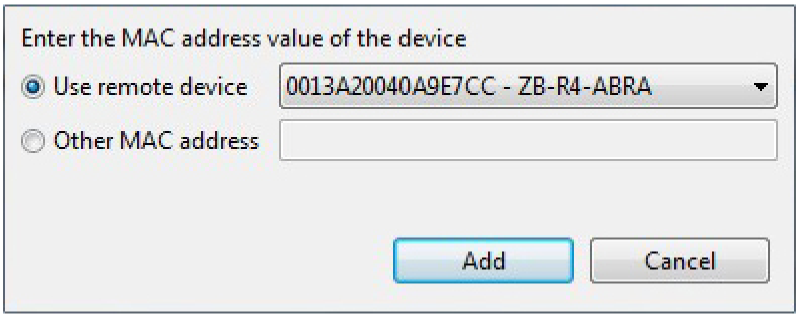
There are three main categories of frames filtering:

Received frames

This section configures how to filter the received frames and has the following options:

- **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 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 click on the + button. A new dialog appears.

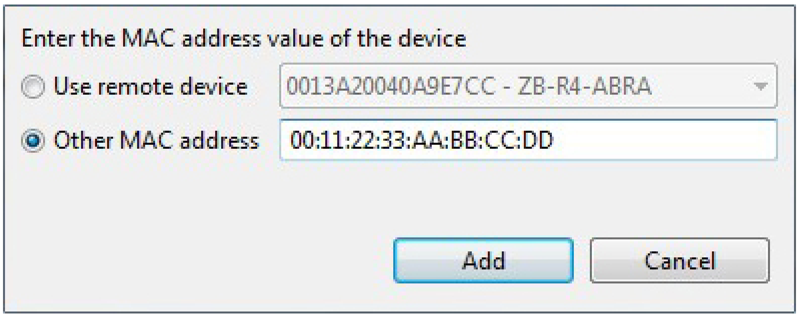
  If any remote devices have been discovered for your local device, you can select the specific device from the combo box:


Enter the MAC address value of the device

● Use remote device: 0013A20040A9E7CC - ZB-R4-ABRA

○ Other MAC address

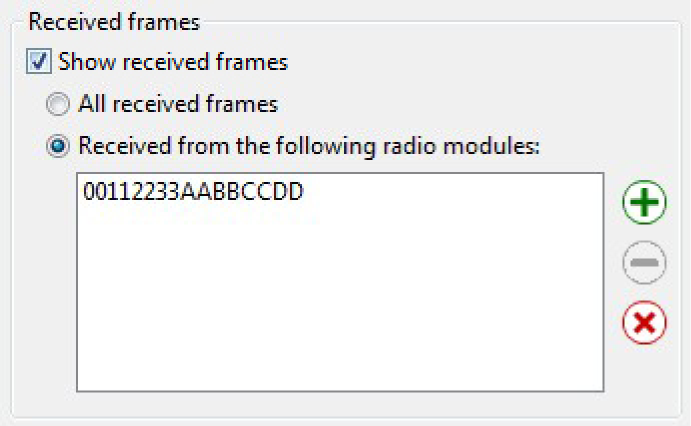
If no remote devices are discovered for the local device, you can enter the MAC address manually:


Enter the MAC address value of the device

○ Use remote device: 0013A20040A9E7CC - ZB-R4-ABRA

● Other MAC address: 00:11:22:33:AA:BB:CC:DD

Click the Add button to add the MAC address to the list:


Received frames

☑ Show received frames

○ All received frames

● Received from the following radio modules:

00112233AABBCCDD

Sent frames

This section configures how to filter the sent frames and has the following options:

- **Show sent frames**: Shows/hides sent frames.
- **All sent frames**: Shows all the sent frames.
- **Sent to the following radio modules**: Shows the sent frames by destination addresses and lists the source MAC addresses that can be filtered.

Frame types

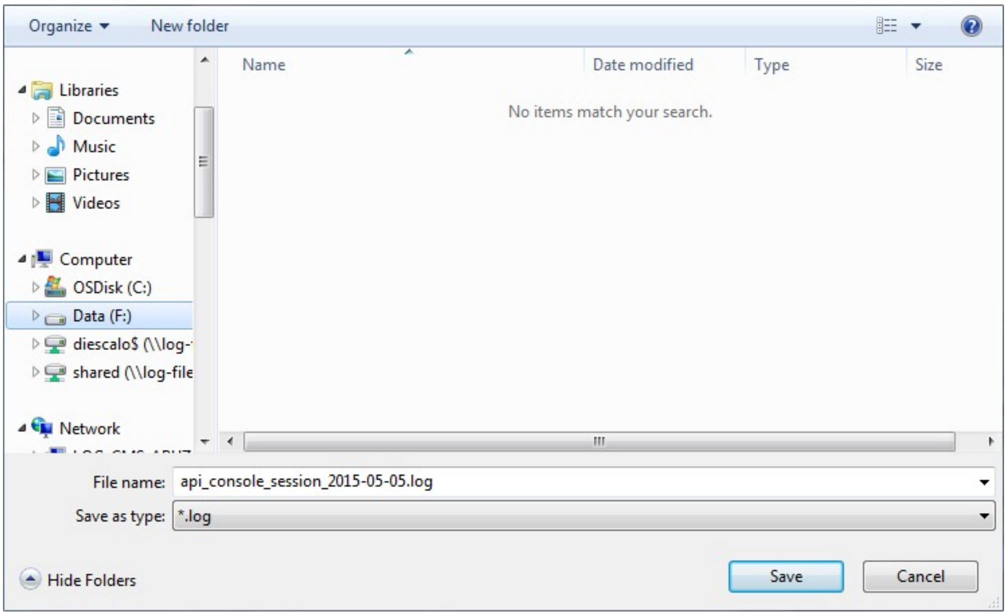
This sections allows you to filter frames by type. All the available frame types are displayed in the list with a checkbox next to each one. By default all frame types are selected. Check/uncheck the desired frame types to configure the list. You can also use the buttons:

- **Select all**: Checks all the frame types.
- **Deselect all**: Unchecks all the frame types.

Saving a 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.

A new **Save file** dialog box asks you for the name and path console session file. XCTU defaults to the physical address of the device followed by the current date for the file name.



No items match your search.

File name: api_console_session_2015-05-05.log

Save as type: *.log

Click **Save** to save the console session file.

Loading a console session

You can open XCTU's **Load console session** tool by clicking the **Load console session** button.

For more information about how to use this tool, see the Load console session tool topic.

Other traffic monitoring features

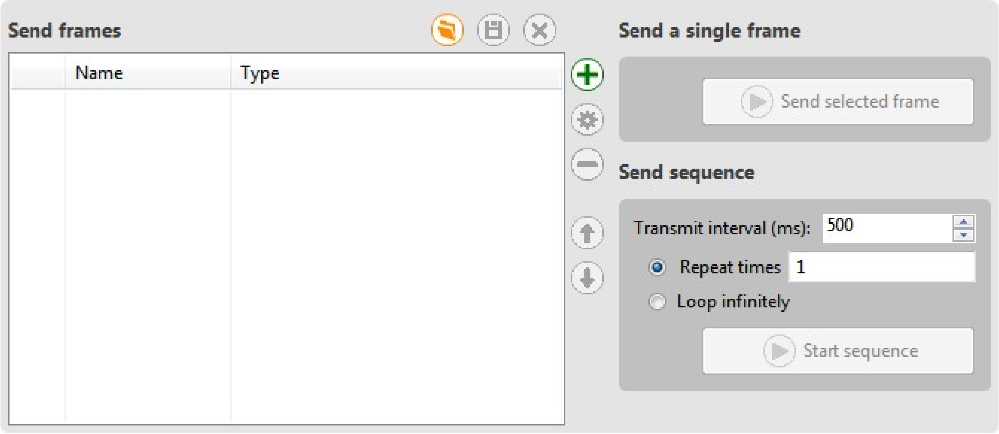
The sent and received API frames table also contains a button to block the scroll.

If you click the **Lock scroll** button, the API frames table won't automatically scroll to the latest API frame when it is sent or received. You will need to manually scroll to the desired frame.

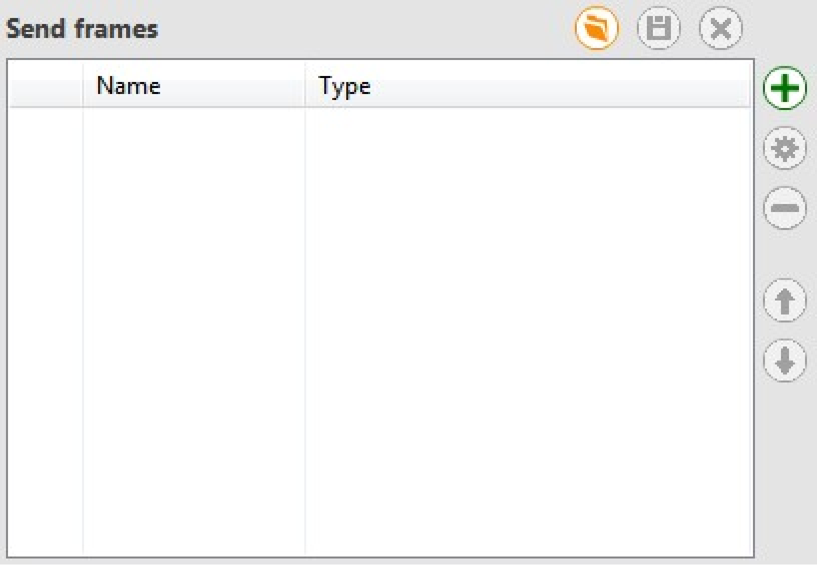
Clicking the button again unlocks the scroll.

You can clear the list of sent and received API frames by clicking the **Clear frames list** button.

Send API frames section



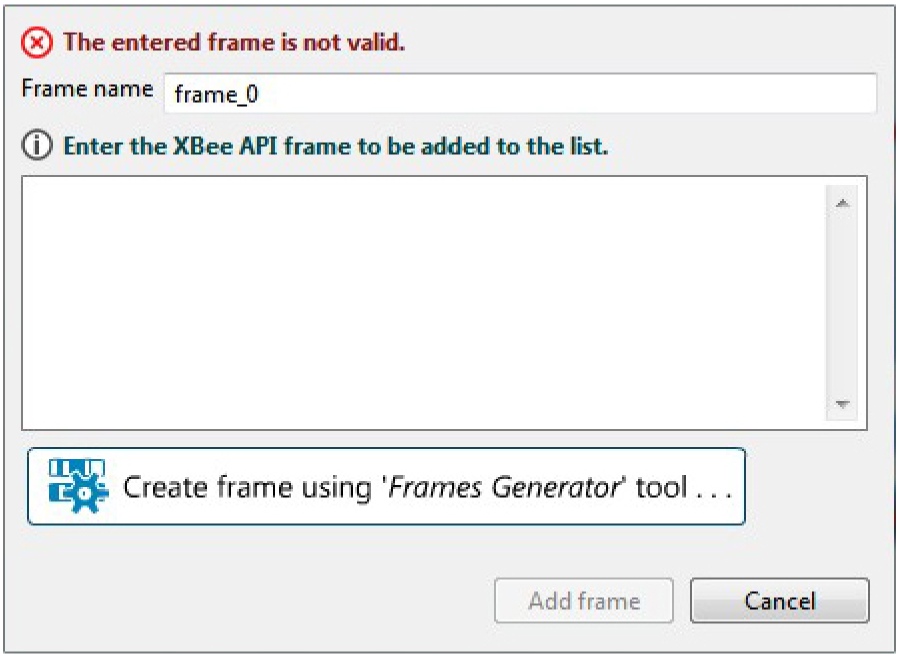
The send API frame section sends a frame or sequence of frames to the radio module. The main control of this section is a list containing the frame or frames to be sent. By default, this list is empty.



Creating an API frame

The first step to send an API frame is to create it. Follow these steps to add a new API frame to the list:

1. Click the **Add new frame** button. ⊕
   The **Add new frame** dialog box appears.



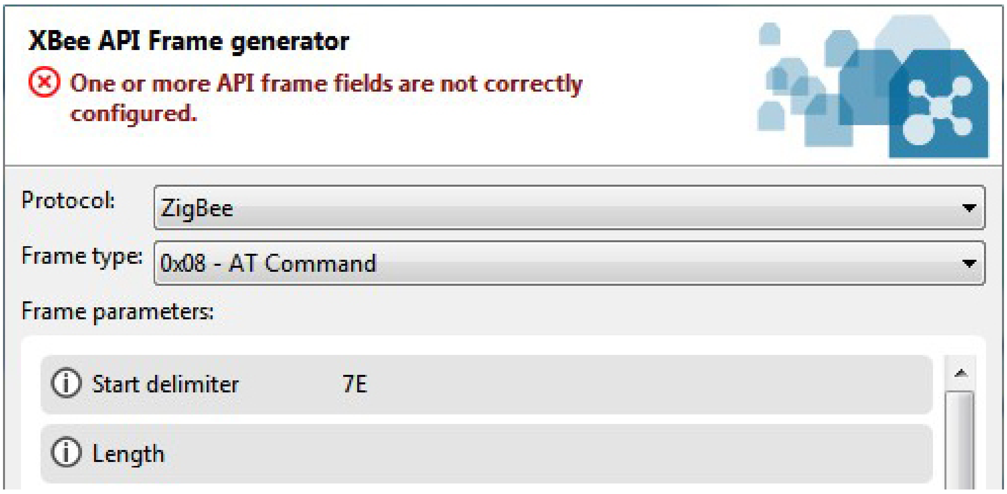
The entered frame is not valid.

Frame name: frame_0

Enter the XBee API frame to be added to the list.

2. Enter a symbolic name for the API frame and fill the contents of the frame. You can type in the byte array of the API frame yourself, or you can click the **Open the frames generator tool** button to open the Frames Generator tool.

3. The Frames Generator tool appears. Configure your API frame and click the **OK** button. For more information about using the Frames generator tool see the Frames generator tool topic.



XBee API Frame generator

One or more API frame fields are not correctly configured.

Protocol: ZigBee

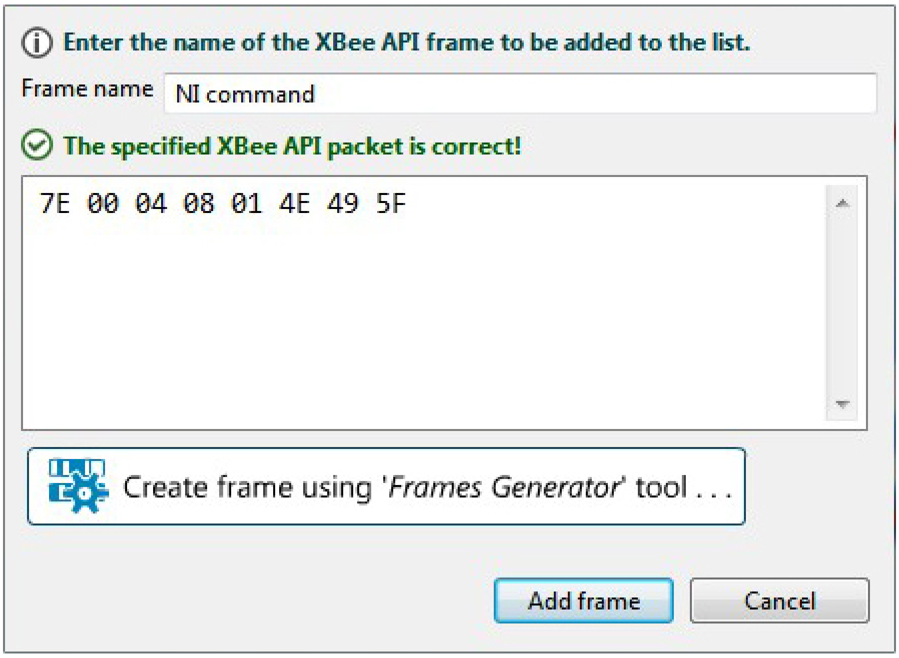
Frame type: 0x08 - AT Command

Frame parameters:

Start delimiter 7E

Length

4. Your generated frame appears in the API frame.



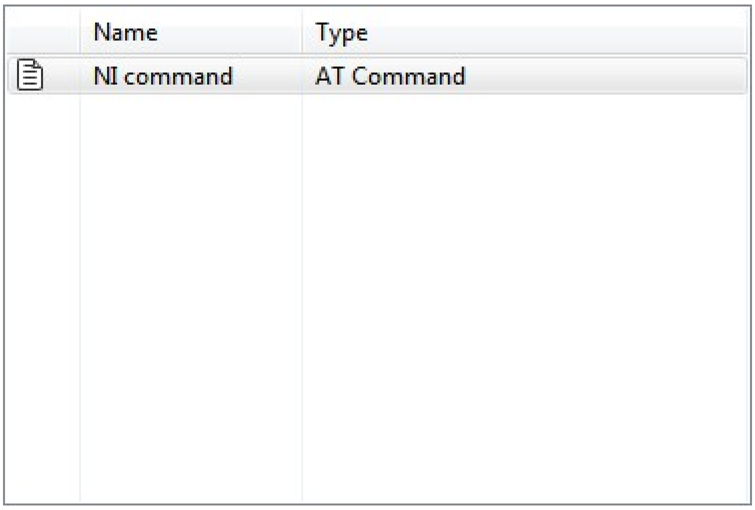
Enter the name of the XBee API frame to be added to the list.

Frame name: NI command

The specified XBee API packet is correct!

```
7E 00 04 08 01 4E 49 5F
```

5. Click the **Add frame** button to add the API frame to the list of frames to send. You will see it in the list of frames.



	Name	Type
🗎	NI command	AT Command

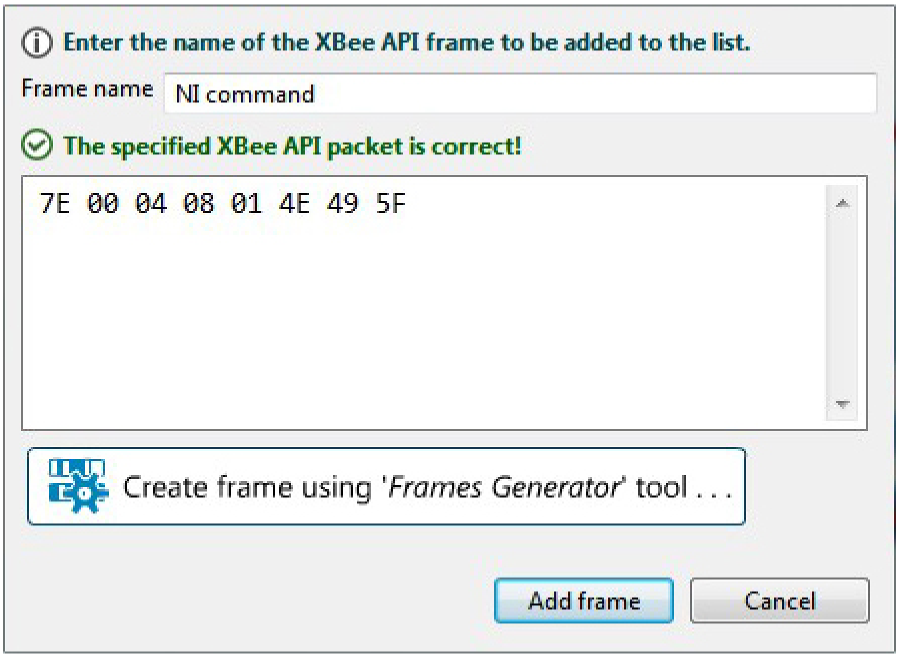
You can repeat the same operation to add more API frames to the list.

Managing API frames

Once an API frame is added to the list, you can perform the following actions:

- **Edit a frame**. To edit a frame, select it and click the **Edit selected frame** button. ⚙

  Within the edit dialog box, you can change the name of the frame and its content. You can also click the **Open the frames generator tool** to edit the content of the frame using the Frames Generator tool.



Enter the name of the XBee API frame to be added to the list.

Frame name: NI command

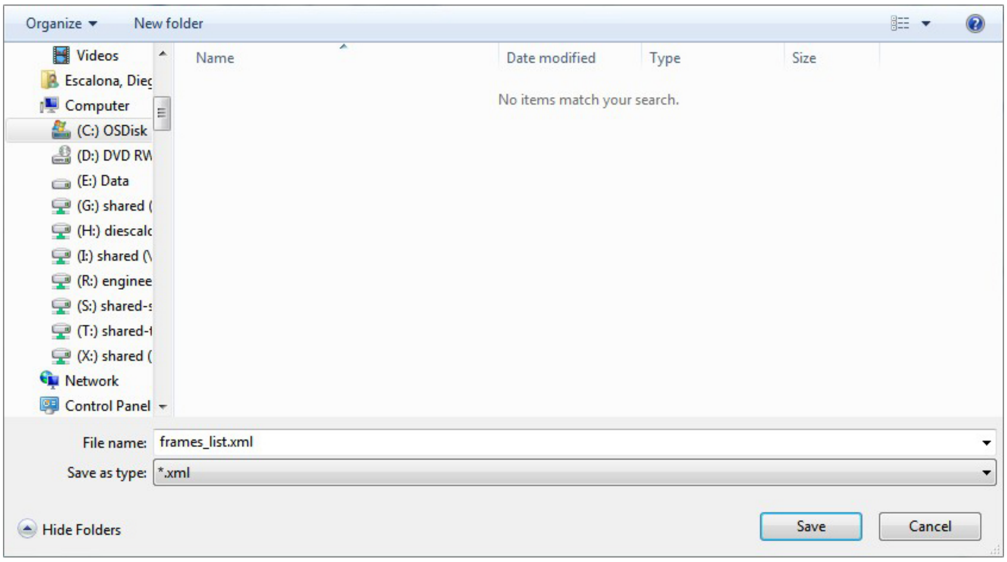
The specified XBee API packet is correct!

```
7E 00 04 08 01 4E 49 5F
```

Click **Apply changes** to save the changes made to the API frame.

- **Remove a frame**. To remove a frame from the list, select it and click the **Remove selected frame** button.
- **Change the order of a frame**. If the list of frames has more than one API frame, you can change the order of the frames. To do so select the API frame you want to move and then click the **Move up** or **Move down** buttons to change the position of the frame in the list.
- **Save the list of frames**. XCTU lets you save the list of frames you have created to be used in future sessions or on different PCs. To save your list of frames, click the **Save frames list** button.

  A new **Save file** dialog box asks you for the name and path of the file containing the list of API frames.

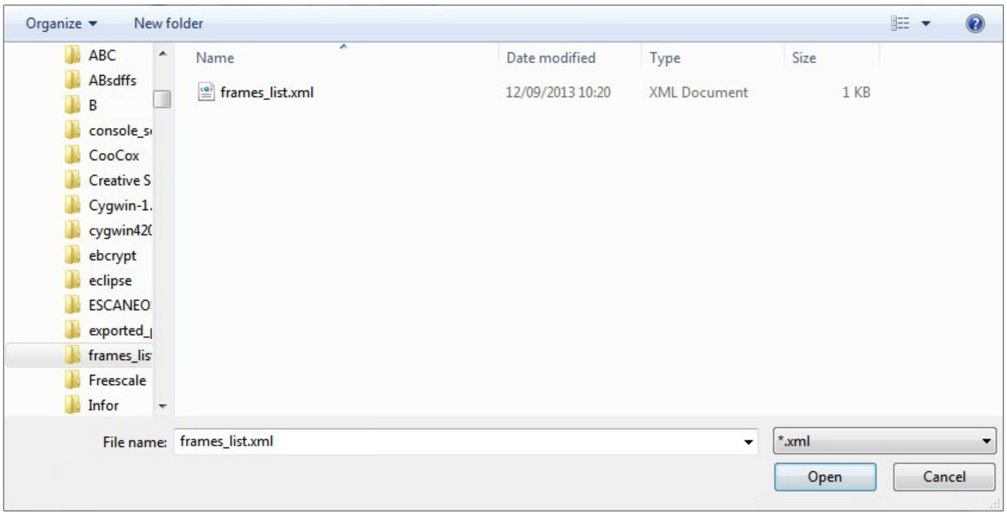
  File name: frames_list.xml

  Save as type: *.xml

  No items match your search.

  Enter a name and a path for the file and click **Save** to save the API frames list file in XML format.
- **Load a list of frames**. Click the **Load frames list** button to load a previously saved list of API frames.

A **Load file** dialog box asks for the file containing the list of frames to load.



Name	Date modified	Type	Size
frames_list.xml	12/09/2013 10:20	XML Document	1 KB

File name: frames_list.xml

*.xml

Select the correct file and click **Open** to load the list of API frames.

If the frames are correctly loaded, they appear in the list. If not, an error message provides the cause of the problem.

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

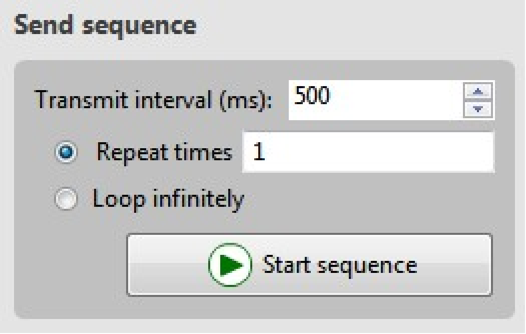
Sending a single API frame

To send a single frame, select it from the list of API frames and then click the **Send selected frame** button on the right.

The sent frame appears in the API frames traffic section.

Sending a sequence of API frames

You can send a sequence of API frames to the radio module. The sequence is made up of the list of frames to send and the send sequence options located next to the list.



Transmit interval (ms): 500

● Repeat times 1

○ Loop infinitely

Configure the following settings in the send sequence options:

- **Transmit interval**: This is the time in milliseconds between frames. The minimum value is 0ms and the maximum value is 60000ms (1 minute)
- **Repeat times**: If you check this option you must enter the number of times the sequence should be repeated or sent. By default this value is 1.

- **Loop infinitely**: If you prefer, you can send the sequence of frames endlessly by checking this option.

> The send sequence options are also saved when you save the list of frames, and are loaded when you load a list of API frames.

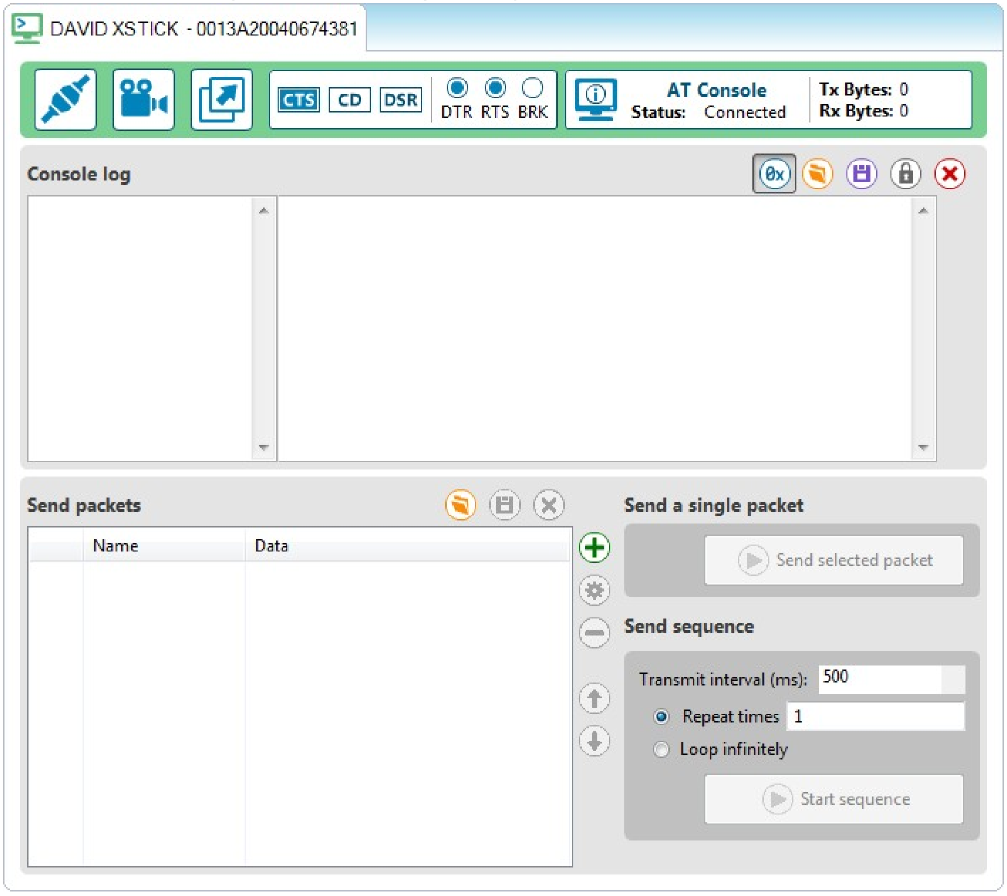
When you have all the send sequence options configured, you can start sending the list of API frames by clicking the **Start sequence** button located below the options.

The sent and received frames appear in the API frames traffic section.

You can also stop the sending process at any time by clicking the **Stop sequence** button.

AT Console

The AT Console allows you to communicate with radio modules running in AT (transparent) operating mode. This console is displayed when the active working mode of XCTU is Consoles and you select a radio module configured with AT operating mode.

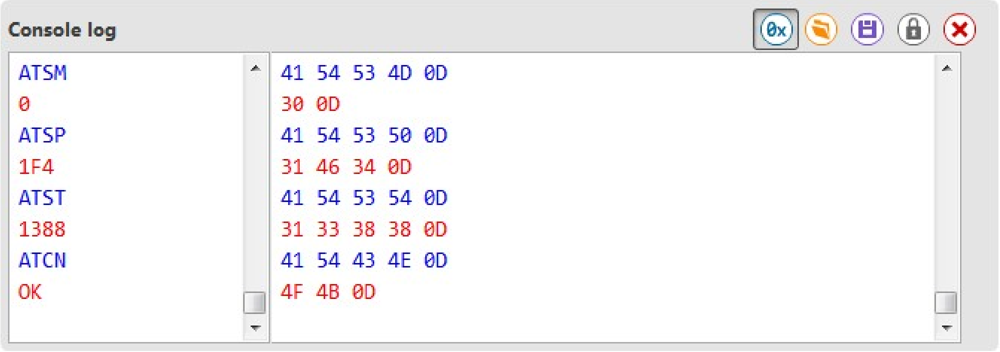

DAVID XSTICK - 0013A20040674381

AT Console
Status: Connected
Tx Bytes: 0
Rx Bytes: 0

Transmit interval (ms): 500
● Repeat times 1
○ Loop infinitely

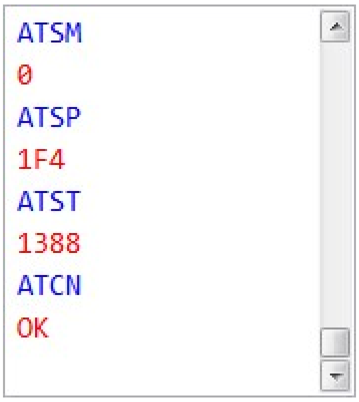
The main difference between this console and the API console is that the communication with the device is direct. All the data you send through the serial interface is queued for RF transmission by the module, and all the RF data received by the module is sent through the serial interface. This console simply displays the data (characters) sent and received through the serial port.

Data traffic monitoring section


```
ATSM    41 54 53 4D 0D
0       30 0D
ATSP    41 54 53 50 0D
1F4     31 46 34 0D
ATST    41 54 53 54 0D
1388    31 33 38 38 0D
ATCN    41 54 43 4E 0D
OK      4F 4B 0D
```

In the AT console, the main control is a data box that displays all the sent and received data characters. Sent characters are blue and received characters are red.

Once data is sent or received by the module, it is added to the end of the data box.

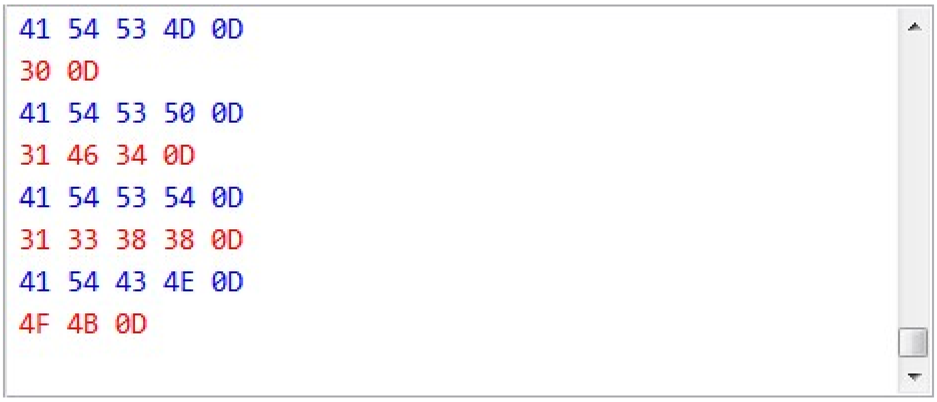


```
ATSM
0
ATSP
1F4
ATST
1388
ATCN
OK
```

The communication with the radio module in AT operating mode does not need any structured data; therefore, you can type directly in the data box control. The characters you enter are automatically sent to the radio module.

Hexadecimal view

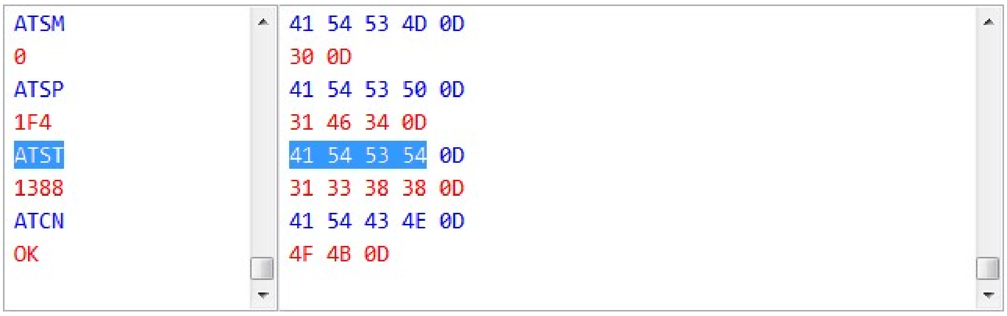
The right side of the data box corresponds to the hexadecimal representation of all the sent and received data characters.



```
41 54 53 4D 0D
30 0D
41 54 53 50 0D
31 46 34 0D
41 54 53 54 0D
31 33 38 38 0D
41 54 43 4E 0D
4F 4B 0D
```

0x	You can show or hide this view by clicking the **Show hexadecimal** button located above the data box.

Note that when you select a character in the data box, the hexadecimal representation is also selected, and vice-versa.

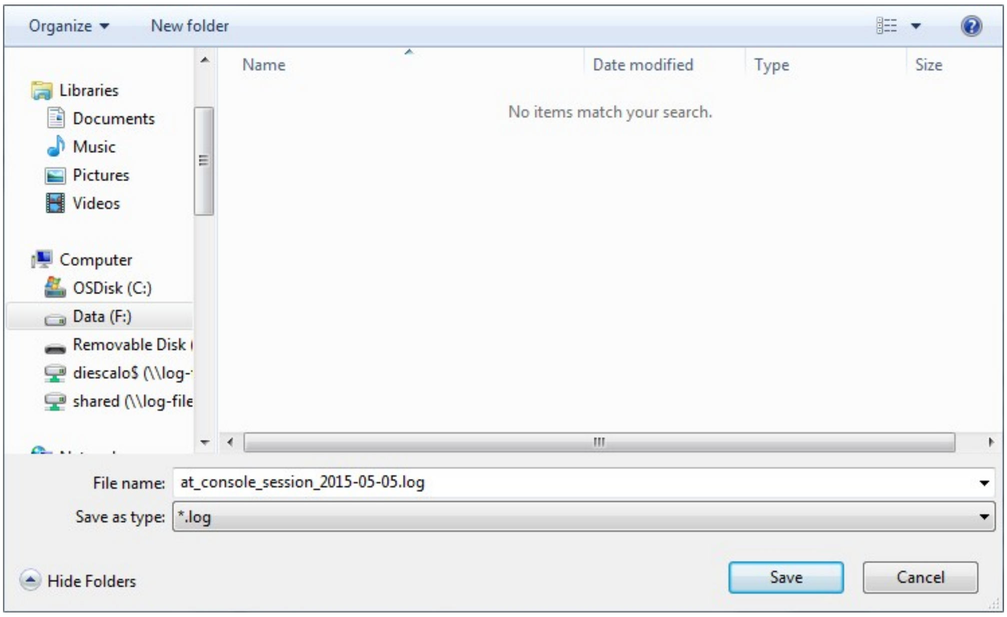


```
ATSM    41 54 53 4D 0D
0       30 0D
ATSP    41 54 53 50 0D
1F4     31 46 34 0D
ATST    41 54 53 54 0D
1388    31 33 38 38 0D
ATCN    41 54 43 4E 0D
OK      4F 4B 0D
```

Saving a console session

💾	Click the **Save console session** button to save the console session, which is the set of sent and received data.

A new **Save file** dialog box requests the name and path of the file containing the console session. By default, the name is the physical address of the device followed by the current date.



File name: at_console_session_2015-05-05.log
Save as type: *.log

Click **Save** to save the console session file.

Loading a console session

📂	You can also load a console session using the **Load console session** button. This button opens the **Load console session** embedded tool of XCTU.

For more information about how to use this tool, see the Load console session tool topic.

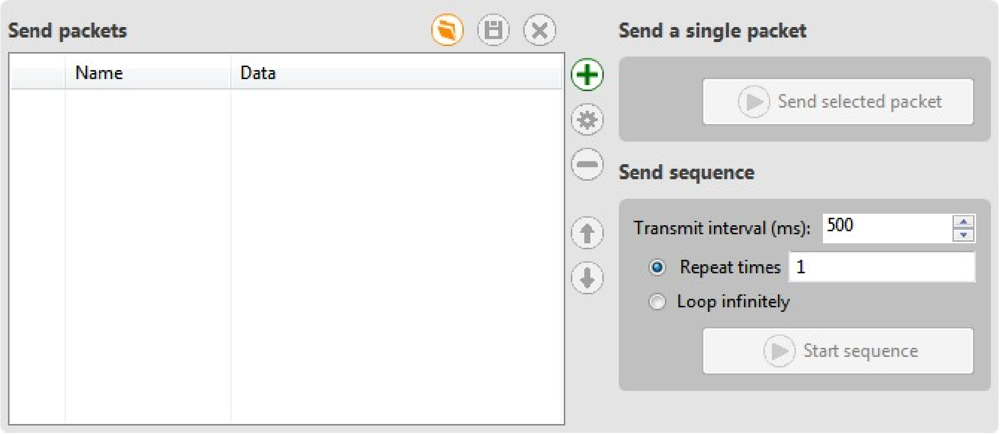
Other traffic monitoring features

🔒	The data box also contains a button to lock the scroll. If you click the **Lock scroll** button, the data box won't automatically scroll to the data character when it is sent or received. You need to manually scroll to it.

Clicking the button again unlocks the scroll.

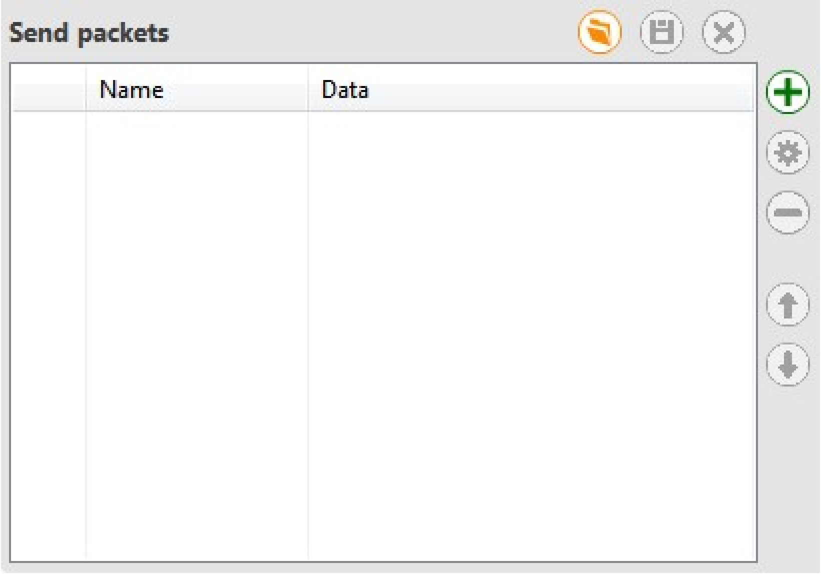
❌	The red X icon is the **Clear data** button. Click the **Clear data** button to clear all the data characters as well as their hexadecimal representation.

Send data packets section



The send data packets section sends a group or groups of characters to the radio module. When you write in the data box, XCTU sends the data as individual characters. Sending a data packet sends all of the characters in a single operation.

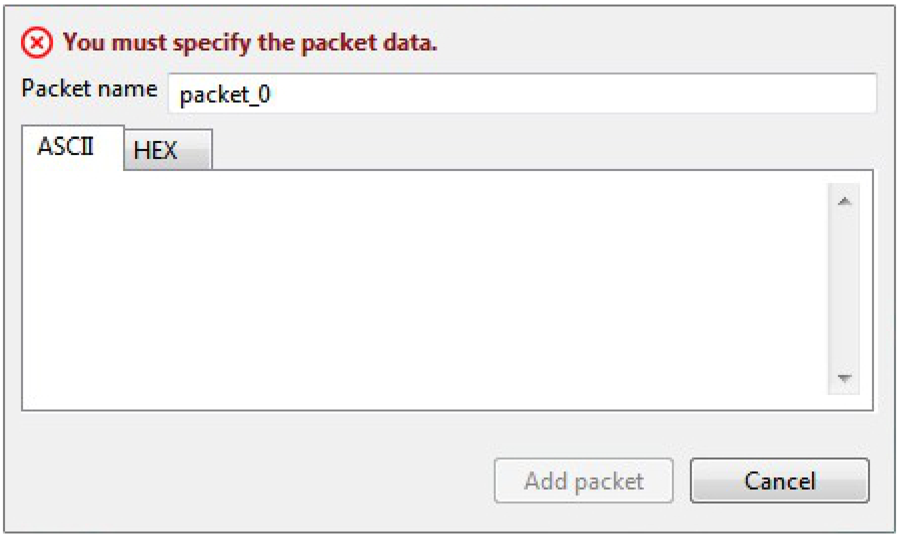
By default, the data packets list is empty.



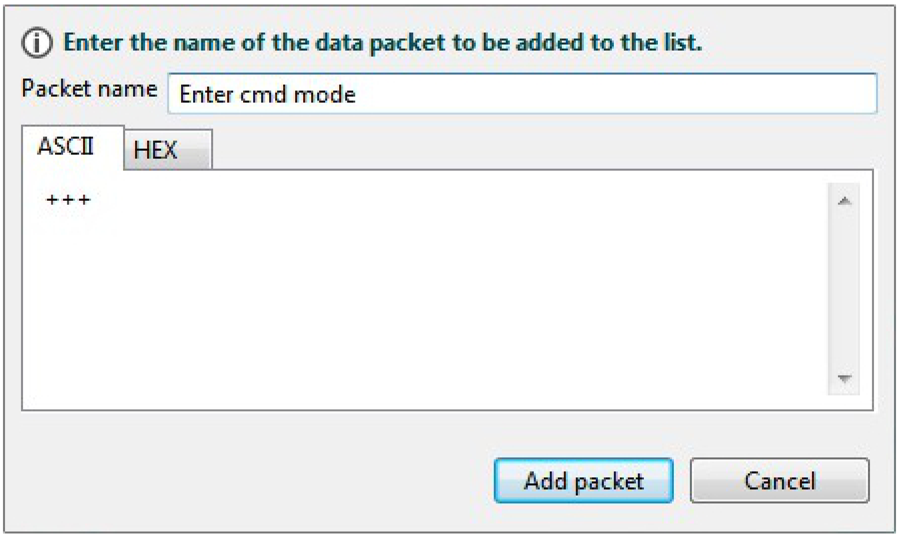
Creating a data packet

Follow these steps to add a new data packet to the list:

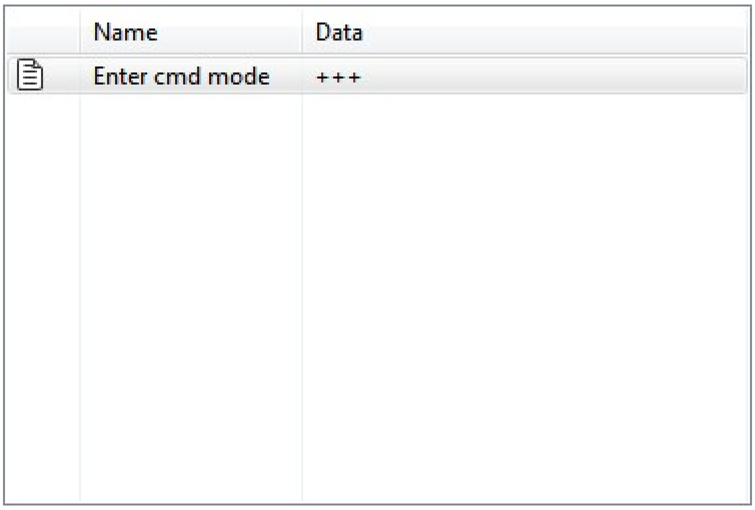
1. Click the **Add new packet** button. ⊕
2. The **Add new packet** dialog appears.



3. Enter a symbolic name for the data packet and then type in your commands. You can change between ASCII and HEX views to enter the contents of the packet in ASCII or hexadecimal.



4. Click the **Add packet** button to add the data packet to the list of packets to send. You will see it in the list of packets.



	Name	Data
	Enter cmd mode	+++

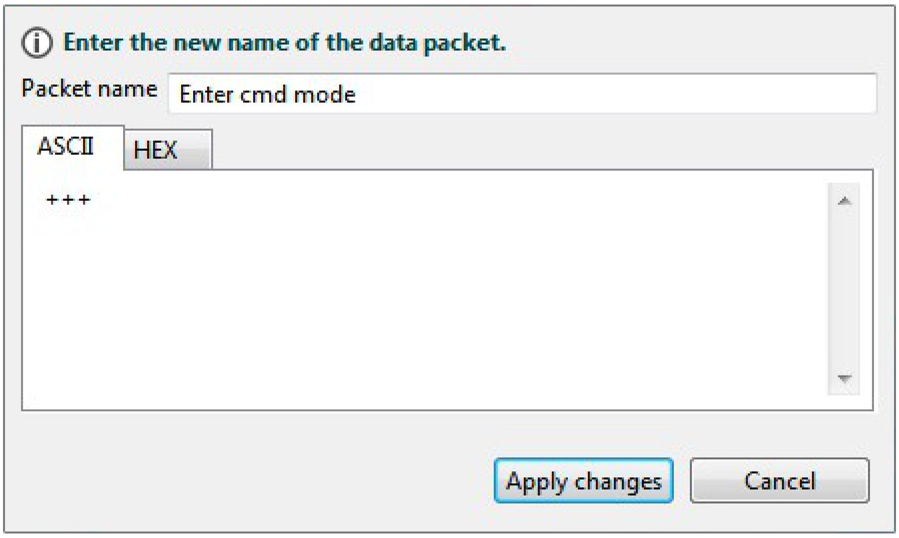
Repeat the same operation to add more data packets to the list.

Managing data packets

Once a data packet is added to the list, you can perform the following actions with it:

- **Edit a packet**. Select a packet and click the **Edit selected packet** button.

  You can change the name of the packet and its content in the edit dialog box.

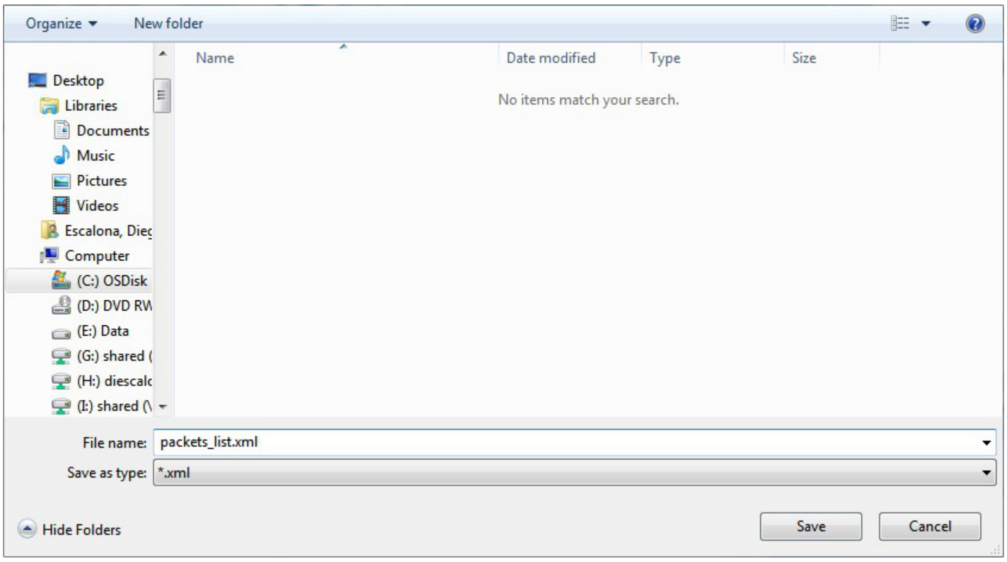
  Enter the new name of the data packet.

  Packet name: Enter cmd mode

  ```
  +++
  ```

  Click **Apply changes** to save the changes of the packet.
- **Remove a packet**. To remove a data packet from the list, select it and click the **Remove selected packet** button.
- **Change the order of a packet**. You can change the order of the packets in a list by selecting a data packet and clicking the **Move up** or **Move down** buttons.
- **Save the list of packets**. You can save the list of packets you have created to be used in future sessions or on different PCs. To save your list of packets, click the **Save packets list** button.

  A new **Save file** dialog box asks for the name and path of the file containing the list of data packets.

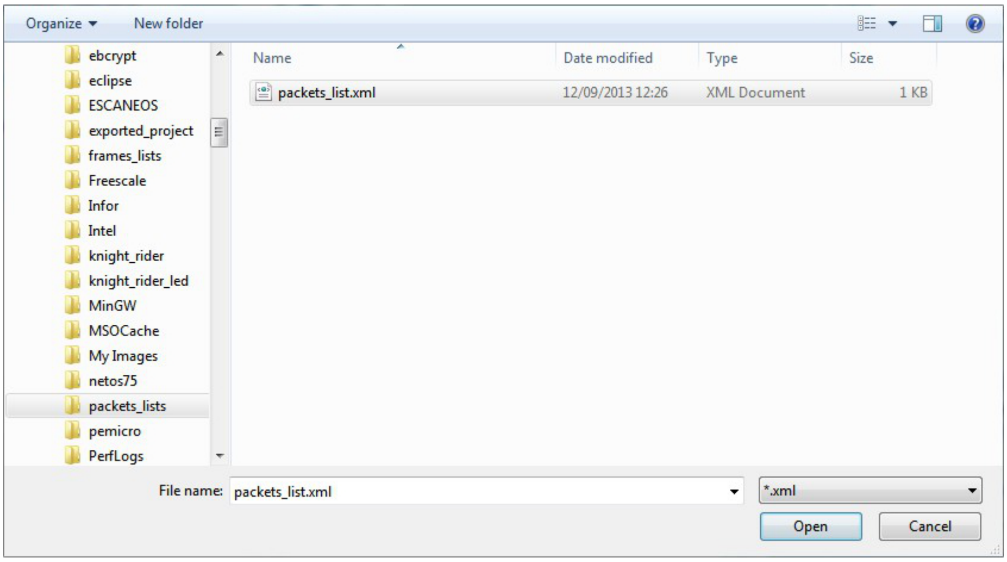
  No items match your search.

  File name: packets_list.xml

  Save as type: *.xml

  Enter a name and a path for the file and click **Save** to save the data packets list file in XML format.
- **Load a list of packets**. Click the **Load packets list** button to load a list of saved data packets.

A **Load file** dialog box asks for the file containing the list of packets to load.



File name: packets_list.xml

*.xml

Select the correct file and click **Open** to load the list of data packets.

If the packets are correctly loaded they appear in the list. If not, an error message provides the cause of the problem.

- **Clear the list of packets**. Click the **Clear list** button to clear the list of data packets.

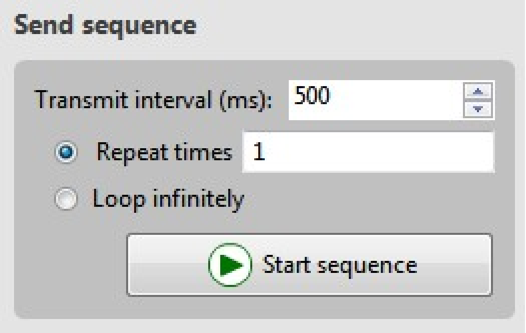
Sending a single data packet

To send a single packet, select it from the list of data packets and then click the **Send selected packet** button on the right.

The sent data appears in the Data traffic section.

Sending a sequence of data packets

You can send a sequence of data packets to the radio module. The sequence is defined by the list of packets to send and the send sequence options located next to the list.



Transmit interval (ms): 500

● Repeat times 1

○ Loop infinitely

Configure the following settings in the send sequence options:

- **Transmit interval**: This is the time in milliseconds between packets. The minimum value is 0ms and the maximum value is 60000ms (1 minute)

- **Repeat times**: If you check this option, you must enter the number of times the sequence should be repeated or sent. By default this value is 1.
- **Loop infinitely**: If you prefer, you can send the sequence of packets endlessly by checking this option.

> The send sequence options are saved when You save the list of packets, andare loaded when you load a list of data packets.

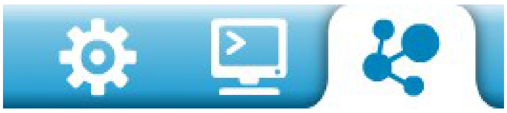
When you have all the send sequence options configured, you can start sending the list of data packets by clicking the **Start sequence button** located below the options.

The sent and received data appears in the Data traffic section.

You can also stop the sending process at any time by clicking the **Stop sequence** button.

Manage your radio network

The Network working mode of XCTU allows you to discover and see the topology of your radio module's network. To do this, select the Network working mode in the toolbar.

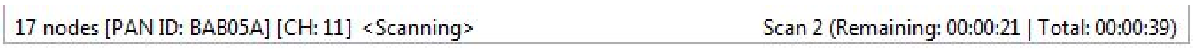


> The Network perspective only works in API operating mode. Radio modules in AT (transparent) mode do not support the network discovery process.

Scanning the network

▶	Clicking the **Start** button scans the radio module's network and returns a list of available modules. It might take some time to finish searching. As the radio modules are found, they are shown in the working area.
■	The discovery process automatically stops when the scans specified in the preferences finish, altough you can stop it by clicking the **Stop** button.

At the bottom, XCTU shows the following:



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

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

On the left side:

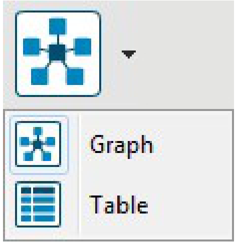
- The number of network nodes (17 nodes)
- The PAN ID ([PAN ID: BAB05A])
- The Channel ([CH: 11])
- The status (<Scanning>) (it can be *Getting information*, *Initializing scan*, *Scanning*, *Waiting for next scan* or *Stopped*)

On the right:

- The current scan number (Scan 2)
- The estimated remaining time to finish the current scan (Remaining: 00:00:21)
- The total time passed since the network scan was started (Total: 00:00:39)

Switching the view

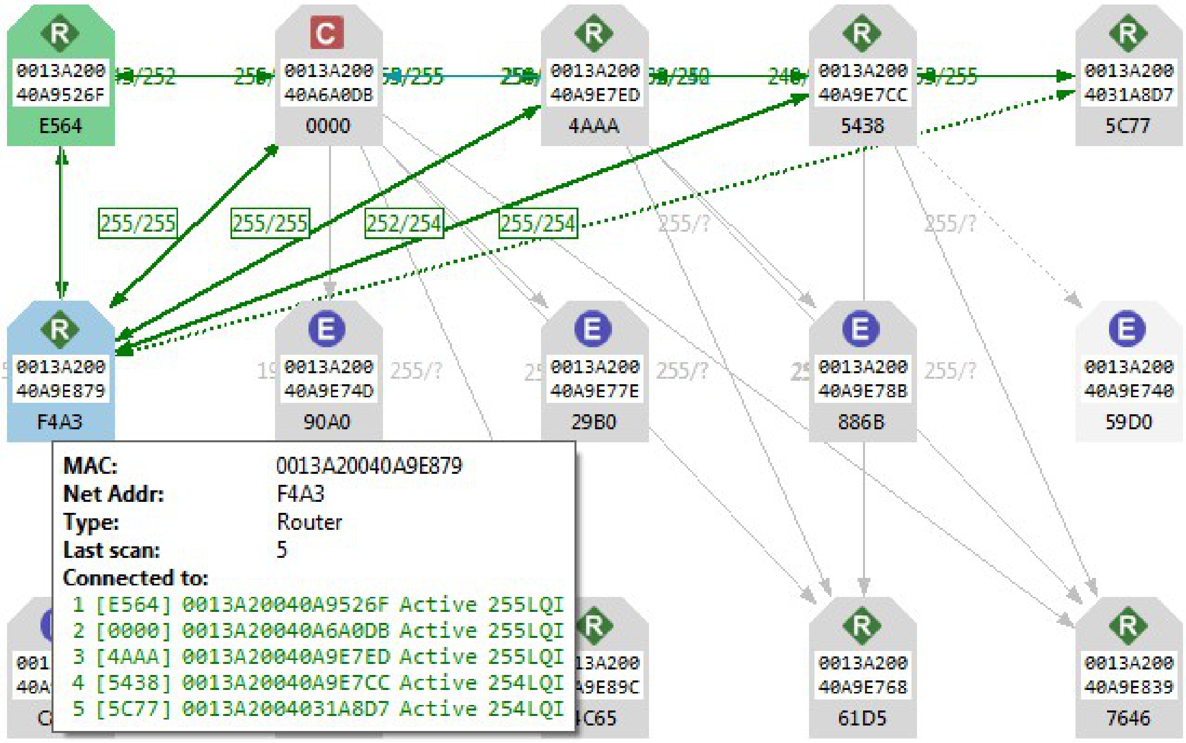
The Network perspective offers two modes for displaying the network of your local radio module:



- Graph view
- Table view

Graph view

The radio modules are graphically displayed as nodes of the network, showing each module's role (Coordinator, Router, or End Device) and MAC address, as well as the 16 bit Network Address for ZigBee networks and Node Identifier for the rest of protocols. Hover the mouse cursor over a node to see more information about its connections.



MAC: 0013A20040A9E879
Net Addr: F4A3
Type: Router
Last scan: 5
Connected to:

```
1 [E564] 0013A20040A9526F Active 255LQI
2 [0000] 0013A20040A6A0DB Active 255LQI
3 [4AAA] 0013A20040A9E7ED Active 255LQI
4 [5438] 0013A20040A9E7CC Active 254LQI
5 [5C77] 0013A2004031A8D7 Active 254LQI
```

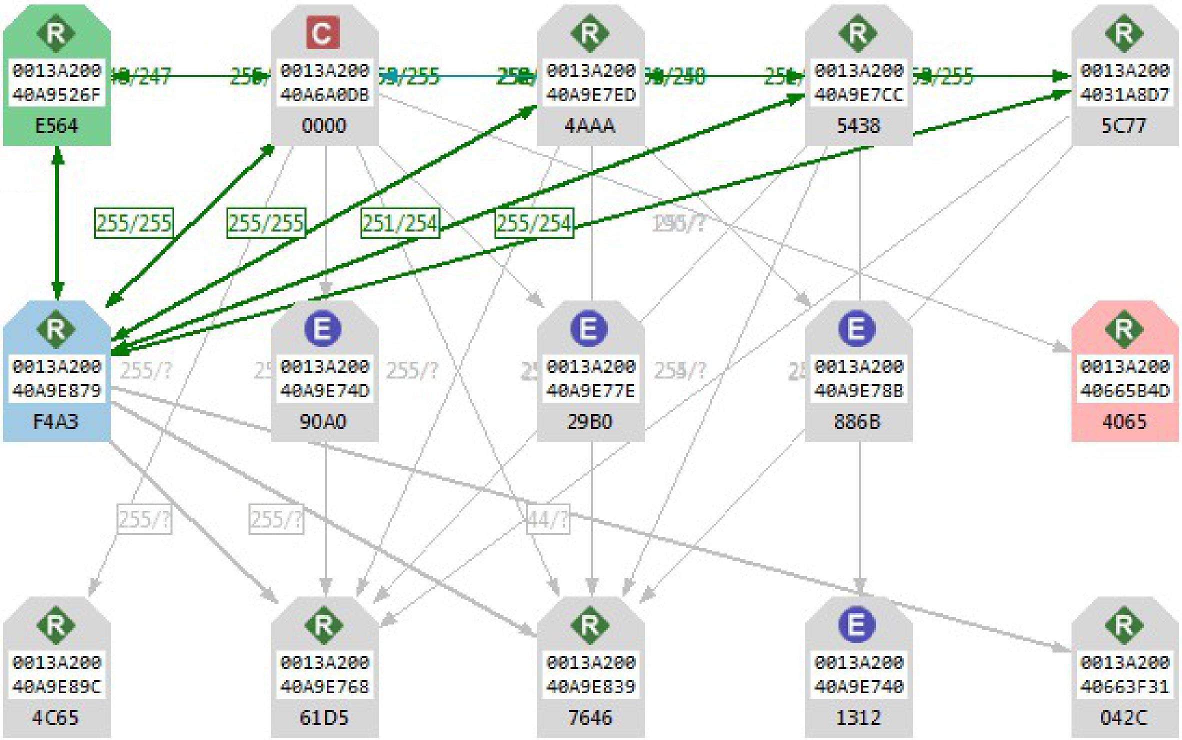
The nodes are connected to their neighbors with lines. The arrows indicate the direction of the communication.

> In a large network, it can be difficult to identify the node connections. Click the node you want to inspect to see its connections.

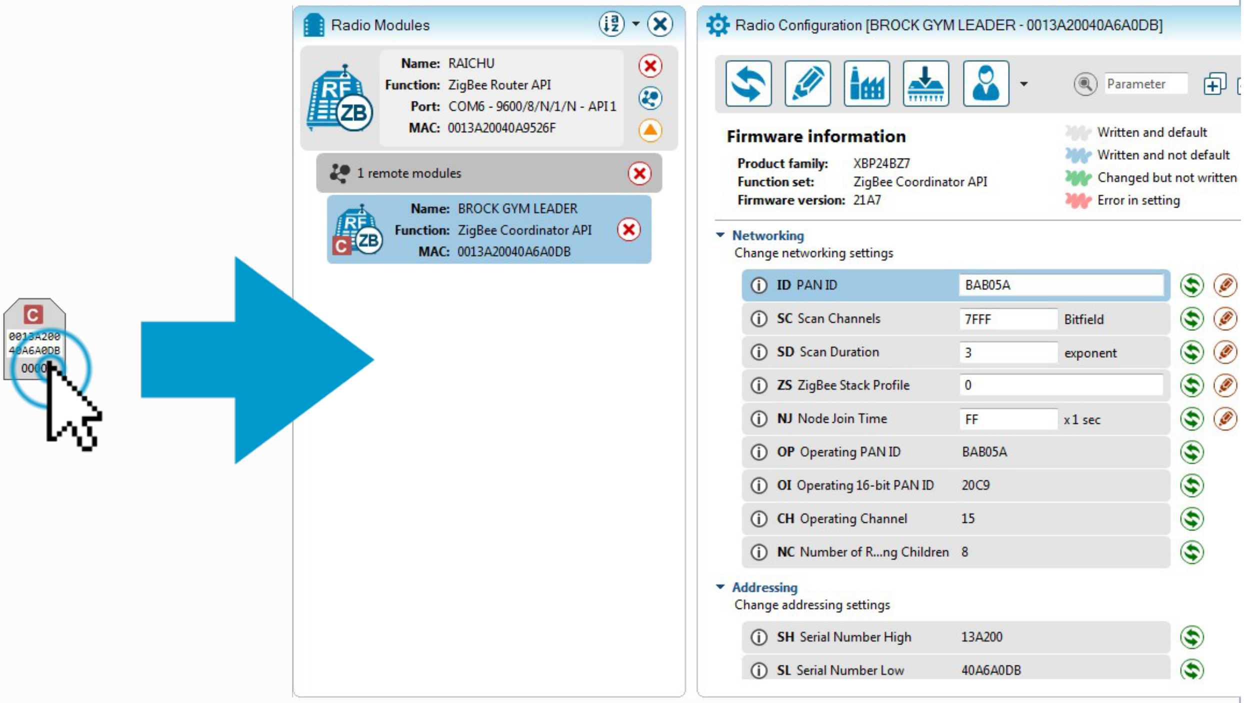
For more information about connections, see the Connections in 802.15.4, DigiMesh, and ZigBee topic.

Nodes discovered in previous scans that have not already been found are displayed in lighter color; they will turn on as they are discovered. The connections that are not discovered yet have a dotted line, while a solid line represents an active connection. Selected nodes are marked in blue and connections in black.

Radio modules colored in red represent devices that are in the network (or have been) but are unreachable now. When a radio module leaves the network, some devices continue to store information about their relationship. So, when a new discovery is performed, the module appears because of the preserved information, but the device is unreachable if you try to verify the connection.



If you double click a node, XCTU switches to the configuration mode displaying the settings of that node. If the node was not in the list of remote nodes, XCTU will add it before displaying its settings.



Name: RAICHU
Function: ZigBee Router API
Port: COM6 - 9600/8/N/1/N - API 1
MAC: 0013A20040A9526F

1 remote modules

Name: BROCK GYM LEADER
Function: ZigBee Coordinator API
MAC: 0013A20040A6A0DB

Firmware information

Product family: XBP24BZ7
Function set: ZigBee Coordinator API
Firmware version: 21A7

Networking
Change networking settings

ID PAN ID	BAB05A	
SC Scan Channels	7FFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x 1 sec
OP Operating PAN ID	BAB05A	
OI Operating 16-bit PAN ID	20C9	
CH Operating Channel	15	
NC Number of R...ng Children	8	

Addressing
Change addressing settings

SH Serial Number High	13A200
SL Serial Number Low	40A6A0DB

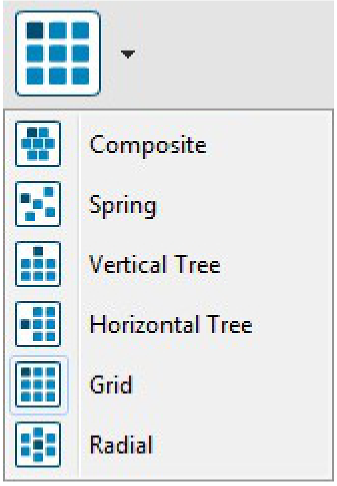
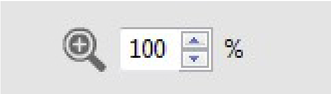
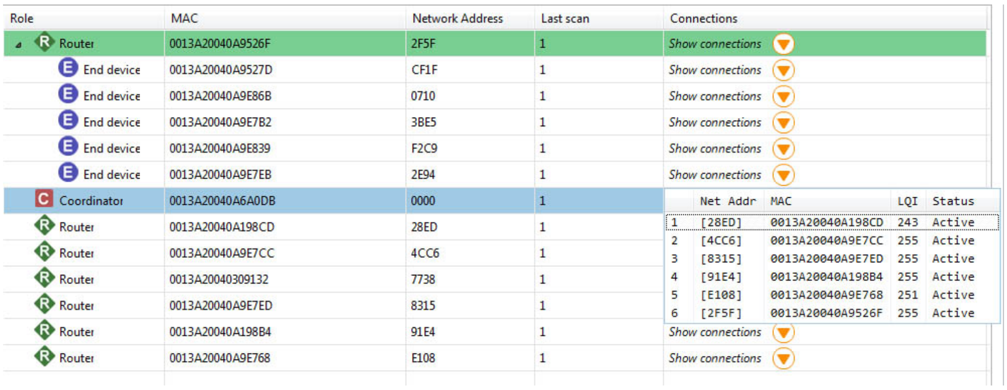
<table>
<tr>
<td> </td>
<td>You can change the network layout by clicking <b>Set layout</b>. There are six types of layouts:
<ul>
<li>Composite</li>
<li>Spring</li>
<li>Vertical tree</li>
<li>Horizontal tree</li>
<li>Grid (default)</li>
<li>Radial</li>
</ul></td>
</tr>
<tr>
<td> </td>
<td>The graphic view includes a <b>zoom level</b>. You can set the zoom level of the network by entering a percentage between 10 and 400.
<blockquote>Select the Graph view, press the <b>Ctrl</b> key and use the <b>mouse wheel</b> to zoom in/out.</blockquote></td>
</tr>
</table>

Table view

In this second view, the radio modules of the network are displayed as rows in a table. For ZigBee devices, the End Devices appear as children of their Coordinators or Routers. In other protocols, all nodes are at the same level.

The table has five columns:

- Role: Coordinator, Router, or End Device.
- MAC: The 64-bit address of the module.
- Network Address for ZigBee devices or Node Identifier for other networks.
- Last scan: the number of scan when the device was discovered for the last time.
- Connections: Click **Show connections** to see another table with information about its connections.

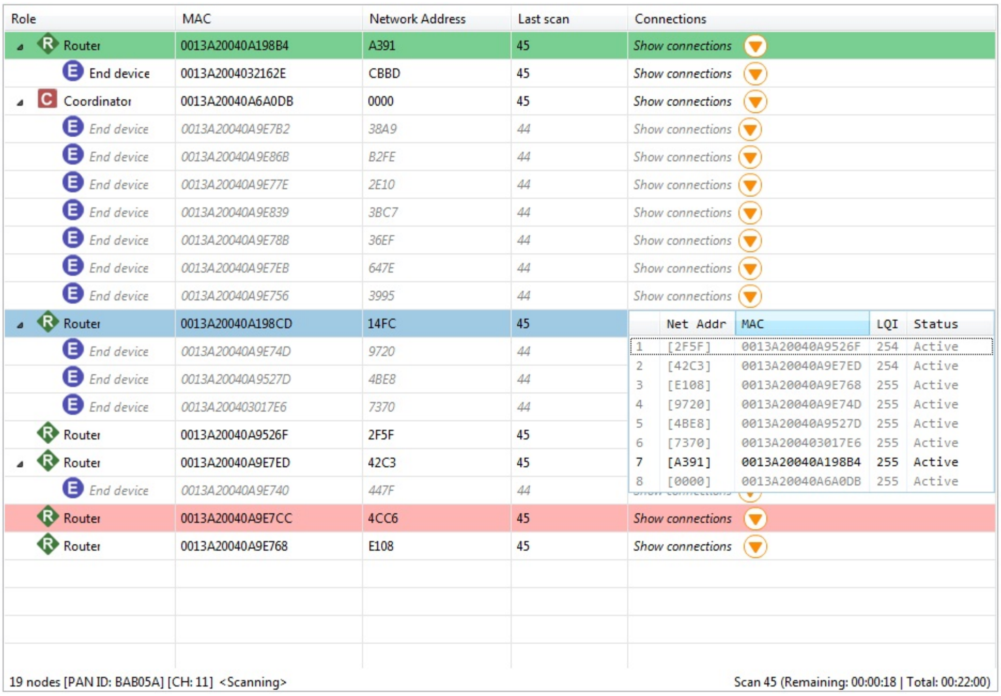


Role	MAC	Network Address	Last scan	Connections
Router	0013A20040A9526F	2F5F	1	*Show connections*
End device	0013A20040A9527D	CF1F	1	*Show connections*
End device	0013A20040A9E86B	0710	1	*Show connections*
End device	0013A20040A9E7B2	3BE5	1	*Show connections*
End device	0013A20040A9E839	F2C9	1	*Show connections*
End device	0013A20040A9E7EB	2E94	1	*Show connections*
Coordinator	0013A20040A6A0DB	0000	1	
Router	0013A20040A198CD	28ED	1	
Router	0013A20040A9E7CC	4CC6	1	
Router	0013A20040309132	7738	1	
Router	0013A20040A9E7ED	8315	1	
Router	0013A20040A198B4	91E4	1	*Show connections*
Router	0013A20040A9E768	E108	1	*Show connections*

	Net Addr	MAC	LQI	Status
1	[28ED]	0013A20040A198CD	243	Active
2	[4CC6]	0013A20040A9E7CC	255	Active
3	[8315]	0013A20040A9E7ED	255	Active
4	[91E4]	0013A20040A198B4	255	Active
5	[E108]	0013A20040A9E768	251	Active
6	[2F5F]	0013A20040A9526F	255	Active

In this case, nodes discovered in previous scans that have not already been found are displayed in grey and italics. The connections that are not discovered yet are also in grey.

Radio modules colored in red represent devices that are in the network (or have been) but are unreachable now.



Role	MAC	Network Address	Last scan	Connections
Router	0013A20040A198B4	A391	45	Show connections
End device	0013A2004032162E	CBBD	45	Show connections
Coordinator	0013A20040A6A0DB	0000	45	Show connections
End device	0013A20040A9E7B2	38A9	44	Show connections
End device	0013A20040A9E86B	B2FE	44	Show connections
End device	0013A20040A9E77E	2E10	44	Show connections
End device	0013A20040A9E839	3BC7	44	Show connections
End device	0013A20040A9E78B	36EF	44	Show connections
End device	0013A20040A9E7EB	647E	44	Show connections
End device	0013A20040A9E756	3995	44	Show connections
Router	0013A20040A198CD	14FC	45	
End device	0013A20040A9E74D	9720	44	
End device	0013A20040A9527D	4BE8	44	
End device	0013A200403017E6	7370	44	
Router	0013A20040A9526F	2F5F	45	
Router	0013A20040A9E7ED	42C3	45	
End device	0013A20040A9E740	447F	44	
Router	0013A20040A9E7CC	4CC6	45	Show connections
Router	0013A20040A9E768	E108	45	Show connections

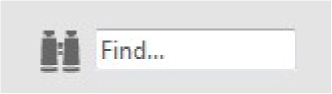
	Net Addr	MAC	LQI	Status
1	[2F5F]	0013A20040A9526F	254	Active
2	[42C3]	0013A20040A9E7ED	254	Active
3	[E108]	0013A20040A9E768	255	Active
4	[9720]	0013A20040A9E74D	255	Active
5	[4BE8]	0013A20040A9527D	255	Active
6	[7370]	0013A200403017E6	255	Active
7	[A391]	0013A20040A198B4	255	Active
8	[0000]	0013A20040A6A0DB	255	Active

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

Scan 45 (Remaining: 00:00:18 | Total: 00:22:00)

> In both views, the node with a green background corresponds to the local radio module, whereas the one with a blue background is the selected module.

Searching a node of the network



If you want to search a particular node of the network, you can use the **Find...** box. It allows you to use an expression to search for nodes.

For example, if you want to search for a module with node identifier (NI) "NODE1" and network address (MY) 0831, you could enter:

`NI:NODE1`, or
`MY:0831`.

The available search prefixes are:

- MAC: (or no prefix) to search by MAC address.
- SH: to search by Serial Number High.
- SL: to search by Serial Number Low.
- NI: to search by Node Identifier (only available in 802.15.4 and DigiMesh).
- MY: to search by 16-bit Network Address (only available in 802.15.4 and ZigBee).

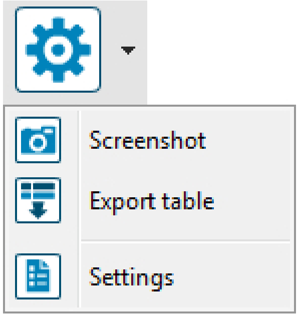
You can also use a wildcard if you do not want to write the entire parameter or you want to find more than one node. The available wildcards for search expressions are:

- * = any string
- ? = any character
- \ = escape for literals, i.e. *, ?, or \

For instance, if you enter `MAC:00*B` you will find all nodes whose MAC starts with 00 and ends with B.

The background color of the search box indicates status. Yellow indicates matches found by XCTU, and red indicates no matches.

Other features



XCTU also offers secondary features inside the **Options** drop-down menu:

- Screenshot
- Export table
- Settings

Taking a screenshot of the network

This function allows you to save an image of the local radio module's network to your computer.

Exporting the network table

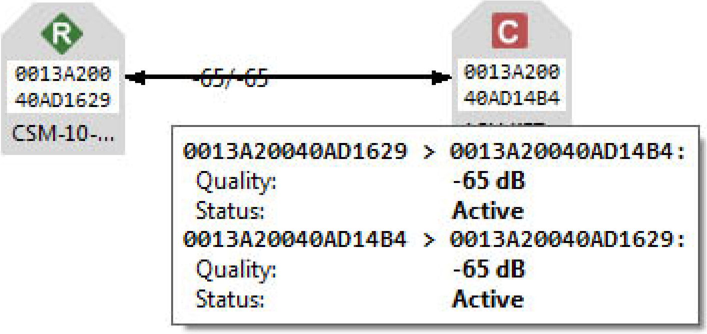
The Export table feature saves the network information in a text file using comma separated value format (csv) that can be opened with spreadsheet applications.

Configuring the Network preferences

For more information about the Network preferences, see the Network preferences topic.

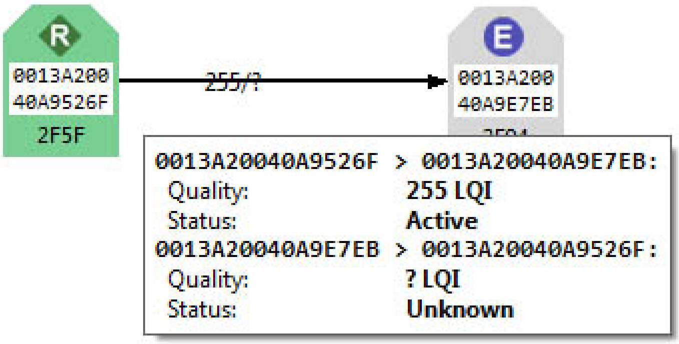
Connections in 802.15.4, DigiMesh, and ZigBee

- **802.15.4**: XCTU does not show the quality of the connection between two nodes. The nodes displayed are in the RF range of the local radio module.
- **DigiMesh**: The quality of the connection between two nodes is shown next to the line that connects them. It displays two numbers separated by a slash. You can hover your mouse cursor over the line that joins the nodes to see the quality and status of both directions of the connection. (If the connection is unidirectional, one of them displays '?' or 'Unknown' because it cannot be determined.) The quality unit is dB.



0013A20040AD1629 > 0013A20040AD14B4:
Quality: **-65 dB**
Status: **Active**
0013A20040AD14B4 > 0013A20040AD1629:
Quality: **-65 dB**
Status: **Active**

- **ZigBee**: The link quality for ZigBee is displayed the same way as it is displayed in DigiMesh, but the link quality is determined by Link Quality Indication, LQI (a number between 0, the worst, and 255, the best).

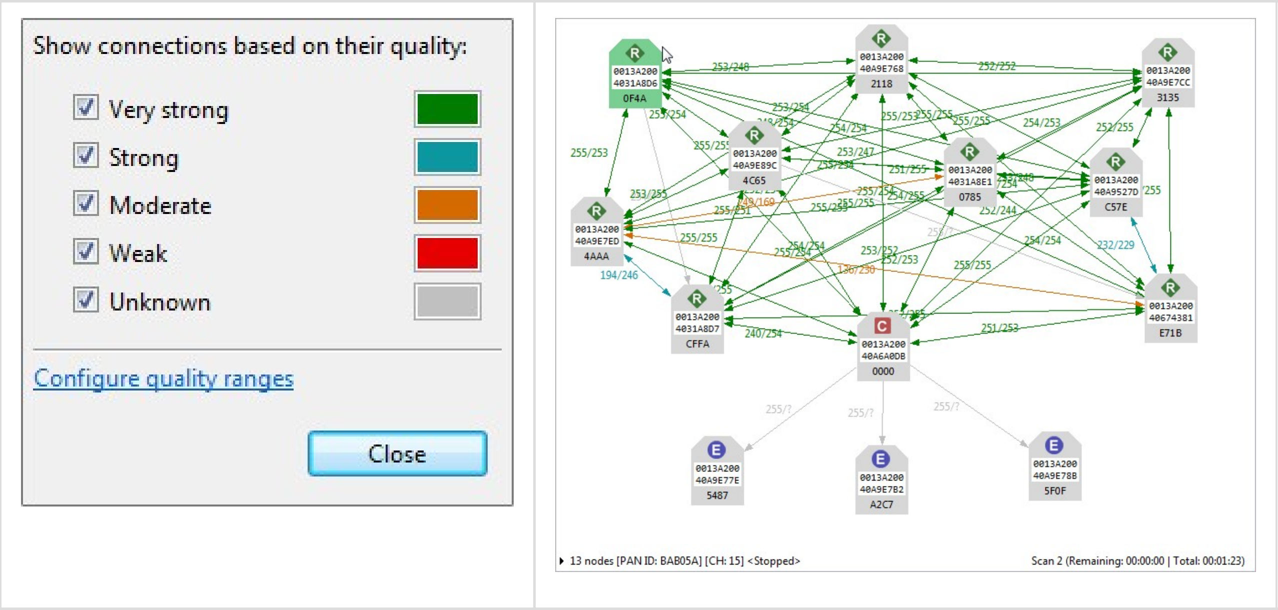


0013A20040A9526F > 0013A20040A9E7EB:
Quality: 255 LQI
Status: Active
0013A20040A9E7EB > 0013A20040A9526F:
Quality: ? LQI
Status: Unknown

Filtering connections by quality

XCTU allows you to hide and show connections between RF modules depending on their quality. The filters icon displays a dialog with all the connection quality filters that you can enable or disable.

By default, all the filters are enabled, thus all the connections are visible.



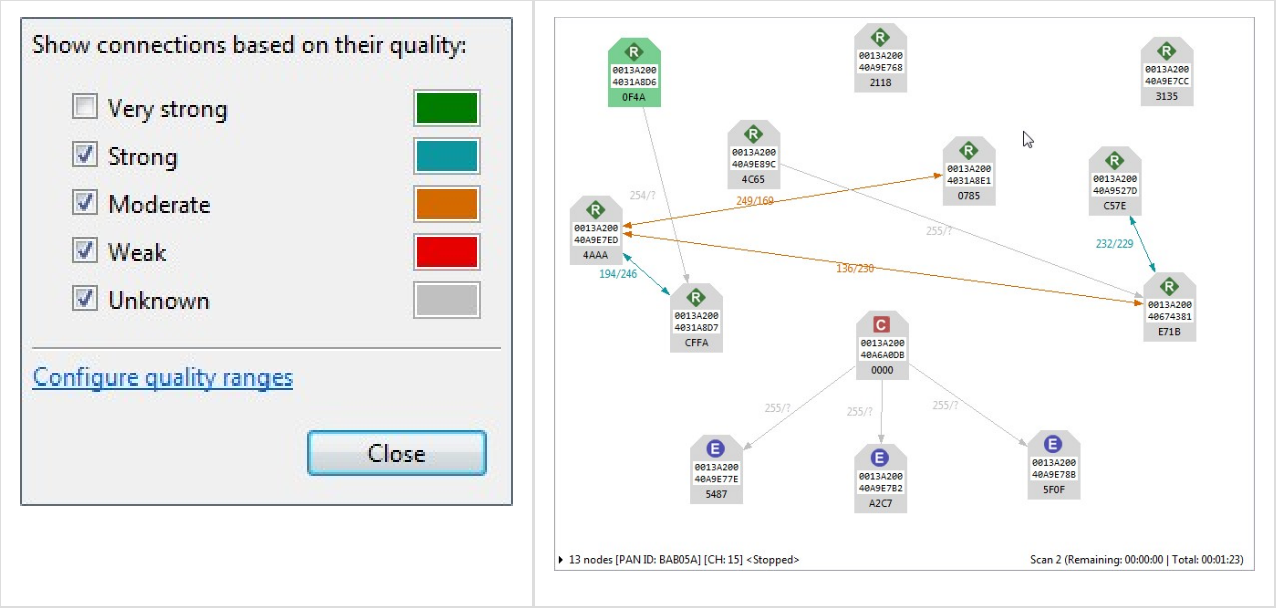
Show connections based on their quality:

- ☑ Very strong 🟩
- ☑ Strong 🟦
- ☑ Moderate 🟧
- ☑ Weak 🟥
- ☑ Unknown ⬜

13 nodes [PAN ID: BAB05A] [CH: 15] <Stopped>

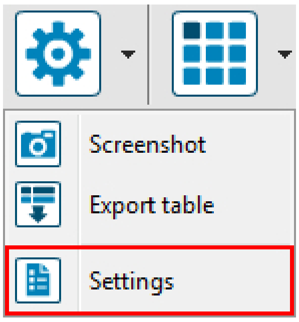
Scan 2 (Remaining: 00:00:00 | Total: 00:01:23)

Unchecking a filter hides all the connections contained in that quality filter.



Connection quality colors and ranges can be established in the **Network > Appearance** section of XCTU settings. You can access them directly by clicking the **Configure quality ranges** link of the **Quality ranges filtering** dialog.

You can also access them by clicking the **Settings** option of the **Tools** menu from the Network view toolbar.

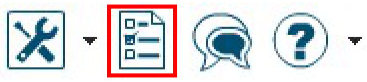


For more information about configuring the link quality ranges values and colors visit the Network appearance preferences section.

XCTU Configuration

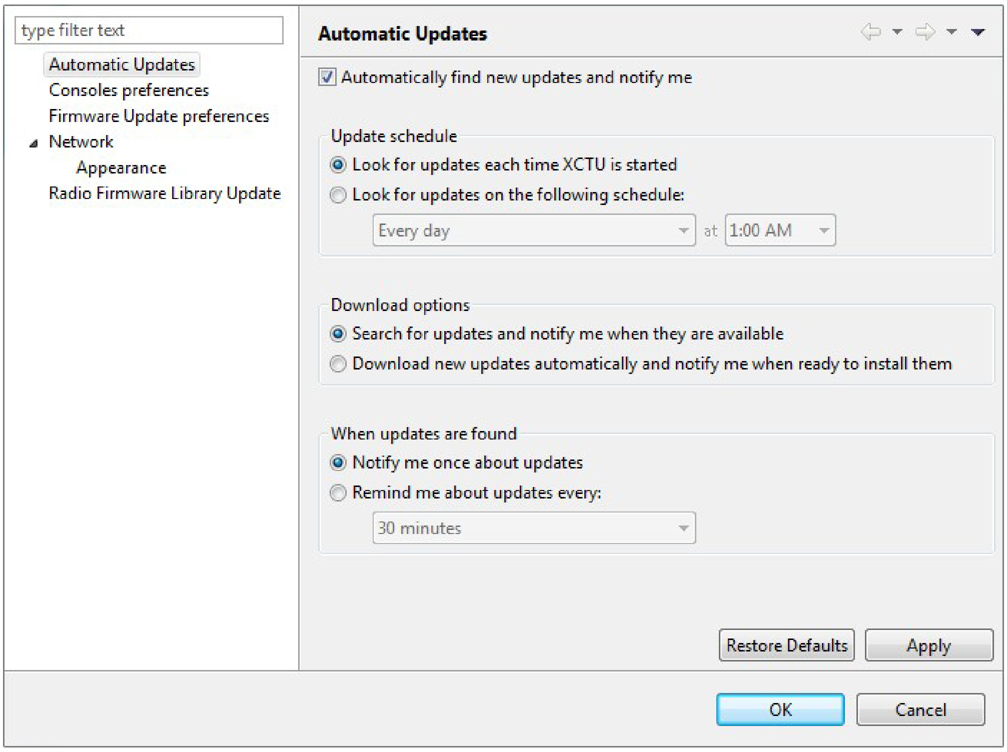
The configuration section of XCTU allows you to configure some settings of certain features of the application.

To open the XCTU configuration dialog box, click the **Preferences** button of the toolbar.



The configuration dialog appears.
The configuration preferences are split in categories listed at the left side of the configuration dialog box. Click a category to view the options. Currently, XCTU allows you to configure five kinds of preferences or categories:

- Automatic Updates preferences
- Consoles preferences
- Firmware update preferences
- Network preferences
- Radio Firmware Library Update preferences



Automatic Updates

☑ Automatically find new updates and notify me

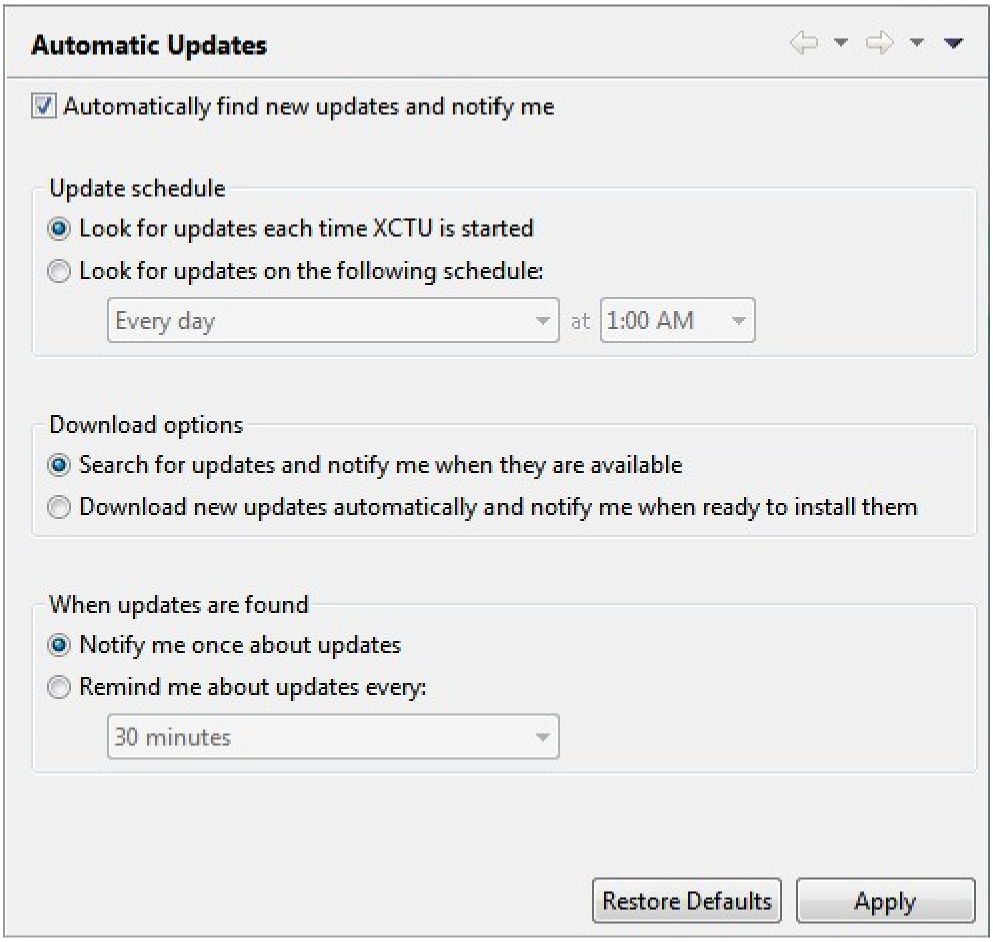
Update schedule
● Look for updates each time XCTU is started
○ Look for updates on the following schedule:
Every day at 1:00 AM

Download options
● Search for updates and notify me when they are available
○ Download new updates automatically and notify me when ready to install them

When updates are found
● Notify me once about updates
○ Remind me about updates every:
30 minutes

Automatic Updates preferences

In this category you can configure when and how new updates for XCTU should be downloaded and installed.



Automatic Updates

☑ Automatically find new updates and notify me

Update schedule
- ● Look for updates each time XCTU is started
- ○ Look for updates on the following schedule:
  Every day at 1:00 AM

Download options
- ● Search for updates and notify me when they are available
- ○ Download new updates automatically and notify me when ready to install them

When updates are found
- ● Notify me once about updates
- ○ Remind me about updates every:
  30 minutes

The first setting of the category enables or disables the automatic updates for XCTU. Un-check this setting if you do not want XCTU to update automatically.

If the **Automatic find new updates and notify me** setting is enabled, you can configure three groups of preferences:

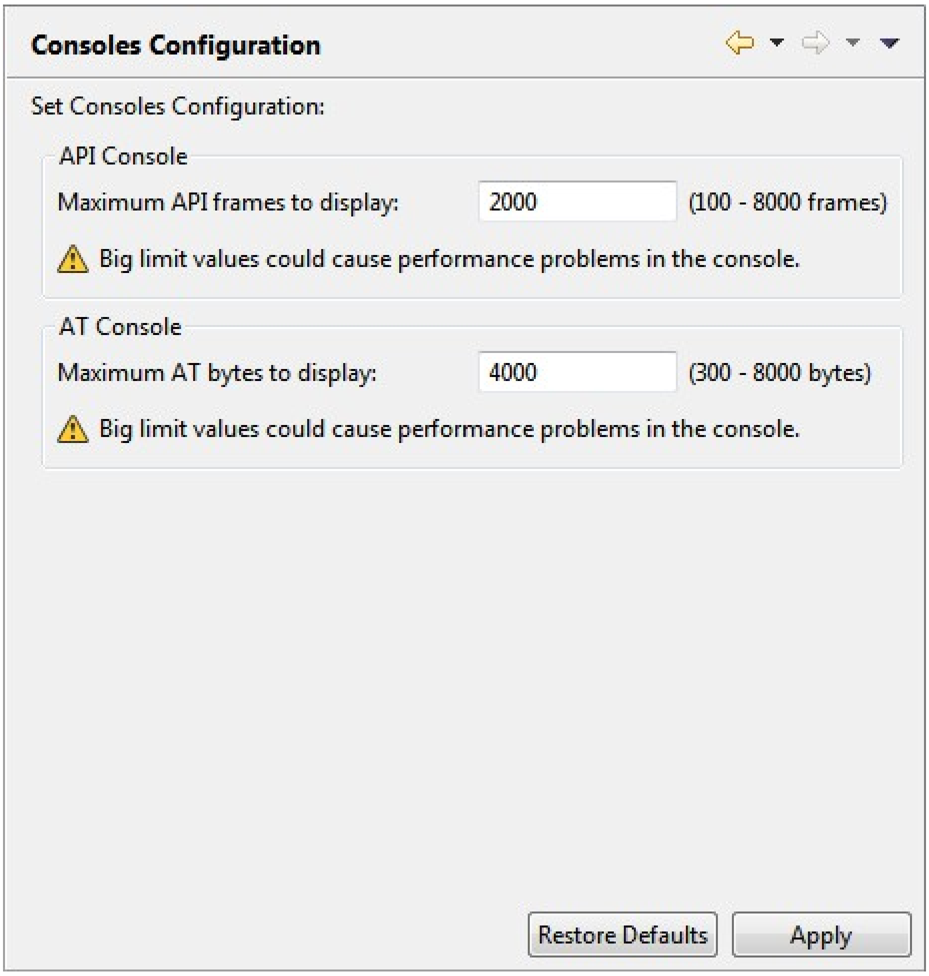
- **Update schedule**: In this group, you can set XCTU to search for new updates. You can set a schedule to search for updates or to update when XCTU is started.

  If you select the **Look for updates on the following schedule** option, you also need to specify the search interval and the hour to search for updates.
- **Download options**: This group establishes when the new updates should download. You can receive notification when new downloads are available, but not download them, or allow XCTU to download automatically and notify you when they are ready to install.
- **When updates are found**: This group allows you to configure the frequency XCTU should notify you about the new updates found. You can receive notification only one time, or periodically.

If the **Automatic find new updates and notify me** setting is disabled, you can only able XCTU manually. See the Updating the XCTU tool topic for more information about updating XCTU.

Consoles preferences

In the Consoles preferences category you can configure the settings related to XCTU consoles.



Maximum API frames to display: 2000 (100 - 8000 frames)

⚠ Big limit values could cause performance problems in the console.

Maximum AT bytes to display: 4000 (300 - 8000 bytes)

⚠ Big limit values could cause performance problems in the console.

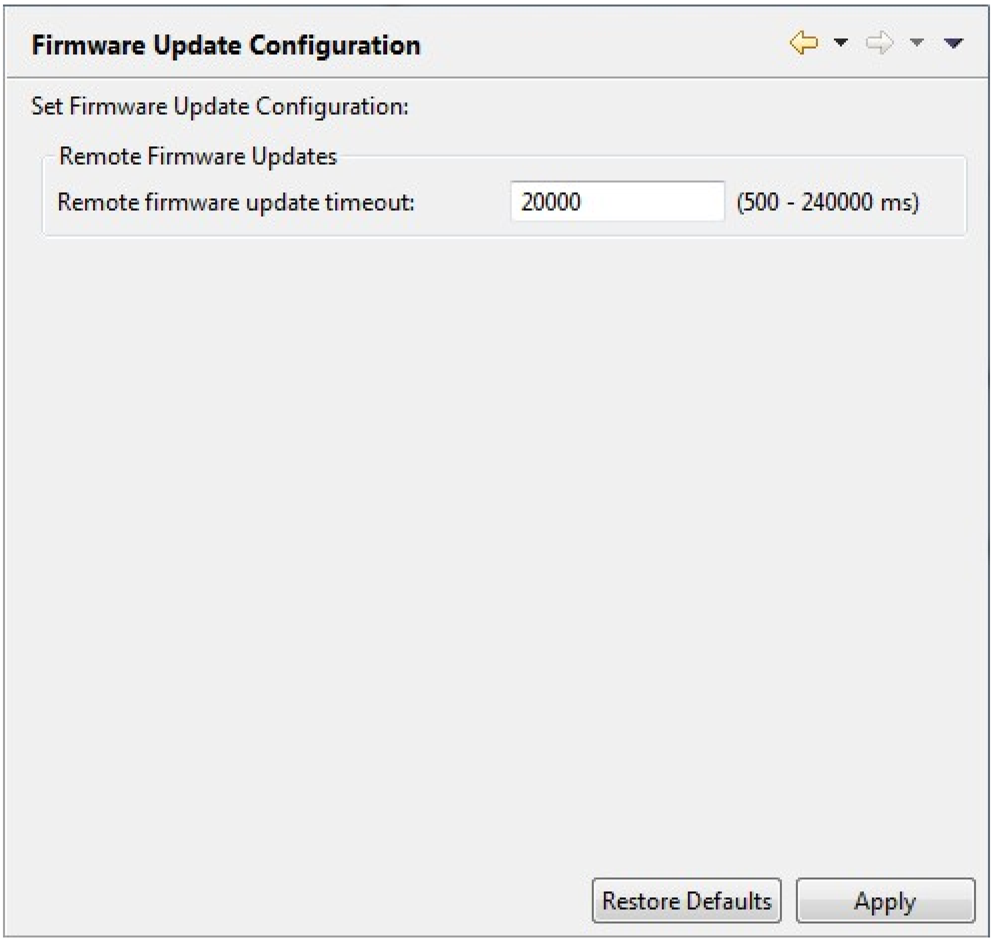
This category is divided into two groups:

- **API Console**: This group allows you to configure the maximum number of API frames that can be stored and displayed in the frames log during a session. When the maximum limit is reached, the session starts overwriting frames.
- **AT Console**: In this group you have to configure the maximum number of bytes that can be stored and displayed during a session. When the maximum limit is reached, the session starts overwriting bytes.

> If you configure the maximum limits with high values, you may notice some performance problems in the consoles.

Firmware update preferences

These preferences allow you to configure firmware update parameters.



Firmware Update Configuration
Set Firmware Update Configuration:
Remote Firmware Updates

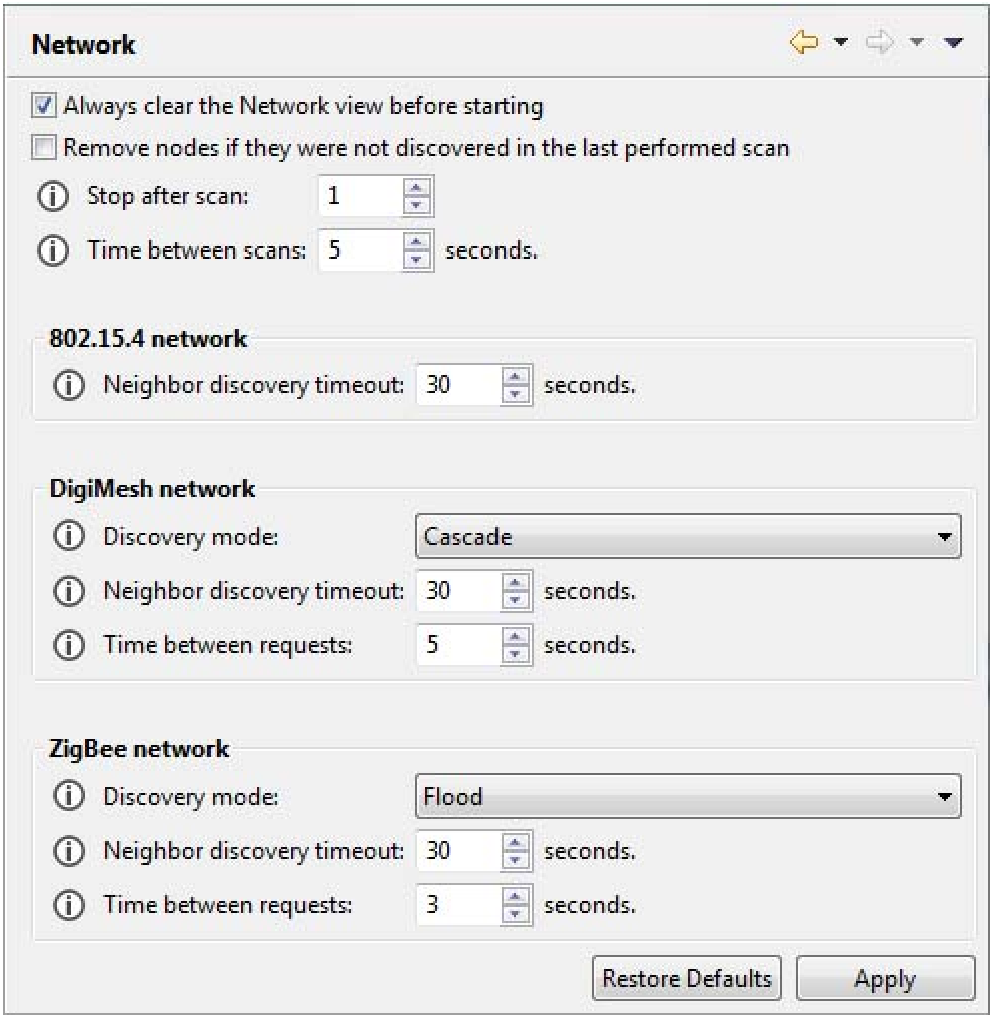
Remote firmware update timeout: 20000 (500 - 240000 ms)

The category has just one section:

- **Remote Firmware Update Timeout:** You can configure 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 the remote firmware update before considering that there was an error during the process.

Network preferences

You can configure all the options related to the network view in the Network preferences. The first set of options are common to all networks; the rest are for specific network types.



Network

☑ Always clear the Network view before starting

☐ Remove nodes if they were not discovered in the last performed scan

Stop after scan: 1

Time between scans: 5 seconds.

802.15.4 network

Neighbor discovery timeout: 30 seconds.

DigiMesh network

Discovery mode: Cascade

Neighbor discovery timeout: 30 seconds.

Time between requests: 5 seconds.

ZigBee network

Discovery mode: Flood

Neighbor discovery timeout: 30 seconds.

Time between requests: 3 seconds.

The first group includes common preferences for all types of network:

- **Always clear the Network view before starting**: If this setting is enabled, every time you start a new network scan the view is cleared.
- **Remove nodes if they were not discovered in the last performed scan**: If this setting is enabled, the nodes that were not discovered in the last scan are removed.
- **Stop after scan**: Number of scans to perform before automatically stopping the discovery process. '0' means the process does not stop automatically.
- **Time between scans**: Time to wait before starting a new network scan. It must be between 0 and 300 seconds (5 minutes) inclusive.

The rest of these preferences are separated according to the type of network: 802.15.4, DigiMesh, or ZigBee.

- **Discovery mode**: Method used by the network discovery process.
  - *Flood*: The neighbor discovery process is performed for every node at the moment it is found. Note that this means several discovery processes might be running at the same time. This might be a faster method, but it may generate a lot of traffic and saturate the network.
  - *Cascade*: The neighbor discovery process is performed for every node as soon as the discovery process finishes. Note that this means only one discovery process runs at the same time. This might be a slower method, but it is likely to generate less traffic.

    > **Cascade** discovery method is **recommended for large networks**.

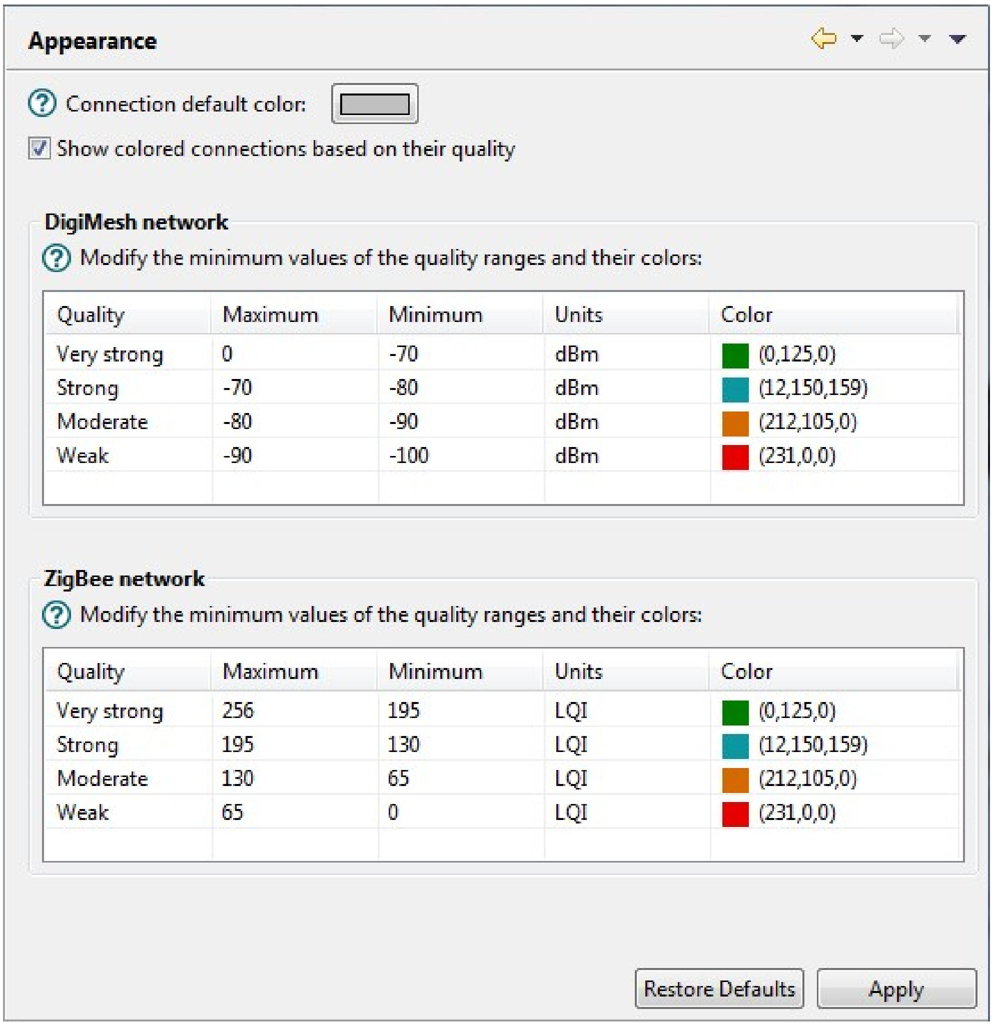
- **Neighbor discovery timeout**: The maximum duration, in seconds, the discovery process should spend finding neighbors of a module. It must be between 5 and 1800 seconds (30 minutes) inclusive.

This timeout is highly dependent on the nature of the network. For DigiMesh, it 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.

- **Time between requests**: The time to wait between node neighbors requests. It must be between 0 and 300 seconds (5 minutes) inclusive. For *Cascade*, this is the number of seconds to wait after completion of the neighbor discovery process of the previous node. For *Flood*, this time is the minimum time to wait between each radio module's neighbor requests.

Appearance

The Network preferences have a sub-category named **Appearance** where you can configure how the node links are represented in the Network Graphic View:



Appearance

Connection default color:

☑ Show colored connections based on their quality

DigiMesh network

Modify the minimum values of the quality ranges and their colors:

Quality	Maximum	Minimum	Units	Color
Very strong	0	-70	dBm	🟩 (0,125,0)
Strong	-70	-80	dBm	🟦 (12,150,159)
Moderate	-80	-90	dBm	🟧 (212,105,0)
Weak	-90	-100	dBm	🟥 (231,0,0)

ZigBee network

Modify the minimum values of the quality ranges and their colors:

Quality	Maximum	Minimum	Units	Color
Very strong	256	195	LQI	🟩 (0,125,0)
Strong	195	130	LQI	🟦 (12,150,159)
Moderate	130	65	LQI	🟧 (212,105,0)
Weak	65	0	LQI	🟥 (231,0,0)

The first section defines common preferences for all the network types:

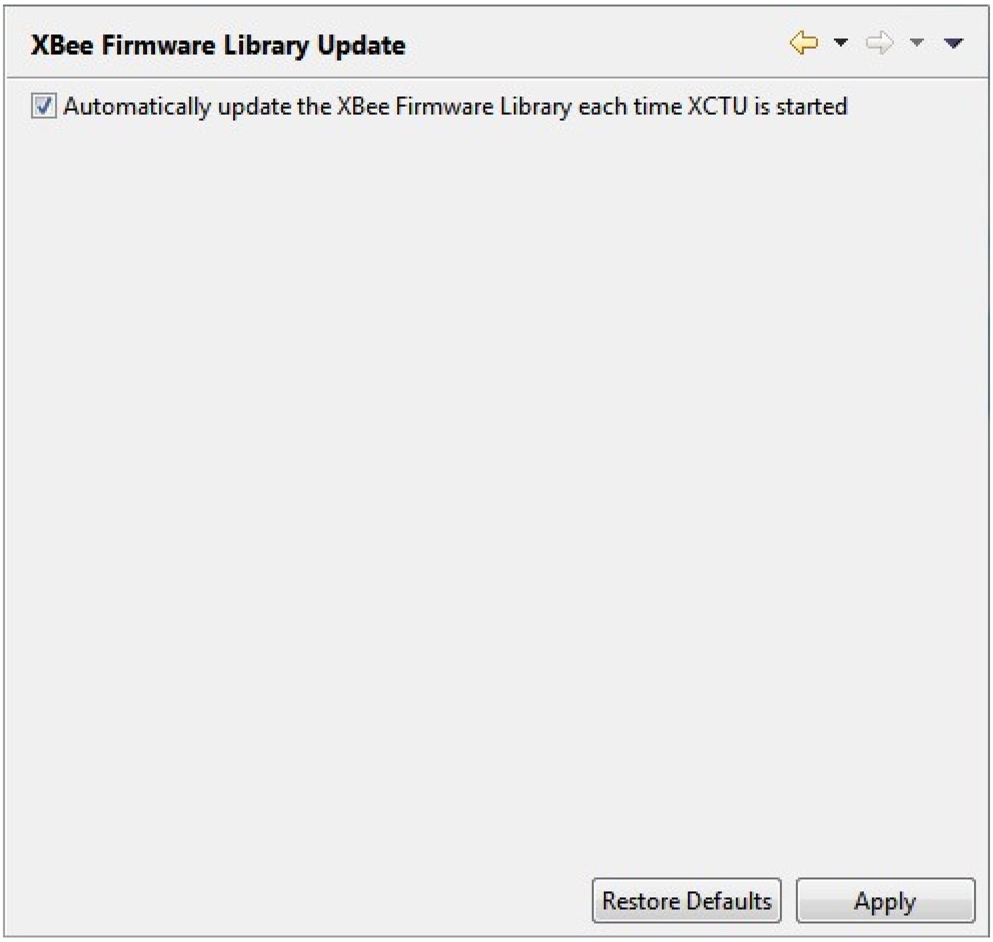
- **Connection default color**: Defines the default color to draw the node's connection lines.
- **Show colored connections based on their quality:** Enables or disables the coloring of the node connection lines based on their link quality.

The rest of the preferences are separated according to the type of network: DigiMesh or ZigBee. The table displays the four quality range categories with their limits and the color that will be used to represent each one. For each quality range you can modify its maximum and minimum values as well as its color by clicking in the corresponding cell and pressing **<ENTER>** after the edition.

> Notice that when you change the minimum value of a quality range, the maximum value of the next range adopts that value. Also, note that minimum values are contained in the ranges but the maximum aren't.

Radio Firmware Library Update preferences

You can instruct XCTU to look for new radio firmware when it starts up by checking the **Automatically update the XBee Firmware Library each time XCTU is started** option.



☑ Automatically update the XBee Firmware Library each time XCTU is started

If the option is disabled, you will be only able to check for new radio firmware manually. See the Updating the radio firmware topic for more information about updating the radio firmware library.

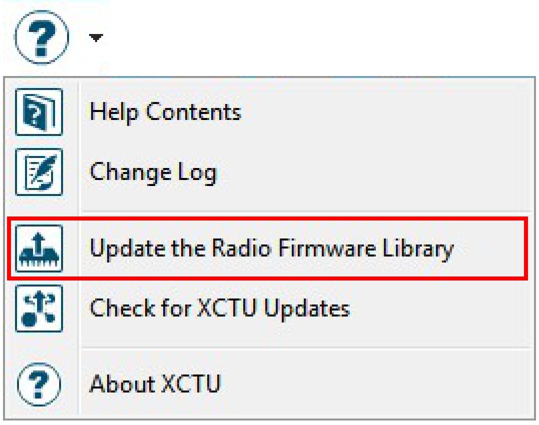
Software Updates

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. See the XCTU Configuration topic for more information about configuring automatic XCTU and radio firmware library updates.

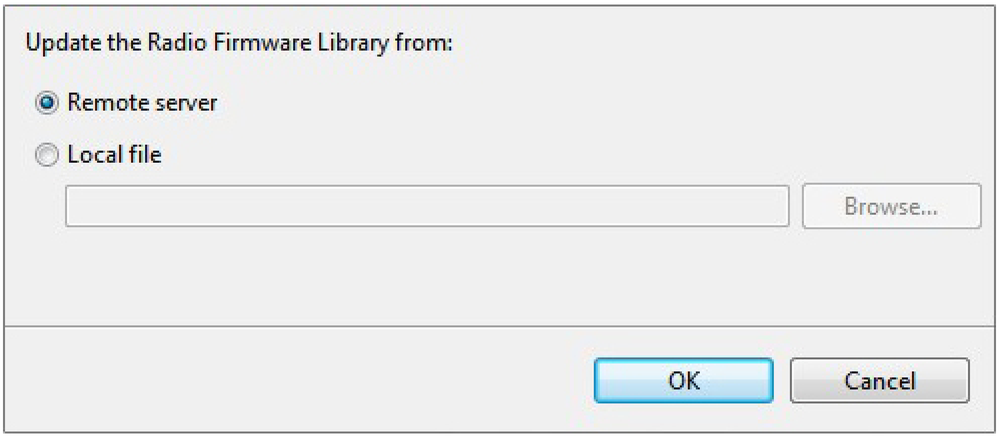
Updating the radio firmware library

Digi periodically releases new versions of the radio firmware to fix bugs or improve functionality. 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.

This process can be configured to be executed automatically (see the Radio Firmware Library Update preferences topic for more information); it can also be executed manually. To do so, go to the help drop-down menu of the toolbar and select the **Update Radio Rirmware Library** option.

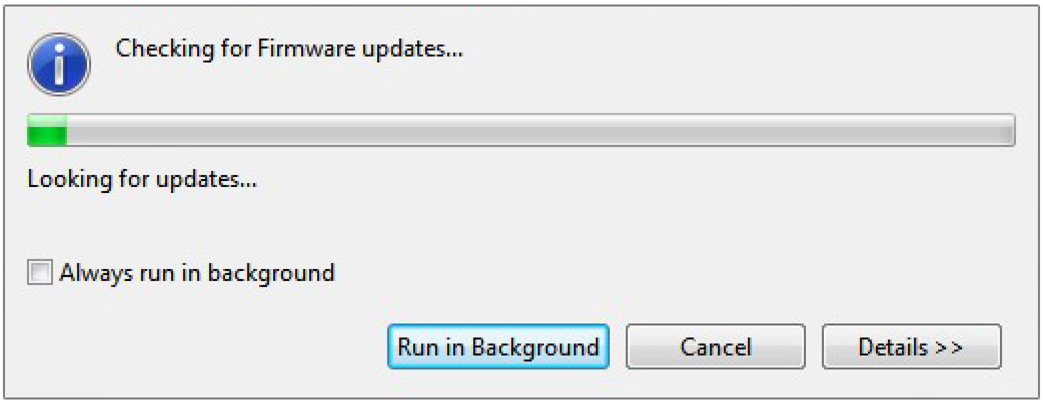


A new dialog allows you to configure the source to look for new radio firmware. This source can be Digi's update server or a local zip file containing the radio firmware.



In cases where you download firmware from Digi's support page, you can choose the **Local file** option, specifying the path of the zip file containing the firmware. Otherwise you should always choose the **Remote server** option, which connects to Digi's update site and looks for new firmware.

Click **OK** to continue with the radio firmware library update. A new progress dialog displays the progress.

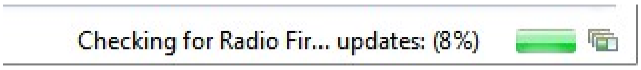


Checking for Firmware updates...

Looking for updates...

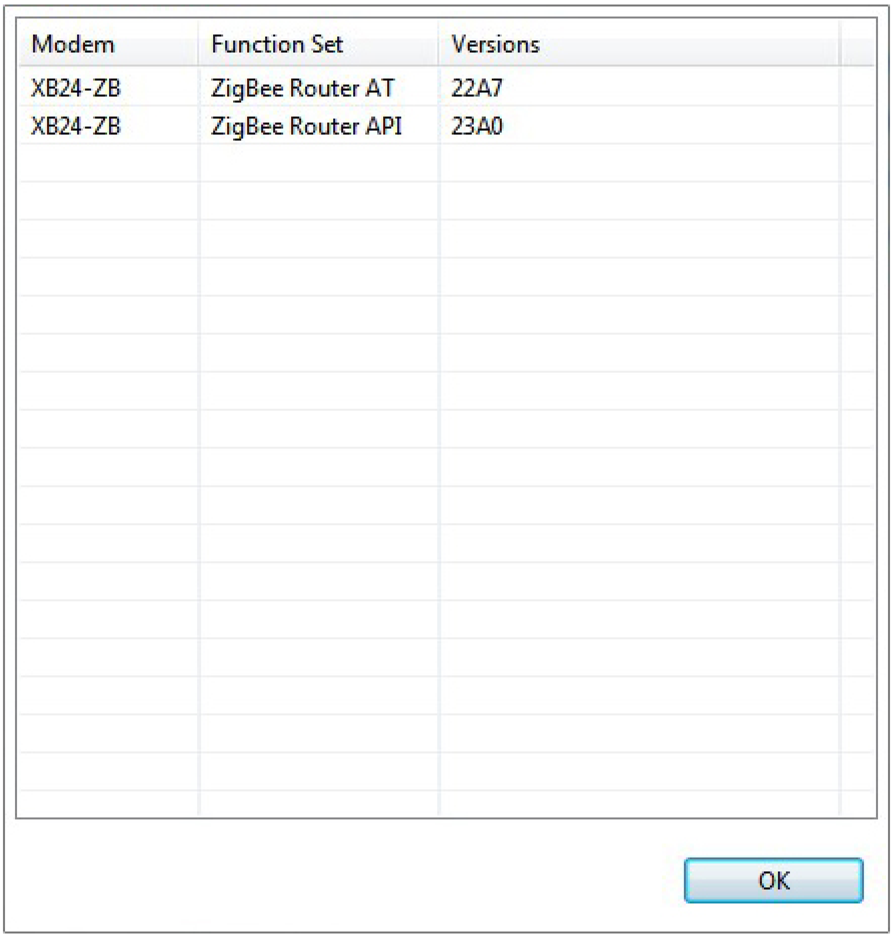
You can click the **Run in background** button to execute this process in the background. This allows you to keep working with XCTU while the new firmware is downloaded.

While the update radio firmware library process is taking place, the status bar displays the progress of the task.



Checking for Radio Fir... updates: (8%)

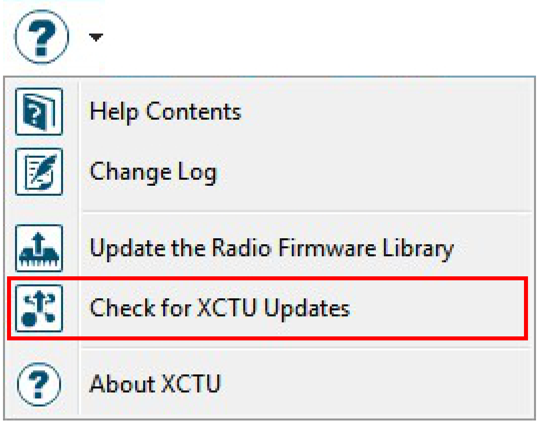
When the process finishes, if new radio firmware has been downloaded, a new dialog box displays the list of downloaded firmware.



Modem	Function Set	Versions
XB24-ZB	ZigBee Router AT	22A7
XB24-ZB	ZigBee Router API	23A0

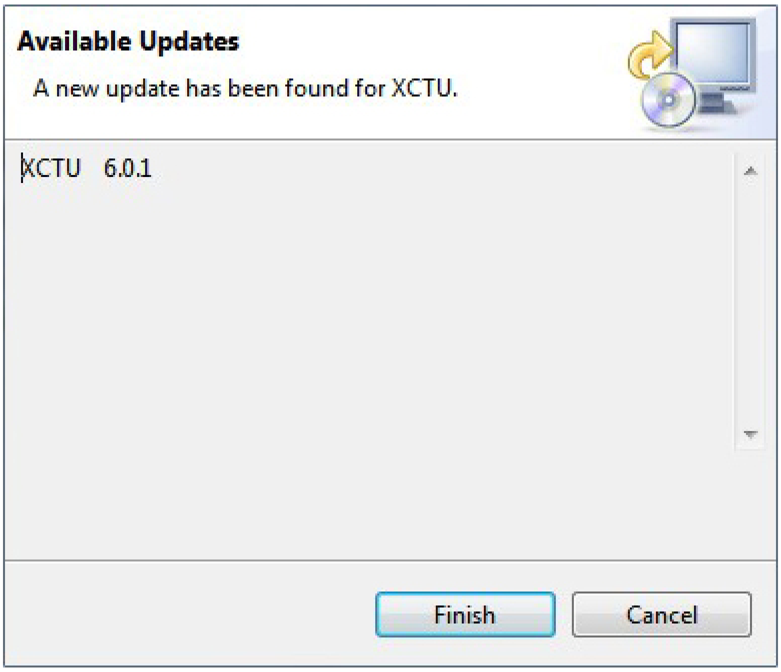
Updating the XCTU tool

You can configure XCTU updates to be executed manually or automatically (see the Automatic Updates preferences topic for more information). To update manually, select the **Check for XCTU Updates** option from the help dropdown in the toolbar.



The update application process displays the progress of the operation in the status bar.

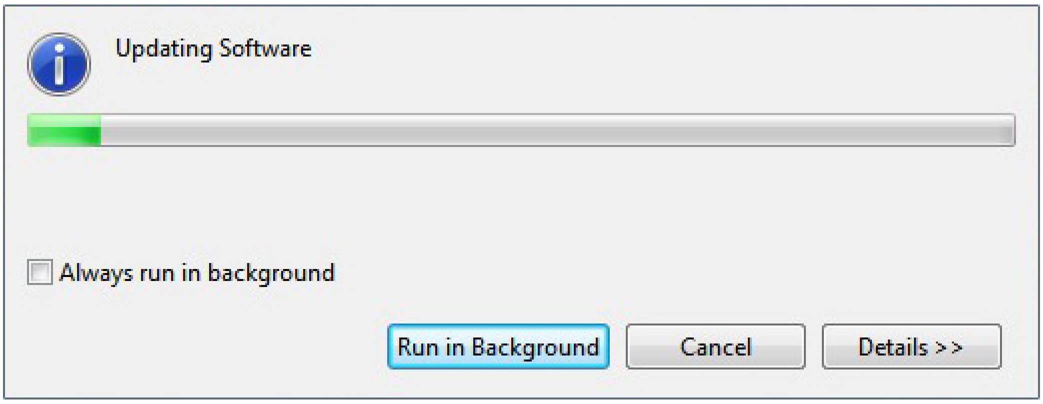
If an update for the application is found, a dialog box displays the update.


Available Updates

A new update has been found for XCTU.

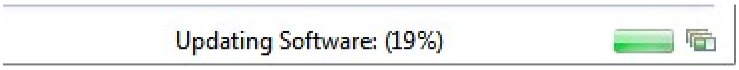
XCTU 6.0.1

Click **Finish** to continue. A new dialog box displays the update progress.


Updating Software

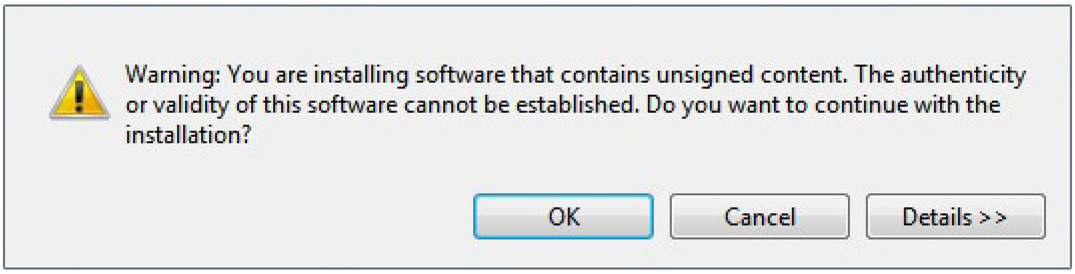
☐ Always run in background

You can click the **Run in Background** button to execute the update process in background mode. The status bar displays the update progress.



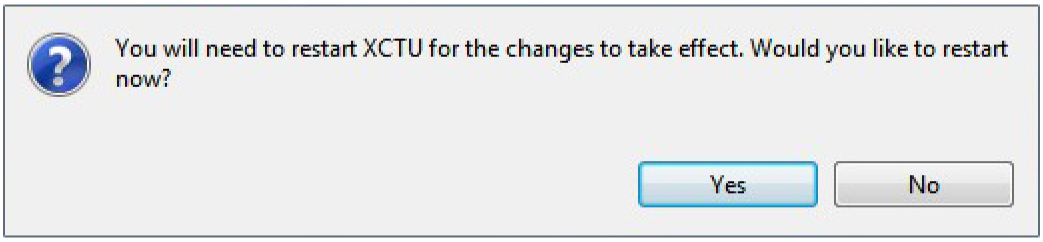
Updating Software: (19%)

If you are asked about installing unsigned software click **OK** to continue.



Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?

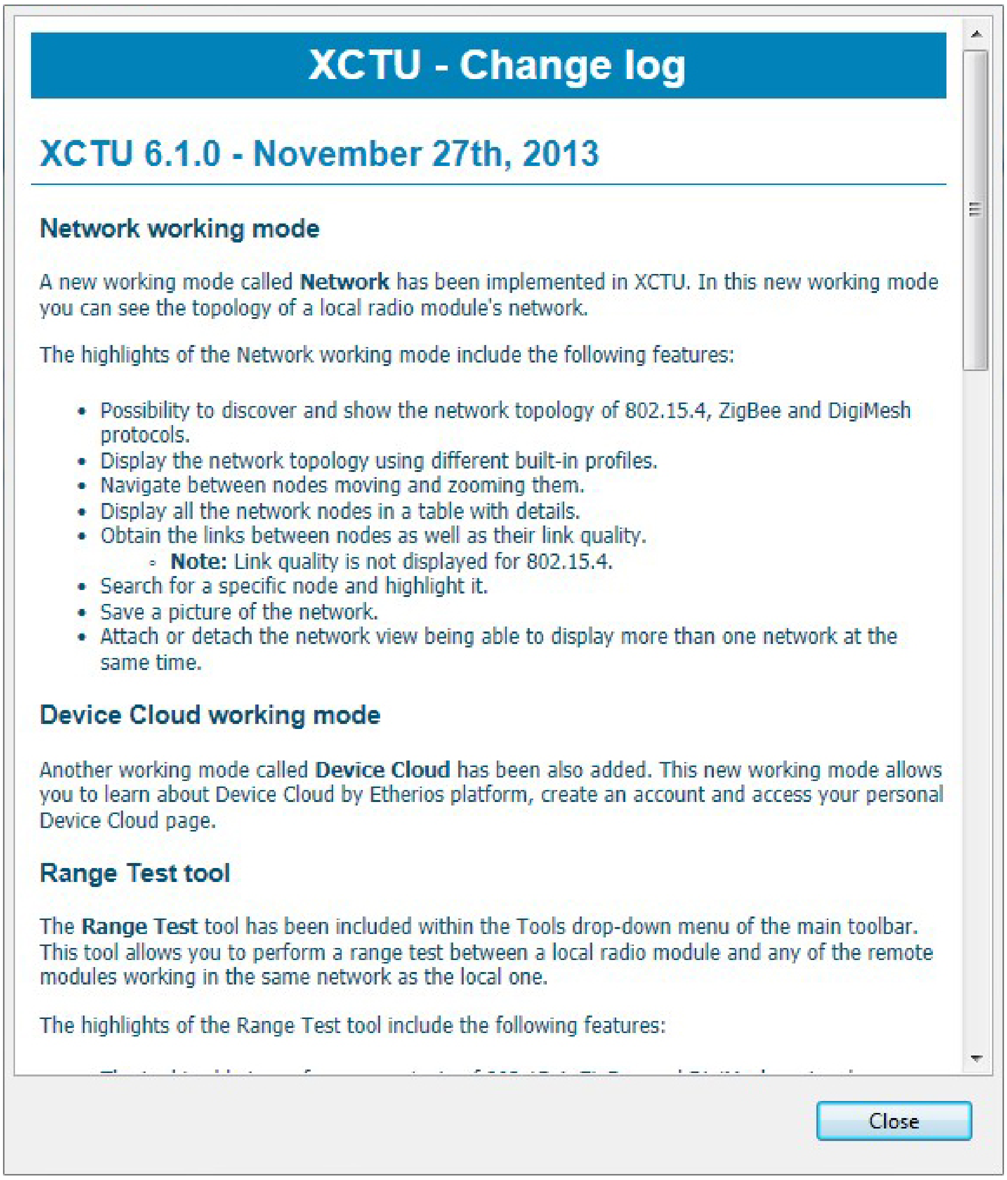
When the process finishes, you need to reset XCTU so new changes can be applied.



You will need to restart XCTU for the changes to take effect. Would you like to restart now?

Click **Yes** and wait for the application to reset.

Once XCTU has started again, a change log dialog displays the changes included in the new version of XCTU.



XCTU - Change log

XCTU 6.1.0 - November 27th, 2013

Network working mode

A new working mode called **Network** has been implemented in XCTU. In this new working mode you can see the topology of a local radio module's network.

The highlights of the Network working mode include the following features:

- Possibility to discover and show the network topology of 802.15.4, ZigBee and DigiMesh protocols.
- Display the network topology using different built-in profiles.
- Navigate between nodes moving and zooming them.
- Display all the network nodes in a table with details.
- Obtain the links between nodes as well as their link quality.
    - **Note:** Link quality is not displayed for 802.15.4.
- Search for a specific node and highlight it.
- Save a picture of the network.
- Attach or detach the network view being able to display more than one network at the same time.

Device Cloud working mode

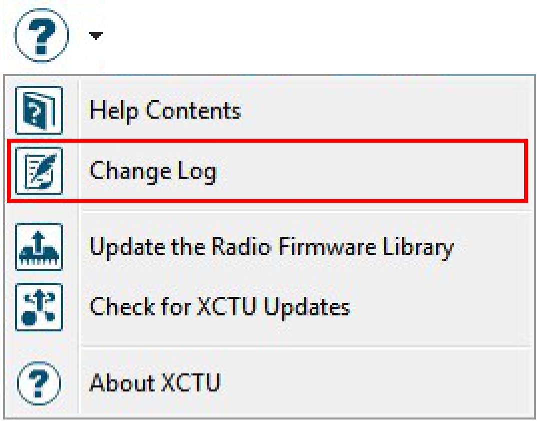
Another working mode called **Device Cloud** has been also added. This new working mode allows you to learn about Device Cloud by Etherios platform, create an account and access your personal Device Cloud page.

Range Test tool

The **Range Test** tool has been included within the Tools drop-down menu of the main toolbar. This tool allows you to perform a range test between a local radio module and any of the remote modules working in the same network as the local one.

The highlights of the Range Test tool include the following features:

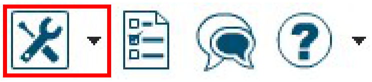
This change log is also accessible from the help drop-down menu of the toolbar.



XCTU tools

XCTU includes a set of embedded tools that can be executed at any time, regardless of the active working mode, and without adding a radio module to the list of devices.

You can access the following tools from the **Tools** drop-down menu of the main toolbar:

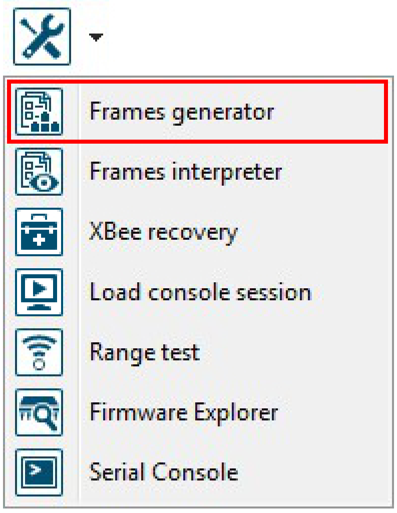


- Frames generator tool
- Frames interpreter tool
- Recovery tool
- Load console session tool
- Range test tool
- Firmware Explorer tool
- Serial Console tool

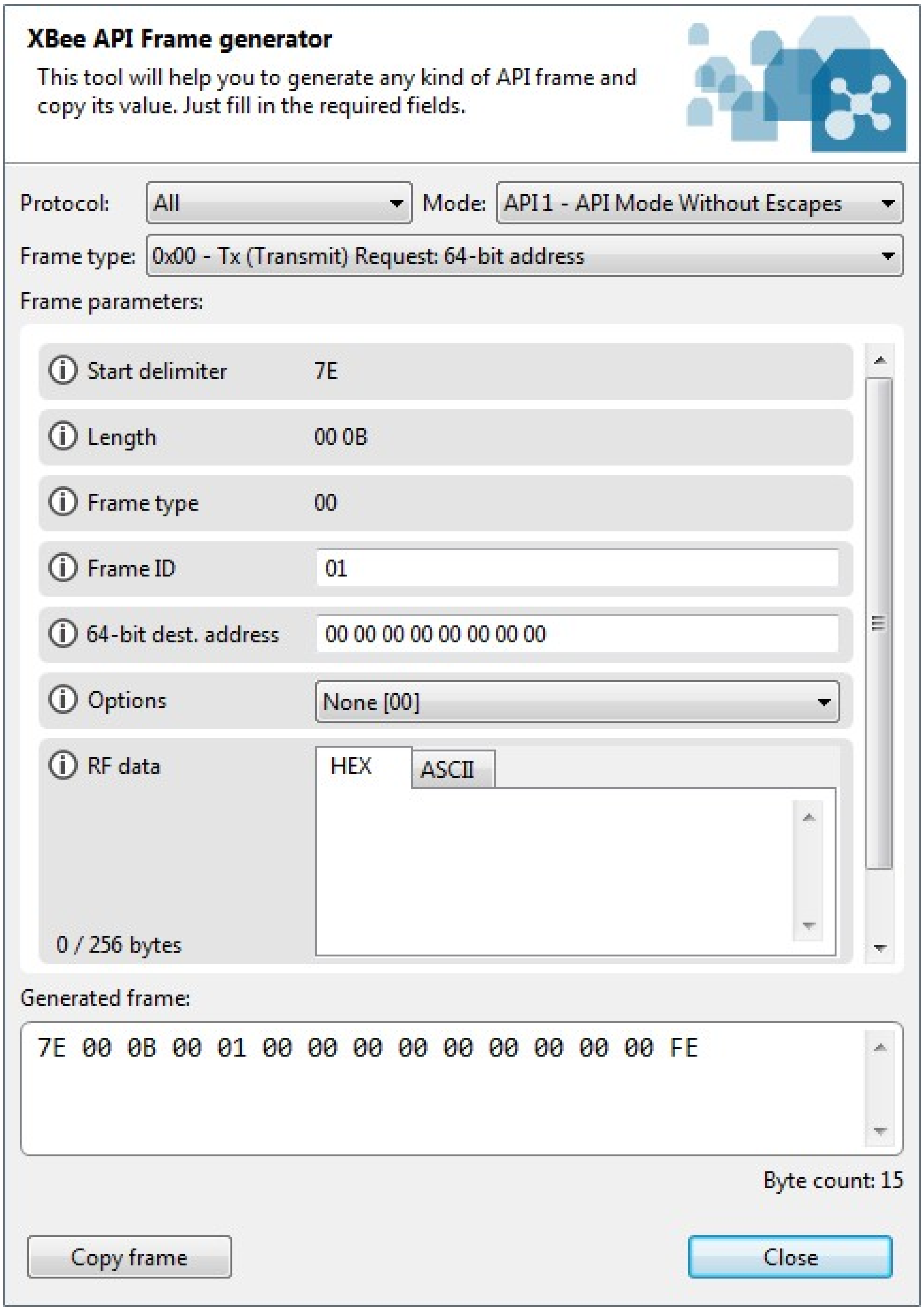
Frames generator tool

The Frames Generator utility is a tool that comes embedded within XCTU. It generates any type of API frame (output or input frames), obtaining the array of bytes that compose the frame. To learn more about API frames see the API frames topic.

Once XCTU is started, you can access the Frames Generator tool without adding a radio module to the list of devices. To open the Frames Generator tool, go to the **Tools** drop-down menu from the main toolbar and select the **Frames Generator** option.



The tool opens in a new floating dialog box.



XBee API Frame generator

This tool will help you to generate any kind of API frame and copy its value. Just fill in the required fields.

Protocol: All

Mode: API 1 - API Mode Without Escapes

Frame type: 0x00 - Tx (Transmit) Request: 64-bit address

Frame parameters:

Start delimiter	7E
Length	00 0B
Frame type	00
Frame ID	01
64-bit dest. address	00 00 00 00 00 00 00 00
Options	None [00]
RF data 0 / 256 bytes	

Generated frame:

```
7E 00 0B 00 01 00 00 00 00 00 00 00 00 00 FE
```

Byte count: 15

The API frame types are filtered by protocol. Select the radio protocol to display the API frames corresponding to the selected protocol. If you select **All** in the protocol, all the frame types are displayed.

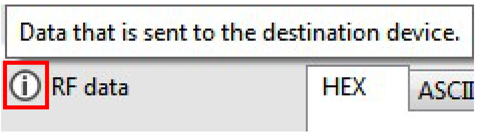
You must configure the API mode (API or API Escaped) of the frame to generate. You can select the mode using the combo box control of the **Mode** field.

> To learn more about these modes refer to the API operating mode and API escaped operating mode topics in the Concepts and terminology section.

Select the API frame type you want to generate.

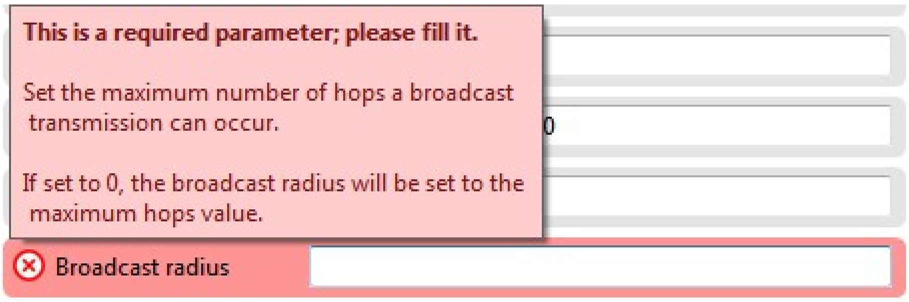
The selected frame's settings appear in the frame parameters list. In this example we have selected the **Zig Bee** protocol and the **ZigBee Transmit Request** API frame type.

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



Data that is sent to the destination device.

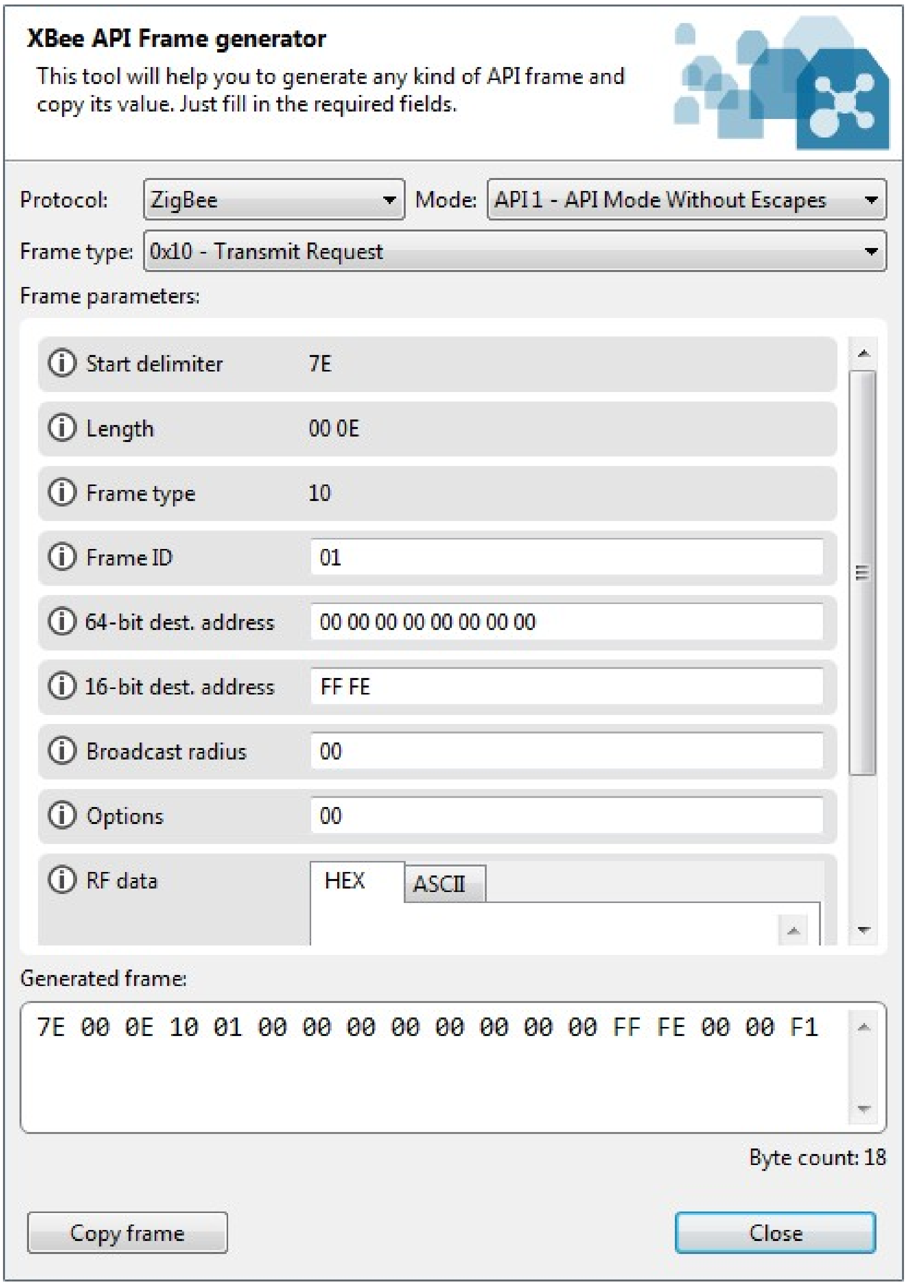
If any setting is not configured correctly, its background color changes to red and the information icon is replaced by a red cross. Hover over the icon to display the error message.



This is a required parameter; please fill it.

Set the maximum number of hops a broadcast transmission can occur.

If set to 0, the broadcast radius will be set to the maximum hops value.



XBee API Frame generator

This tool will help you to generate any kind of API frame and copy its value. Just fill in the required fields.

Protocol: ZigBee

Mode: API 1 - API Mode Without Escapes

Frame type: 0x10 - Transmit Request

Frame parameters:

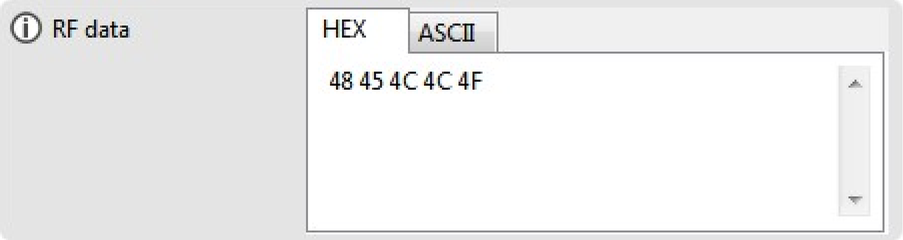
Start delimiter	7E
Length	00 0E
Frame type	10
Frame ID	01
64-bit dest. address	00 00 00 00 00 00 00 00
16-bit dest. address	FF FE
Broadcast radius	00
Options	00
RF data	

Generated frame:

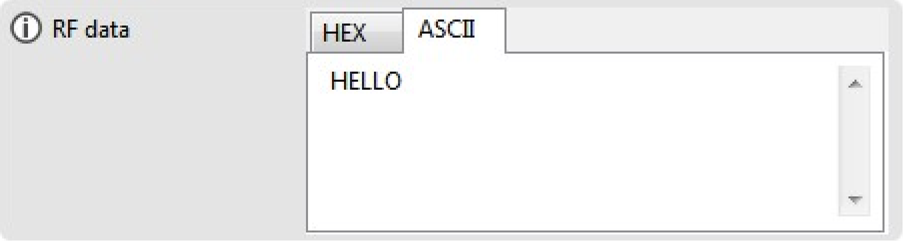
```
7E 00 0E 10 01 00 00 00 00 00 00 00 00 FF FE 00 00 F1
```

Byte count: 18

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.

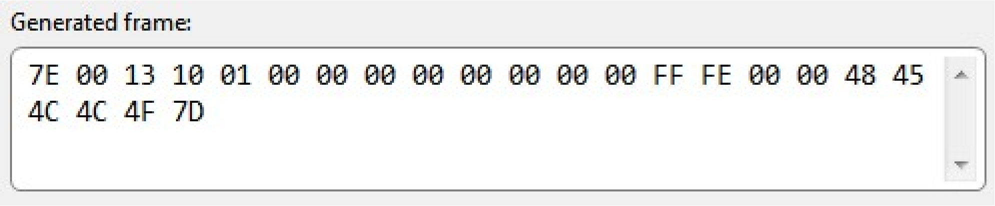


48 45 4C 4C 4F



HELLO

When all the settings are filled in 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 there or by clicking the **Copy frame** button.



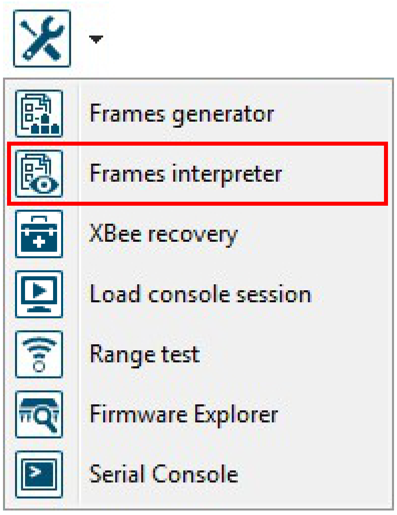
```
7E 00 13 10 01 00 00 00 00 00 00 00 00 FF FE 00 00 48 45
4C 4C 4F 7D
```

> The Frames Generator tool can be also accessed from the API console, when you add a new API frame to the list of frames to send. When this happens, the tool includes an **OK** button.
>
> You can use the OK button to automatically copy the generated byte array to the content of the frame to be added.

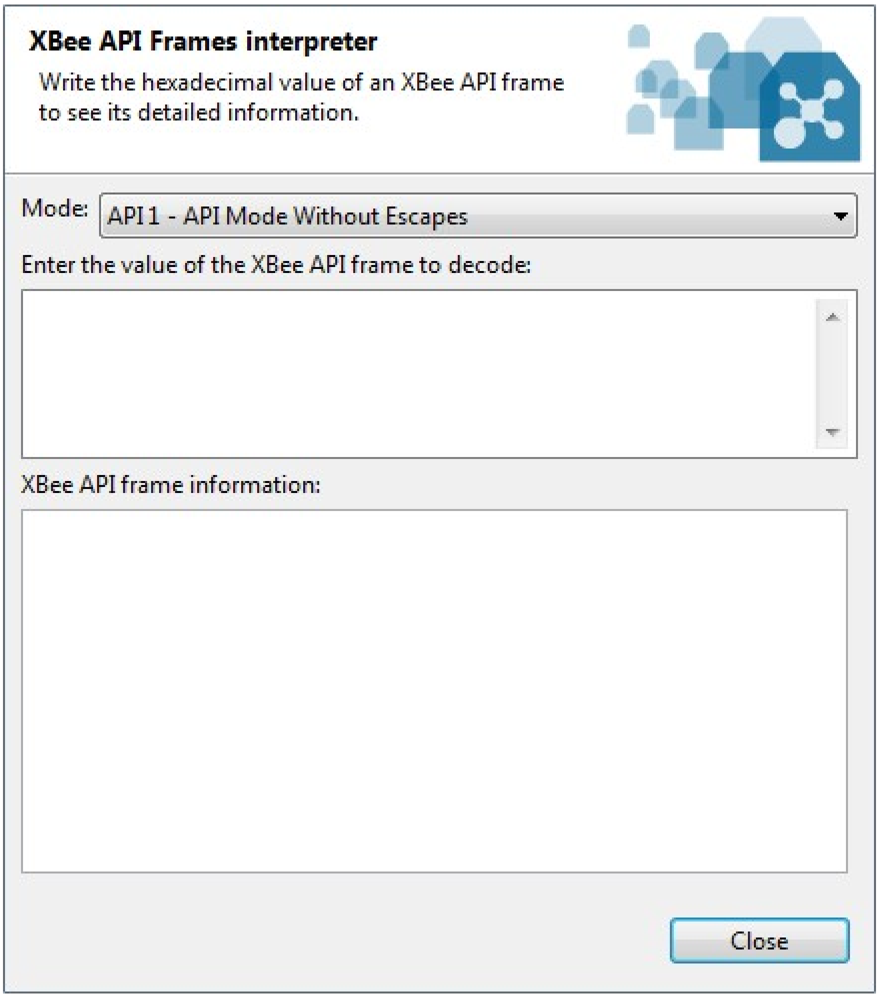
Frames interpreter tool

The Frames interpreter tool, which is embedded in XCTU, 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. If you don't know what an API frame is, see the API frames topic.

To open the Frames interpreter tool, select the **Frames Interpreter** option from the **Tools** drop-down menu in the main toolbar.



The tool opens in a new floating dialog box.



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:

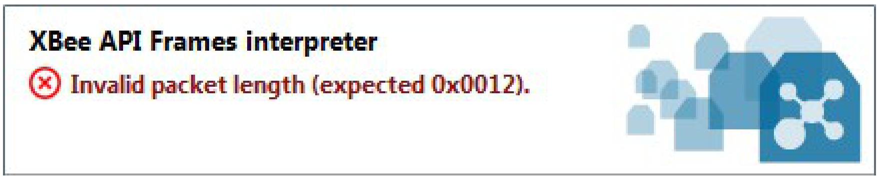
XBee API frame information:

The Frames interpreter allows you to choose the API mode (API or API Escaped) of the frame before decoding it. You can select the mode using the combo box control of the **Mode** field.

> To learn more about these modes refer to the API operating mode and API escaped operating mode topics of the Concepts and terminology section.

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.

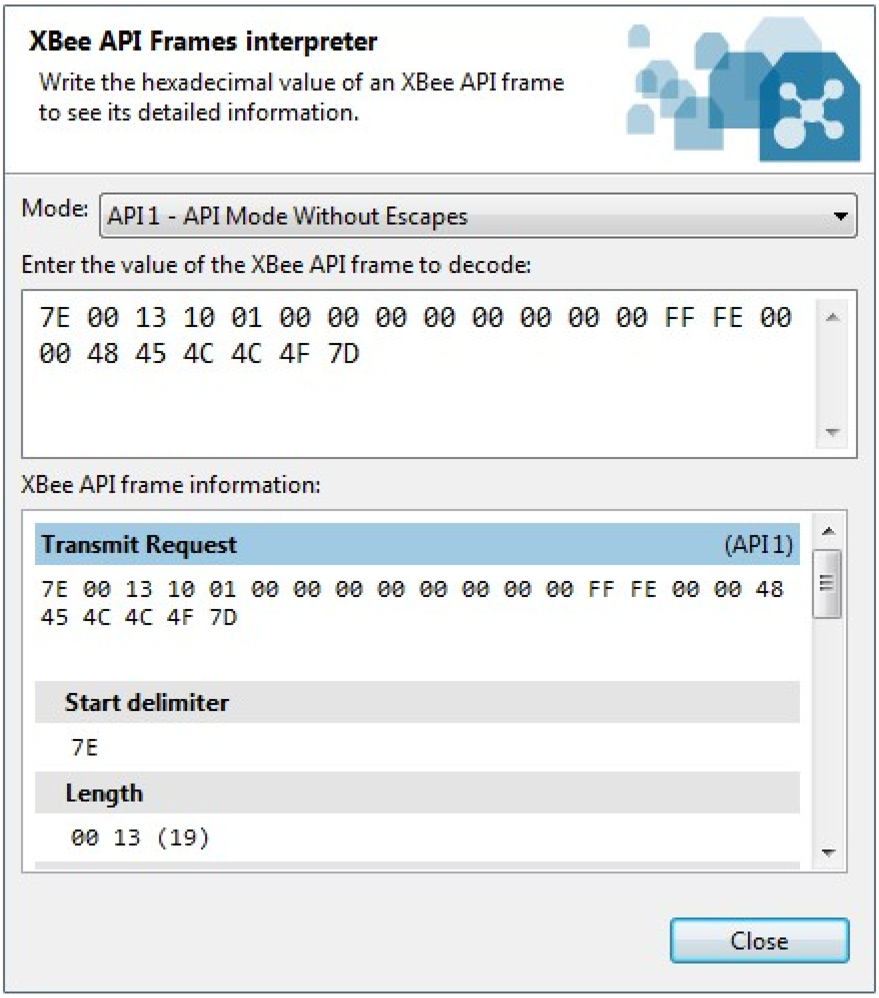
If the byte array is not valid, the tool displays the corresponding error at the top.



XBee API Frames interpreter

Invalid packet length (expected 0x0012).

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 FF FE 00
00 48 45 4C 4C 4F 7D
```

XBee API frame information:

Transmit Request (API 1)

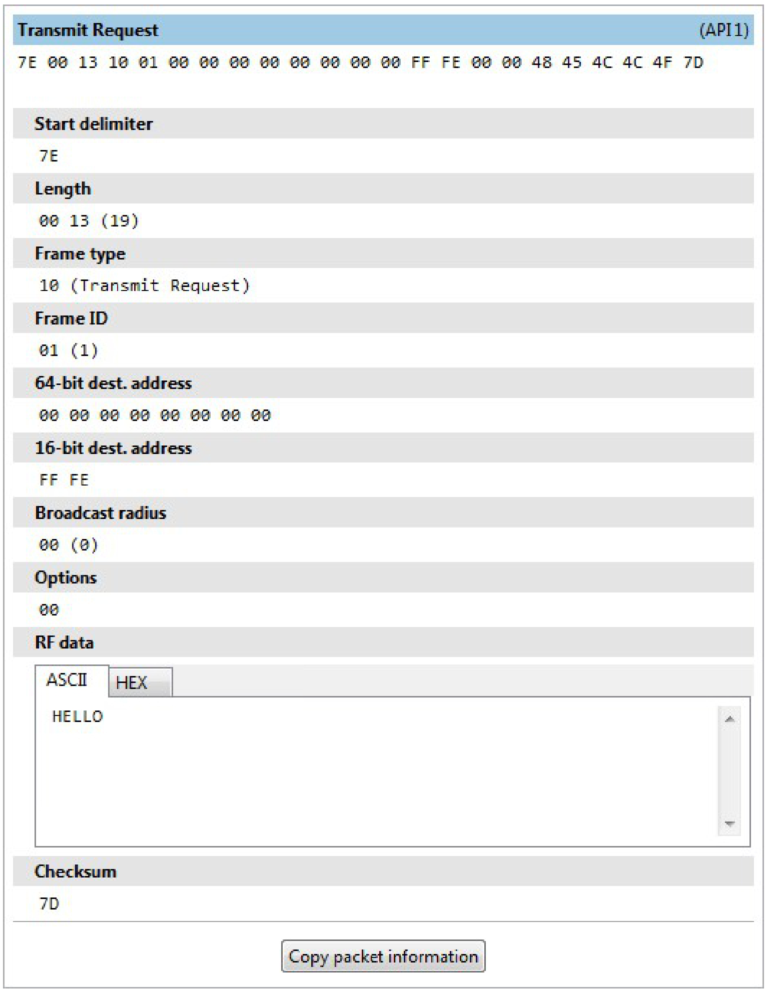
```
7E 00 13 10 01 00 00 00 00 00 00 00 00 FF FE 00 00 48
45 4C 4C 4F 7D
```

Start delimiter

7E

Length

00 13 (19)



Transmit Request (API1)

```
7E 00 13 10 01 00 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

HELLO

Checksum

7D

The decoded information includes the type of the API frame with all the 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.

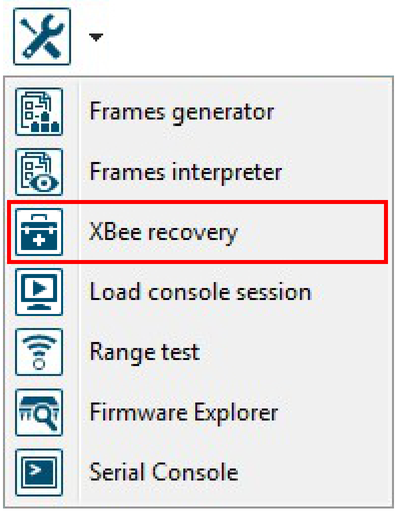
The **Copy packet information** button copies the packet information to the clipboard in plain text.

Recovery tool

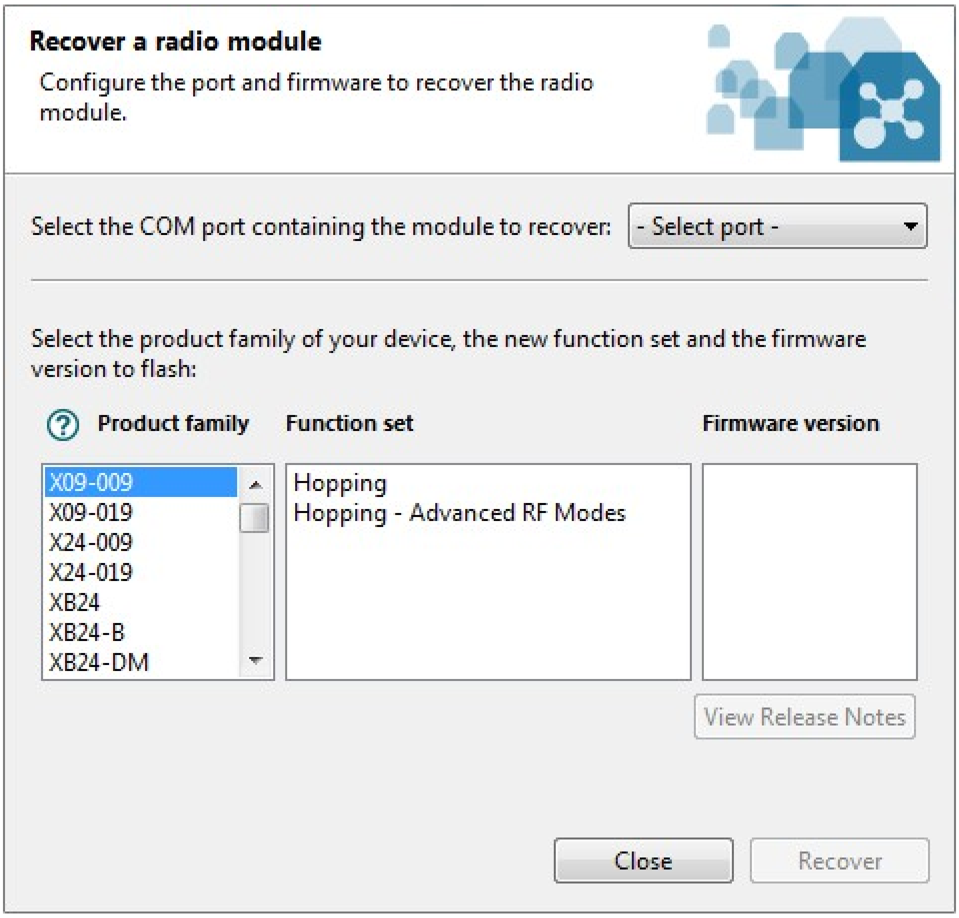
The purpose of the Recovery tool is to force a firmware update to recover radio modules that have damaged firmware or are in programming mode.

> Due to the internal architecture, the programmable XBee radio modules and XLR PRO Radio Solution do not support the recovery feature provided by XCTU.

To open the Recovery tool, select the **XBee recovery** option from the **Tools** drop-down menu in the main toolbar.

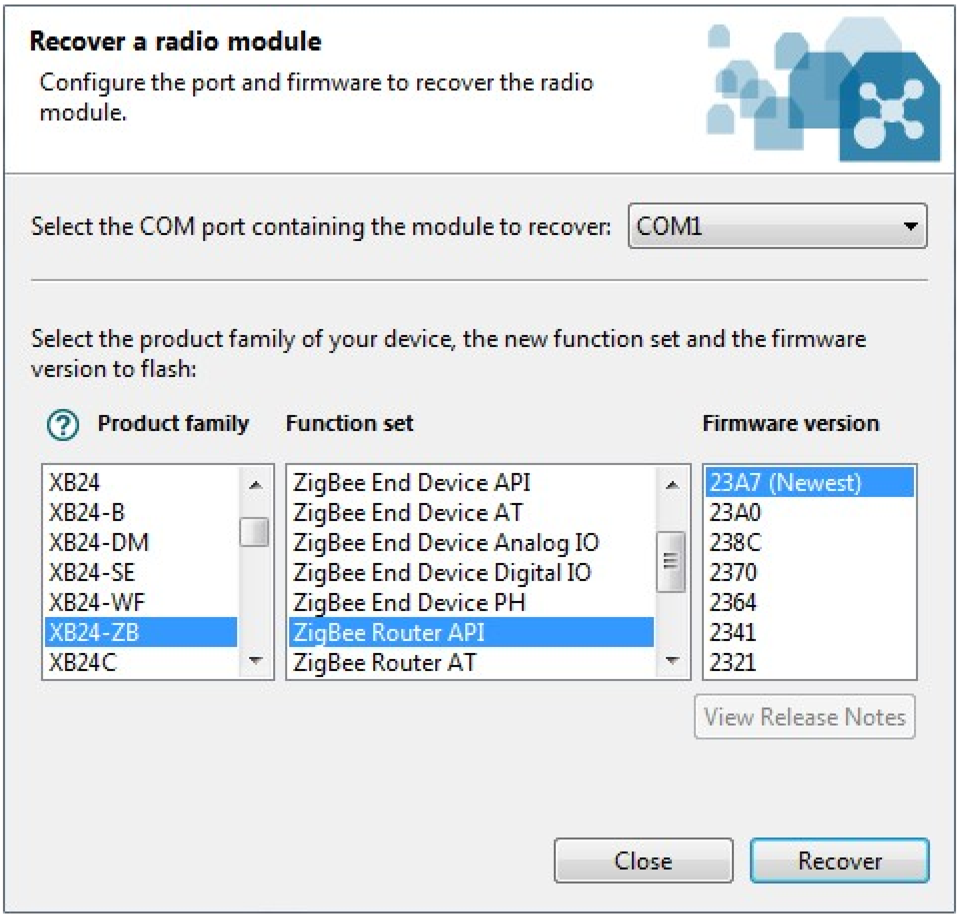


The tool opens in a new floating dialog box.


Recover a radio module
Configure the port and firmware to recover the radio module.
Select the COM port containing the module to recover:
- Select port -
Select the product family of your device, the new function set and the firmware version to flash:
Product family
Function set
Firmware version
X09-009
X09-019
X24-009
X24-019
XB24
XB24-B
XB24-DM
Hopping
Hopping - Advanced RF Modes
View Release Notes
Close
Recover

First, select the serial port where the damaged radio module is attached.

Then select the firmware you want to use, specifying its family, function and version from the corresponding lists.



Recover a radio module

Configure the port and firmware to recover the radio module.

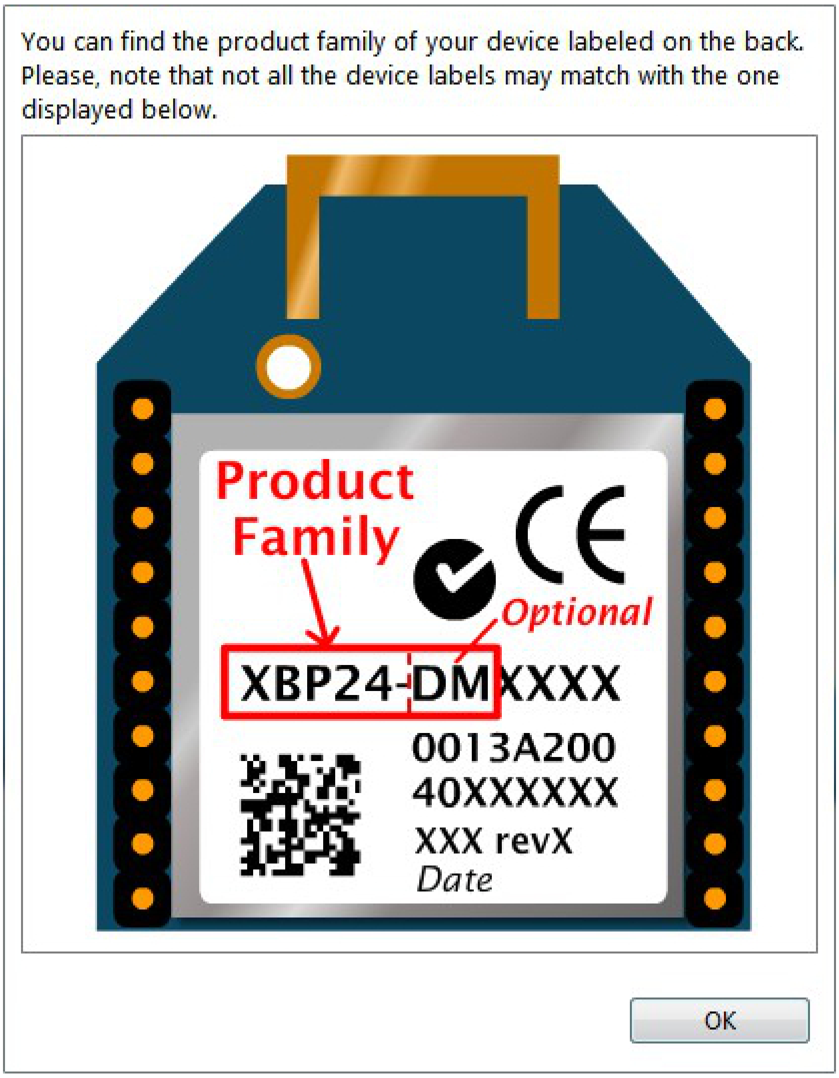
Select the COM port containing the module to recover: COM1

Select the product family of your device, the new function set and the firmware version to flash:

? Product family	Function set	Firmware version
XB24	ZigBee End Device API	23A7 (Newest)
XB24-B	ZigBee End Device AT	23A0
XB24-DM	ZigBee End Device Analog IO	238C
XB24-SE	ZigBee End Device Digital IO	2370
XB24-WF	ZigBee End Device PH	2364
XB24-ZB	ZigBee Router API	2341
XB24C	ZigBee Router AT	2321

? **Product family**	If you don't know which Product family your radio module belongs to, click the question mark icon next to the Product family label.

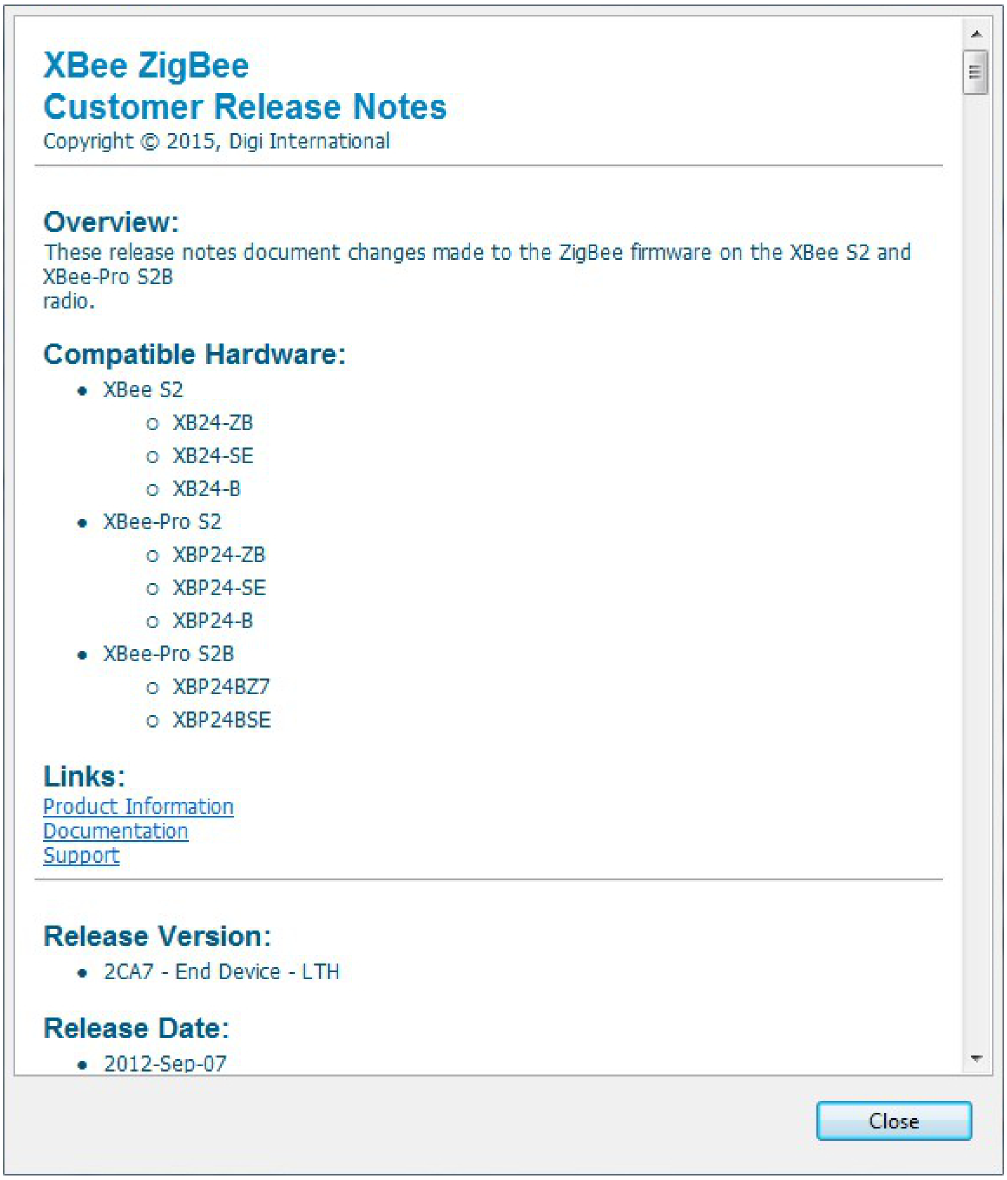
A dialog box appears indicating the location of your device's product family.



You can find the product family of your device labeled on the back. Please, note that not all the device labels may match with the one displayed below.

View Release Notes	If the selected firmware has any Release Notes available, the **View Release Notes** button below the firmware list is automatically enabled.

Clicking on the button, a new window displays the selected firmware Release Notes:



XBee ZigBee
Customer Release Notes
Copyright © 2015, Digi International

Overview:
These release notes document changes made to the ZigBee firmware on the XBee S2 and XBee-Pro S2B radio.

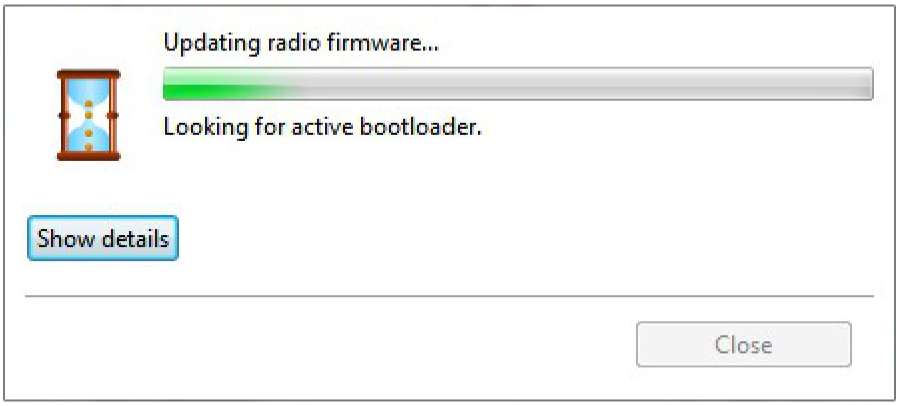
Compatible Hardware:
- XBee S2
  - XB24-ZB
  - XB24-SE
  - XB24-B
- XBee-Pro S2
  - XBP24-ZB
  - XBP24-SE
  - XBP24-B
- XBee-Pro S2B
  - XBP24BZ7
  - XBP24BSE

Links:
Product Information
Documentation
Support

Release Version:
- 2CA7 - End Device - LTH

Release Date:
- 2012-Sep-07

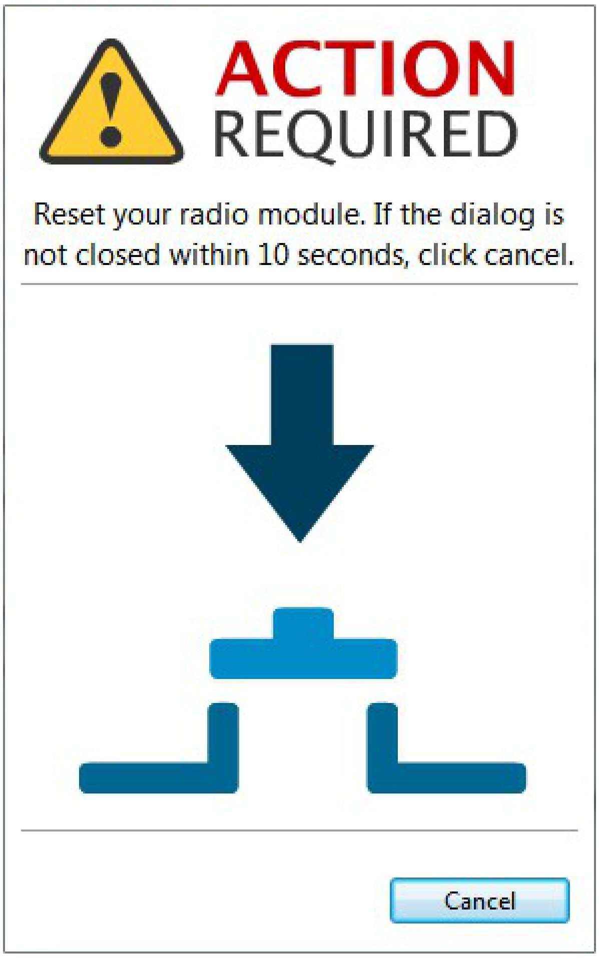
When finished, click the **Recover** button to try to flash the new firmware in the radio module. A new progress dialog box displays the details.



Updating radio firmware...

Looking for active bootloader.

The recover process may display a dialog asking you to reset the radio module:

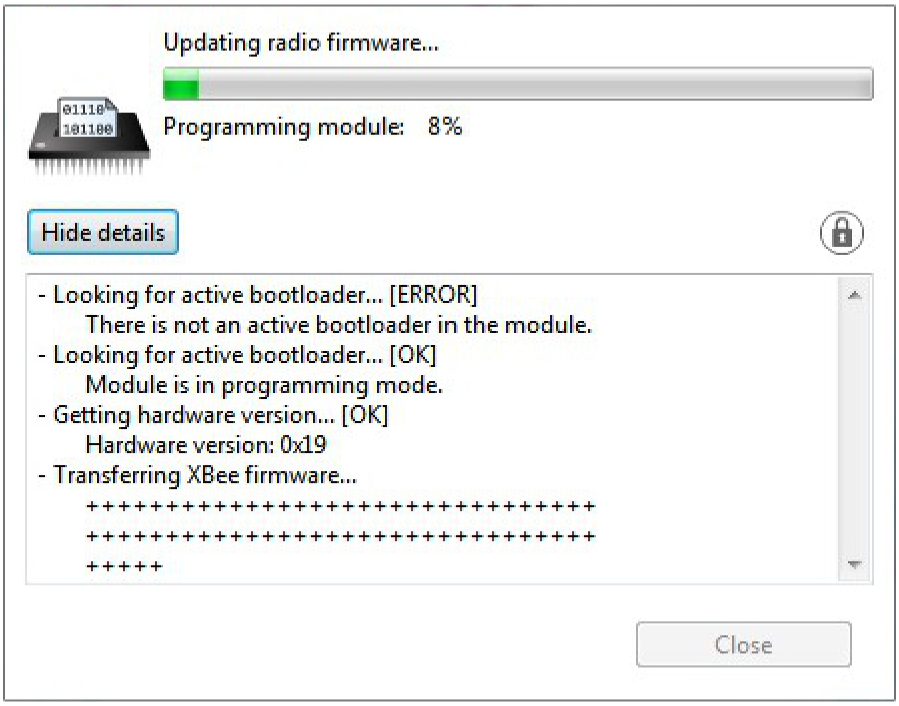


ACTION REQUIRED

Reset your radio module. If the dialog is not closed within 10 seconds, click cancel.

If that happens, reset your radio module and wait for the dialog to close.

The process is similar to the Update firmware process from the Configuration working mode. You can click the **Show details** button to see a simple log with the actions taking place.

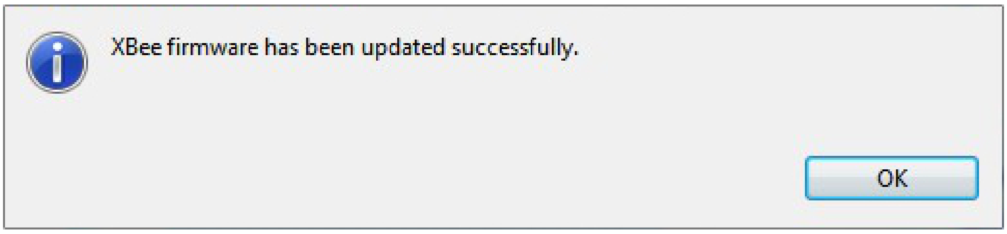


Updating radio firmware...

Programming module: 8%

```
- Looking for active bootloader... [ERROR]
      There is not an active bootloader in the module.
- Looking for active bootloader... [OK]
      Module is in programming mode.
- Getting hardware version... [OK]
      Hardware version: 0x19
- Transferring XBee firmware...
      ++++++++++++++++++++++++++++++++
      ++++++++++++++++++++++++++++++++
      +++++
```

If there are any errors during the recovery process, an error message appears, and the error is logged in the details box. If the recovery process completes successfully, a validation message appears.

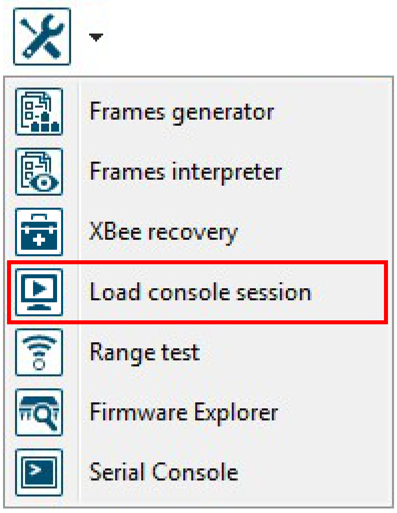


XBee firmware has been updated successfully.

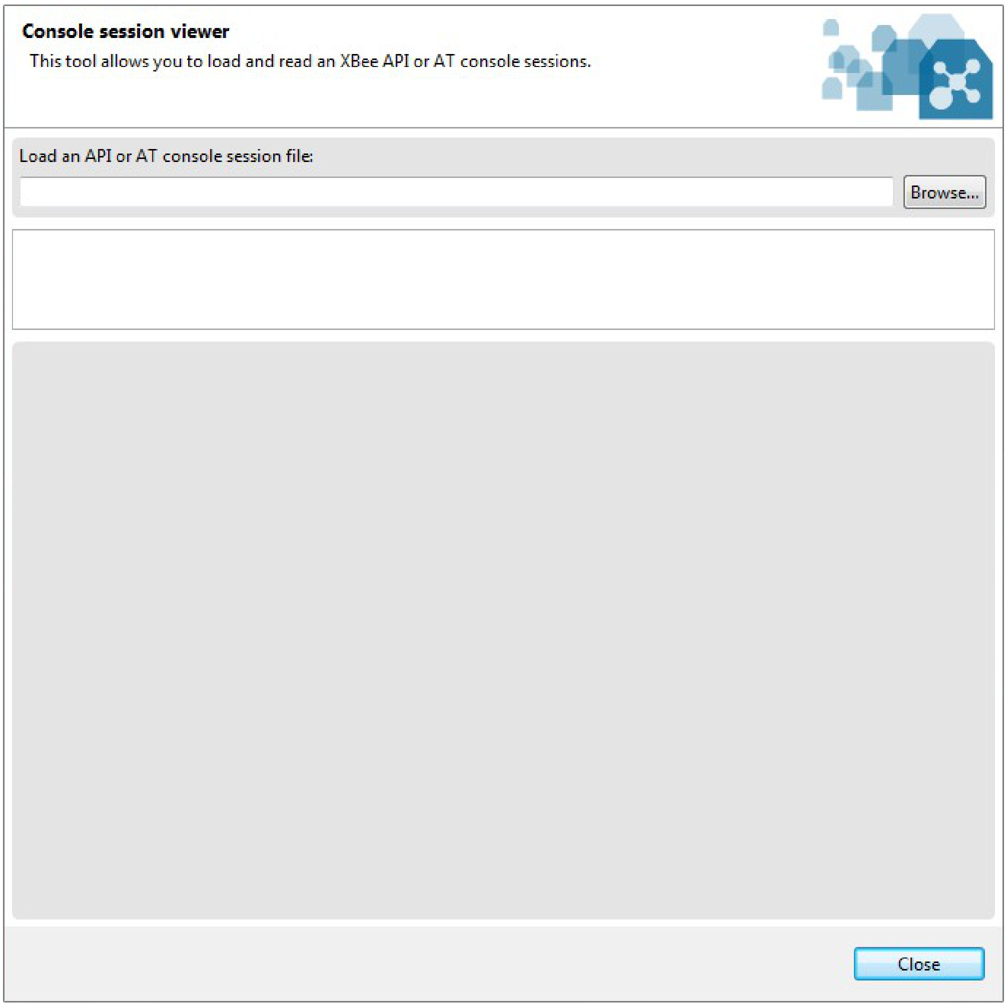
Load console session tool

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 the **Load console session tool** option from the **Tools** drop-down menu in the toolbar.



The tool opens in a new floating dialog box.

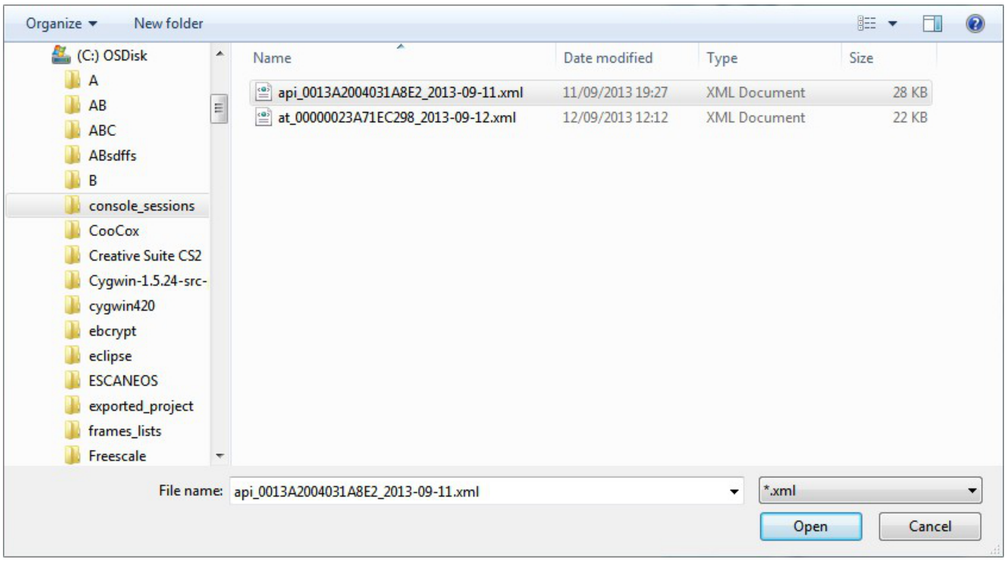


Console session viewer

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

Load an API or AT console session file:

To load a session, click the **Browse** button. A new **Open file** dialog box asks you for a saved XML console session file.



Name	Date modified	Type	Size
api_0013A2004031A8E2_2013-09-11.xml	11/09/2013 19:27	XML Document	28 KB
at_00000023A71EC298_2013-09-12.xml	12/09/2013 12:12	XML Document	22 KB

File name: api_0013A2004031A8E2_2013-09-11.xml

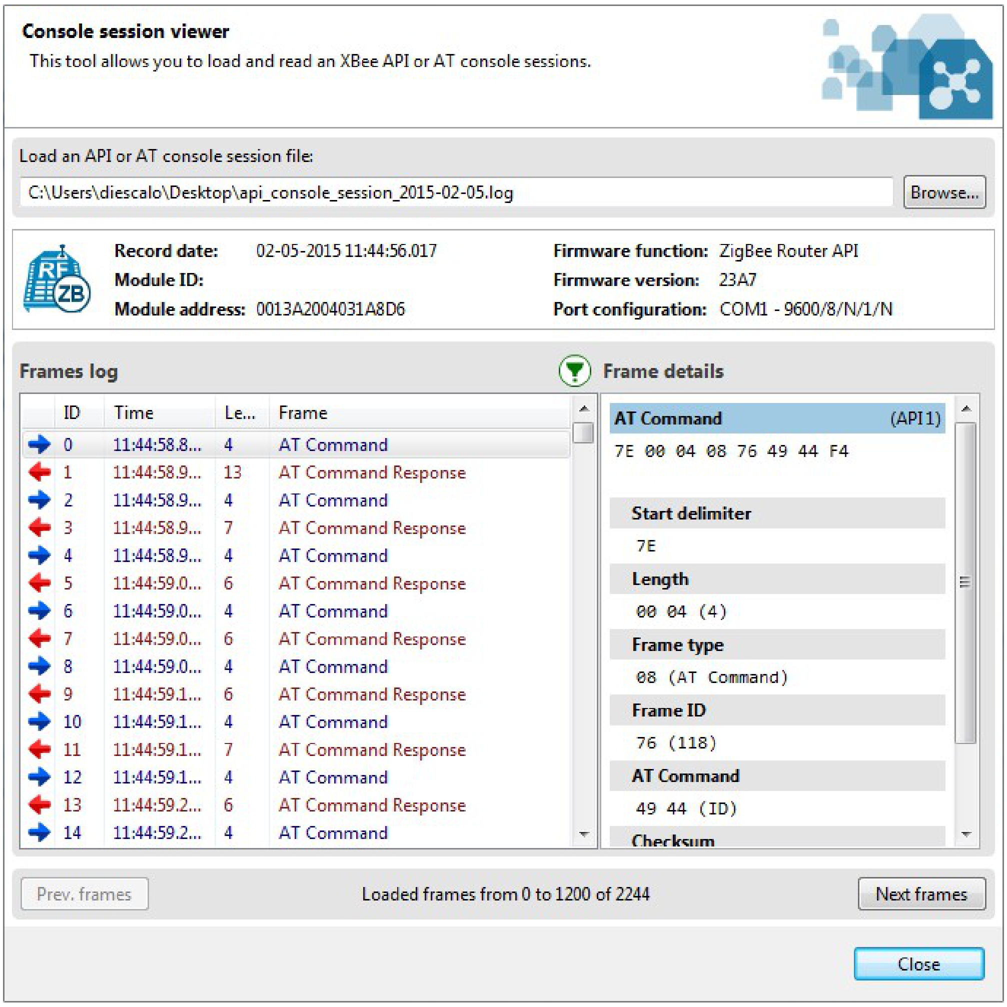
*.xml

Select your console session file and click **Open** to continue.

> 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, this tool maintains compatibility.

If the tool is not able to load the selected file, you will get an error stating the cause of the problem. If the console session file was loaded successfully, the tool will display three new controls:

- Radio module information box
- Console session control
- Navigation control



Load an API or AT console session file: C:\Users\diescalo\Desktop\api_console_session_2015-02-05.log

Record date: 02-05-2015 11:44:56.017
Module ID:
Module address: 0013A2004031A8D6
Firmware function: ZigBee Router API
Firmware version: 23A7
Port configuration: COM1 - 9600/8/N/1/N

ID	Time	Le...	Frame
0	11:44:58.8...	4	AT Command
1	11:44:58.9...	13	AT Command Response
2	11:44:58.9...	4	AT Command
3	11:44:58.9...	7	AT Command Response
4	11:44:58.9...	4	AT Command
5	11:44:59.0...	6	AT Command Response
6	11:44:59.0...	4	AT Command
7	11:44:59.0...	6	AT Command Response
8	11:44:59.0...	4	AT Command
9	11:44:59.1...	6	AT Command Response
10	11:44:59.1...	4	AT Command
11	11:44:59.1...	7	AT Command Response
12	11:44:59.1...	4	AT Command
13	11:44:59.2...	6	AT Command Response
14	11:44:59.2...	4	AT Command

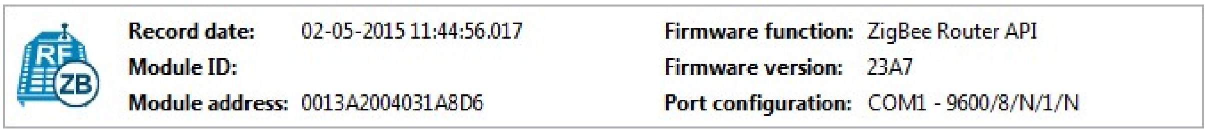
AT Command (API 1)
7E 00 04 08 76 49 44 F4

Start delimiter
7E
Length
00 04 (4)
Frame type
08 (AT Command)
Frame ID
76 (118)
AT Command
49 44 (ID)
Checksum

Loaded frames from 0 to 1200 of 2244

Radio module information box

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



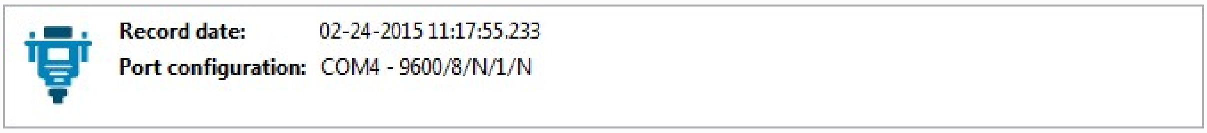
Record date: 02-05-2015 11:44:56.017
Module ID:
Module address: 0013A2004031A8D6
Firmware function: ZigBee Router API
Firmware version: 23A7
Port configuration: COM1 - 9600/8/N/1/N

This information includes the following:

- **Icon**. Represents the protocol of the radio module.
- **Record date**. The date the console session was saved.
- **Module ID**. Displays 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.

> If the session to load corresponds to a Serial console session, the Radio module information box

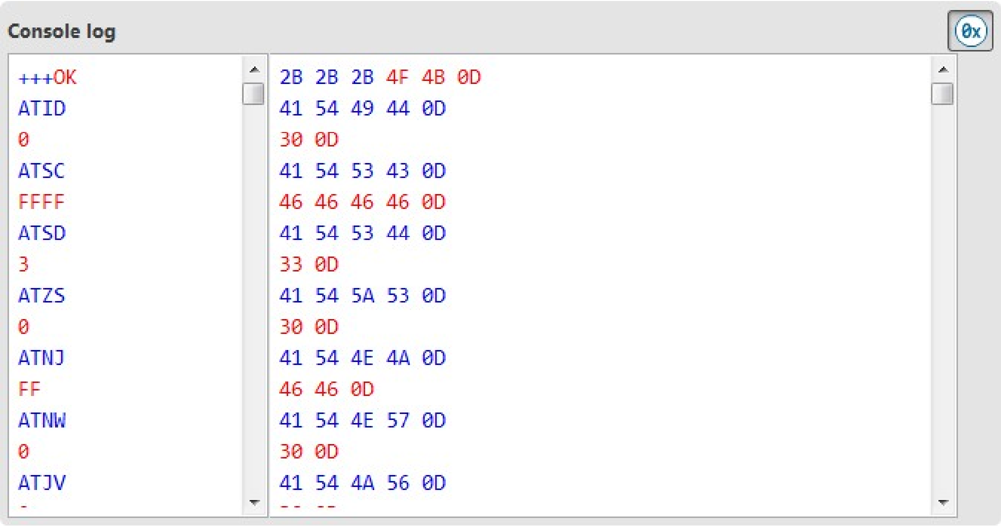
only displays the **Record date** and the **Port configuration** fields.



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

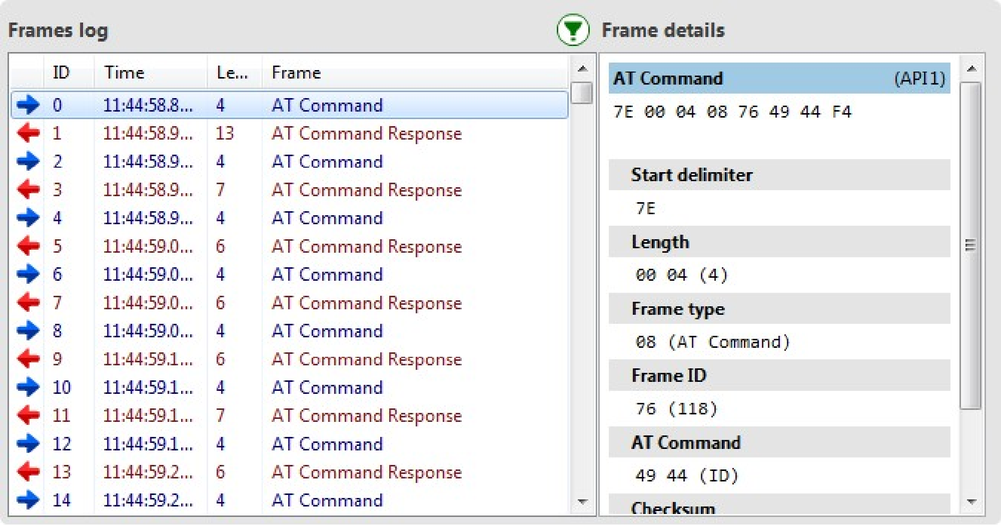
Console session control

This control varies depending on the type of console session. If the saved console session was 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 will be a data text box with the hexadecimal representation of the data. See the **Data traffic monitoring section** topic of the AT Console or Serial Console Tool for more information about this control.



```
+++OK    2B 2B 2B 4F 4B 0D
ATID     41 54 49 44 0D
0        30 0D
ATSC     41 54 53 43 0D
FFFF     46 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
ATNW     41 54 4E 57 0D
0        30 0D
ATJV     41 54 4A 56 0D
```

If the saved console session was an API session, the control will be the same as the one found in the API frames traffic monitoring section of the API Console, or which is the same, it will be an API frames table with an API frame details view attached in the right side. See the API frames traffic monitoring section for more information about this control.



	ID	Time	Le...	Frame
→	0	11:44:58.8...	4	AT Command
←	1	11:44:58.9...	13	AT Command Response
→	2	11:44:58.9...	4	AT Command
←	3	11:44:58.9...	7	AT Command Response
→	4	11:44:58.9...	4	AT Command
←	5	11:44:59.0...	6	AT Command Response
→	6	11:44:59.0...	4	AT Command
←	7	11:44:59.0...	6	AT Command Response
→	8	11:44:59.0...	4	AT Command
←	9	11:44:59.1...	6	AT Command Response
→	10	11:44:59.1...	4	AT Command
←	11	11:44:59.1...	7	AT Command Response
→	12	11:44:59.1...	4	AT Command
←	13	11:44:59.2...	6	AT Command Response
→	14	11:44:59.2...	4	AT Command

AT Command (API1)
7E 00 04 08 76 49 44 F4

Start delimiter
7E

Length
00 04 (4)

Frame type
08 (AT Command)

Frame ID
76 (118)

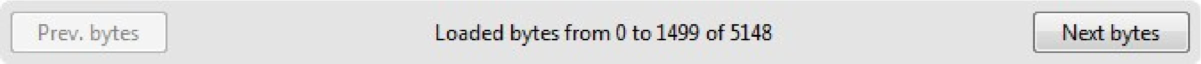
AT Command
49 44 (ID)

Checksum

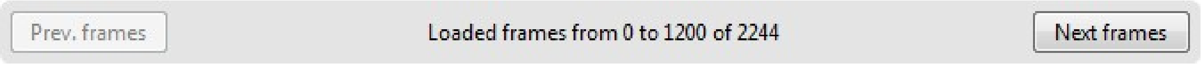
Regardless of the control displayed, you can only review the session data; any other functionality with the controls is disabled in this tool.

Navigation control

Sometimes the session loaded is too long and the tool cannot display all the data at once. When this happens, the tool splits the data in blocks of bytes (when the session is an AT or Serial session) or blocks of frames (when the session is an API session) and enables a navigation control just below the Console session control. This control provides information about the bytes or frames that are currently displayed as well as two buttons to load the next or previous blocks of data.



Loaded bytes from 0 to 1499 of 5148



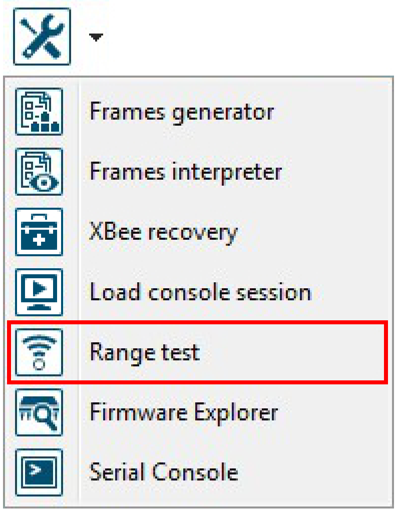
Loaded frames from 0 to 1200 of 2244

> If the session has more than one block of bytes or frames, you can also load the next and previous blocks using the Console session control scroll bar. (The next block of data loads automatically when you scroll down to the bottom of the control and try to scroll down again.)

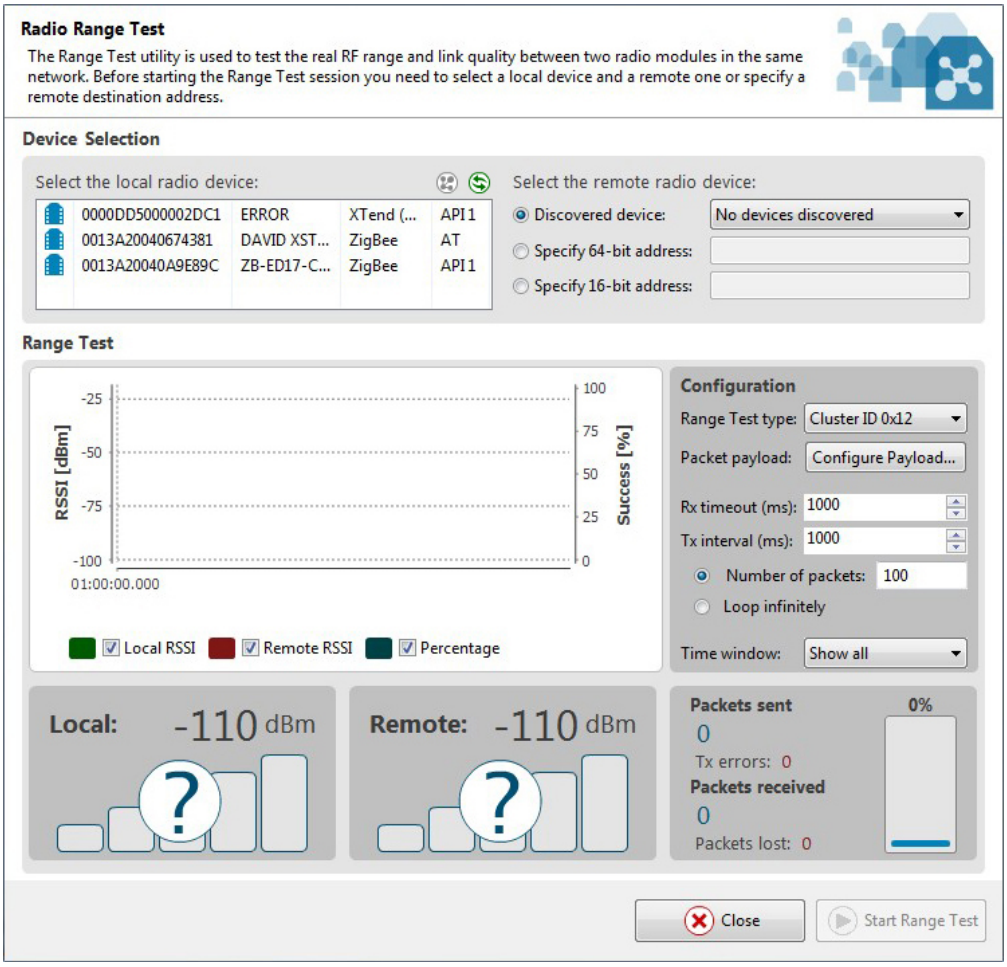
Range test tool

The Range Test utility, embedded within XCTU, tests the real RF range and link quality between two radio modules in the same network. To perform a range test, you need to have a local radio module connected to your PC and added to XCTU, and a remote device in the same network as the local device. To learn how to add local devices to XCTU, see the Add radio modules topic.

Once XCTU is started and you have added at least one local radio device, select the **Range Test** option from the **Tools** drop-down menu in the main toolbar.



The tool opens in a new floating dialog box:



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:

0000DD5000002DC1	ERROR	XTend (...	API 1
0013A20040674381	DAVID XST...	ZigBee	AT
0013A20040A9E89C	ZB-ED17-C...	ZigBee	API 1

Select the remote radio device:

● Discovered device: No devices discovered

○ Specify 64-bit address:

○ Specify 16-bit address:

Range Test

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

Time window: Show all

Local: -110 dBm

Remote: -110 dBm

Packets sent 0

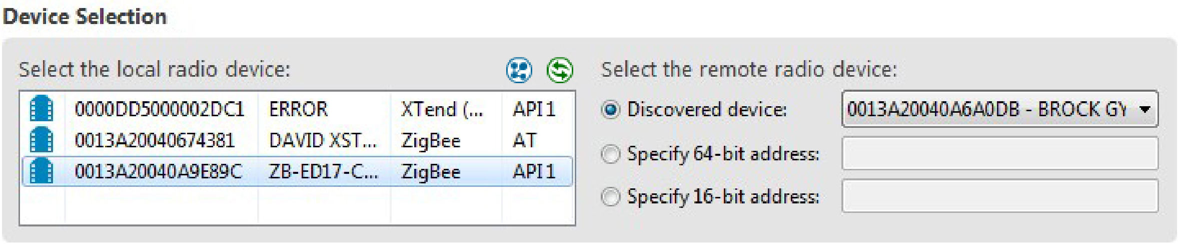
Tx errors: 0

Packets received 0

Packets lost: 0

Device selection

This section allows you to select the local device that will perform the range test and the remote device against which the range test will be performed:



Device Selection

Select the local radio device:

0000DD5000002DC1	ERROR	XTend (...	API 1
0013A20040674381	DAVID XST...	ZigBee	AT
0013A20040A9E89C	ZB-ED17-C...	ZigBee	API 1

Select the remote radio device:

● Discovered device: 0013A20040A6A0DB - BROCK GY

○ Specify 64-bit address:

○ Specify 16-bit address:

Local device

The local device list populates with the devices that you have added to XCTU. Select the device you want to use in the range test. The list provides the following information about each local device:

- **MAC address**
- **Device name**
- **Protocol**
- **Operating mode**

	You can refresh the local devices table anytime by clicking the **Refresh** button located above it.

You can discover remote devices for the selected local device by clicking the **Discover remote devices** button.

Remote device

The remote device selection can be performed in three ways:

- **Discovered device:** If any remote devices are discovered for the selected local device, this combo is filled with the available remote devices.

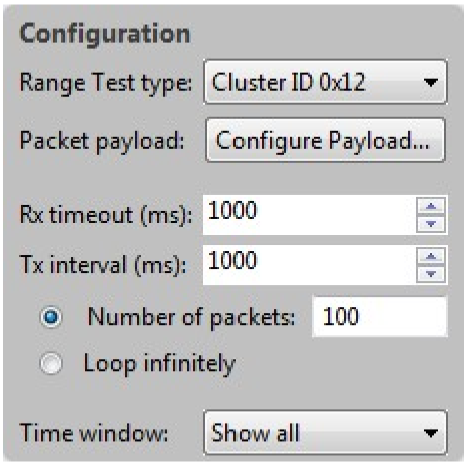
> Not all protocols support node discovery. Only devices that support node discovery list remote devices.

- **Specify 64-bit address:** Manually enter the 64-bit address of the destination device.
- **Specify 16-bit address:** Manually enter the 16-bit address of the destination device.

> Not all protocols support 64- and 16-bit addressing. Devices that do not support any of these mechanisms display an error at the top of the page.

Session configuration

The range test session configuration sectione appears below the device selection, to the right. This section allows you to configure the range test process:



Configuration

Range Test type: Cluster ID 0x12
Packet payload:
Rx timeout (ms): 1000
Tx interval (ms): 1000
● Number of packets: 100
○ Loop infinitely
Time window: Show all

The following are the available settings.

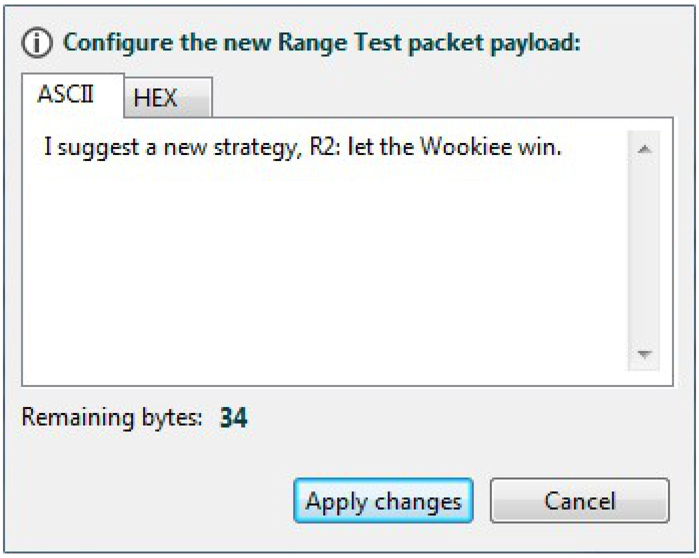
- **Range Test type:** Determines the type of range test to perform. Available options include:
    - **Cluster ID 0x12**: The range test is performed using explicit addressing frames/packets directed to the Cluster ID 0x12 which returns the sent message.

    > Not all protocols and operating modes support **Cluster ID 0x12 Range Test** type. Devices that do not support it display an error at the top of the page.

    - **Loopback**: The range test is performed using the serial port/USB hardware loopback capabilities. Specific user actions are required to use this mechanism. See Special Considerations.

    > Loopback range tests require the remote device to be in AT (transparent) mode.

- **Packet payload:** Allows you to configure the packet payload that will be sent to the remote device. This button opens a new window where you can enter the new value of the payload:



I suggest a new strategy, R2: let the Wookiee win.

Remaining bytes: 34

- **Rx Timeout (ms):** The maximum time in milliseconds to wait for response from the remote device before considering a packet to be lost.
- **Tx interval (ms):** The minimum time in milliseconds to wait before sending a new packet to the remote device.
- **Number of packets:** Check this option to set the number of packets to send in this session.
- **Loop infinitely:** Check this option to send packets infinitely until the range test session is stopped manually.
- **Time window:** Configures the visible time window of the RSSI data measured by the range test.

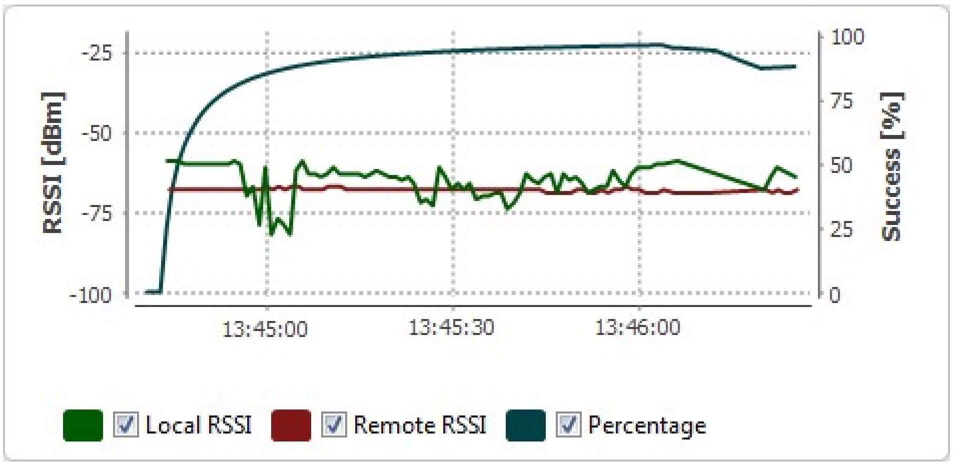
	When you have configured all the options, click **Start Range Test** to start sending packets and measure the signal strength.
	You can stop the process any time by pressing the same button, now showing the text **Stop Range Test.**

Data presentation

Once the range test process has started, you can see the retrieved data represented in several ways.

RSSI Chart

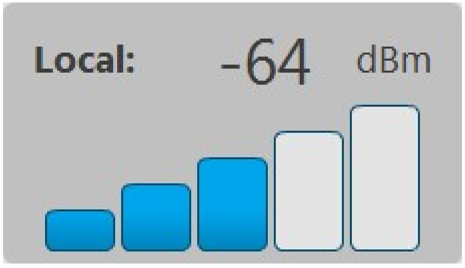
This chart represents the RSSI values of the local and remote devices during the range test session. The chart also contains the value of the percentage of success for the total packets sent:



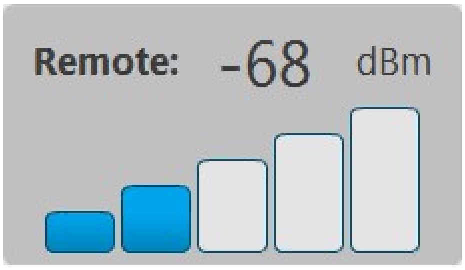
You can hide and show any of the chart data by checking or unchecking the data checkboxes.

Instant RSSI values

These two components display the instant RSSI value of the local and remote devices. This value is retrieved for the last packet sent/received:



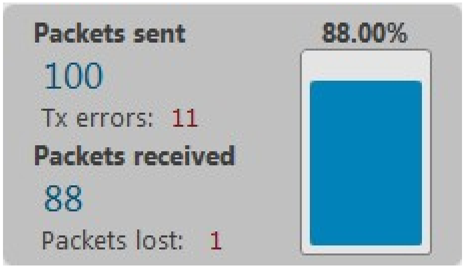
Local: -64 dBm



Remote: -68 dBm

Packet summary

This control displays the total amount of packets sent, packets received, transmission errors and packets lost. It also displays the success rate (as a percentage) for sending and receiving packets during the range test session:

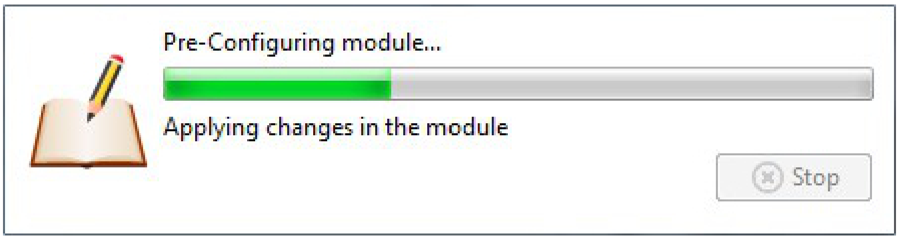


Packets sent 100
Tx errors: 11
Packets received 88
Packets lost: 1
88.00%

Special considerations

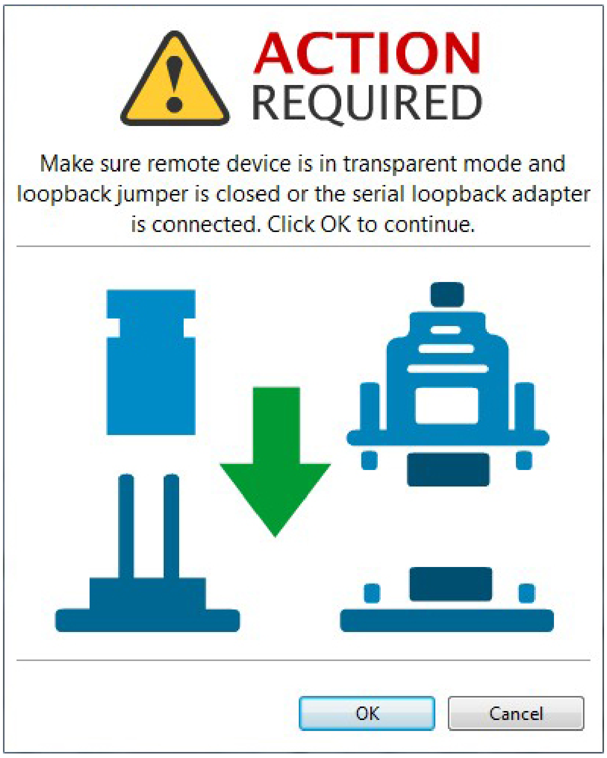
There are some special considerations to be aware of while working with the Range test tool:

- Range test is only supported in these protocols:
  - **ZigBee**
  - **Digi Mesh**
  - **XTend**
  - **XTend - Digi Mesh**
  - **XC/XSC**
  - **XLR PRO**
  - **802.15.4**
  - **Dig Point**
- 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:

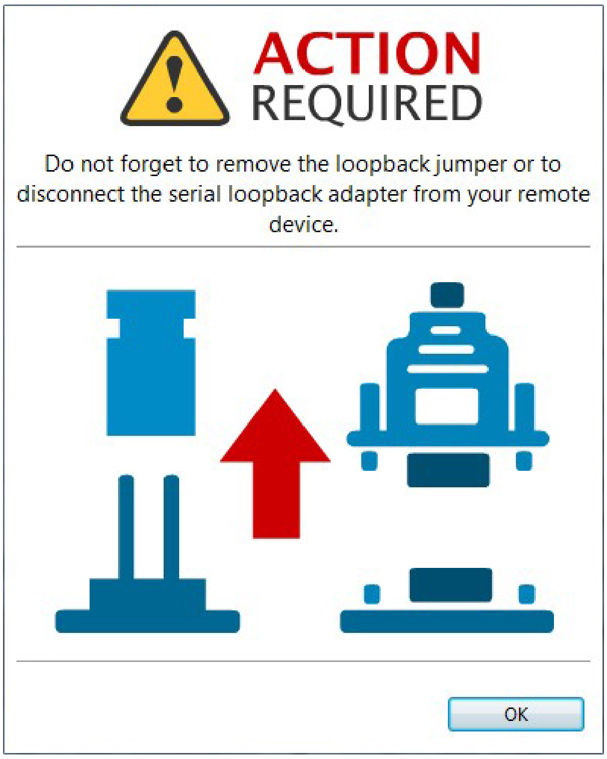

Pre-Configuring module...

Applying changes in the module

- 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:


ACTION REQUIRED

Make sure remote device is in transparent mode and loopback jumper is closed or the serial loopback adapter is connected. Click OK to continue.

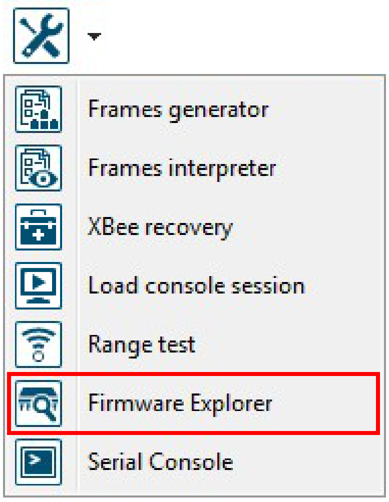


Do not forget to remove the loopback jumper or to disconnect the serial loopback adapter from your remote device.

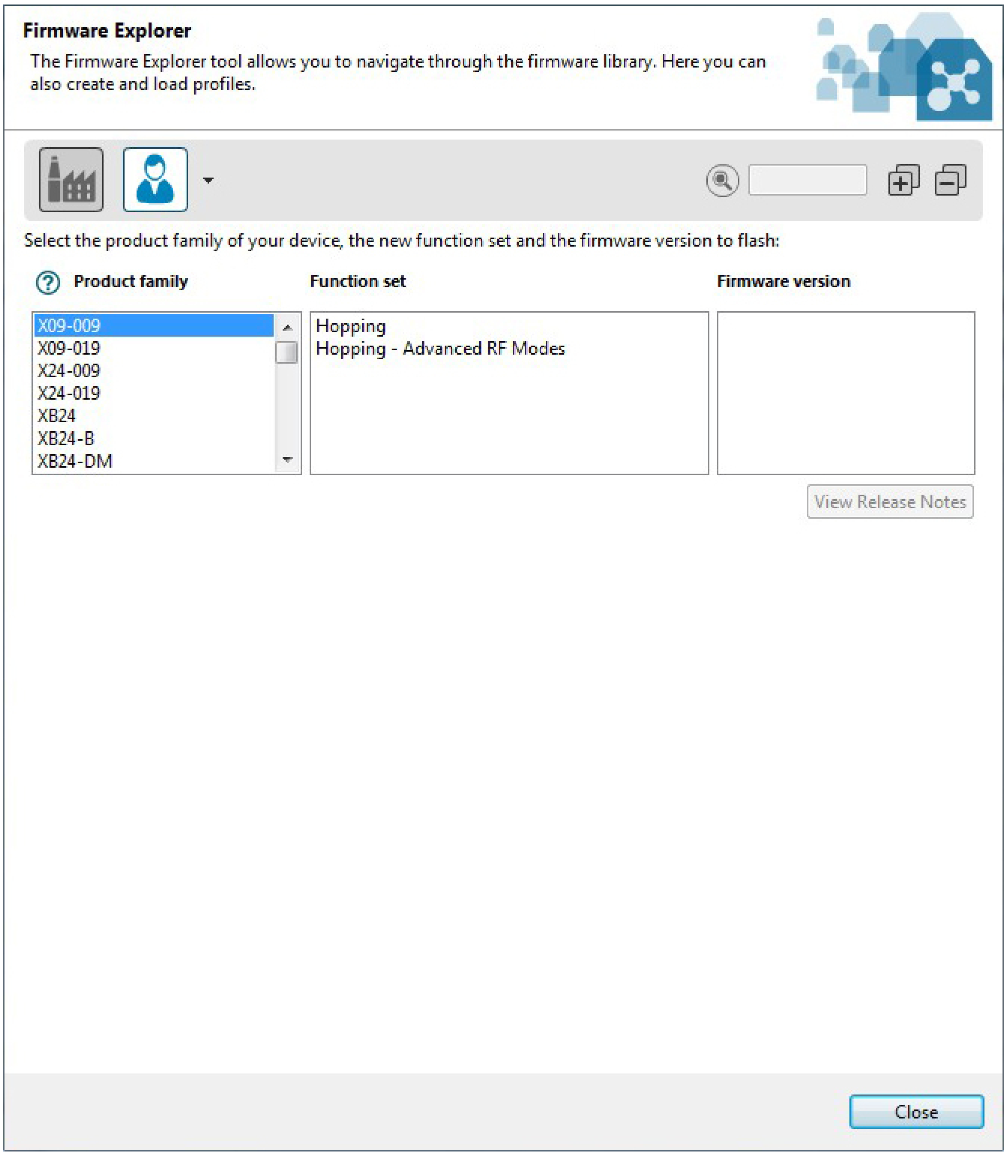
Firmware Explorer tool

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. See the Radio firmware topic for more information.

To open the Firmware Explorer tool, select the **Firmware Explorer** option from the **Tools** drop-down menu in the main toolbar.



The tool opens in a new floating dialog box.



Firmware Explorer

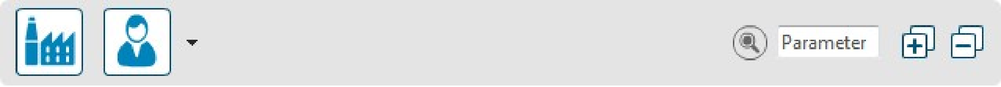
The Firmware Explorer tool allows you to navigate through the firmware library. Here you can also create and load profiles.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
X09-009	Hopping	
X09-019	Hopping - Advanced RF Modes	
X24-009		
X24-019		
XB24		
XB24-B		
XB24-DM		

Toolbar

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



Loading default firmware settings

	The **Load default firmware settings** button allows you to load the default values established by the firmware.

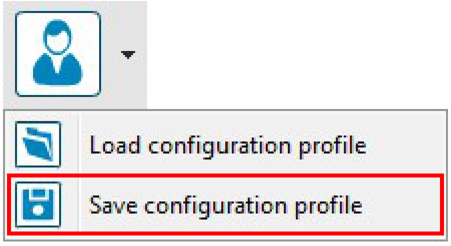
Configuration profiles

A configuration profile contains the information and settings values of a radio firmware version. The firmware explorer tool lets you save and write configuration profiles the same way as in the Configuration working mode (Configuration profiles). This feature is useful in a production environment when the same parameters need to be set on multiple radios.

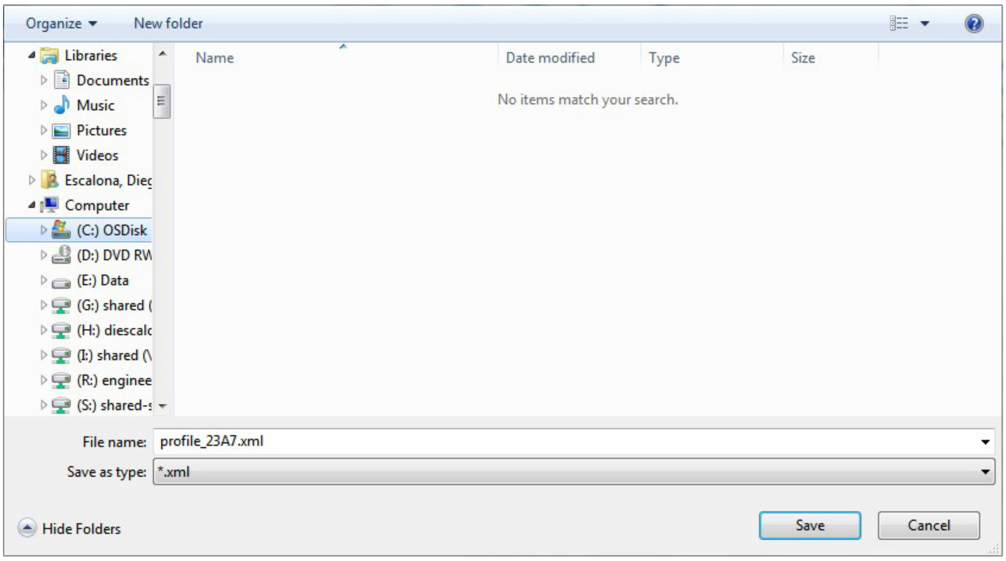
Saving a configuration profile

To save a configuration profile, first configure all the settings of the firmware with your desired values.

Next, go to the **Configuration profiles** drop-down menu and select **Save configuration profile**.



A new **Save file** dialog box asks you for the name and destination of the profile file. Choose a name and path and click **OK** to save your configuration profile.

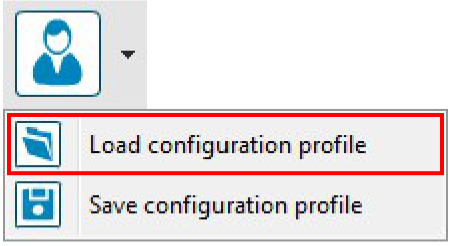


File name: profile_23A7.xml

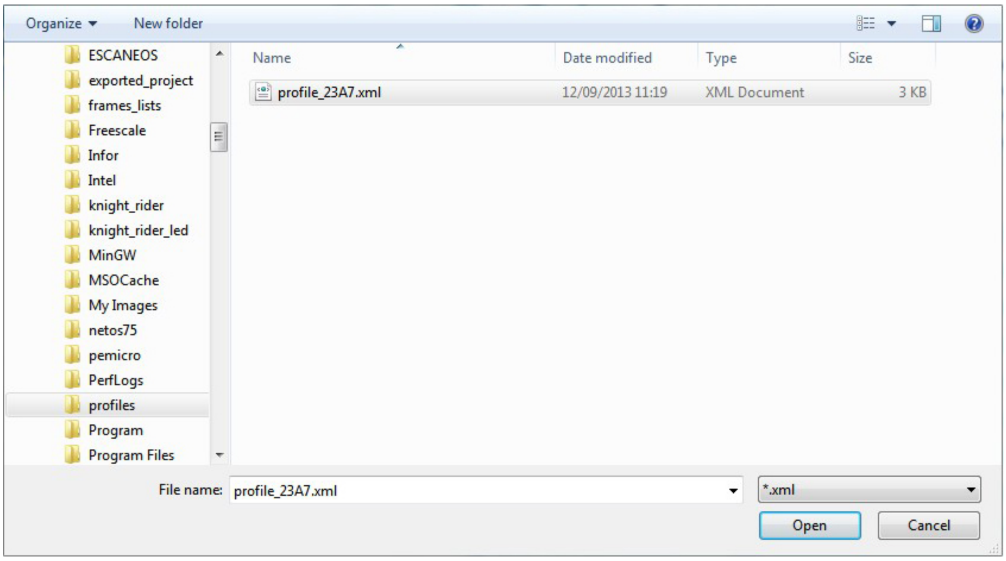
Save as type: *.xml

Loading a configuration profile

To load a configuration profile, select **Load configuration profile** from the **Configuration profiles** drop-down menu.



A new **Open file** window asks you for the configuration profile file to load. Look for a previously saved configuration profile and click **Open**.



Name	Date modified	Type	Size
profile_23A7.xml	12/09/2013 11:19	XML Document	3 KB

File name: profile_23A7.xml

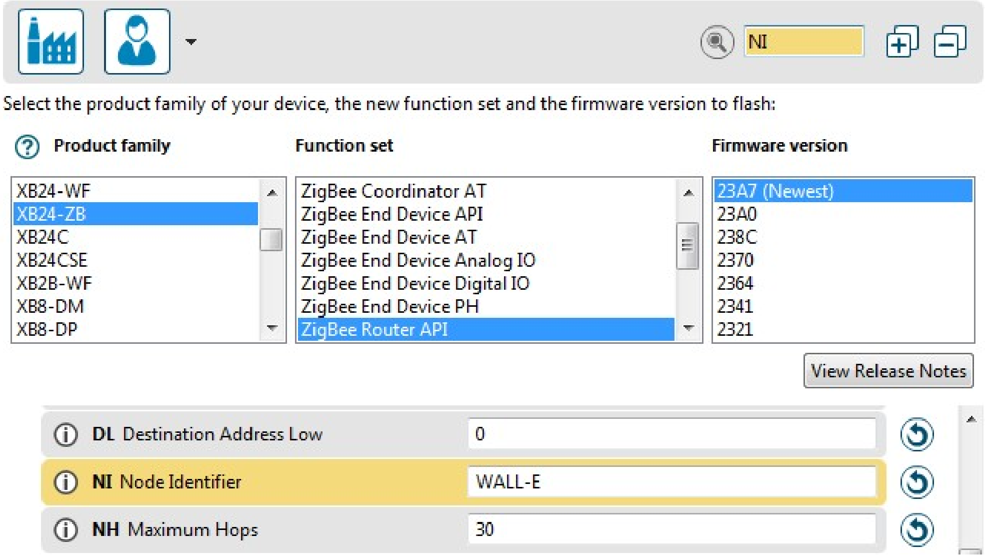
*.xml

The firmware corresponding to the profile is selected automatically and the settings are loaded.

> You can load a profile after opening the Firmware explorer tool without selecting a firmware version from the lists.

Searching a setting

You can enter an AT command name in the toolbar search box to find the command's parameters for a specific firmware version. If the command is found, it is highlighted in yellow.


NI

Select the product family of your device, the new function set and the firmware version to flash:

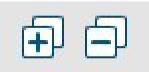
DL Destination Address Low: 0

NI Node Identifier: WALL-E

NH Maximum Hops: 30

Expanding and collapsing sections

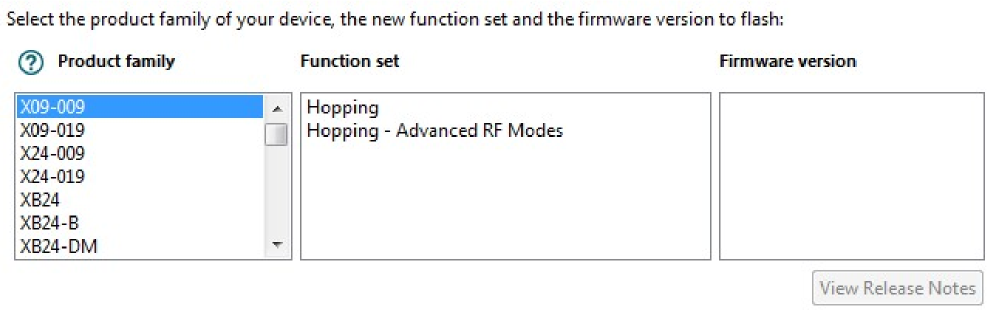
The icons on the far right side of the configuration toolbar allow you to Expand and Collapse sections.



Clicking the **Expand sections** button expands all the settings that were collapsed; clicking the **Collapse sections** button collapses all expanded settings.

Firmware selection panel

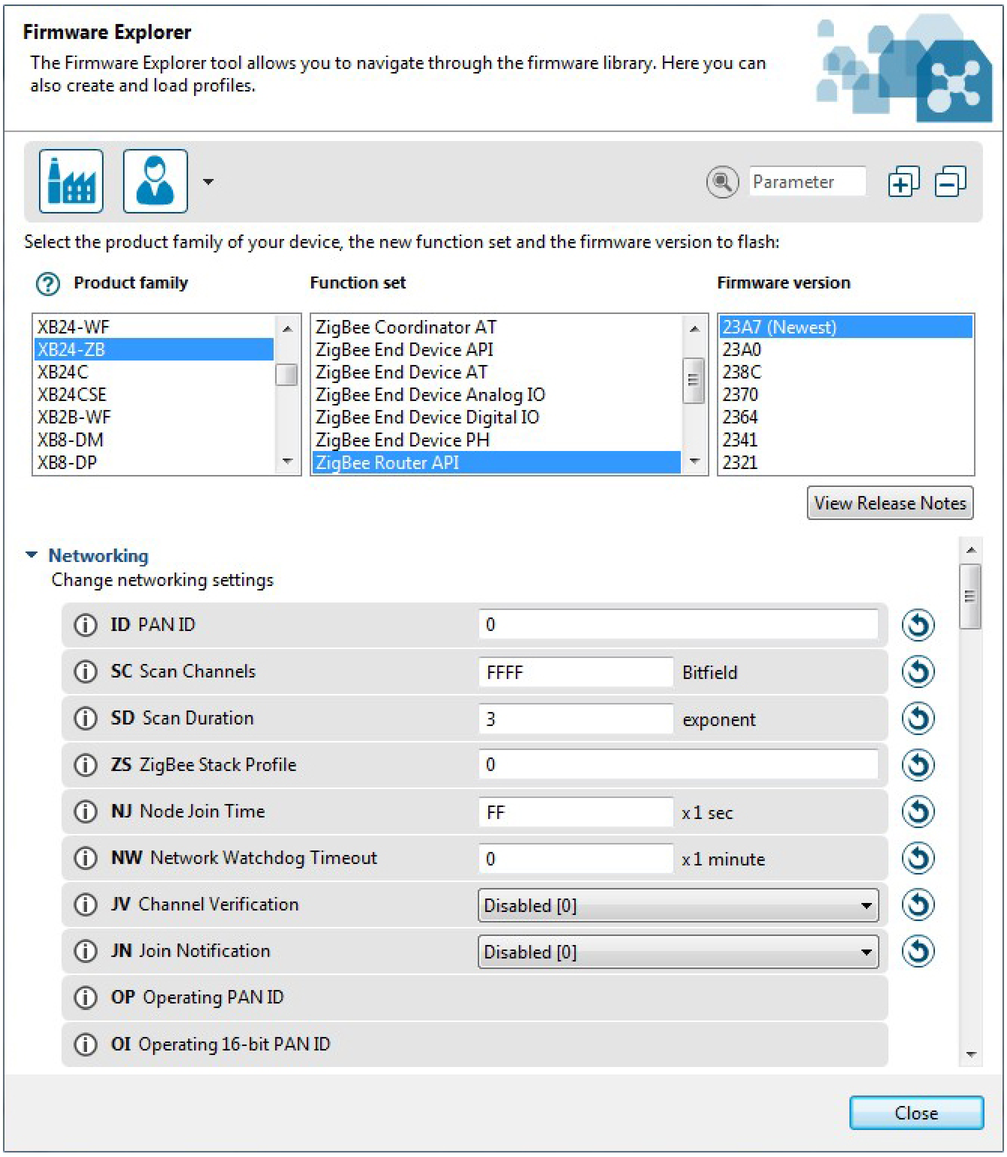
The firmware selection panel allows you to specify the firmware you are loading onto a device. To display the settings of a specific radio firmware, specify the firmware family, function, and version from the corresponding lists.



Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
X09-009	Hopping	
X09-019	Hopping - Advanced RF Modes	
X24-009		
X24-019		
XB24		
XB24-B		
XB24-DM		

Once a specific firmware version has been selected, the settings are displayed in the firmware settings panel.



Firmware Explorer

The Firmware Explorer tool allows you to navigate through the firmware library. Here you can also create and load profiles.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
XB24-WF	ZigBee Coordinator AT	23A7 (Newest)
XB24-ZB	ZigBee End Device API	23A0
XB24C	ZigBee End Device AT	238C
XB24CSE	ZigBee End Device Analog IO	2370
XB2B-WF	ZigBee End Device Digital IO	2364
XB8-DM	ZigBee End Device PH	2341
XB8-DP	ZigBee Router API	2321

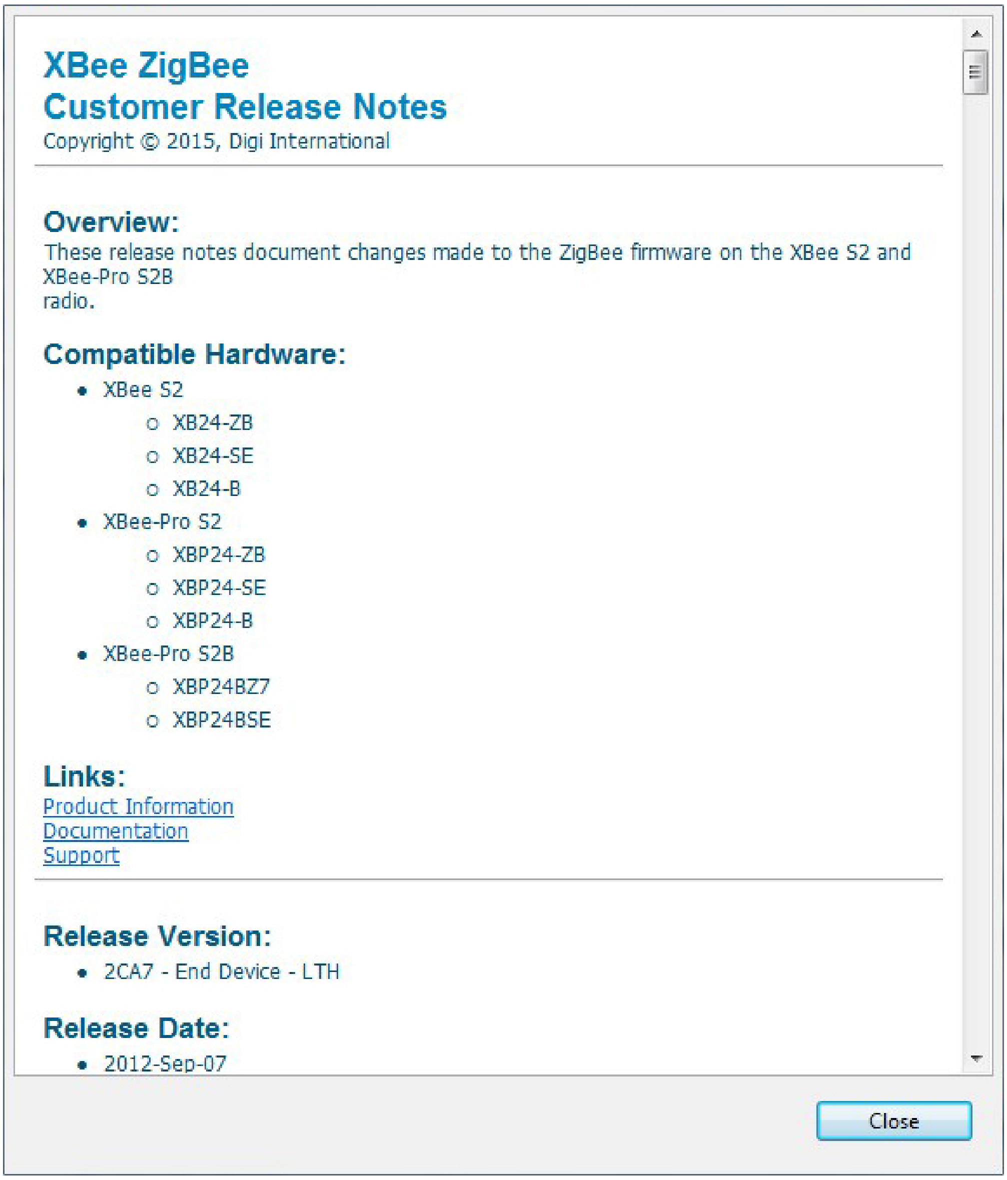
Networking

Change networking settings

ID PAN ID	0	
SC Scan Channels	FFFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x 1 sec
NW Network Watchdog Timeout	0	x 1 minute
JV Channel Verification	Disabled [0]	
JN Join Notification	Disabled [0]	
OP Operating PAN ID		
OI Operating 16-bit PAN ID		

View Release Notes	If the selected firmware has any Release Notes available, the **View Release Notes** button below the firmware list is automatically enabled.

When you click the **View Release Notes** button, a new window displays the selected firmware Release Notes:



XBee ZigBee
Customer Release Notes

Copyright © 2015, Digi International

Overview:

These release notes document changes made to the ZigBee firmware on the XBee S2 and XBee-Pro S2B radio.

Compatible Hardware:

- XBee S2
    - XB24-ZB
    - XB24-SE
    - XB24-B
- XBee-Pro S2
    - XBP24-ZB
    - XBP24-SE
    - XBP24-B
- XBee-Pro S2B
    - XBP24BZ7
    - XBP24BSE

Links:

Product Information
Documentation
Support

Release Version:

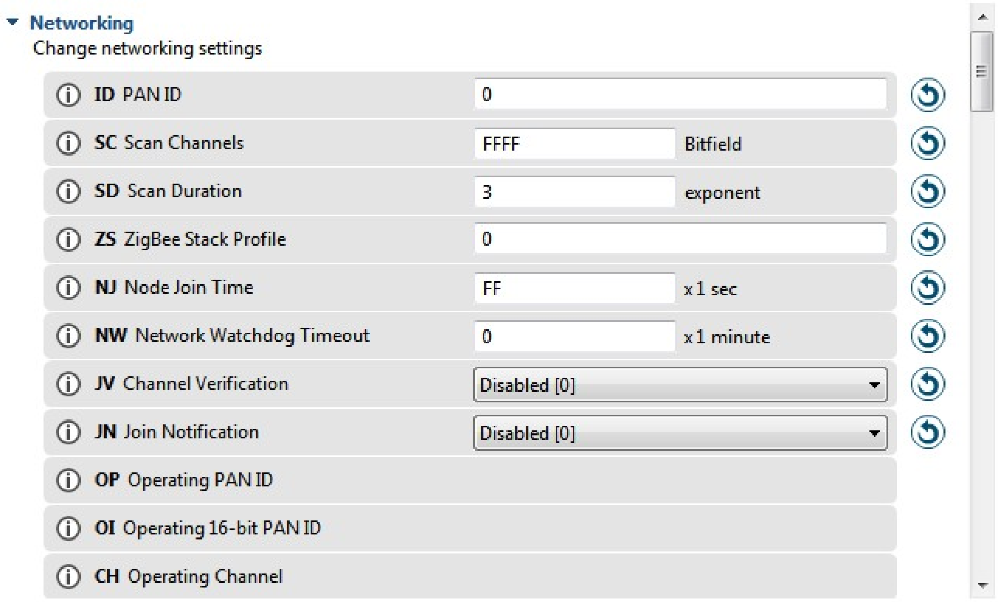
- 2CA7 - End Device - LTH

Release Date:

- 2012-Sep-07

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.



Networking
Change networking settings

ID PAN ID: 0
SC Scan Channels: FFFF Bitfield
SD Scan Duration: 3 exponent
ZS ZigBee Stack Profile: 0
NJ Node Join Time: FF x 1 sec
NW Network Watchdog Timeout: 0 x 1 minute
JV Channel Verification: Disabled [0]
JN Join Notification: Disabled [0]
OP Operating PAN ID
OI Operating 16-bit PAN ID
CH Operating Channel

You can see the default value of each setting, and also change them to edit configuration profiles.

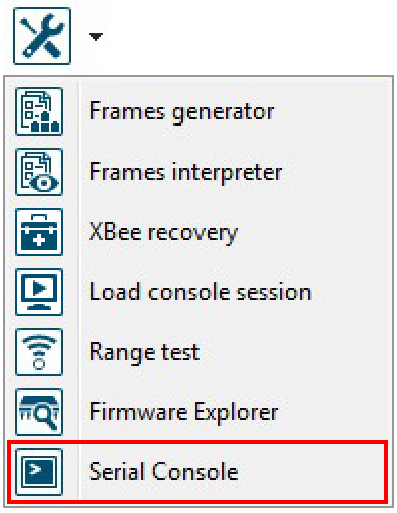
↺	Click the blue arrow button next to a setting to load the default value of that setting.

To learn more about configuration settings see the Firmware settings topic.

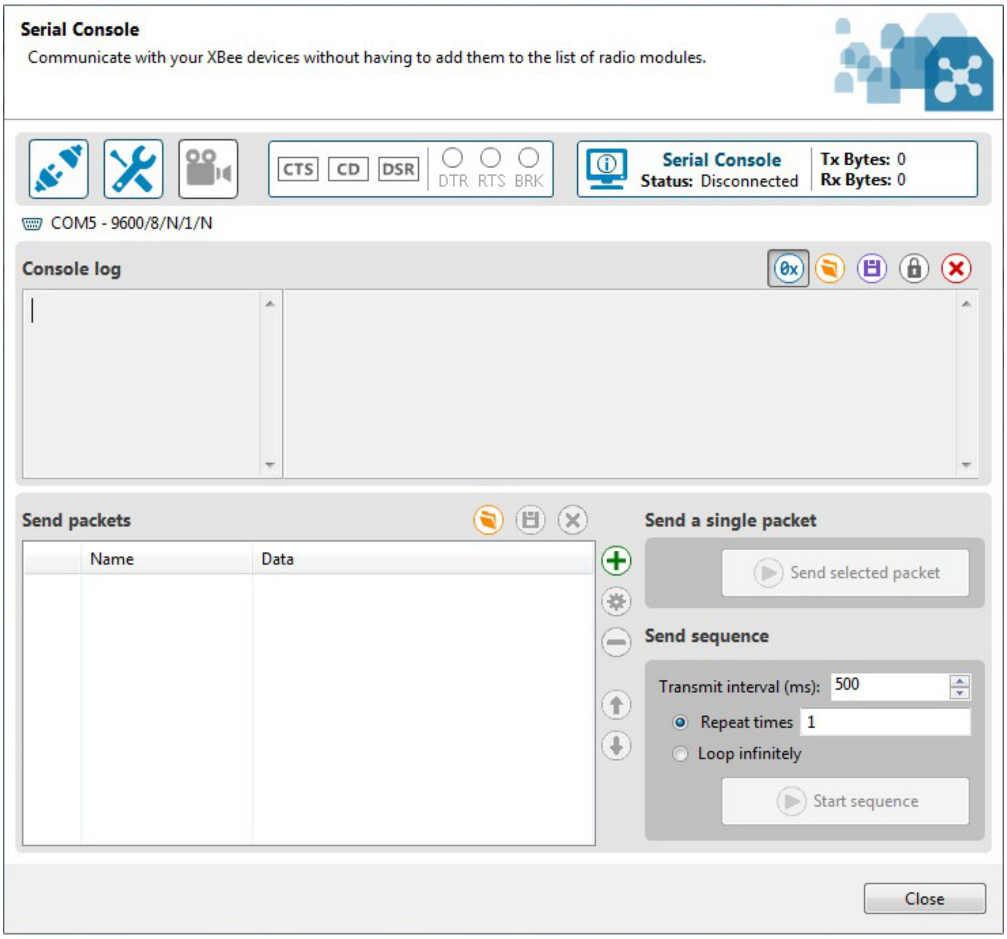
Serial Console tool

The Serial Console tool allows you to interact with your radio modules without first discovering them and adding them to the list of radio modules. The layout and functionality of the tool is similar to the AT Console view.

To open the Serial Console tool, select Serial console from the Tools drop-down menu in the toolbar.



The Serial console opens in a new floating dialog box.


Serial Console

Communicate with your XBee devices without having to add them to the list of radio modules.

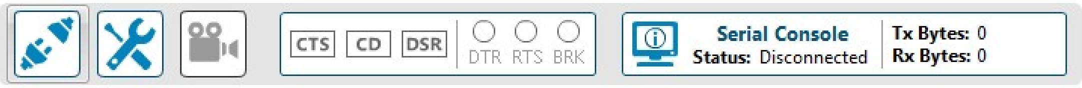
Serial Console
Status: Disconnected
Tx Bytes: 0
Rx Bytes: 0

COM5 - 9600/8/N/1/N

Transmit interval (ms): 500
● Repeat times 1
○ Loop infinitely

Toolbar

The Serial Console has a toolbar control that allows you to perform actions related to the console:



The console toolbar provides the following functionality:

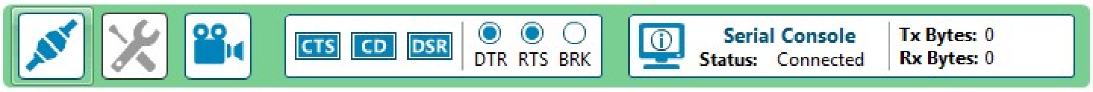
- Connecting and disconnecting the console
- Serial port configuration
- Recording the console session
- Line status indicators control
- Console overview control

Connecting and disconnecting the console

The first time you open a console, it is disconnected by default. This is indicated by the the gray background of the toolbar and the status text reading "Disconnected."

	Click the **Connect** 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.

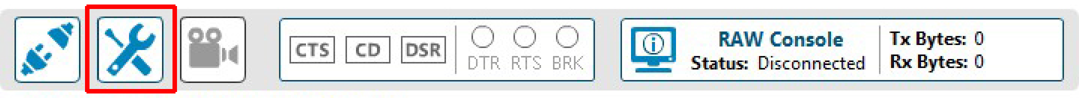
The background color of the toolbar changes to green and the status text changes to "Connected." When the console is connected, all the data traffic of the radio module is captured by the console and displayed in the corresponding controls.



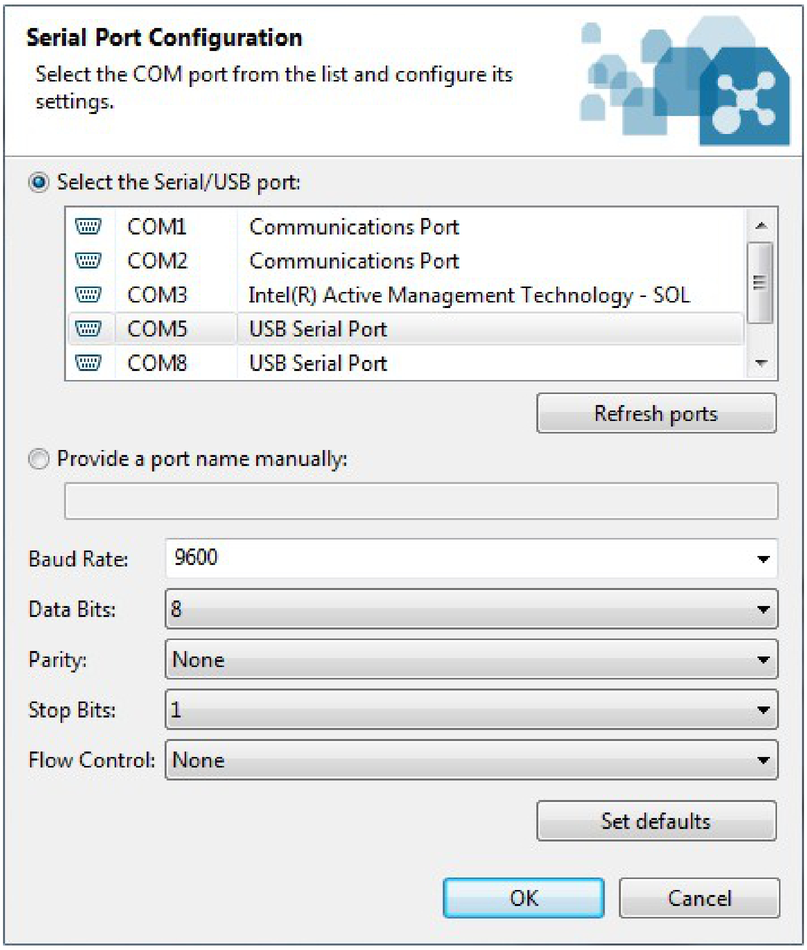
	Click the **Disconnect** button if you want to disconnect the console from the module.

Serial port configuration

Before working with the serial console, click the **Configure button** from the Serial Console toolbar to configure the serial connection (if the console is connected), or click the **Connect button** (if the console is not yet connected):



A new dialog lists all 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
COM2	Communications Port
COM3	Intel(R) Active Management Technology - SOL
COM5	USB Serial Port
COM8	USB Serial Port

○ Provide a port name manually:

Baud Rate: 9600

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: None

Select the XBee device's serial port, or provide it manually. The rest of the parameters are loaded with their default values and you may need to change them.

> Most common serial configuration is:
>
> - **Baud rate**: 9600 or 115200
> - **Data bits**: 8
> - **Stop bits**: 1
> - **Parity**: None
> - **Flow control**: None

> Custom baud rates can only be typed under Windows OS.

Recording the console session

The video icon on the toolbar allows you to record all the console's upcoming and outgoing data as it is sent or received. To start or stop recording, the console must be connected.

When you click the **Start Recording** button, a save file dialog asks for the destination file for the sent and received data.

Data is periodically written to a log file using the **Comma Separated Values** format (CSV), so it can be opened and interpreted by other consumer, business or scientific applications.

A console log file consists of a first record with the console session information and multiple data records with the data sent and received. The console session record has the following fields:

- **Date**: The date and time when the session record started.
- **ID**: Not used in this record; displays a "-" character.
- **Type**: One of **API**, **API2**, **AT** or **Serial**.
- **Description**: Other comma separated console session information such as the Node Identifier of the module associated with the console, the module's MAC Address, Function set and Firmware version, and the COM port information of the module's port.

Example

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

The data records have the following fields:

- **Date**: The date and time when the data was sent or received.
- **ID**: ID number of the packet or API frame.
- **Type**: One of **SENT** or **RECV**.
- **Data**: Data sent or received in Hexadecimal string format.

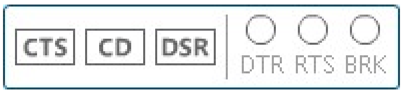
Example

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

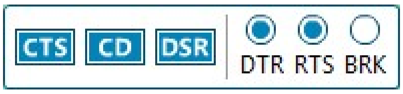
Click the **Stop Recording** button at any time to stop saving any sent or received data.

Line status indicators control

The status line indicators control, located in the middle of the console toolbar, displays the status of the RS-232 hardware flow control lines. Blue status inficates that the line is asserted, while white indicates that it is de-asserted. The status is not displayed until you open the connection of the console; meanwhile the control is disabled.



Connection closed



Connection open

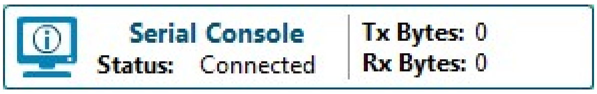
This is a short description of the lines that can be viewed and managed from this control:

CTS	Clear to send	Indicates that the connected device is ready to accept data.
CD	Carrier Detect	Detects the presence of a connection.
DSR	Data Set Ready	Indicates that the connected device is ready for communications.
DTR	Data Terminal Ready	Indicates to the connected device that the terminal is ready for communications.

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.

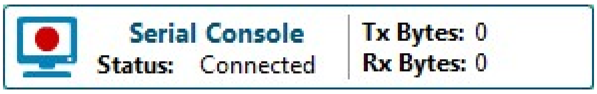
Console overview control

The console status panel, on the right side of the toolbar, displays the console type (AT or API), its status (Connected or Disconnected) and the number of sent and received frames or bytes.



Serial Console
Status: Connected
Tx Bytes: 0
Rx Bytes: 0

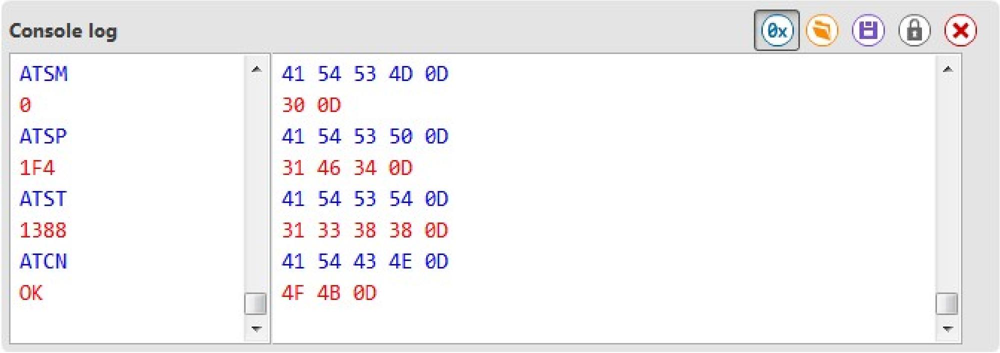
The icon of the console overview control changes when the console session is being recorded, so you can identify when the recording option is activated.



Serial Console
Status: Connected
Tx Bytes: 0
Rx Bytes: 0

Data traffic monitoring section

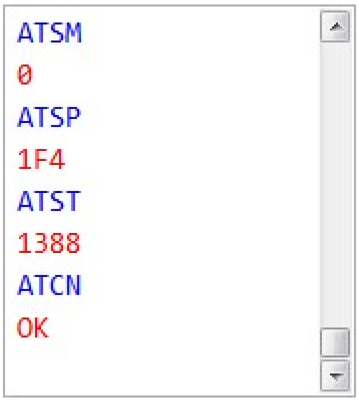
In the **Serial Console**, the main control is a data box that displays all the sent and received data characters. Sent characters are blue and received characters are red.



Console log

```
ATSM     41 54 53 4D 0D
0        30 0D
ATSP     41 54 53 50 0D
1F4      31 46 34 0D
ATST     41 54 53 54 0D
1388     31 33 38 38 0D
ATCN     41 54 43 4E 0D
OK       4F 4B 0D
```

Once data is sent or received by the module, it will be added to the end of the data box.

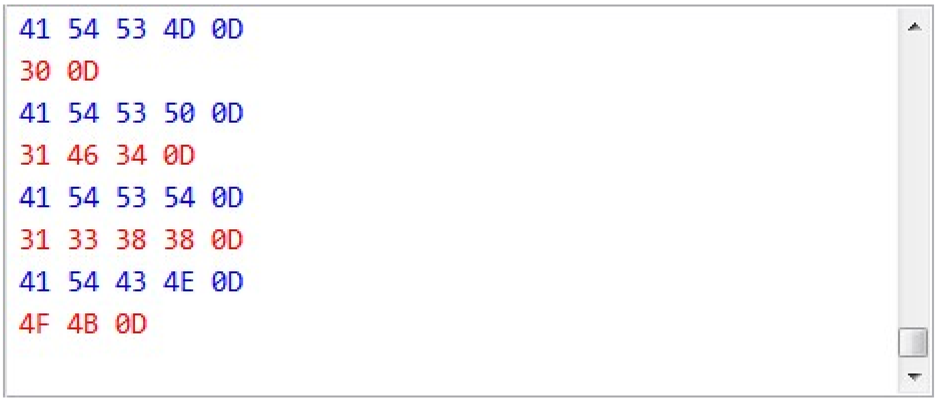


```
ATSM
0
ATSP
1F4
ATST
1388
ATCN
OK
```

As the communication with the radio module does not need any structured data, you can type directly in the data box control. All the characters you enter are automatically sent to the ratio module.

Hexadecimal view

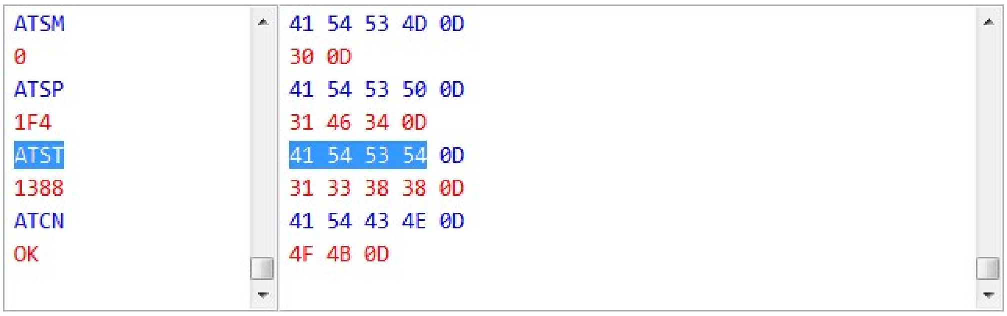
The right side of the data box corresponds to the hexadecimal representation of all the sent and received data characters.



```
41 54 53 4D 0D
30 0D
41 54 53 50 0D
31 46 34 0D
41 54 53 54 0D
31 33 38 38 0D
41 54 43 4E 0D
4F 4B 0D
```

You can show or hide this view by clicking the **Show hexadecimal** button located above the data box.

Note that when you select a character in the data box, the hexadecimal representation is also selected, and vice-versa.



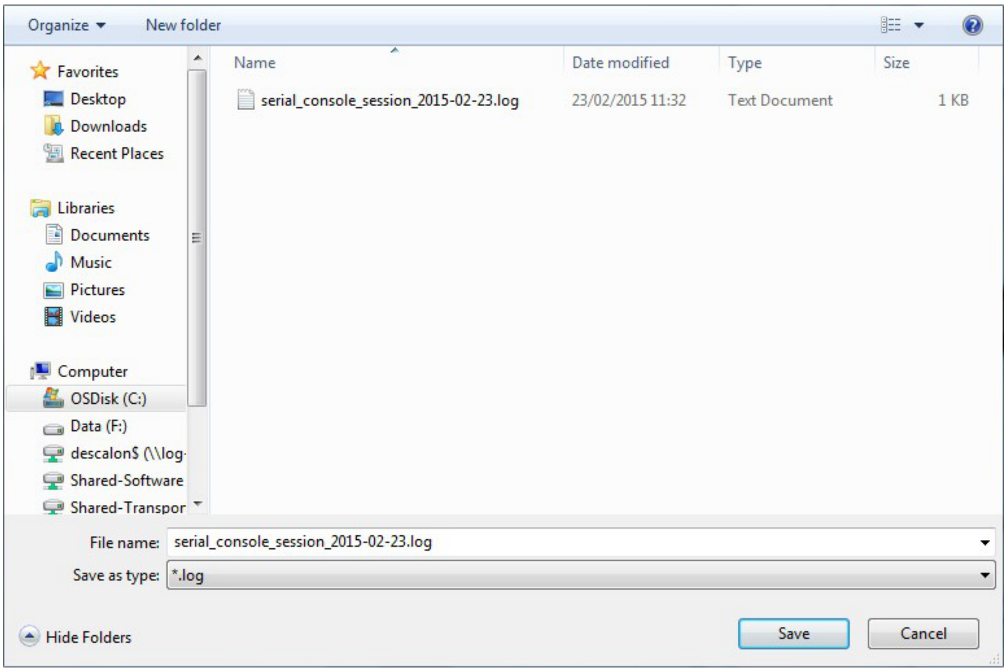
```
ATSM
0
ATSP
1F4
ATST
1388
ATCN
OK
```

```
41 54 53 4D 0D
30 0D
41 54 53 50 0D
31 46 34 0D
41 54 53 54 0D
31 33 38 38 0D
41 54 43 4E 0D
4F 4B 0D
```

Saving a console session

Click the **Save console session** button to save the sent and received data from the console session.

A new **Save file** dialog requests the name and path of the file containing the console session. By default, XCTU gives the physical address of the device followed by the current date.



Name	Date modified	Type	Size
serial_console_session_2015-02-23.log	23/02/2015 11:32	Text Document	1 KB

File name: serial_console_session_2015-02-23.log

Save as type: *.log

Click **Save** to save the console session file.

Loading a console session

📂	You can also load a console session using the **Load console session** button. This button opens XCTU's **Load console session** tool.

For more information about how to use this tool, see the Load console session tool topic.

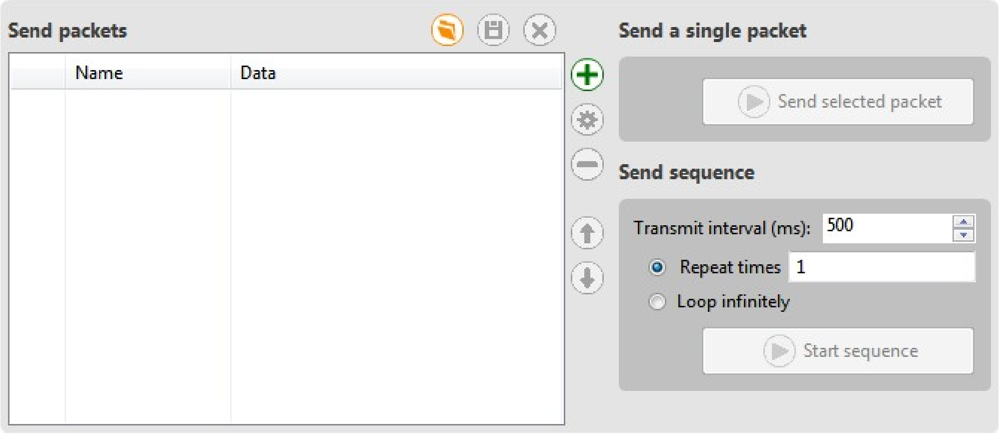
Other traffic monitoring features

🔒	The data box also contains a button to lock the scroll. If you click the **Lock scroll** button, the data box won't automatically scroll to the data character when it is sent or received. Instead, you will have to manually scroll to it.

Clicking the button again will unlock the scroll.

✖	The red x icon lets you clear the data box. If you click the **Clear data** button, all the data characters are cleared. Note that this action will also clear their hexadecimal representation.

Send data packets section



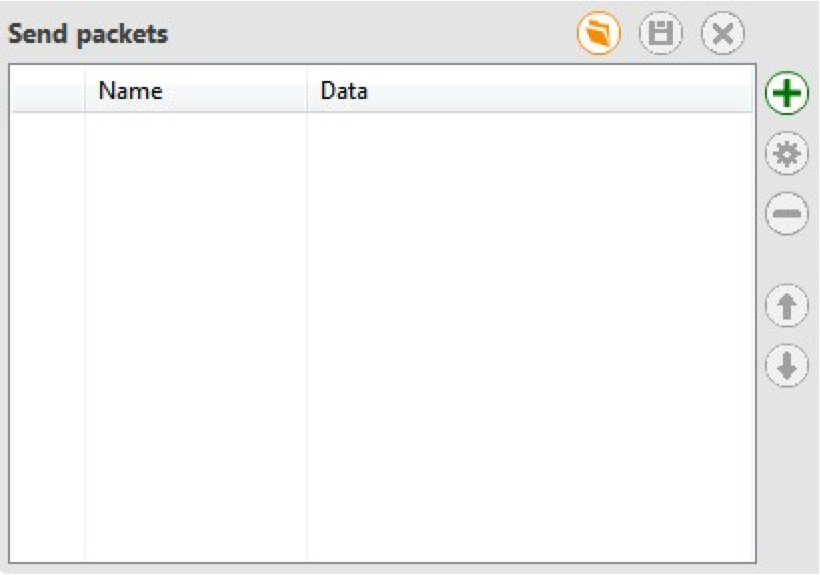
Transmit interval (ms): 500

● Repeat times 1

○ Loop infinitely

The send data packets section lets you send a group or groups of characters to the radio module. When you write in the data box, XCTU sends the data as individual characters. Sending a data packet sends all the characters in a single operation.

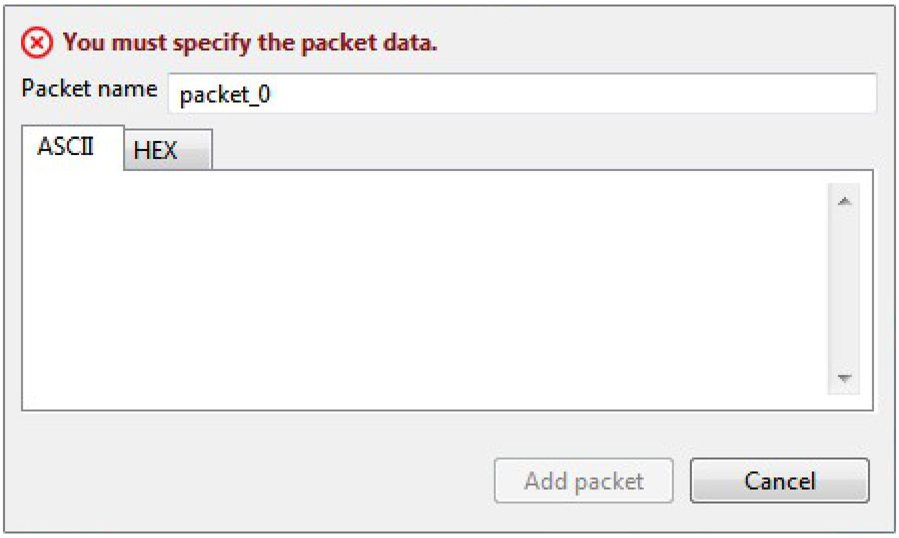
By default, the data packets list is empty.



Creating a data packet

Follow these steps to create a new data packet and add it to the list:

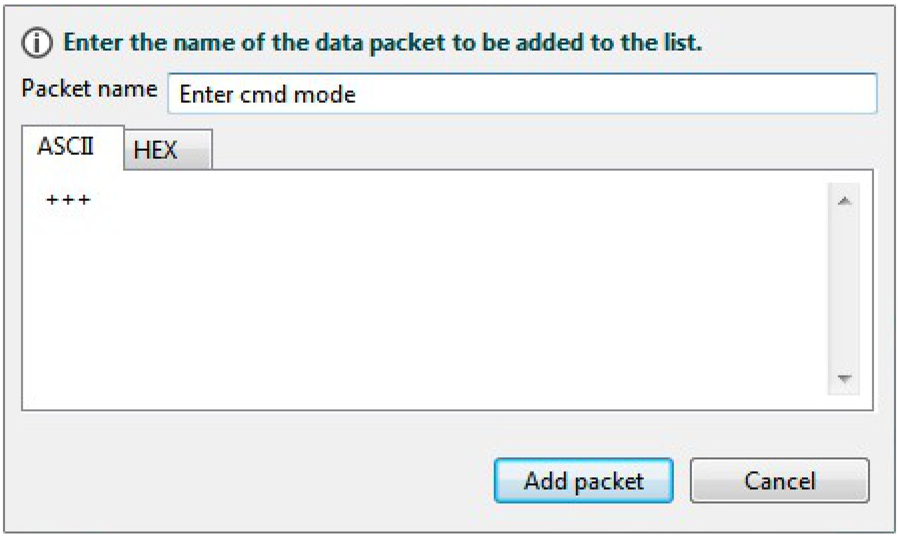
1. Click the **Add new packet** button. ⊕
2. The **Add new packet** dialog appears.



You must specify the packet data.

Packet name: packet_0

3. Enter a symbolic name for the data packet and then type in your commands. You can view the contents of the packet in ASCII or hexadecimal using the tabs.

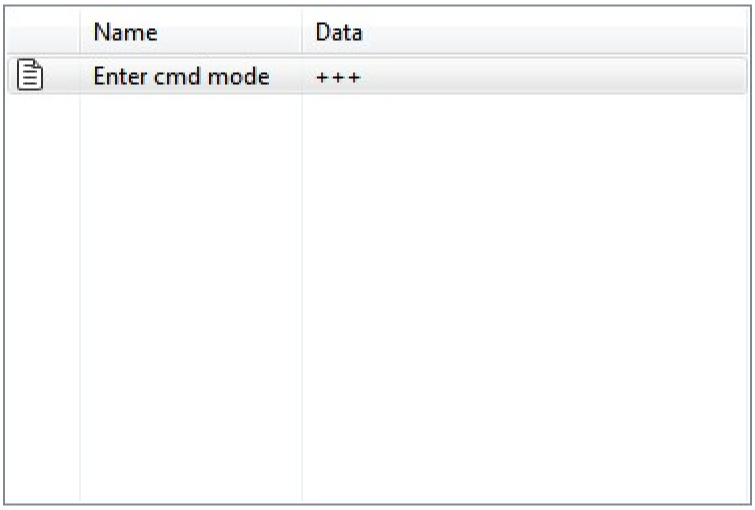


Enter the name of the data packet to be added to the list.

Packet name: Enter cmd mode

```
+++
```

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



Name	Data
Enter cmd mode	+++

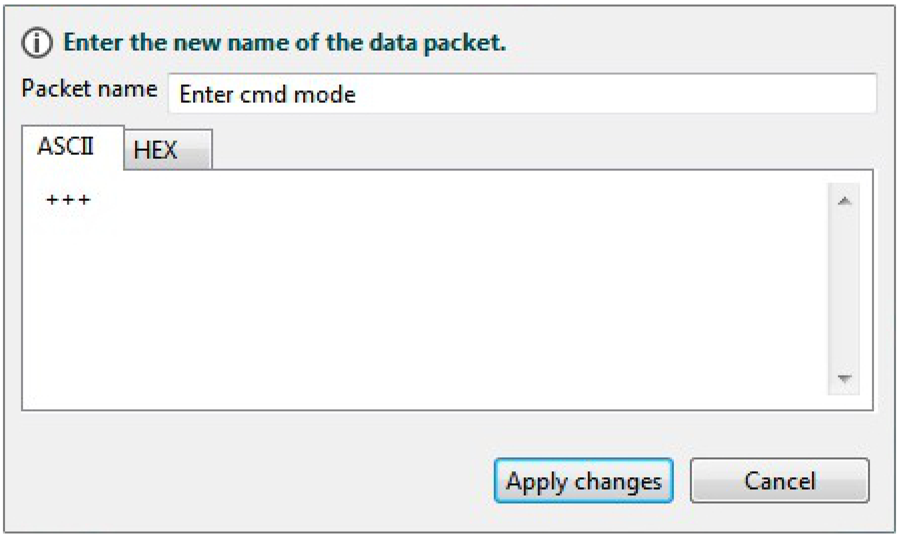
You can repeat the same operation to add more data packets to the list.

Managing data packets

Once a data packet is added to the list, you can perform the following actions with it:

- **Edit a packet**. To edit a packet, select the desired packet name and click the **Edit selected packet** button.

  You can change the name of the packet and its content in the edit dialog box.

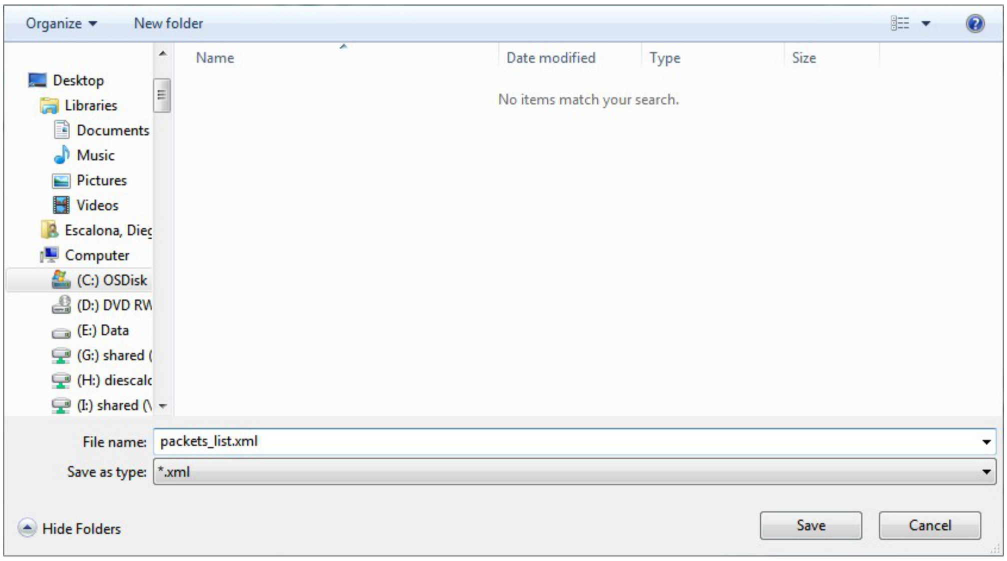
  Enter the new name of the data packet.

  Packet name: Enter cmd mode

  +++

  Click **Apply changes** to save the changes of the packet.
- **Remove a packet**. To remove a data packet from the list, select it and click the **Remove selected packet** button.
- **Change the order of a packet**. To change the order of a list of packets, select the data packet you want to move and click the **Move up** or **Move down** button.
- **Save the list of packets**. To save the list of packets you have created to be used in future sessions or on different PCs, click the **Save packets list** button.

  The **Save file** dialog box requests the name and path of the data packets file.



No items match your search.

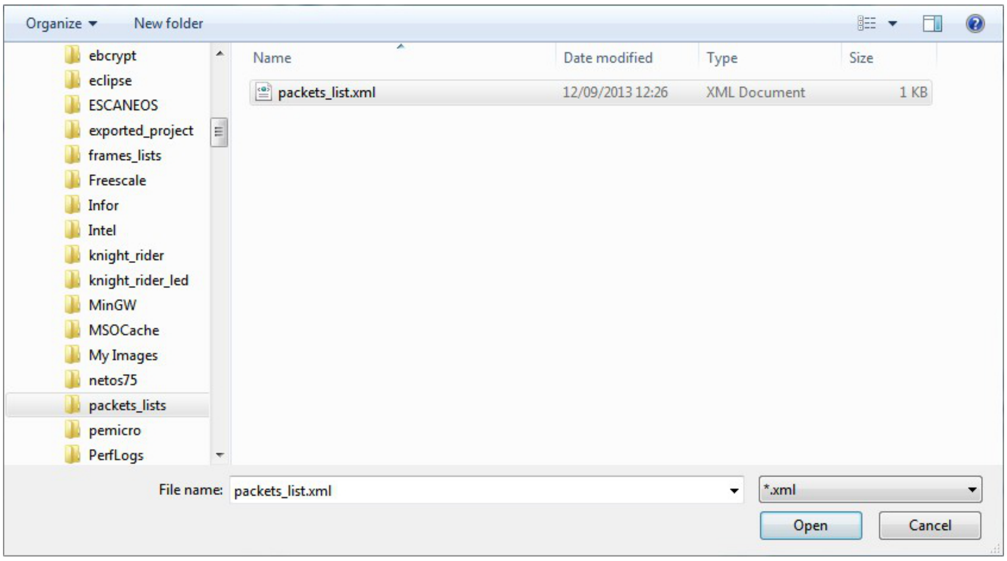
File name: packets_list.xml

Save as type: *.xml

Enter a name and a path for the file and click **Save** to save the data packets list file in XML format.

- **Load a list of packets**. To load a list of saved data packets, click the **Load packets list** button.

  A **Load file** dialog box requests the file containing the list of packets to load.



Name	Date modified	Type	Size
packets_list.xml	12/09/2013 12:26	XML Document	1 KB

File name: packets_list.xml

*.xml

  Select the correct file and click **Open** to load the list of data packets.

  If the packets are correctly loaded they appear in the list. If not, an error message states the cause of the problem.

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

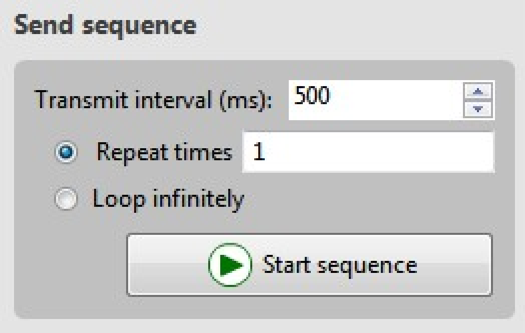
Sending a single data packet

To send a single packet, select it from the list of data packets and then click the **Send selected packet** button on the right.

The sent data appears in the Data traffic section.

Sending a sequence of data packets

You can send a sequence of data packets to the radio module. The sequence is defined by the list of packets to send and the send sequence options located next to the list.


Send sequence

Transmit interval (ms): 500

● Repeat times 1

○ Loop infinitely

Configure the following settings in the send sequence options:

- **Transmit interval**: This is the time in milliseconds between packets. The minimum value is 100ms and the maximum value is 60000ms (1 minute)
- **Repeat times**: If you check this option, you must enter the number of times the sequence should be repeated or sent. By default this value is 1.
- **Loop infinitely**: If you prefer, you can send the sequence of packets endlessly by checking this option.

> The send sequence options are also saved when you save the list of packets, and are loaded when you load a list of data packets.

The sent and received data appears in the Data traffic section.

When you have all the send sequence options configured, you can start sending the list of data packets by clicking the **Start sequence button** located below the options.

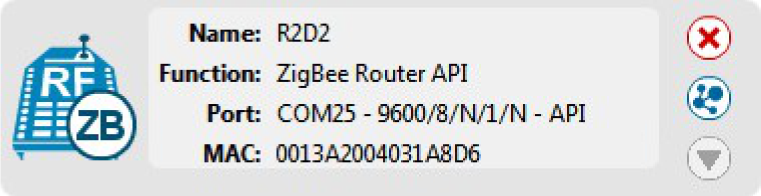
You can also stop the sending process at any time by clicking the **Stop sequence** button.

> You can open as many Serial console dialogs as you want.

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.



Name: R2D2
Function: ZigBee Router API
Port: COM25 - 9600/8/N/1/N - API
MAC: 0013A2004031A8D6

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 the role that it has configured.

After executing a remote node discovery process with a local DigiMesh radio module, the device becomes unresponsive.

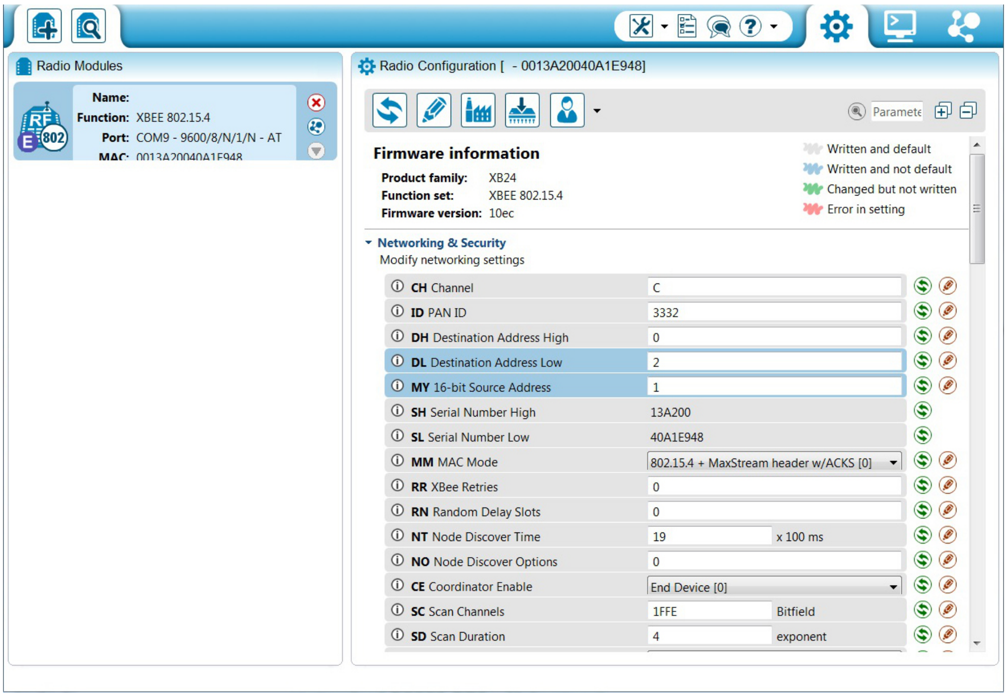
In DigiMesh protocol, when you perform a Node Discovery (ND) or a Find Neighbors (FN) operation, the module will not process any AT command until its configured NT time expires.

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 lower than 1 second and XCTU is not able to find it or read its settings.

It is very complicated to communicate with end devices configured in API operating mode that are awake for less than 1 second. 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 awake the module for 30 seconds and XCTU will be able to communicate with it.

Texts within XCTU are too big in Windows OS and they affect the usability of the application.



Radio Configuration [- 0013A20040A1E948]

Name:
Function: XBEE 802.15.4
Port: COM9 - 9600/8/N/1/N - AT
MAC: 0013A20040A1E948

Firmware information

Product family: XB24
Function set: XBEE 802.15.4
Firmware version: 10ec

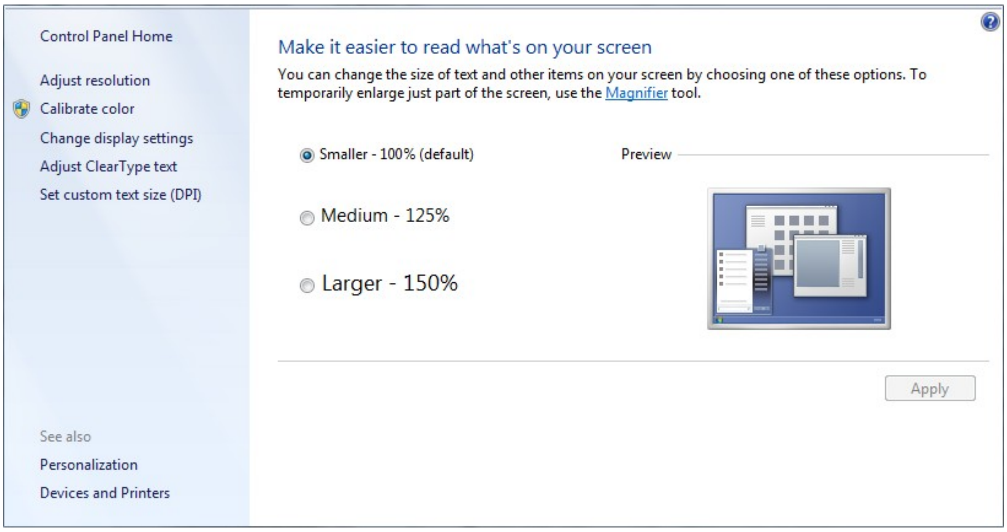
Networking & Security
Modify networking settings

CH Channel	C
ID PAN ID	3332
DH Destination Address High	0
DL Destination Address Low	2
MY 16-bit Source Address	1
SH Serial Number High	13A200
SL Serial Number Low	40A1E948
MM MAC Mode	802.15.4 + MaxStream header w/ACKS [0]
RR XBee Retries	0
RN Random Delay Slots	0
NT Node Discover Time	19 x 100 ms
NO Node Discover Options	0
CE Coordinator Enable	End Device [0]
SC Scan Channels	1FFE Bitfield
SD Scan Duration	4 exponent

Some devices (most likely laptops and tablets) use to have small screens that support high resolution values. In order to maintain the optimal resolution of the screen, Windows automatically modifies the size of some objects and texts of the applications setting them to 125% of their normal size.

There are objects and controls within XCTU that were designed with a fixed size, so configuring the size of texts, icons and other objects with a value greater than 100% can cause to see some cut texts and controls within XCTU. For an optimal user experience of the tool we recommend you to calibrate their size with a value of 100%. You can do it by following these steps:

1. Go to **Start > Control Panel**
2. Select the **Appearance and Personalization** category
3. Select **Display** option
4. Choose the **Smaller - 100%** option and click **Apply**



Make it easier to read what's on your screen

You can change the size of text and other items on your screen by choosing one of these options. To temporarily enlarge just part of the screen, use the Magnifier tool.

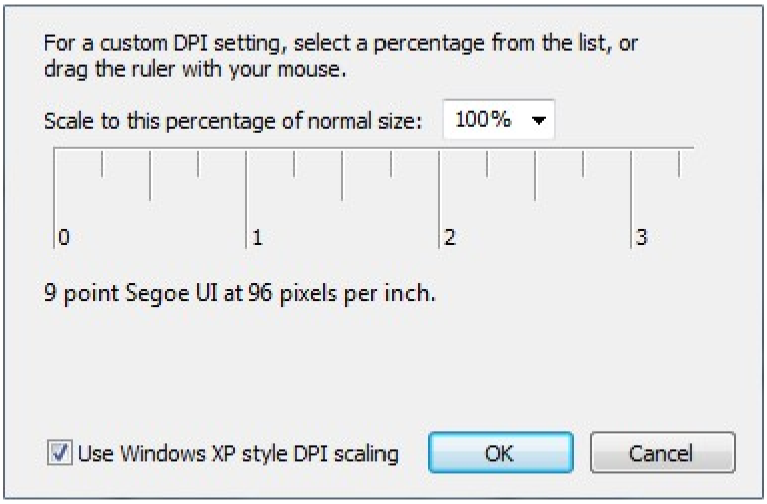
● Smaller - 100% (default)

○ Medium - 125%

○ Larger - 150%

Apart from the three preset choices for making text, icons, and other items larger than normal size, Windows also gives you the possibility to give a custom value between 100% and 150%. To do so follow these steps:

1. Go to **Start > Control Panel**
2. Select the **Appearance and Personalization** category
3. Select **Display** option
4. Select the **Set custom text size (DPI)** option in the left pane
5. Click the scale (ruler), drag the setting to whatever percentage size increase you want. Then click **OK**



For a custom DPI setting, select a percentage from the list, or drag the ruler with your mouse.

Scale to this percentage of normal size: 100%

9 point Segoe UI at 96 pixels per inch.

☑ Use Windows XP style DPI scaling

You will need to log off and log in again to make the changes effective.

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 XBee 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.

Networking

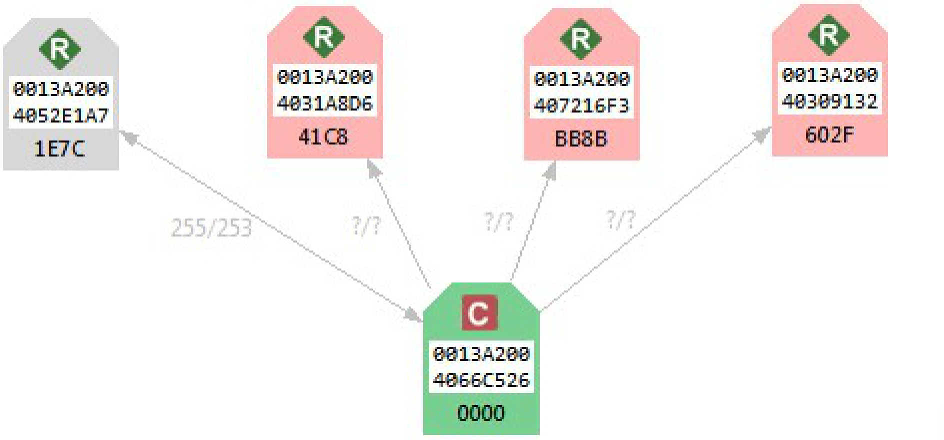
In a relatively big network, not all remote devices are found after clicking the search button of a local radio module.

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:

- Add those that have been found
- Click again the search button of the local device
- When asked whether to clear the list of remote modules, click **NO**
- Wait for new modules to be discovered

Repeat this process until all the modules on your network are found.

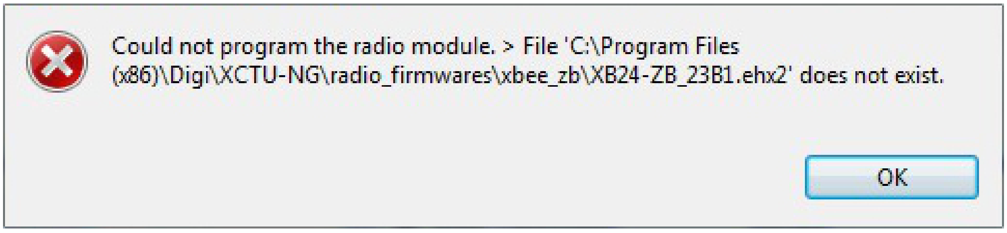
When discovering a network, some of the nodes are displayed in light red color. What does it mean?.



Radio modules painted red 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. That's why, when a new discovery is performed, the process thinks that the module is still there but when trying to verify the connection it does not answer or is unreachable.

Firmware Update

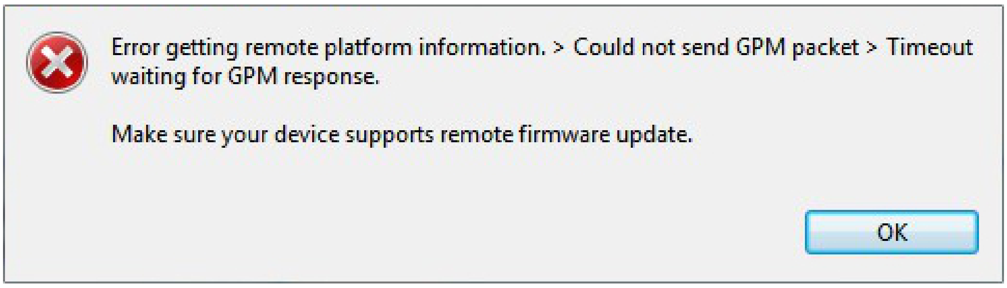
When attempting to update the firmware of a local radio module the following error appears indicating that the firmware binary does not exist:



Could not program the radio module. > File 'C:\Program Files (x86)\Digi\XCTU-NG\radio_firmwares\xbee_zb\XB24-ZB_23B1.ehx2' 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 the XCTU. You should contact with Digi support so they can provide you with an updated firmware version.

When attempting to update the firmware of a remote radio module, following error message appears:



Error getting remote platform information. > Could not send GPM packet > Timeout waiting for GPM response.

Make sure your device supports remote firmware update.

Probably your remote radio module does not support remote firmware update. Here is a complete list with the radio modules that support the remote firmware update feature:

- XBee/XBee-PRO ZB
- Programmable XBee-PRO ZB
- XBee/XBee-PRO ZB SMT
- Programmable XBee-PRO ZB SMT
- XBee-PRO 900HP
- Programmable XBee-PRO 900HP
- XBee 865LP
- Programmable XBee 865LP
- XBee 868LP
- Programmable XBee 868LP
- XLR PRO Radio Solution

Known issues and limitations

This is the list of current known issues and limitations of XCTU:

- XCTU is not compatible with the following Digi RF devices:
    - **XTream** 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, range testing, and so on. 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 your 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 will not be able to perform Network discovery in the Network working mode.
- XLR PRO Radio Solution and programmable XBee radio modules do not support the recovery feature provided by XCTU.
- UNIX-based operating systems allow you to open multiple instances of the same port. While Windows avoids this scenario, it can produce unexpected behaviors if you have the same port open simultaneously by different apps in Linux.

How-to articles

- How-to: Update the firmware of your modules
- How-to: Visualize your network

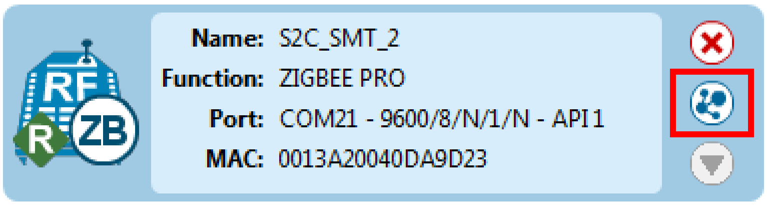
How-to: Update the firmware of your modules

You can use the Update firmware tool of XCTU to update firmware in both local and remote modules. Refer to the Updating firmware section for more information.

Add the XBee to XCTU

You must have a local XBee 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 XBee in the same network. With a local module properly attached to your computer, follow these steps:

1. Add the XBee attached to your computer to XCTU so it is displayed in the Radio Modules list.
2. To add your remote XBee 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.



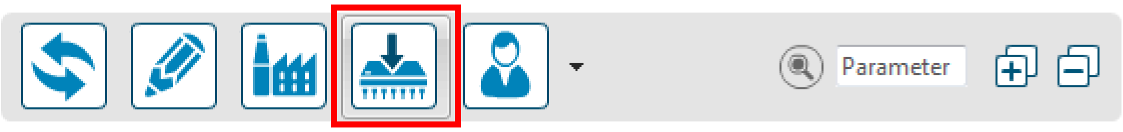
Name: S2C_SMT_2
Function: ZIGBEE PRO
Port: COM21 - 9600/8/N/1/N - API 1
MAC: 0013A20040DA9D23

    c. When a remote XBee 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.

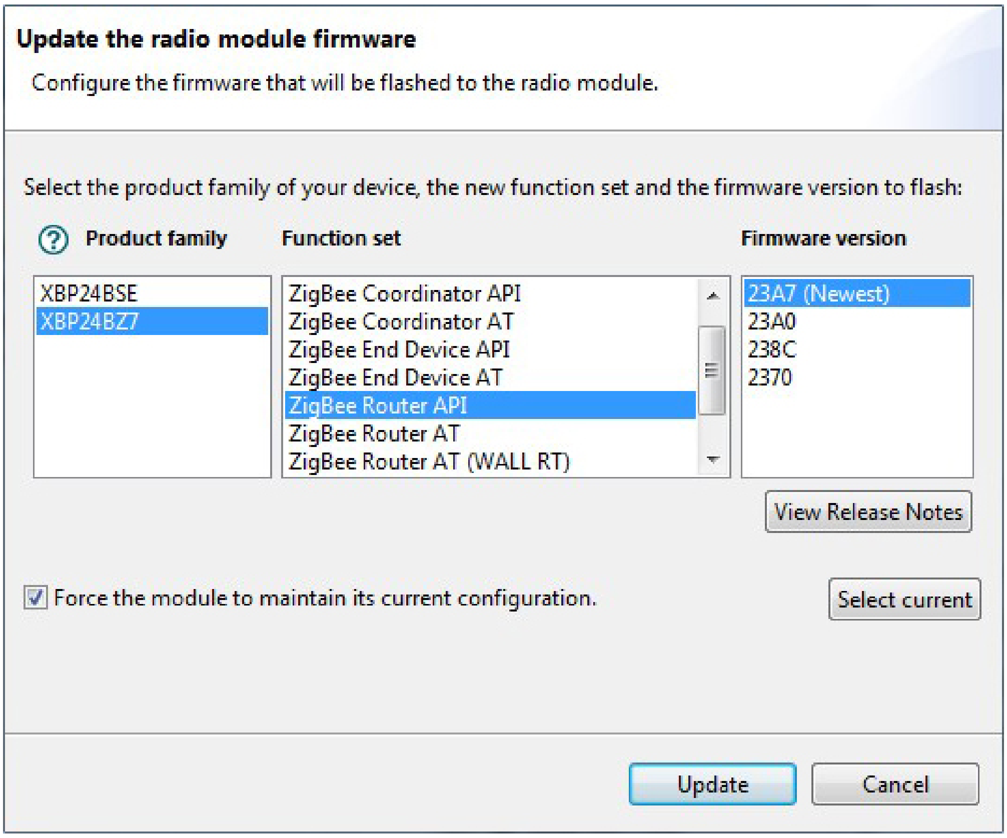
> Once you add your XBee to the Radio Modules list of XCTU, the update process is exactly the same regardless of whether it is a local or remote module.

Update the firmware

1. Select a local or remote XBee from the Radio Modules list. Click **Update firmware**.



2. The Update the radio module firmware dialog is displayed. Select the product family of the XBee, the function set, and firmware version.



Update the radio module firmware

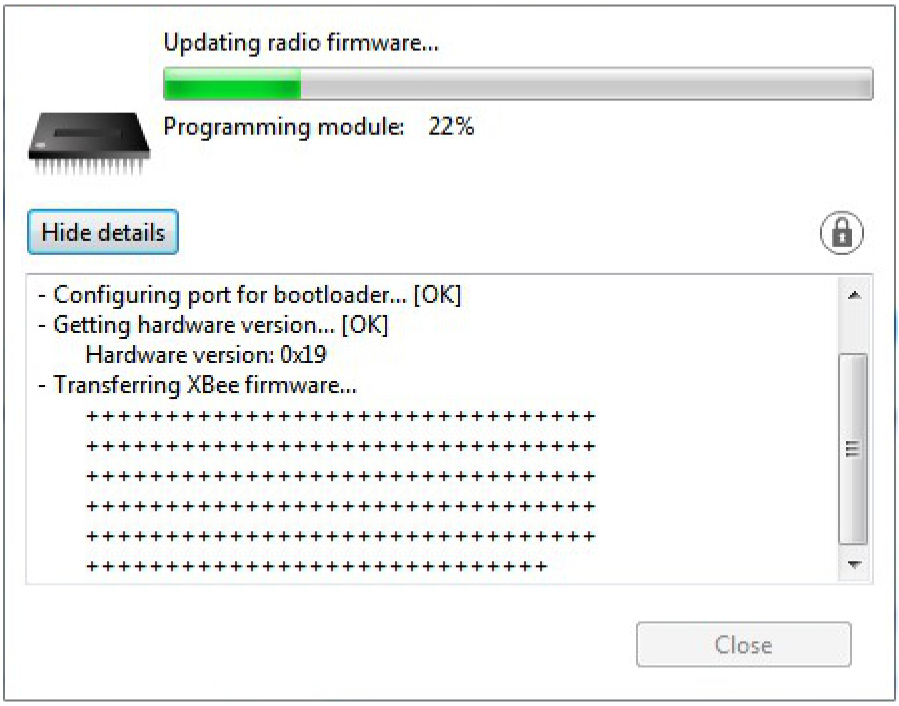
Configure the firmware that will be flashed to the radio module.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
XBP24BSE XBP24BZ7	ZigBee Coordinator API ZigBee Coordinator AT ZigBee End Device API ZigBee End Device AT ZigBee Router API ZigBee Router AT ZigBee Router AT (WALL RT)	23A7 (Newest) 23A0 238C 2370

☑ Force the module to maintain its current configuration.

3. Click **Update**. A progress dialog displays the status of the update. Click **Show details** to review the steps of the firmware update process.

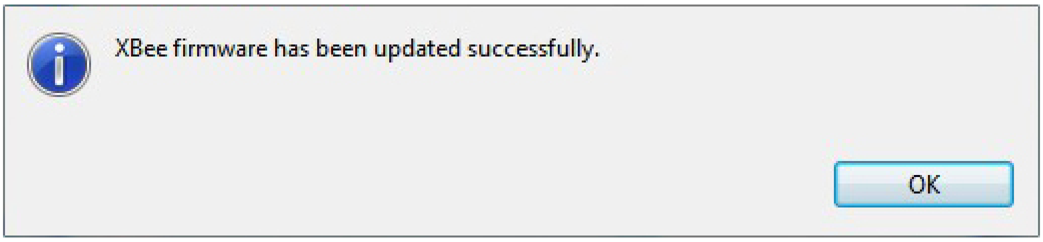


Updating radio firmware...

Programming module: 22%

```
- Configuring port for bootloader... [OK]
- Getting hardware version... [OK]
    Hardware version: 0x19
- Transferring XBee firmware...
    ++++++++++++++++++++++++++++++++
    ++++++++++++++++++++++++++++++++
    ++++++++++++++++++++++++++++++++
    ++++++++++++++++++++++++++++++++
    ++++++++++++++++++++++++++++++++
    +++++++++++++++++++++++++++++
```

4. If the process finishes without an error, a message indicates the firmware was updated successfully.



XBee firmware has been updated successfully.

The following video shows you how to perform local and remote firmware updates: https://www.youtube.com/embed/EYB5P6YZeQk

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 ZB
   - Programmable XBee-PRO ZB
   - XBee/XBee-PRO ZB SMT
   - Programmable XBee-PRO ZB SMT
   - XBee-PRO 900HP
   - Programmable XBee-PRO 900HP
   - XBee 865LP
   - Programmable XBee 865LP
   - XBee 868LP
   - Programmable XBee 868LP
   - XLR PRO Radio Solution
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. To do so, follow the instructions described in the Recovery tool topic.

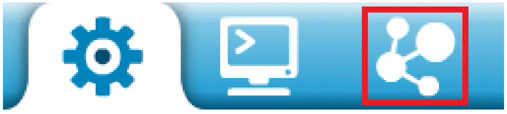
How-to: Visualize your network

XCTU's Network working mode allows you to discover and visualize the topology and interconnections of your network. Refer to the Manage your radio network section for more information.

Scan the network

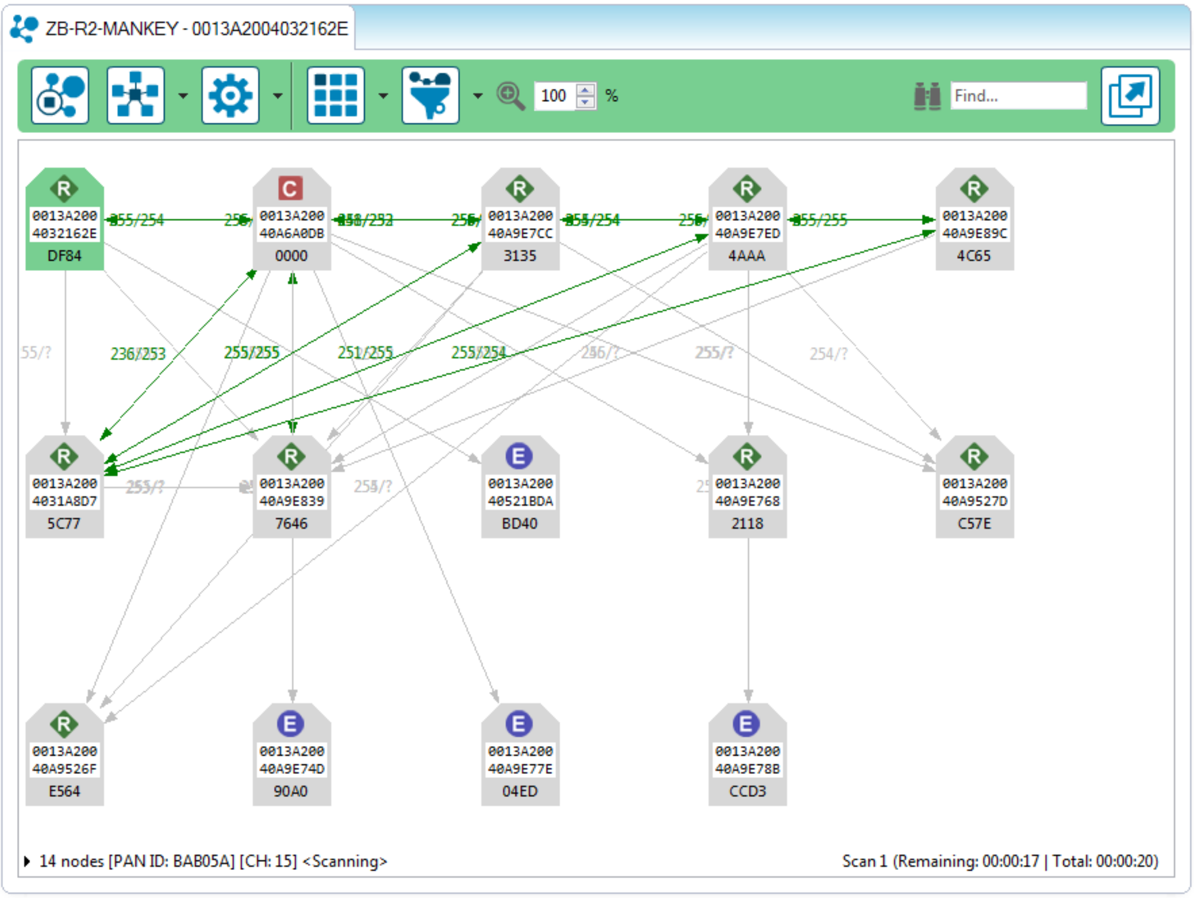
To discover the network, follow these steps:

1. Once you have added the XBee module to XCTU, switch to **Network working mode**.



2. Click **Scan the radio module network** to start the network discovery process.

The modules are dynamically added as they are discovered, showing their connections and link quality.



14 nodes [PAN ID: BAB05A] [CH: 15] <Scanning>

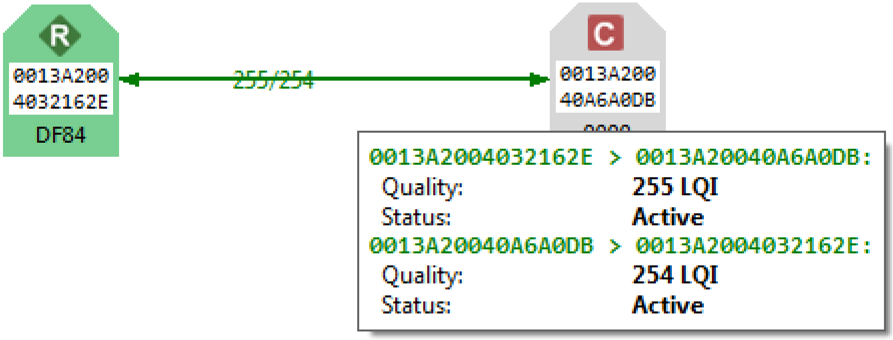
Scan 1 (Remaining: 00:00:17 | Total: 00:00:20)

Sometimes one scan is not enough to discover the whole network if, for example, modules are sleeping or the network is very large. XCTU will continue to perform consecutive scans until you click **Stop scanning the network** .

Explore the network

Each XBee module displays some information about itself, such as its role or its address. Hover the cursor over one of them to view more detail.

The modules are connected by lines which indicate 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.



0013A2004032162E > 0013A20040A6A0DB:
Quality: 255 LQI
Status: Active
0013A20040A6A0DB > 0013A2004032162E:
Quality: 254 LQI
Status: Active

Other features

Network mode provides features to help you examine the network:

	Table view	Display the XBee modules as rows in a table.
	Screenshot	Save an image of the network to your computer.
	Export table	Export network information in a tabular file format.
	Settings	Configure the network-related parameters.
	Layout	Change the network layout. There are six types of layout: composite, spring, vertical tree, horizontal tree, grid (default), and radial.
	Filter links	Hide or display connections between RF modules based on their quality.
	Zoom	Change the zoom level of the network.
	Find box	Search for a particular module of the network.

The following video demonstrates how to scan and explore a network: https://www.youtube.com/embed/QVF8HvJ0vAE