UBIQUA 2.0 USER GUIDE

UPDATED MAY 2018

Contents

Chapter 8: Graphic View

Chapter 9: Properties View

Chapter 10: Event View

Chapter 11: Output View

Chapter 12: Ubiqua Services

Chapter 15: Setting Preferences

Chapter 16: Supported Hardware

Chapter 17: Remote Sniffers

Chapter 18: Supported Protocols

Chapter 19: Custom Payloads

Chapter 20: Troubleshooting

UBIQUA 2.2

Document generated: May 9th 2018

Chapter 1: Overview

Welcome to Ubiqua

Ubiqua Protocol Analyzer is a tool designed to assist in the various phases of Wireless Sensor Network application development: debugging, testing, and deployment. Ubiqua integrates the top IEEE 802.15.4-based protocol decoder and a wide set of analysis features to provide a powerful, user-friendly, fast, and scalable debugging environment.

This document guides you through the installation process and allows you to learn the basic and advanced features of Ubiqua. If you have additional questions or comments we will be happy to assist you at support@ubilogix.com.

Chapter 2: Getting Started

To start following this guide you need to download and install Ubiqua on your computer. To download the latest version please visit the website at: http://www.ubilogix.com/ubiqua.

The website will ask you to sign in using your Ubilogix account. Remember to read the End User License Agreement before using Ubiqua and its related services. If you don't have an account you can create one at: http://www.ubilogix.com/register. The registration is free and it will take you no more than a couple of minutes.

Requirements

Ubiqua requires the following resources to work properly in your system. Make sure your system meets the minimum requirements before you install the software.

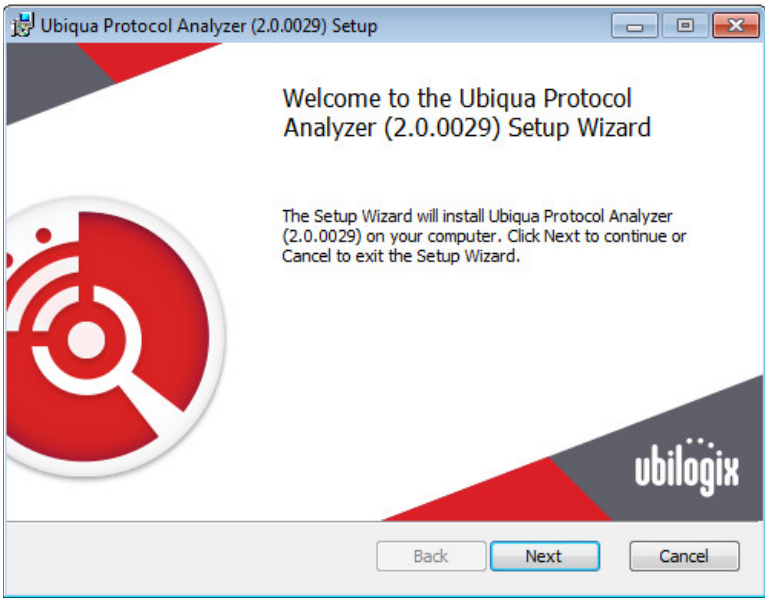
- Windows 7, Windows 8.1, Windows 10 or later.
- Microsoft .NET Framework 4.6.1 Available as a free download at: https://www.microsoft.com/en-us/download/details.aspx?id=49981.
- Internet access.

Windows XP users have the option to install Ubiqua from the 1.3 version (build 2142) until the 1.5 version, `Microsoft .NET Framework 4.0` is required.

Installation

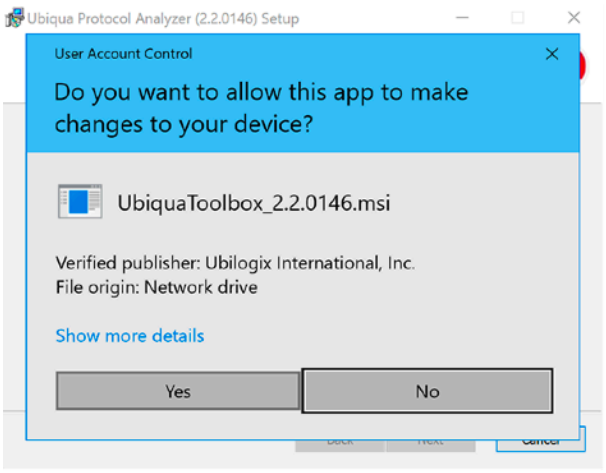
After downloading the Ubiqua Installer, navigate to the location where it was saved and double-click on it. The installation process will begin with the Setup Wizard window.

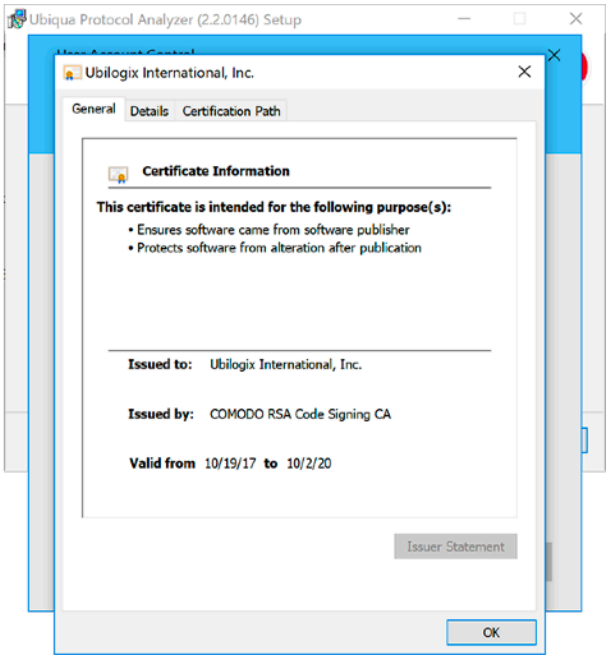
Click on the Next button and then read the End-User License Agreement. You need to accept to the License Agreement by clicking the "I accept the terms in the License Agreement" check box in order to continue with the installation. Following this action a 'User Control Dialog' will appear asking you to accept the changes on your device, in this window you will find details about the Ubilogix certificate, click on the 'Show more details' link to expand the information about this.



Next you have to choose a location on your hard disk drive where Ubiqua will be installed. By default it will create a folder named "Ubilogix" and a subfolder "Ubiqua Protocol Analyzer" within the Program Files folder and place all needed files for the software to operate in there. We recommend you don't change the default destination folder unless you have a better option that suits your needs. To continue installing Ubiqua, click on the Next button.

Now the wizard is ready to copy all the files. Click on the Install button. Depending on your user credentials the installer might ask you for permission to make changes, in such case



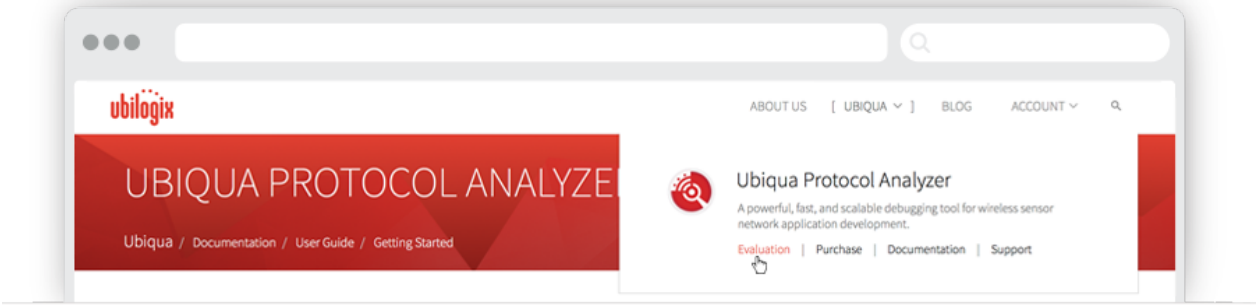


please follow the instructions on screen. The installation process should not take more than a few seconds. On the final screen you will be informed that the installation process is completed, click the Finish button to close the setup wizard window.

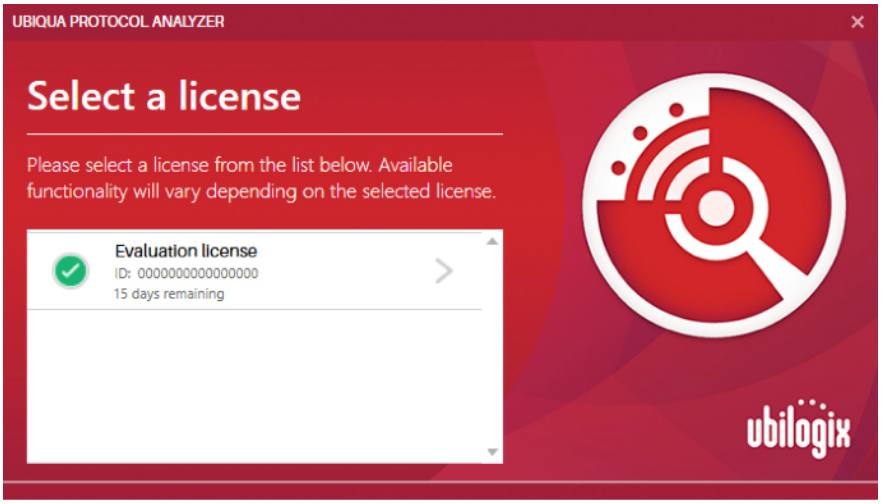
Congratulations, you now have Ubiqua installed on your system!

Starting a 21/day evaluation

To activate your free 21/days evaluation you need a Ubilogix account; if you don't have it, go to the Ubilogix website and from the top menu go to the "Products" option, select the "Evaluation" link and follow the instructions, if you decide to buy a license select the “Purchase” link, then click the “Create a subscription” button and configure your plan, for more details about our licensing methods you can read about it our FAQ Section.



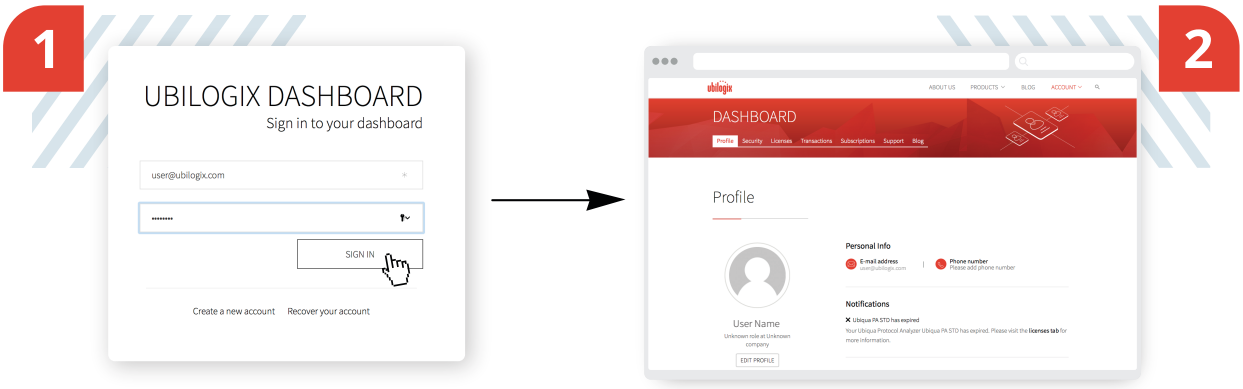
You should get your activation code by email within a few minutes. To be able to use the evaluation you have to activate the access code; for this just follow the instructions in "Validating an access code" section. Please note that only one evaluation-license is provided per user account. The evaluation version of Ubiqua is the fully functional version, but with a 1000 packets limit when capturing, or opening capture files. Once the evaluation period ends, the Welcome window will only show the options to upgrade or purchase a license.



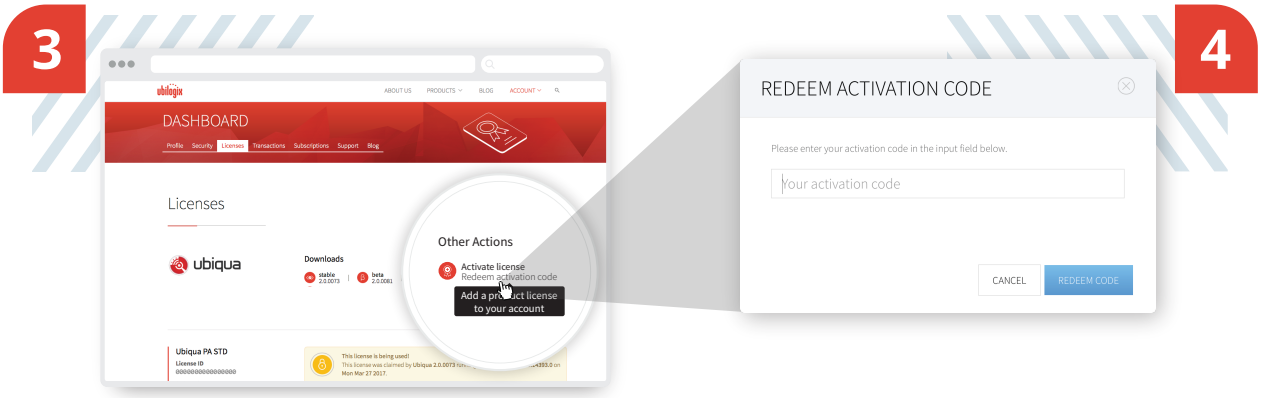
Validating an access code

To activate your copy of Ubiqua, you need a valid Activation Code and a working Internet connection. The Activation Code is sent to you by e-mail after a successful Ubiqua purchase. If you have purchased Ubiqua but you haven't received your Activation Code, please check first the spam folder of your mailbox; then (if needed) write us to: support@ubilogix.com

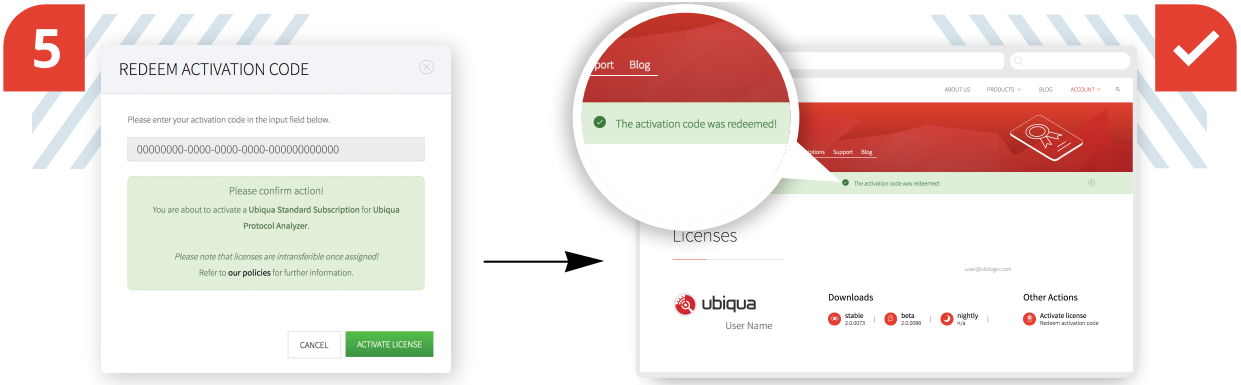
To proceed with the activation process, login to your Ubilogix account pointing your web browser to www.ubilogix.com/login and type your access credentials (email and password) Click the Sign in button, by default you will be presented with your profile page.



From the Dashboard sub menu click the "Licenses" tab and then the "Redeem activation code" link. A dialog will appear where you can type or paste your activation code, following this action click the “Redeem Code" button.

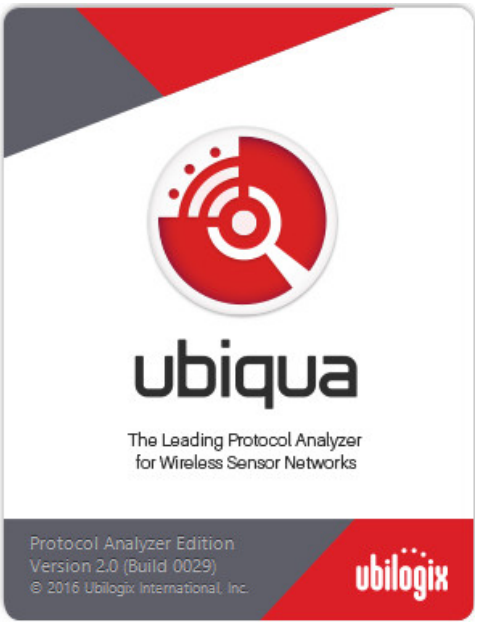


Finally click the “Activate License” button, if the process succeeds your new Ubiqua License will be displayed, and a success message will be shown.



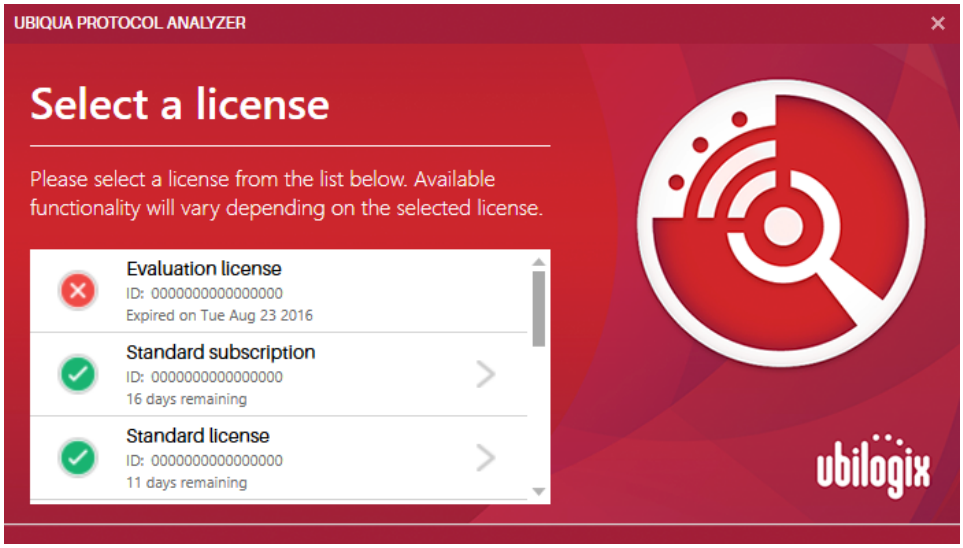
Starting Ubiqua

To start Ubiqua either double-click on the Ubiqua Protocol Analyzer shortcut icon on your desktop or click on the Ubiqua Protocol Analyzer shortcut located in the Start > All Programs > Ubilogix group. A Splash screen featuring the Ubiqua logo will appear.

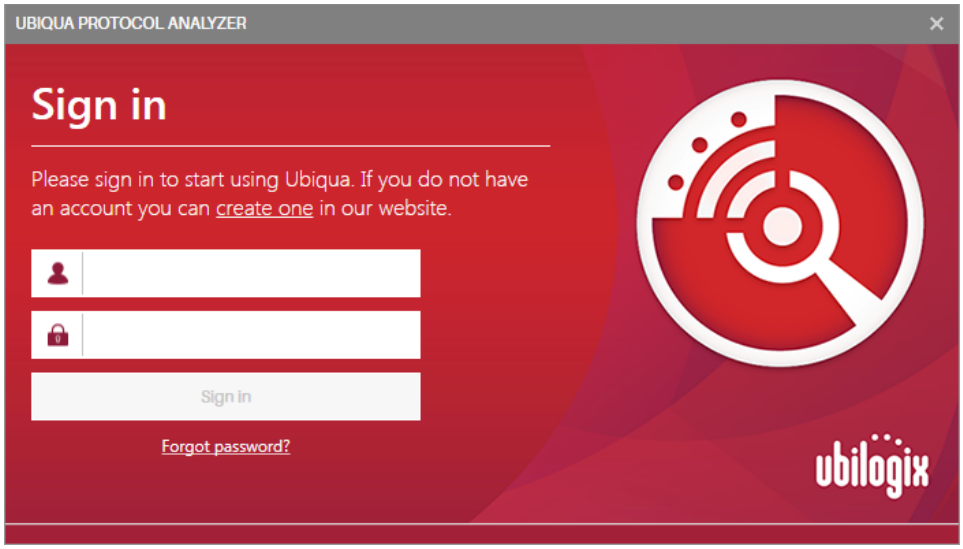


The first time you run Ubiqua the Ubiqua Authentication window and the Welcome window will appear. The Ubiqua Authentication window will ask for your Ubilogix account, write the e-mail and password of your Ubilogix account in the respective "E-mail" and

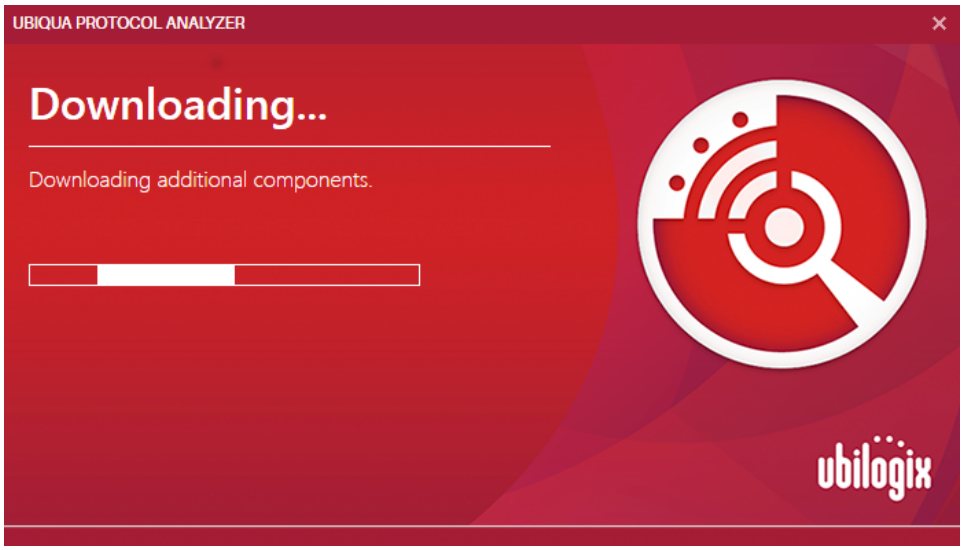
"Password" text boxes. If you have an Ubilogix account but you can't remember your password, start the recover password process at: https://www.ubilogix.com/recover.



From here you can select the available licenses.

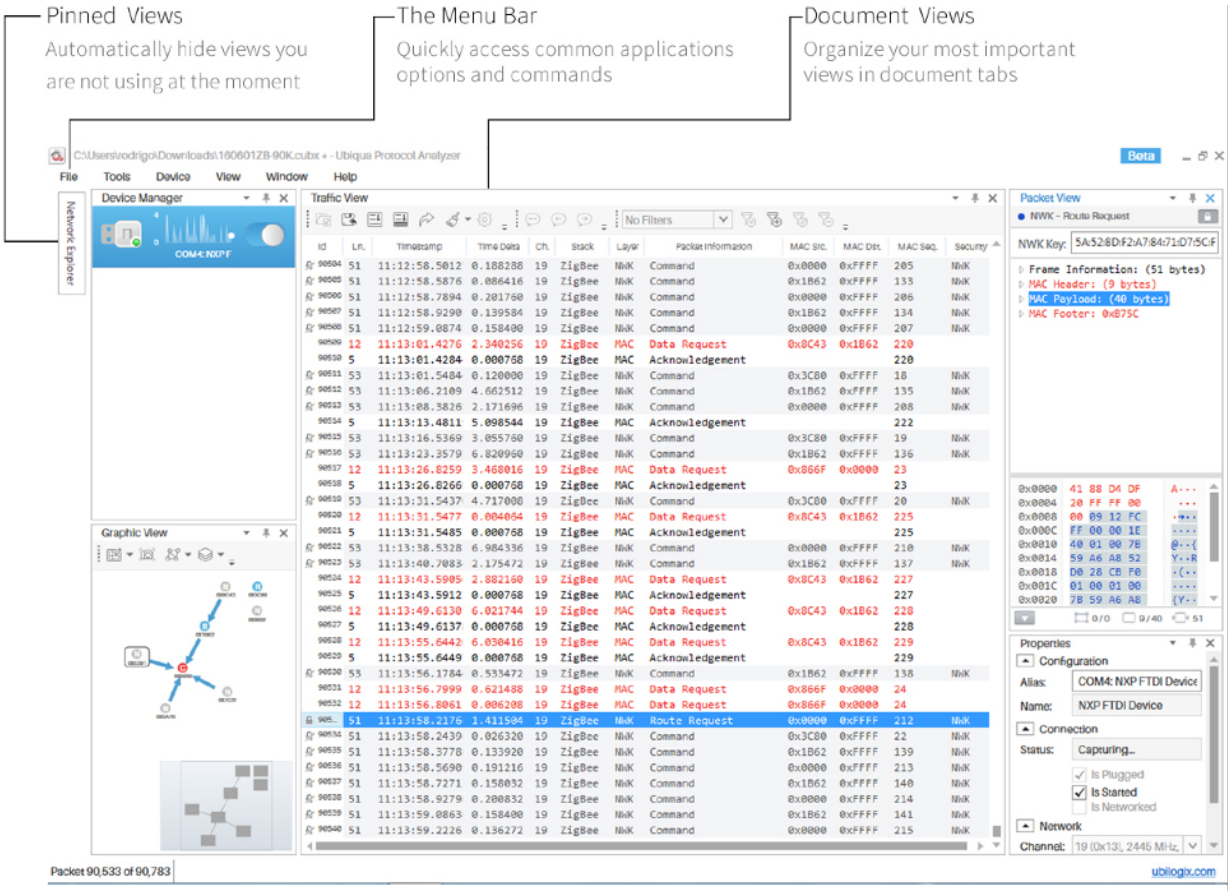


If it is the first time you run Ubiqua it may need to connect to the Internet to download additional updated components; this process may take a few moments and then Ubiqua will be ready to run.



The User Interface

The Ubiqua user interface is organized in dockable panels. Each panel provides access to different sets of information and tools of the Ubiqua software. This section introduces you



to this environment and will guide you through each of the visual components in the application. Further sections of this user guide discuss each component in more detail.

The main window of Ubiqua is composed of 3 sections: the menu bar, the workspace, and the status bar. The menu bar, as in other applications, offers you access to the most common options and commands available in the application. The status bar displays the current application status or shows information about the current selected packet. The most complex section is the workspace. As explained above, the many views of Ubiqua are actually dockable panels you can arrange to form the layout of your preference.

Docking Panels

Each panel has 4 dock modes: floating, dockable, auto hide, and hide. The layout of the workspace is changed by dragging and dropping the individual panels.

For instance, while dragging the panel over the workspace, icons will appear over the existing panels giving you the option to put the dragged panel on top, on the right, on the bottom, on the left, or as a tab of the target panel. If you drag the panel out the main window, the mode will change to "floating". To auto hide a panel in dockable mode, just click the button with the pin icon on the top right area of the panel (to the left of the close button).

Auto hidden panels will show up when the mouse cursor is over the panel tab (which is located at the far left side of the main window), and will automatically hide when you move the mouse out of the panel. In the above figure, the Watch View is in auto hide mode. To change the mode of an auto hidden panel, you must click the button with the pin icon again (it will show a rotated pin icon).

Once you have set the layout to your convenience, Ubiqua will keep the views arranged in the way you specified. You can always reset the layout to the default by using the Window > Reset Window Layout menu item. You have also the option to save the layout on a file and recover it later, to do so use the File > Save Environment... and File > Load Environment... menu items. The environment file does not only contain the layout but all application settings (for details see Setting Preferences).

The Menu Bar

The menu bar offers access to the most common actions in Ubiqua. The following is a brief description of what each menu item does and where you can find more information.

File

- **Open Capture...** – Opens a capture file and loads all the stored data which includes raw packets, security keys, addresses, graph layouts, and others (details in Capture Files).

- **Open Partial Capture** – Obtains a consecutive selection of packets of a CUBX capture file. To open partial capture, click on the Open Partial Capture option of the File menu. Select the capture file to open and click the Open button. The Select Capture Packets dialog opens and the user can select a packets range . The date and time are visible when the user hovers over the slider of the selected packet.

- **Save Capture As...** – Saves a new capture file with all the present data in the application.

- **Save Filtered Capture...** – Saves a new capture file containing all the present data in the application but only the packets available through the currently enabled filter (details in Filtering Packets).

- **Export Packets...** – Exports only the packets in the capture; excluding the Security Keys and collected Network Addresses (details in Exporting Packets).

- **Open Environment...** – Opens an environment file and loads all application settings such as views layouts, Traffic View column customizations, and others (details in Setting Preferences).

- **Save Environment...** – Saves a new environment file and stores all application settings.

- **Recent Captures** – Allows you to access the recently opened capture files; the list is limited to the last 32 files. Unexisting files will be dimmed and the "(not found)" text added at the end. To reset the list click the Clear menu item you will find at the end of this submenu.

- **Exit** – Closes the application.

Tools

- **Change Protocol Stack...** – This menu item allows you to change the protocol stack of captured packets by channel and PAN ID combinations (details in Changing Protocol Stacks).

- **Go To Packet...** – Selects the packet with the specified ID in the Traffic View (details in Go To Packet).

- **Options...** – Shows the options dialog where the application settings are set (see Setting Preferences).

Device

- **Start Device** – Starts the currently selected device on the Device Manager.
- **Protocol Stack** – Allows you to set the protocol stack used to decode incoming packets from the selected device on the Device Manager (see Configuring Devices for details).
- **Channel** – Allows you to set the channel of the selected device on the Device Manager.
- **Add Device...** – Opens the dialog used to add a new device in the manager (see Adding Devices).
- **Remove Device** – Removes the currently selected device on the Device Manager (see Removing Devices).

View

- **Traffic View** – Shows a view containing a grid with the available packets (see Traffic View).
- **Packet View** – Shows a view containing the details of the currently selected packet (see Packet View).
- **Graphic View** – Shows a view containing a graphical representation of the available packets. See Graphic View for more details about the functionality of this view.
- **Network Explorer** – Shows a view containing a tree of the identified network nodes (see Network Explorer).
- **Device Manager** – Shows a view containing a list of devices (see Device Manager).
- **Properties** – Shows a view containing the details and options for the current selection (see Configuring Devices and Customizing Nodes for more details).
- **Watch View** – Shows a view containing the current values of fields and variables (see Watch View for more details about the functionality of this view).

Window

- **Reset Window Layout** – Resets the workspace layout to its default as explained in Docking Panels.

- **Minimize To Tray** – Hides the Ubiqua window leaving only an icon in the system taskbar where you can show the window again. This option is useful for capturing packets in background mode or to control Ubiqua by the means of the services it provides (see Ubiqua Services for details).

Help

- **Documentation** – Allows you to open documentation files such as this guide.

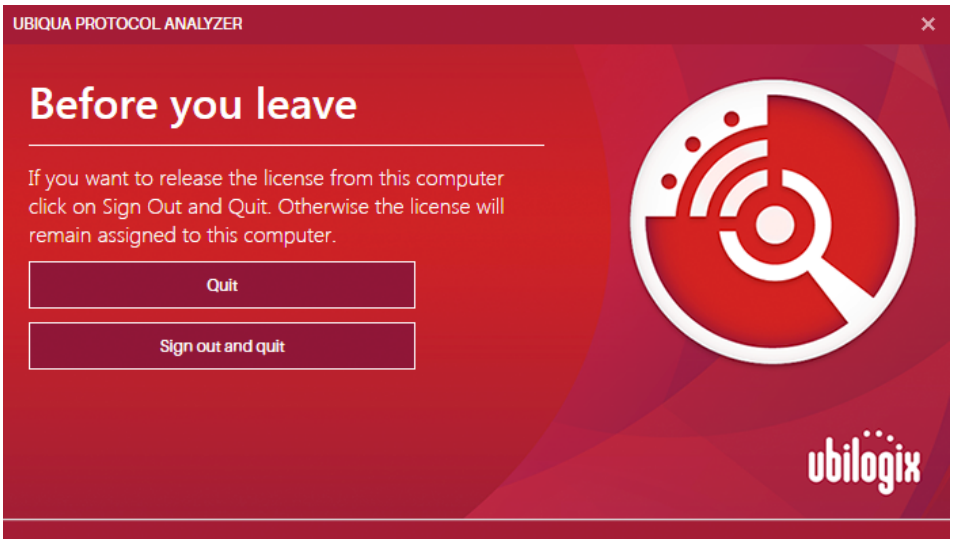
- **Ubiqua Support Page** – Opens the Ubilogix support webpage.

- **Check for Updates...** – Checks if a new version of Ubiqua is available for download and opens a dialog showing the latest release notes and if you are eligible for the update according to your license status (see Check for Updates).

- **About Ubiqua** – Opens the application about window.

Exiting Ubiqua PA

When you have a Standard Subscription, Ubiqua PA will present a dialog with the options avaiable before clossing the application.



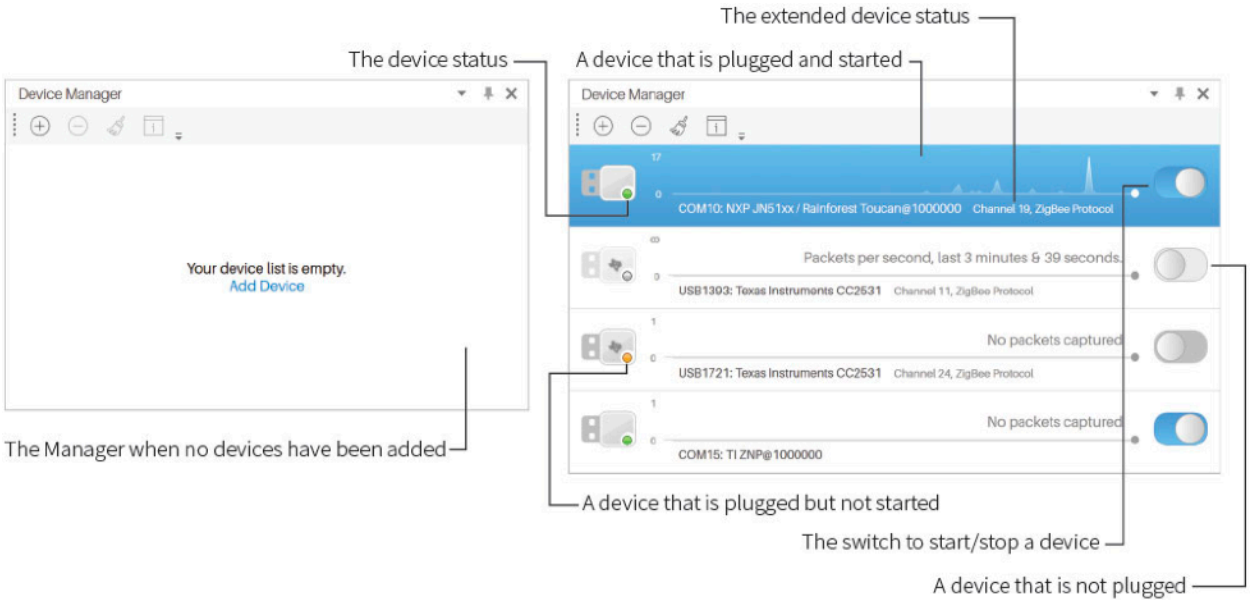
1. **Quit:** this option will close Ubiqua PA leaving your user authenticated so you don't have to sign in next time you start the application.

2. **Sign out and quit:** this option will release your license from the current system and let you login to another computer with the same license.

Chapter 3: Device Manager

The Device Manager lists and controls the devices used as a source of packets in the application. If the view is not visible, show it by selecting the Views > Device Manager menu item. The devices list will be empty the first time you open the manager. You need to populate this list by manually adding each device as needed (see the "Adding Devices" section for instructions).

Once the list has been populated, each device will be displayed in an individual control containing the following information: vendor icon, alias, status, extended status, and a switch to change the device status (see the figure below). The status will display "Offline" if the device is not plugged to your computer, "Idle" if it is plugged but not started, and "Capturing..." when it is plugged and started. The extended status will display additional information related to the role of the device in the application. For instance, the extended status of a sniffer device capturing packets will show the channel and protocol stack used for decoding.



Adding Devices

To add a new device to the list, right click inside the Device Manager view and select the Add Device... menu item, or click the Add Device link on the "Your device list is empty" message (if that is your case). Before opening the Add Device dialog make sure your device is compatible with Ubiqua (see Supported Sniffer Hardware for details), then open the dialog (see the figure below), and follow the next steps:

1. Connect the device to your computer.

2. The Found New Hardware wizard might appear. In such case, follow the instructions on the screen to install the device drivers. If you need to manually specify the location of drivers, you will find them in the **Program Files\Ubilogix\Drivers** folder.

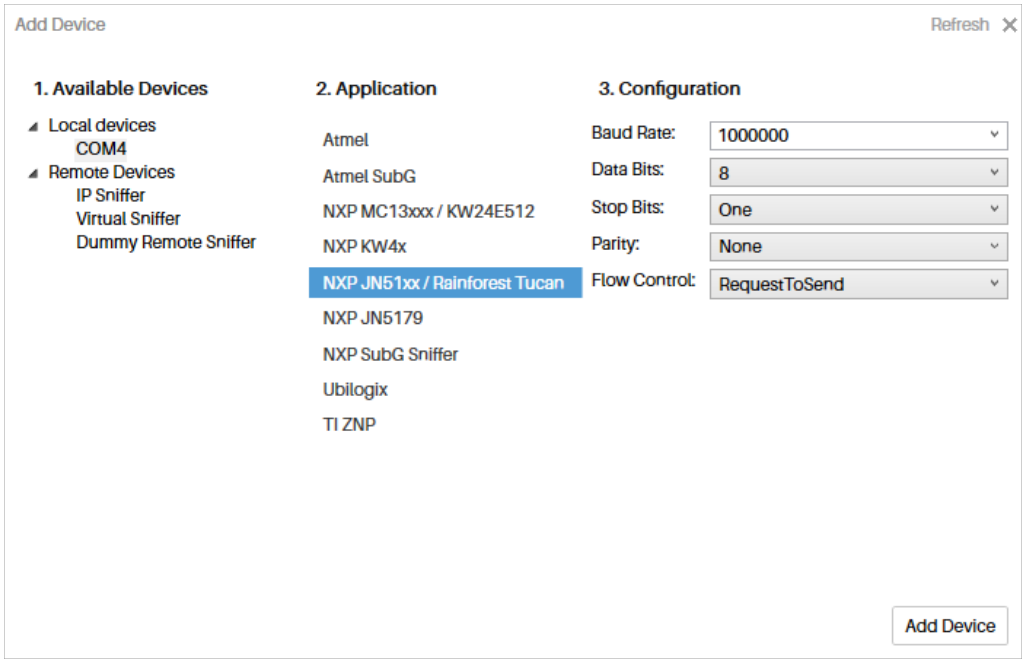
3. Wait a couple of seconds while the driver is installed and the device recognized.

4. Available Devices section, select your device. After the selection is made, Ubiqua will fill in the Application section with specific vendors.

5. Select the device you want to add. If no devices are found try searching again by clicking the Refresh button at the top right corner of the Add Device window.

6. From the Application combo box select the appropriate device application.

7. Click the Add Device button.



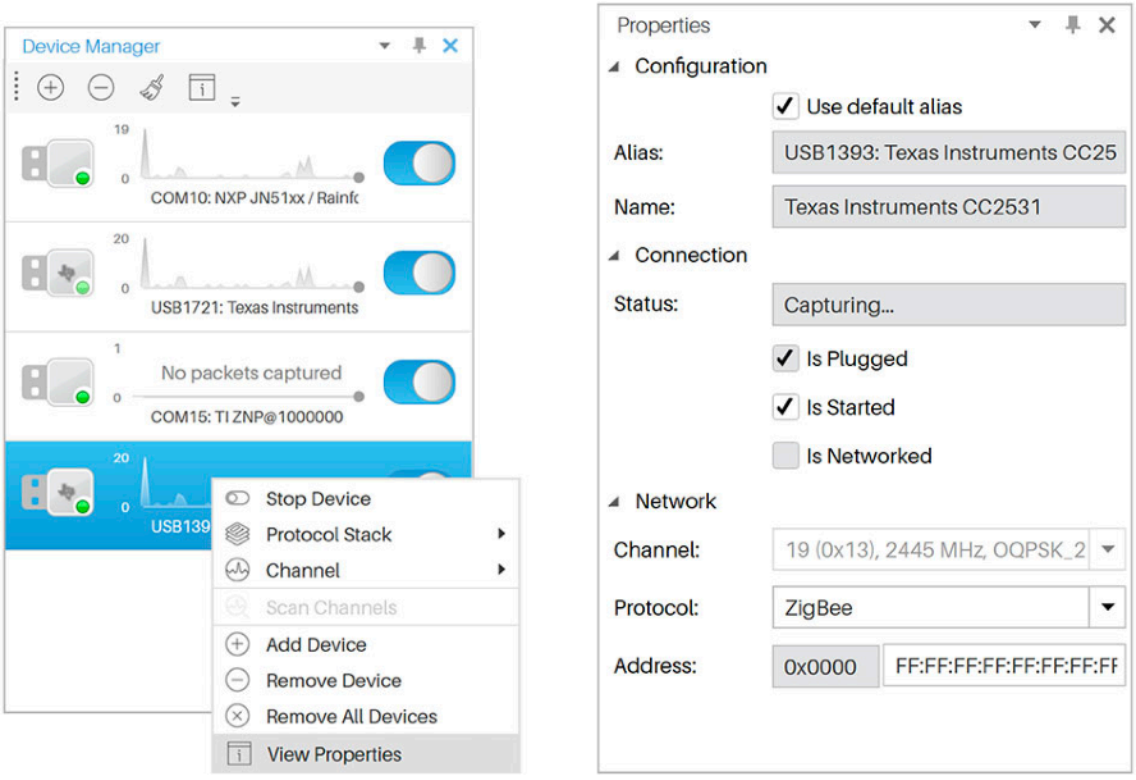
It is also possible to add devices by specifying custom values for properties such as baud rate, data bits, stop bits and others.

When setting up Sewio Open sniffer for the first time, we must configure the sniffer via its web interface, aditional information can be found in http://www.ubilogix.com/ubiqua/documentation/guide/sewio-hardware/.

Configuring Devices

Once you have installed the drivers and added the device to the list, you can configure its properties by right-clicking on the device and selecting the View Properties menu item (see the figure below). This will open the Properties view, where the properties can be edited. As long as the Properties view is open, it will show the properties of the selected device in the Device Manager.

Please keep in mind that if the protocol or channel is changed, and the device is running, you must stop and start it again. Ubiqua provides a shortcut for this, the Protocol Stack and Channel menu items of the device contextual menu. Right click on a device and make your selection, Ubiqua will automatically restart the device if it was running.items of the device contextual menu. Right click on a device and make your selection, Ubiqua will automatically restart the device if it was running.



Capturing Packets

To start capturing packets just click on the device switch button or select the Start/Stop Device on the contextual menu. If a problem was found starting the device and you can't stop or start it anymore, proceed to unplug the device and plug it back to your system;

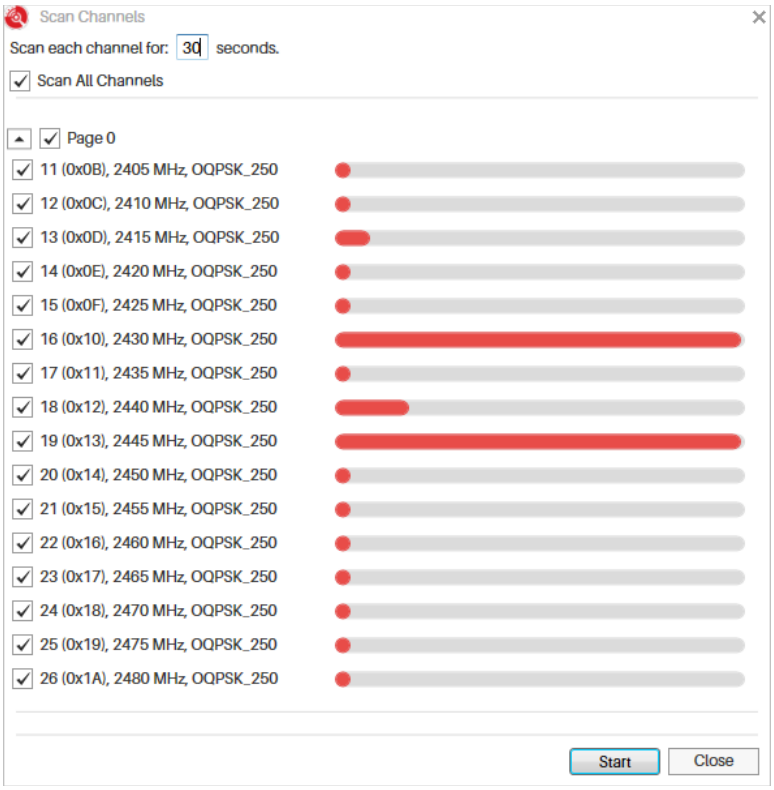
usually, this will resolve the problem. If you keep experiencing problems, make sure the port is not being used by another program in your system and try again. Also, please note that according to your license you could be limited to a certain number of devices started at the same time. Whatever the cause of the problem is, the extended status of the device in question will display a possible solution.

Removing Devices

Devices on the list are maintained by the application and stored in environment files. To remove a device right click on it and select the Remove Device menu item. If you want to remove all devices right click on the Device Manager and select the Remove All Devices menu item. Please note that this action can't be undone.

Scanning Channels

Scan channels allows capturing through all channels or the selected channels. Each selected channel will capture during a specific time, at the end, Ubiqua will show the number of captured packets per channel on a graph.



To scan channels, right click on the device and select the Scan Channels... menu item. After the Scan Channels dialog opens, select the channels to scan and set the scanning time per

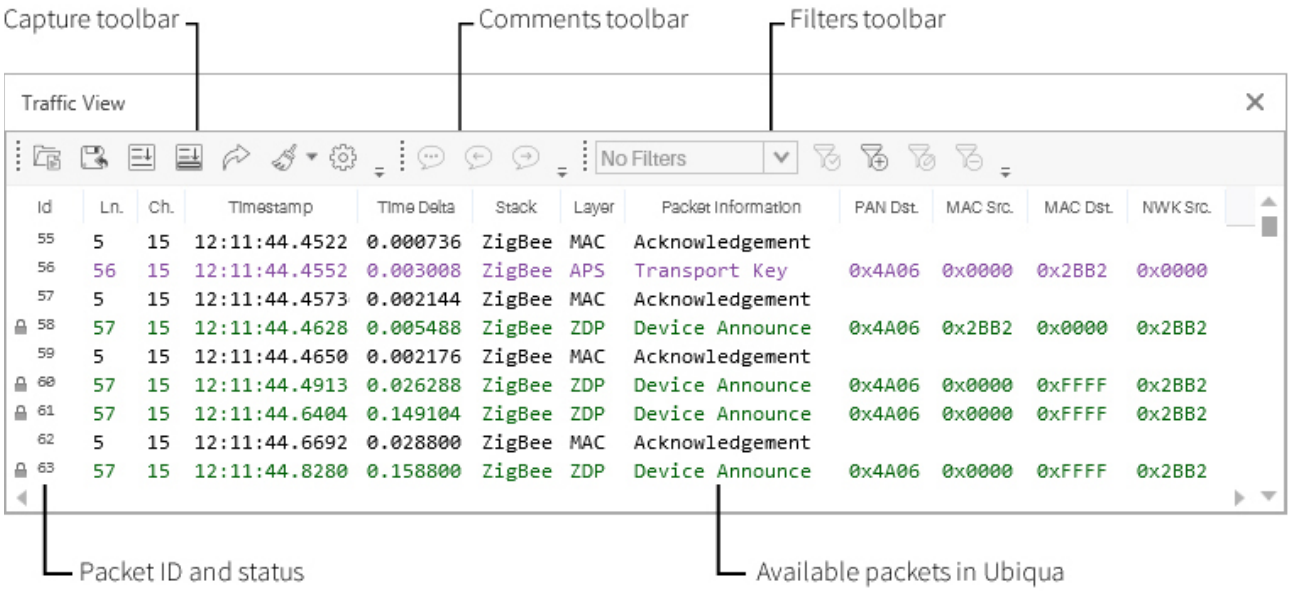
channel; to start the scanning process press the Start button. The graph will display the packets captured per channel when the you place the mouse over any channel. Note that the packets captured during the scanning process may appear in the Traffic View and that the channel list may be different, depending on the capabilities of the device used for scanning.

Chapter 4: Traffic View

The Traffic View is the most feature-rich component in Ubiqua. This chapter describes the full functionality of the Traffic View including instructions on topics such as: how to manage capture files, the actions that can be performed with packets selected on the grid, how to filter packets, and other related features.

The Traffic View is mainly composed of 2 components: a grid, and a set of toolbars (see the figure below). The grid shows all the packets captured with devices or loaded from capture files. Toolbars provide access to most of the functionality available throughout the system and because of this, the Traffic View can be seen as the central point of Ubiqua. Many of the actions performed on it produce changes or updates in other views. For instance, selecting a packet on the grid changes the Packet View contents. Also, actions in other views could cause the Traffic View to update its contents. For instance, starting a sniffer in the Device Manager will cause the grid to update itself to show incoming packets.

The following image depicts a traffic view with some packets captured. Also, note that the first column shows an icon that depicts additional information about the packet on its corresponding row, for example whether there's a comment for the packet (call out icon), if there was an error when decoding (cross mark icon) or if the packet was encrypted (closed lock) and the decryption was successful (open lock).



Color Codes

The Traffic View provides a helpful coloring scheme to easily identify the layer and protocol for each packet captured. The following table lists the available options:

Layer	Color Name	Protocol
Dark Green	TCP	Thread, Zigbee IP, IP
Dark Green	UDP	Thread, Zigbee IP, IP
Dark Green	ICMPv6	Thread, Zigbee IP, IP
Dark Green	ICMPv6	Thread, Zigbee IP, IP
Dark Green	PANA	Thread, Zigbee IP, IP
Dark Green	mDNS	Thread, Zigbee IP, IP
Dark Green	MLE	Zigbee IP, IP
Orange	MLE	Thread
Light Green	CoaP	Thread, Zigbee IP, IP
Dark Blue	Ethernet	Thread, Zigbee IP, IP
Purple	HTTP	Thread, Zigbee IP, IP
Orange	SE2	Zigbee IP, IP
Orange	JenNet-IP	JenNet-IP
Gray	MAC-Beacon	Zigbee IP, Thread, Zigbee
Brown	Mac-Data	Zigbee IP, Thread, Zigbee
Black	MAC-Acknowledgement	Zigbee IP, Thread, Zigbee
Red	MAC-Command	Zigbee IP, Thread, Zigbee
Red	NetBios	Zigbee IP, Thread, Zigbee
Gray	PopNet- Beacon	PopNet
Brown	PopNet- Mac-Data	PopNet
Black	PopNet- MAC-Acknowledgement	PopNet
Red	PopNet- MAC-Command	PopNet
Light Green	PopNet-APP	PopNet
LightBlue	PopNet-NWK	PopNet

Layer	Color Name	Protocol
Black	Pop-Nwk_Acknowledgement	PopNet
Purple	DHCPv6	Thread, Zigbee IP, IP
Purple	DTLS	Thread, Zigbee IP, IP
DarkGreen	ZDP	Zigbee
Light Green	ZCL	Zigbee
Purple	APS	Zigbee
Dark Blue	NWK	Zigbee
Dark Blue	NWK-GP	Zigbee
LightBlue	6LowPAN	Zigbee IP, Thread
DarkGreen	EAP	Zigbee IP
LightBlue	IPv4	Thread, Zigbee IP, IP
Red	IPv6	Zigbee IP, Thread, Zigbee, IP

Timestamp & Time Delta

The Timestamp columns displays the exact date and time when a data packet was captured. You have the possibility to choose one of the 2 formats in which the information can be displayed, the 'Date and time' or just the 'Time' format. To configure it, click the Tools > Options menu item, and then select the 'Traffic' tab, at the bottom of the tab body are the 'Date and Time' and 'Only time' radio buttons, select one of these options and then click the 'OK' button to determine the format in which information will be displayed in the Timestamp column.

The Time Delta is the time interval between 2 captured packets, which is calculated based on the elapsed time from the previous packet regardless of its source network or channel. In case a data filter is applied to the capture, the time intervals will be recalculated based on the result.

Capture Files

Ubiqua uses capture files not only to store the packets you see in the Traffic View, but also to store other data such as the layout and settings of the nodes in the Graphic View, or the security keys used for decoding. When saving a capture, the data available in all views is

retrieved and stored into a new file. Note that this process does not store decoded data, so when you open a capture file all the stored packets will be decoded again to populate data in all views.

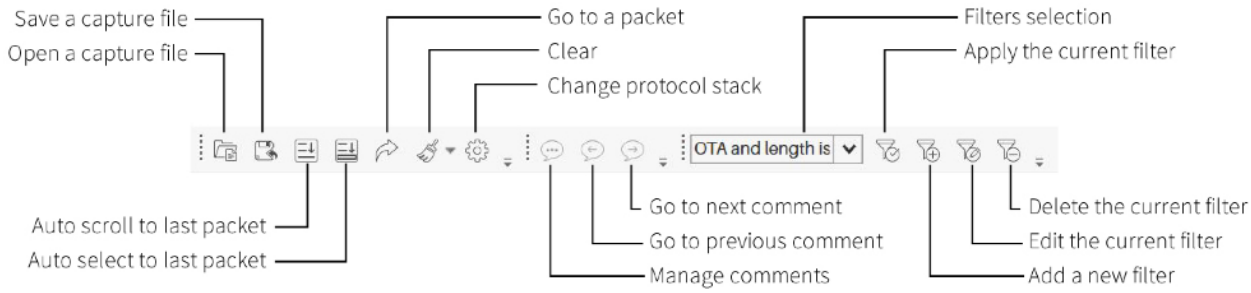
Saving Capture Files

To save the available capture data into a new file, follow the next steps:

1. Start the Save As dialog by either selecting the File > Save Capture As... menu item, clicking the Save Capture toolbar button (see the figure below), or pressing Ctrl+S on your keyboard.

2. Select the location where you want to store the new capture file, specify the file name, and press the Save button. Ubiqua capture files have the `.cubx` file extension but you can also save the capture in the `.cubx` and `.pcap` file formats.

3. A progress dialog will appear showing the status of the save process.



Opening Capture Files

The process to open a capture file is very similar to saving:

1. Start the Open dialog by either selecting the File > Open Capture menu item, clicking the Open Capture toolbar button (see the above figure), or pressing Ctrl+O on your keyboard.

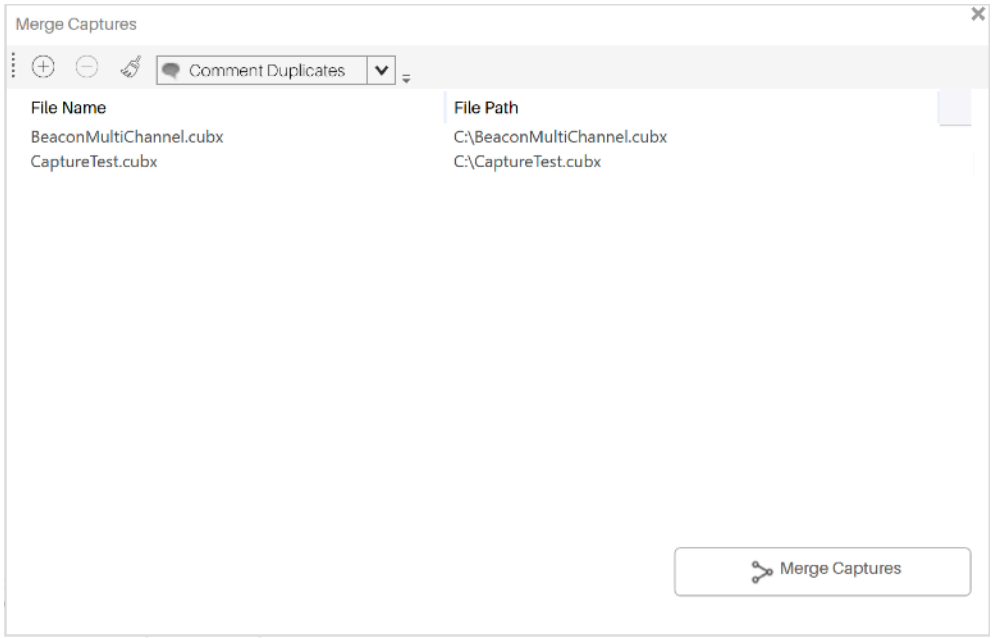
2. Select or specify the capture file and press the Open button. Additionally to its own `.cubx` format, Ubiqua also supports opening captures in a number of other file formats such as `.dcf` and `.pcap`. If the capture file does not hold Protocol information at the time a dialog opens with the available protocols and the user can select one of them to decode the capture.

3. A progress dialog will appear showing the status of the opening process. Note that depending on the file size (directly related to the number of packets stored), this process may take some time as the contained packets are being decoded on the fly to populate data in the corresponding views.

Merge Capture Files

This feature allows the user to merge `.cubx` files. This process consists in bringing together in one file all the packets from the different source files, order them chronologically and — for the case of Zigbee frames — detect duplicates and mark them with a comment or delete them.

To use this feature clock on menu Tools > Merge Captures or press Ctrl+M on your keyboard, after this action a dialog window will appear on your screen; In the toolbar area of the window there is the Add Capture button, that allows you to add the capture files to merge to a list, a maximum of ten files is allowed to be added to the list, as a second option is the Remove button that works once the capture files have been loaded, and has the function of remove one of the files of the list, just clicking the item you want to delete and then clicking the Remove button on the toolbar, next to it is the Remove All button, that clears the element list just pressing this button or with the Ctrl+Delete combination on your keyboard. Finally there is a combo box with 2 options, the default option Comment Duplicates that adds a comment to those packets that are duplicated, and the Remove Duplicates option that delete those packets that appear more than once in the merged capture file.



Once you have a list of selected files click the Merge Captures button at the bottom right of the window, after this action, a browse window will appear for you to indicate where do you want to save the merged file and how you want to name it, then save it.

Once the new file is generated a notification will appear asking if you want to open the new merged file, if you accept press the Yes button, and the new file will be loaded on the Traffic View, if you want to open it later, press the No button or just close the notification, following this action the merged the file list will be cleared, letting you to make a new merge if needed.

Auto Scroll and Selection

The Traffic View features two options that are useful to track the latest packet on the grid, this options are Auto Scroll and Auto Select. The Auto Scroll option moves the scroll position to bring into view the latest packet captured, while the Auto Select option selects the latest captured package (which also brings its contents into the Packet View). To activate or deactivate this options, use its corresponding toolbar toggle buttons (see the above figure).

Clear

The clear functionality resets all the information available for 4 different types of data (see below). You can choose any or all types of data to clear by clicking the down arrow next to the Clear button.

This functionality is accessible from the Clear toolbar button, and you can choose among what types of data to clear by clicking the down arrow next to the Clear button (see the figure above).

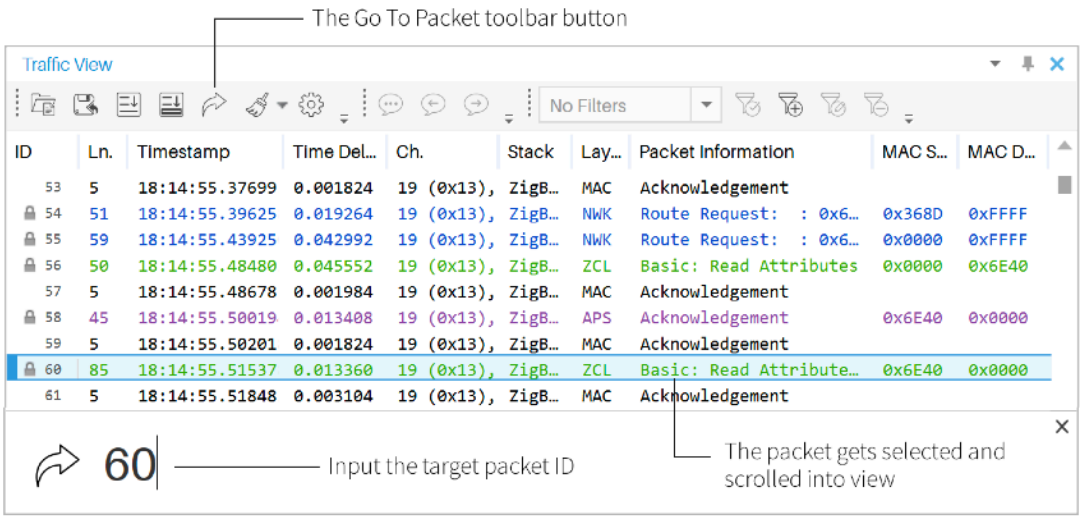
The types of data to clear are:

- **Clear Packets** – Removes all the packets in the Traffic View.
- **Clear Nodes** – Clears the nodes information in the Graphic View, the Network Explorer and the properties window.
- **Clear Security Keys** – Removes the security keys stored in the keychain. (For more info see security keys).

- **Clear Addresses** – Removes the network addresses listed in the addresses table in the Security tab of the Options window.

Go To Packet

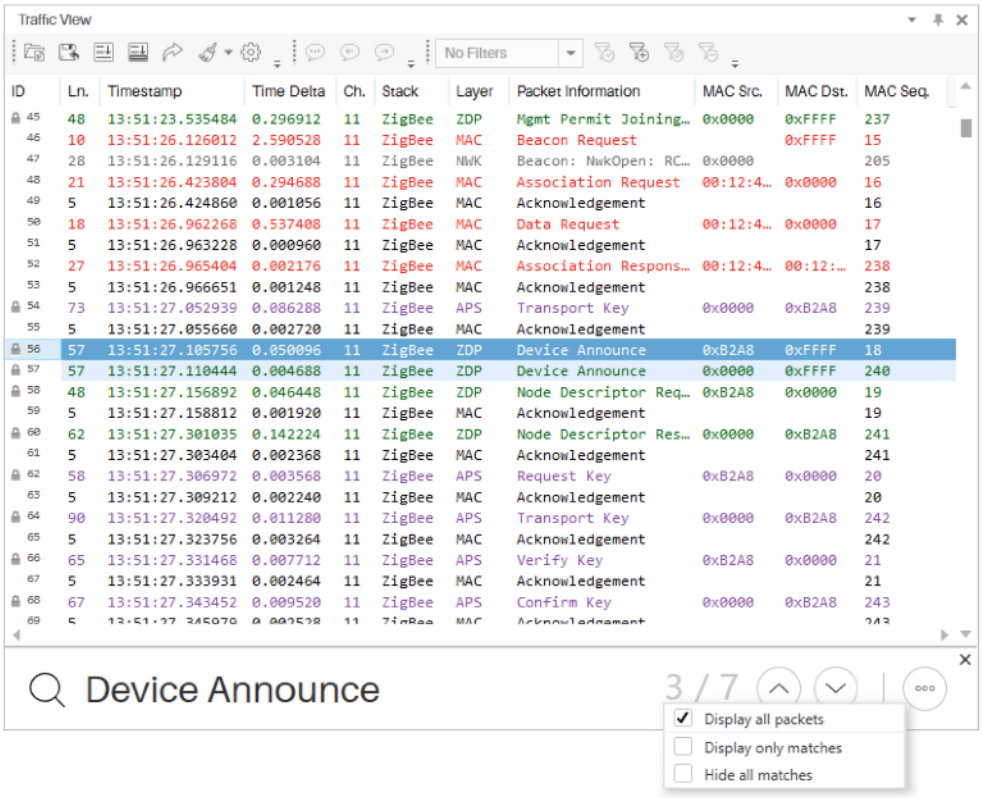
This feature is similar to the Auto Scroll and Auto Select options. The difference is that this option will scroll and then select, not the latest packet, but the packet with the ID you specify. To start using this option, click the Go To Packet toolbar button (see the figure above), or press the Ctrl+G keys on your keyboard. A pop-up box will appear at the top right area of the Traffic View. Type the ID of the packet you want to look and then press Enter. The grid will scroll and the packet will be selected. If the ID you specified does not correspond to any of the available packets you will hear an exclamation sound. Close the pop-up by clicking the × button or pressing the Esc key.



Find Packets

This tool allows you to quickly make a detailed string search of the information contained in each packet of the Traffic View.
To start using this feature click the Tools > Find menu item or press Ctrl+F on your keyboard, after this a search bar will appear at the bottom of the Traffic View, next to the magnifying glass icon there is a text input where you have to type the key word you are looking for which must have a minimum length of 3 characters, then click the arrow buttons at the right of the text input for select next/previous packet that match with search criteria, if the typed word does not match with the information contained in the Traffic View, a label will display the message “No match found”, in the opposite case, it will appear

a label indicating a number of a list of matches where you are at, and the total number of existing matches with that key word.



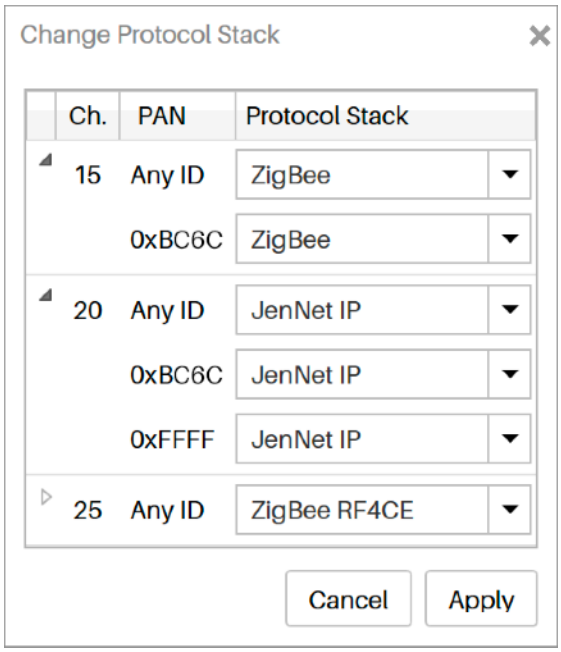
When you are navigating through the matches, the packets that contain the searched word will be highlighted, making an easier search.

To the right side end of the Find bar there is a three dotted icon button that being pressed shows a context menu with 3 different checkbox options, “Display all packets” which is the default option and show all the packets, “Display only matches” that will only let you see those packets that match with the string search typed in the text input, and the third option “Hide all matches” which hides those matching packets with the search criteria. To re-display all packets check the default option “Display all packets” or just close the Find bar.

Changing Protocol Stacks

This feature allows you to change the protocol stack of the packets already available in Ubiqua. The change protocol process will clear all the current capture data, apply the changes you selected, and then re-decode all the packets, producing new capture data. To start the process you must stop all capturing devices, then click the Change Protocol Stack

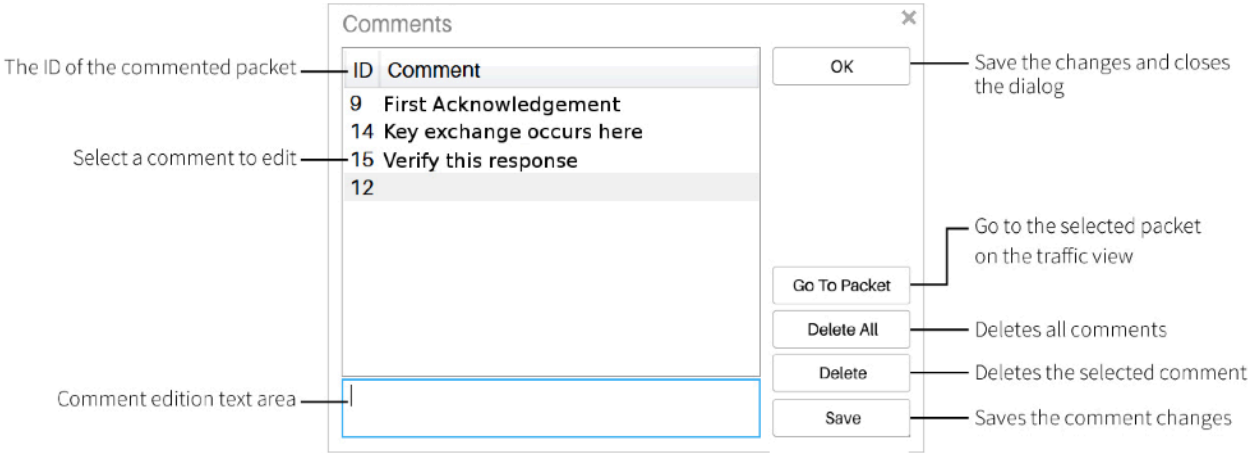
tool bar button or press the Ctrl+H keys on your keyboard. The Change Protocol Stack dialog will appear (see the figure below). On it, you will have the choice to set what protocol stack will be used to re-decode the packets on a given channel (for a list of the available stacks see Supported Protocols). There is also the option to set a new protocol stack in combinations of target channels and PAN IDs, giving you a fine-grained control over the changes. Once you have selected the new stacks, press the Apply button. After this point, all capture data will be cleared and the packets will be re-decoded using your settings (a progress bar will show you the status).
Please note that devices retain their own protocol setting. The change protocol stacks feature only affects previously captured packets.



Commenting Packets

Ubiqua features the ability to annotate the packets shown in the Traffic View. Capture files store this information so you will be able to share insights about packets with other Ubiqua users, or just give you the opportunity to store notes about your capture analysis.

To start using comments, select a packet on the grid. Right click on it to open the contextual menu and select the Add Comment menu item. The Comments dialog will appear (see the figure below), write your notes and press the OK button. To see the comment, just move the mouse cursor over the row header and it will display the comment on a tooltip. To delete a packet's comment from the grid, right click on it, and select the Delete Comment menu item.



The comments toolbar has 3 buttons (see image below), the first one opens the Comments dialog (where you can edit or delete the comments) and the Go To Previous/Next Comment buttons that you can use to move the scroll of the grid and select the previous or next commented packet.

Filtering Packets

There are certain times in which you don't want to see all the captured packets but a subset of them. To help with overload, Ubiqua features the ability to filter packets and show only the ones that fulfill a certain logical expression.

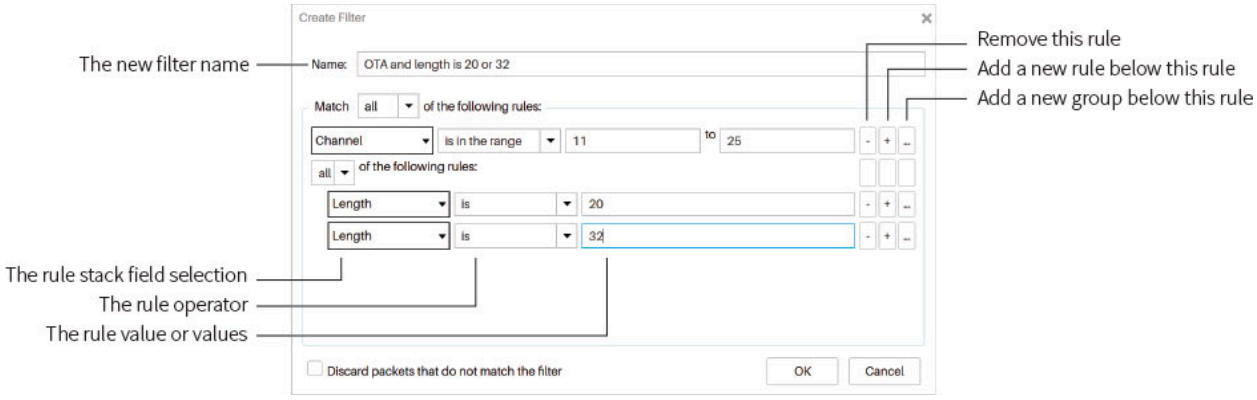
Filters are managed through the filters toolbar at the top of the Traffic View (see below).

To create a new filter click the Add Filter toolbar button. The Add Filter dialog will appear (see the figure below). You will find it very similar to the ones used to create playlists on music players. The dialog allows you to define filters by combining a set of rules grouped by a "match all" or "match any" operations. Each rule is composed of a field, an operator, and a set of values. Also, you have the possibility to specify subsets of rules for complex scenarios. A field is an element whose value can be obtained by decoding the packet, the operator defines the condition the field value must comply, and the set of values (which can vary depending on the used operator and the data type of the field) specify what are the expression values that will be used to evaluate the rule.

The figure below shows an example. The "OTA and length is 20 or 32" filter is composed of a simple rule and a rule group. Both rules must be matched. The first one asks to filter packets whose Channel is in the range of 11 to 25. The rule group asks to match any of 2 sub-rules, the first is that Length is equal to 20, and the second is that Length is equal to 32.

Once that filters have been created, select one from the Filters selection at the Filters toolbar and click the Apply Filter toolbar button. If there is not enough decoded information available to compute the filter, packets will be decoded again to retrieve any additional data needed to finally apply the filter. Once that filters have been applied, Ubiqua will only show the packets that the filter allows even when capturing new packets from devices. To edit the selected filter click the Edit Filter toolbar button, to remove it, click the Delete Filter toolbar button.

Filters are not stored on capture files. They will be stored on the environment and they will be available as long as you don't delete them. As with the layout of the views, they can be stored on an Environment File (more details on this in Setting Preferences).



Exporting Packets

f you need to extract packets of your capture and you need them on a different file format (not just a capture file), you can export the selected packets on the Traffic View. To start the export process, select the packets you want to export, Go to the File menu and choose the Export Packets item. Note that in order to select packets in the Traffic View, you must disable both the "Auto Select Last Packet" and the "Auto Scroll" options.

You can export either all the packets or only the selected packets in the following file formats: `.xls` (Microsoft Excel spreadsheets), `.csv` (Comma Separated Values), `.opml` (Outline Processor Markup Language), and `.txt` (simple text files). The `.opml` and `.txt` file formats will include all the decoded data of the packets while the `.xls` and `.csv` file formats will only include the data shown in the columns of the Traffic View.

Copying a Packet

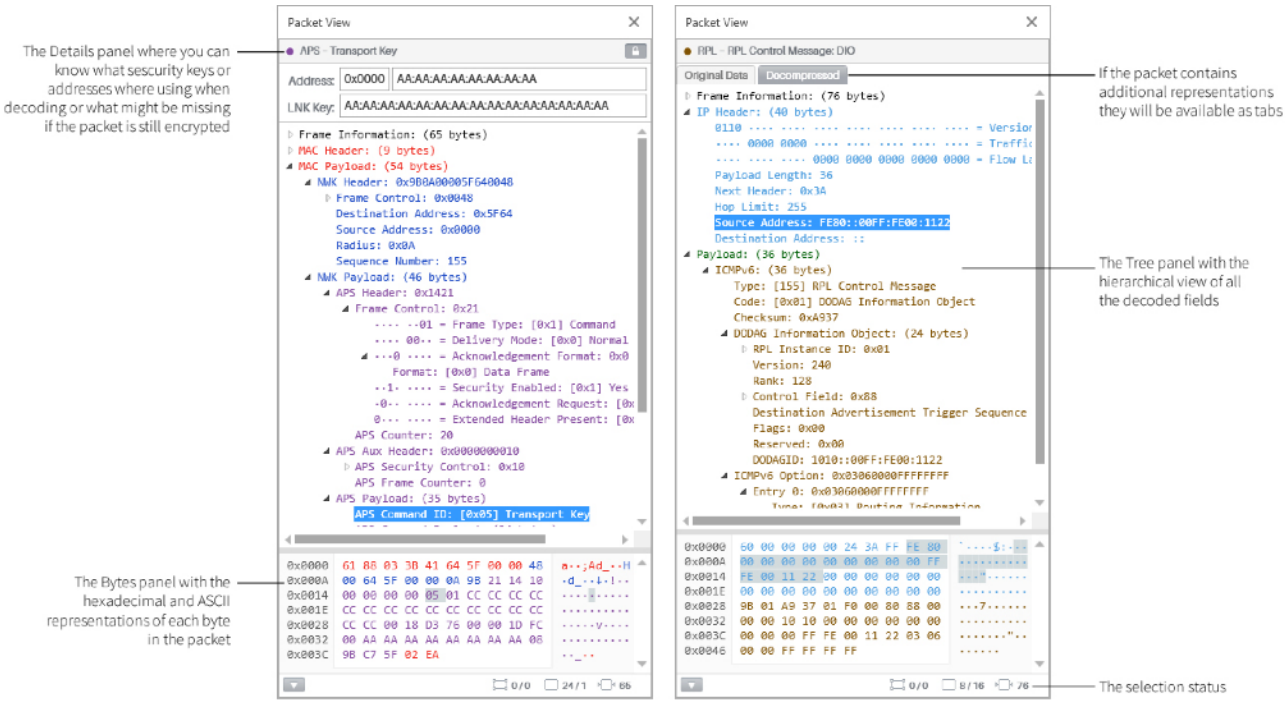
You can copy a packet as it appears in the Traffic View or the Packet View's tree by selecting the packet you want to copy; right click on the selection and choose the Copy menu item. The clipboard will store the selected packet data and it can be pasted in any text editor. When traffic is being captured in fast rates, it is recommended to disable the "Auto Scroll" and "Auto Select Last" options to easily select the packet from the Traffic View.

Chapter 5: Packet View

The Packet View displays the decoded packet data along with its raw contents. The view is composed of two panels, a status bar and an optional Description panel for when the packet contents are still encrypted. (see figure below). At the top of the Packet View, the Title bar shows details such as the security keys and network addresses used to decrypt the packet. Below the Title bar, the First panel contains a tree; the tree is a hierarchical representation of all the decoded fields of the packet currently selected on the Traffic View. Finally, the second panel is the Bytes panel; this panel displays the hexadecimal and ASCII representations for each byte of the packet.

The panels are linked together so if you select a tree node, the corresponding bytes for the field will be highlighted on the Bytes panel and the other way around. Finally, the status bar of the view indicates the offset and length values relative to the full packet, to the selected fields of the tree node, or to the selected panel cells.

In the Title bar there's a "lock" icon (see the figure below), when clicked it shows the optional panel that shows the key for decrypting the packed data or if not accessible allows the user to type the decryption key.



Each panel in the Packet View has its own context menu. Through the tree context menu you can copy (Ctrl+C) the selected field in the default format or in additional formats by using the Copy As options, also, you can create or edit filters (see Creating Quick Filters). The context menu in the Bytes panel allows copying the selected bytes and also you can navigate to the parent field of the selection by pressing the Esc key.

You are not restricted to only one Packet View instance at a time, you can open as many as you need. Just double click on a packet in the Traffic View to launch a Packet View of the same.

Testing Decryption

The Title bar will always display the row color in the form of a colored circle on the top left and the packet information as it is displayed in the Traffic View (see Traffic View for more information). Whenever a packet is encrypted, regardless of whether they were decrypted or not, an expansion button will appear on the top right. If the packet was decrypted the security keys used for decryption and the related addresses will be listed, if the packet could not be decrypted and the missing item is known, you will be provided with auto-complete text boxes to provide either the missing security key or address relationship. After providing the missing items, a Redecode button will appear. Click the button to decode the packet again, if the decryption if successful, you will be asked if the complete capture must be re-decoded or not.

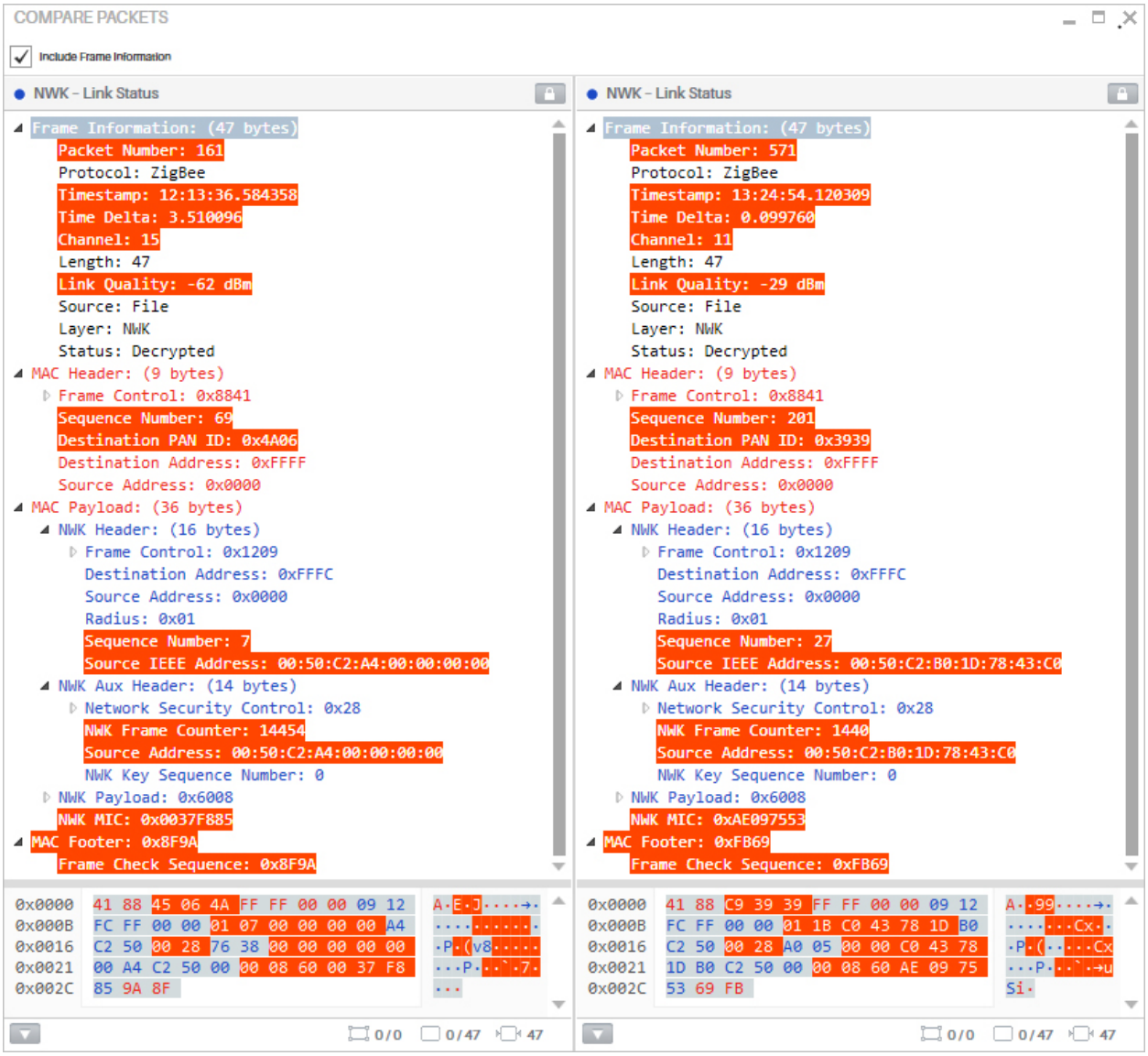
Creating Quick Filters

From this view you can create filters using the values of the displayed field values on the tree nodes. Just right click on a tree node and move your mouse cursor over the Create Filter menu item. Depending on the data type of the packet field you selected, different options will appear. Select one of the options and then the filter will be created and applied. Use the filter toolbar to edit or remove you newly created filter (see Filtering Packets for details).

Comparing Packets

The Packet View can also be used to compare packets and highlight the differences between them. To do so, click on the first packet on the Traffic View. Press and hold the Ctrl key on your keyboard. Click on the packet to compare to. Right click on one of the selected packets. Select the Compare Packets menu item. The Compare Packets window will appear highlighting the differences between both packets in red (see the figure below). The fields with different values will be already expanded on the tree representation although, only

OTA fields will be compared. Select the checkbox at the top of the window to also include the Frame Information (non OTA fields) in the comparison.



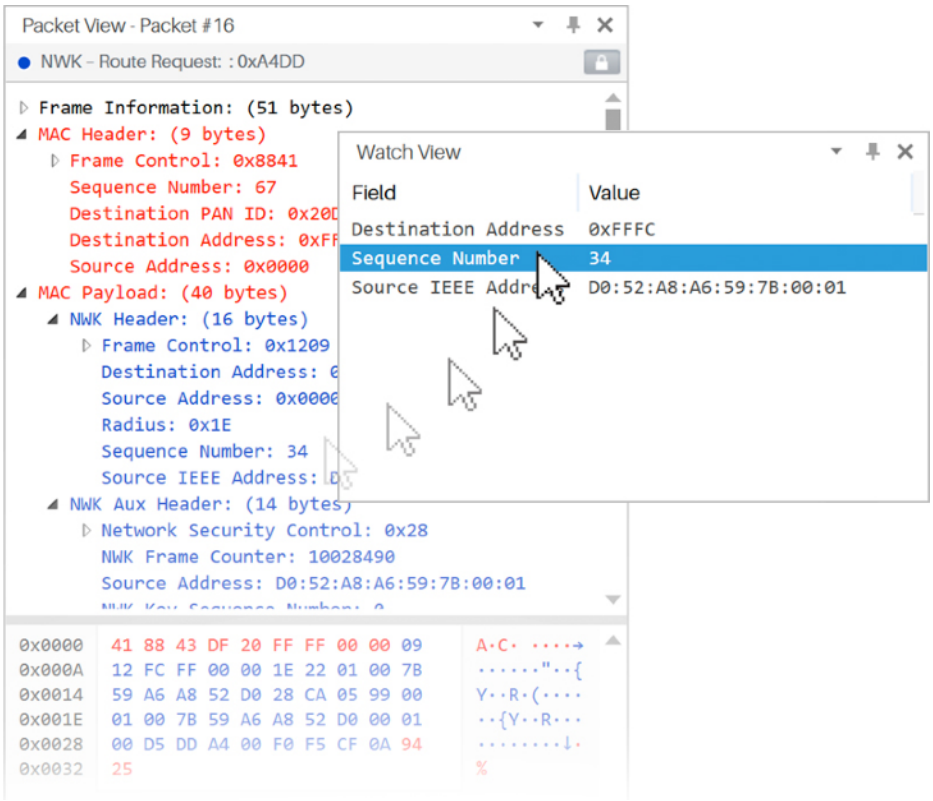
Note that is recommended to disable the "Auto Scroll" and "Auto Select Packet" options before selecting the packets to be compared from the Traffic View.

Chapter 6: Watch View

This view aims to assist you to track value of a field shown in the ‘Packet View’ tree, which is useful when capturing since it allows you to see how the value is changing as more packets arrive. To use this feature, just select a field from the ‘Packet View’ and drag it to the ‘Watch View’ (see the figure below), the value will be updated each time you select a different packet on the Traffic View. Combine this feature with the Auto Select option to see the value changing with incoming packets.

The information displayed in the ‘Watch View’ can be copied or deleted by the user, for this, you have to select one or several fields and then right click on them, following this action a context menu will appear with the ‘Remove’, ‘Copy’ and ‘Copy Value(s)’ options. To eliminate a field select ‘Remove’ option or click the ‘delete’ key on your keyboard, selecting the ‘Copy’ item will save both ‘Field’ and ‘Value’ to clipboard, unlike the 'Copy Value (s)' alternative that only copies the 'Value' of the field.

The contents on the Watch View are stored on the Environment, so they will be the same each time you open Ubiqua (for more information see Setting Preferences).



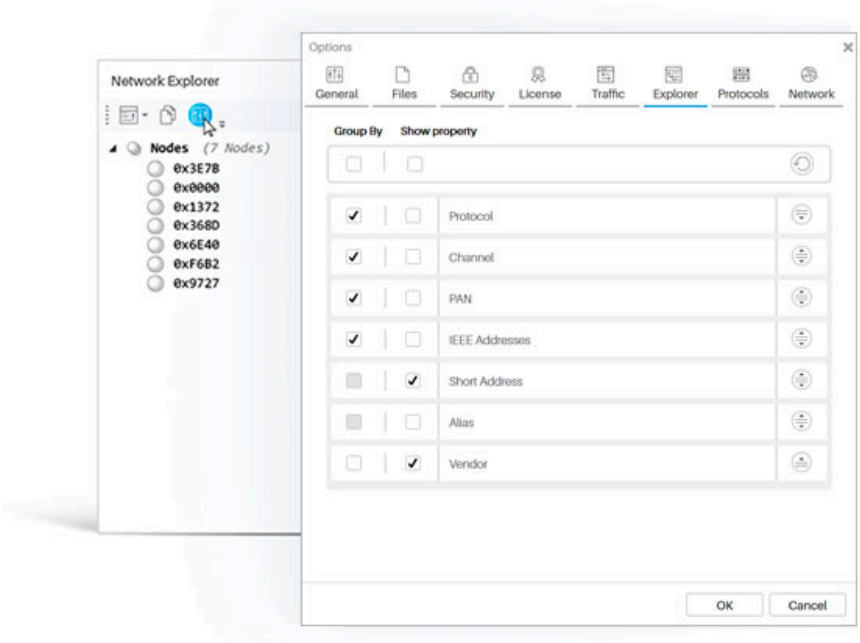
Chapter 7: Network Explorer

The Network Explorer view shows a tree representation of the network devices found on a capture log. You can customize the hierarchy of the tree grouping your devices by protocol, PAN ID, channel or vendor, besides having the option to select the properties that you would like to display, and sort the entries according to your preferences.

In the Network Explorer toolbar, the first option allows you to sort your tree nodes, in order to visualize your data in ascending or descending order. The second option copies the network list and its addresses to clipboard, and the last option will open the Options window with the Explorer tab selected to configure the Network Explorer. Another way to reach to this configuration window is clicking the Tools > Options menu item, and then selecting the Explorer tab.

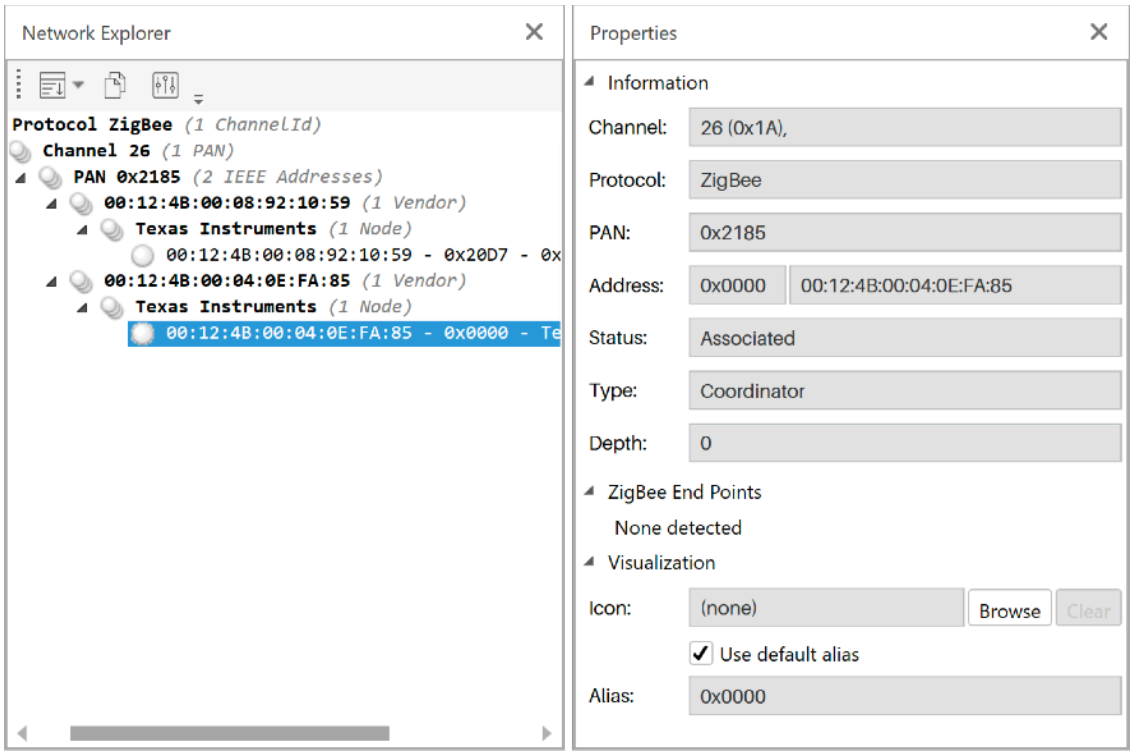
Once the Explorer tab is open you will be presented with different node properties, each one of them has 2 column with check boxes. The 'Group By' column allows you to group your devices by Protocol, Channel Id, PAN, IEEE Address, and Vendor; this mentioned properties plus Short Address and Vendor, are the options enabled in the 'Show property' column, so you can check those you need to be displayed in the Network Explorer window.

If you need to change the hierarchical order of your tree view, you can also drag and drop the properties in the Explorer tab, in order to customize your information display needs.

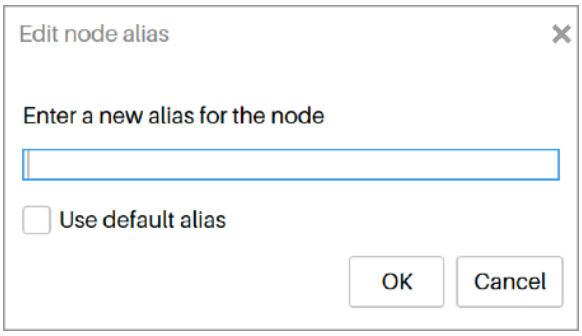


The Network Explorer is directly connected to the Graphic and Properties views. By the time you select a node on the Network Explorer, its equivalent nodes will be selected on the Graphic View as well as in the Properties View.

If you need to see the full information of a node, you have to right click on it, and select the 'View Properties' option, after this the Properties View window will open showing the selected node information, if this window is already open, you just have to click over the node you need to know about, and the information will be automatically loaded in the Properties View.



'Edit Alias', 'Go to packet first time seen' and 'Go to packet updated' are the other displayed options when you right click on a node. Selecting the 'Edit Alias' option, gives you the possibility to change this node property, a dialog window will be open with the current alias of the selected node loaded in a text input, to modify it delete the input data and enter the new alias you want to give to the selected element, to set the changes press the 'OK' button at the bottom right of the dialog, if you want to keep the same node alias just press the Cancel button. You can change the node alias anytime you need following the previously mentioned steps. If you want to revert the alias to its default value, check the 'Use default alias' option and press the 'OK' button in the dialog window and the node short address will be set as the node alias how it used to be at the beginning.



If you select the 'Go to packet first time seen' option, the correspondent row of the selected packet in the Traffic View will be highlighted, an equivalent action will happen if you select the 'Go to packet updated' option, the highlighted row will be that where the selected packet was updated.

The network nodes can be also a tool to filter your data captures in the Traffic View, right click on one of the nodes, select the 'Create New Filter' menu item, and a customized filter option for your selected node will appear for you to click it, following this action the Traffic View will only display those packets related with the selected node in the Traffic View.

Automatic vendor detection for network devices

The vendor name of your network devices can be seen in the tree view structure of the Network Explorer, making easier for you to identify them in the hierarchical representation. To be able to see this property, the long address of the device must have beed identified in the current capture, and only those vendors who are registered in the IEEE list (http://standards-oui.ieee.org/oui.txt) will be able to show this property in the Network Explorer view. The IEEE list will be loaded automatically in Ubiqua the first time you install the software, nevertheless the list can be updated every indeterminate lapse of time.

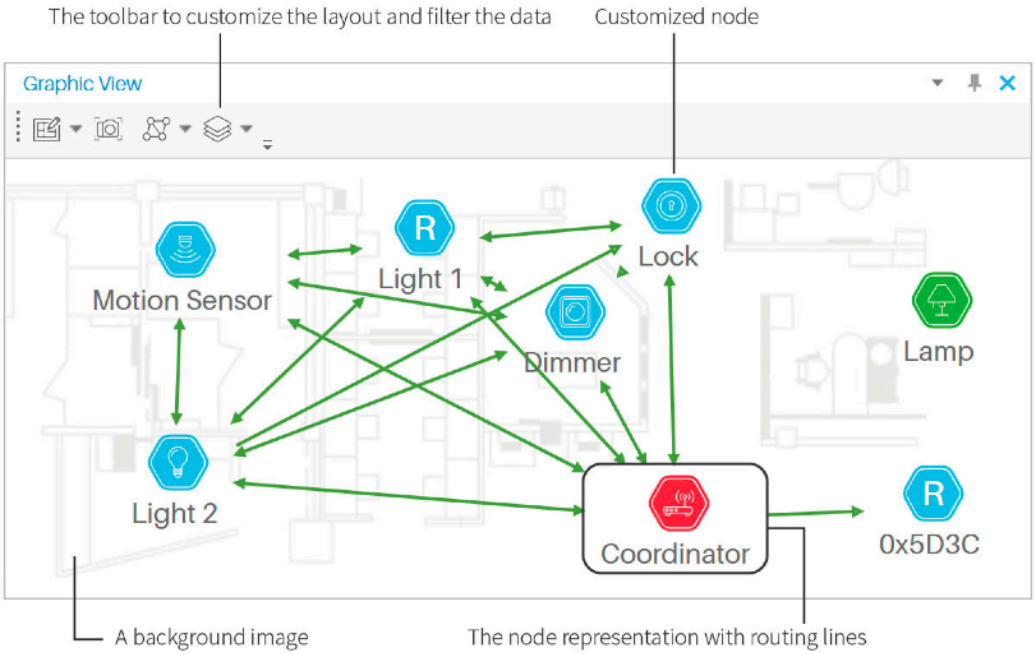
You can update your list directly from the IEEE website http://standards-oui.ieee.org/oui.txt, or download the .txt file and keep this information stored in your computer, to both update and load this local file, click the Tools > Options menu item, in the 'General' tab, navigate to the 'IEEE OUI (Vendors)' section at the bottom of the window. To update directly from the IEEE site click the 'Synchronize from IEEE' button, following this action an alert window will appear indicating the number of elements that have been added and updated, once you have done an online update, you can do another one within 2 hours, since this type of information is not updated constantly. To load the .txt file check the 'Synchronize from File' check box, after this load your local file, 2 update options will appear for you to choose, the 'Update' option which adds the information to the existing list, and the 'Replace' option that deletes the existing data and replaces it with the content of the document that is being loaded, once you have selected one of the options press the button 'Synchronize from File' at the bottom right of the window to finish this task.

Short address list

Every time a node joins the network, it is assigned a short address. If the node leaves and then rejoins the network, a new short address may be assigned to it, sometimes this maintains the same one. The record of this events is saved by Ubiqua and can be found in the ‘Network Explorer’ window, listed in the last level of every branch in the tree representation of this view. The list is compound by the current short address of the node which appears at the bottom of this, in case the node has been assigned short addresses that are no longer valid during the capture, these will appear strikethrough marked in the list. Note that to de able to display the 'Short Address’ list, this option must be checked in the ‘Network Explorer’ configuration window, under the ‘Show property’ column.

Chapter 8: Graphic View

This view is a graphical representation of the network topology as interpreted by the current capture data. Each time a new node and address is detected on an incoming packet, such node is included into the representation. The Graphic View is composed of a canvas and a toolbar (see the figure below). The canvas contains a network graph showing the current network nodes. The toolbar offers access to options such as adding a background, customizing the network graph layout, selecting what data is shown, and controlling the zoom. This chapter details each of those options.



Changing The Background

To change the background of the canvas, click on the Load Background Image toolbar button (see the figure below). The Open dialog will appear. Select the new background

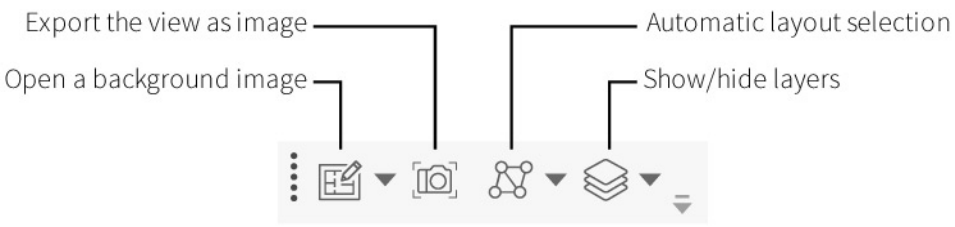
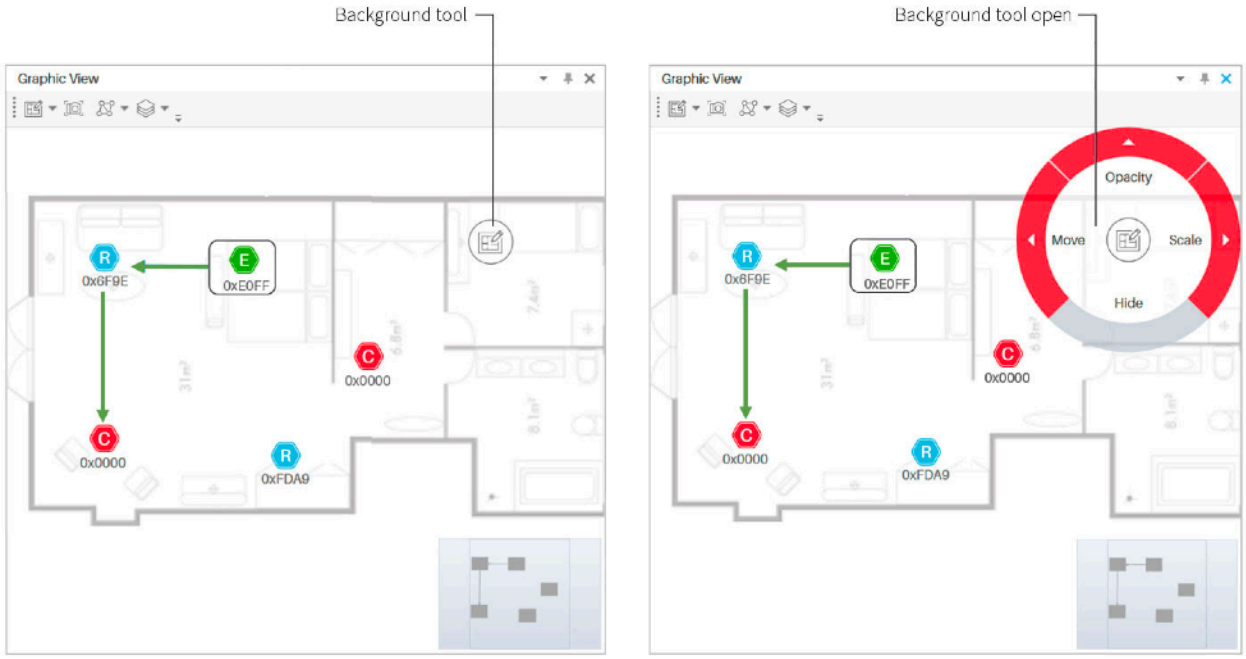


image and click the Open button. You can use images in the .bmp, .gif, .jpg, or .png file formats.

After adding a background to the canvas a Background Tool button appears in the top right section of the canvas area, this tool allows the user to move, zoom and change the transparency of the background image. This tool button can be hidden after use by clicking on the Hide button in the control or in the Show Tool menu item in the Background Tool button in the Graphic View Toolbar.



If the Background tool is not visible, you can show it again by clicking the Background button in the Toolbar and selecting the Show Tool Option.

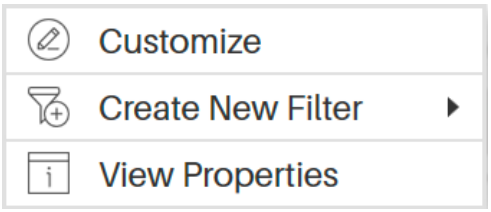
Customizing Nodes

Ubiqua has 4 different default node statuses icons for Zigbee networks, represented with colored hexagons in 4 different tones. A blue icon represents a router node, a green one represents an end device, a red icon represents a coordinator node and a gray one represents any other case.

For Thread networks, every node state is represented with the initials of every node's role and type. The L letter represents a Leader node, the R letter represents a Router node, the End Devices are represented with an E letter, the BR acronym represents a Border Router, and the LBR acronym is for a Leader Border Router.

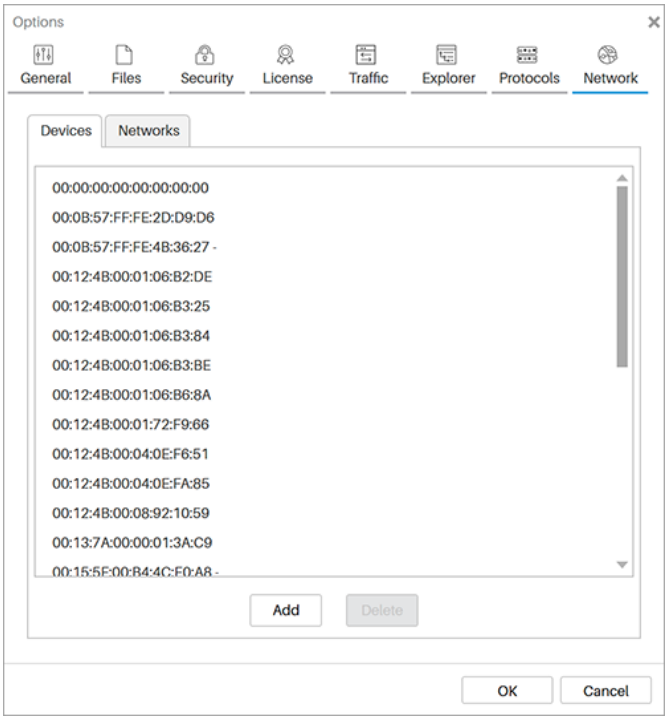
The appearance, position and alias of the nodes can be customized by you. You can change the position of the nodes by clicking and dragging each one within the Graphic View. The connecting arrows will move as the node does for persistence. Also, note that for this action is recommended that the Graphic and the Network Explorer Views are not hidden or auto-hidden.

If you want to easily identify the nodes of your network, Ubiqua allows you to visually customize them by letting you change their alias and their icon. To do so, right click on one of your Graphic View's nodes and select the 'Customize' option in the context menu to open this window. The node's 'IEEE Address' and 'Alias' will be displayed in it, being the second one only available for you to modify it.

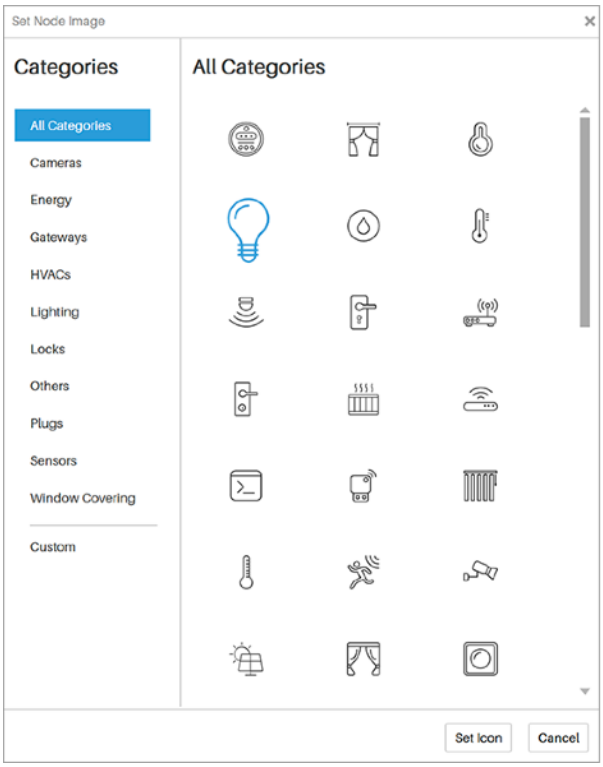


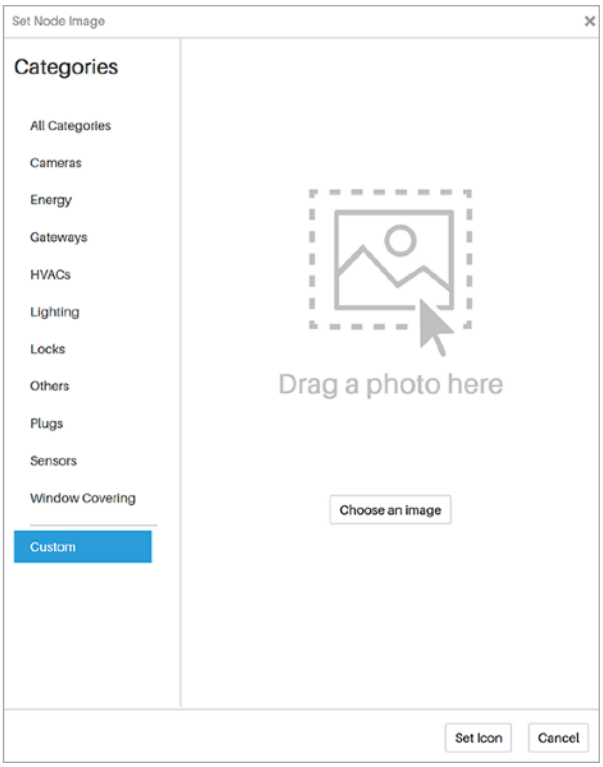
The 'Customize' window can also be opened from the 'Properties' view, clicking the 'Customize' button located in the 'Customizable Properties' expander below the 'Alias' input, as well as from the 'Options' window, under the 'Network' > 'Devices' tabs, a list of devices is displayed where each of them can be identified by their 'IEEE Address'. Right clicking on them and selecting 'Customize' option will also open that window, unlike the window opened from the 'Graphic View', the window opened from the device list displays the 'Short Address' of the selected device. Note that if the device has previously appeared in the network with different short addresses, they will be displayed in a list under the 'Existing Short Addresses' expander.

The 'Customize' window also has the function to let you change the node's icon. In the left side of the window you will see a circle with the legend 'Change Icon', click on it to open the



'Set Node Image' dialog, which is compound of 2 columns, the left one lists the different categories of the 'Ubiqua Icon Set' for your devices, the right one shows the correspondent icons of the selected category. To assign an icon to your node, click on one of the listed icons and then click on the 'Set Icon' button at the bottom right of the dialog, following this action the icon will be set and previewed in the 'Customize' window. If you prefer to add

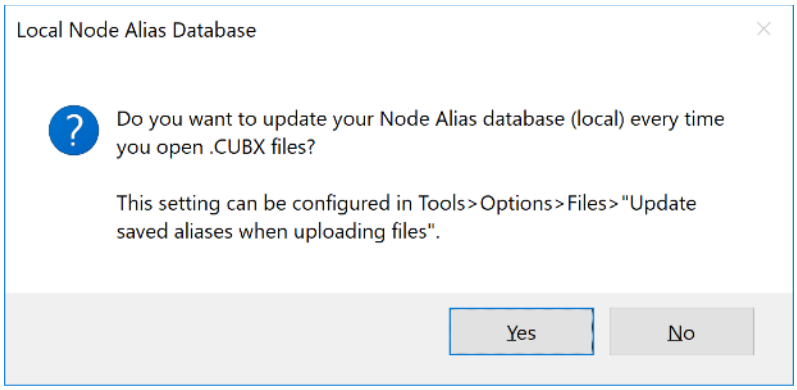




your personal image files to the network nodes, click on the 'Custom' option at the bottom of the 'Categories' column, after this a drag and drop area will be displayed for you to place your images, below this a button with the 'Choose an image' legend will also appear as an alternative option to open a file locator in your computer to load your images. The chosen images with the 'Custom' option will also be previewed in the 'Customize' dialog.

Once you have performed the previously mentioned steps, click the 'Save Changes' button to visualize the node customization, or 'Cancel' to keep the node's default properties. If you saved the changes the Graphic View will be updated with your new given Alias and Icon properties. When you modify a node alias or icon, this is saved in your configuration, this will help to remember your choice, and when the node appears again in the capture it will use the previously saved data.

By the time you have nodes aliases saved in your local configuration, the first time you open an external `.cubx` file, you will be asked in a dialog window if you want to update your local nodes aliases every time you open a `.cubx` file, if you press the Yes button those nodes that appear repeated in your local saved configuration as well as in the external `.cubx` file, will be renamed as they appear in the external file, those nodes that do not appear in the opened file will keep their local given alias. If you want yo change this configuration you can go to the Tools > Options > Files, uncheck the "Update saved aliases when uploading files" option and press the OK button.



Right clicking on your nodes also gives you the possibility to filter your Traffic View capture, select the 'Create New Filter' menu item, and a customized filter option for your selected node will appear for you to click it, following this action the Traffic View will only display those packets related with the selected node in the Traffic View.

Changing the Network Layers

Ubiqua shows two kind of relationships between network nodes: Topology and Routing. Each one of these are represented in the Graphic view as layers. The Topology layer for

Zigbee captures is constructed by examining the information of the MAC 2003 and 2006 association request/response message and are represented by blue color edges. The Routing layer for Zigbee captures is constructed by examining the information contain in the Zigbee Network Link Status messages and are represented by green color edges.

Spanning and Zooming

To span the Graphic View, click on any free space on the canvas and drag the mouse cursor. For zooming use the Overview map control or use the mouse wheel button.

Exporting Images

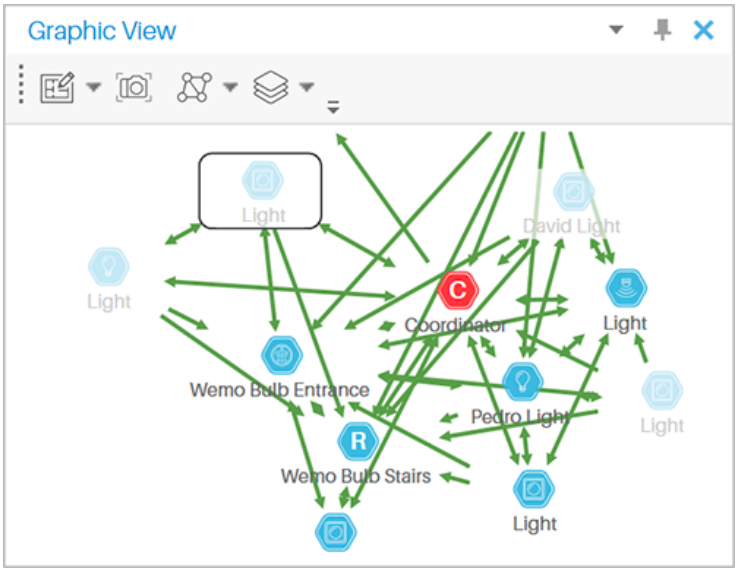
The Graphic View has a feature useful to share the visual representation of the network. You can export the contents of the view as an image. To do so, click on the Save Image toolbar button. The Save dialog will appear, select the location and the file format and click the Save button. You can save images in the .bmp, .gif, .jpg, .png, or .tif file formats.

Filtering Nodes

The Graphic View has the ability to hide and show nodes based on their PAN id or corresponding communication channel. To do this select the "Network Nodes" option from the "Layers" toolbar in the Graphic view, and select the way you want to Show/Hide the nodes, either by channel or Pan ID; next a list of the selected items will appear. To show an item just click to show a check mark, to hide it click the item for uncheck.

Ghost Nodes

The 'Graphic View' is the visual representation of the nodes that are part of the network, Ubiqua also allows you to see those nodes that have joined and then leaved the network during a current capture, this nodes are visualized in the topology with a clearer opacity than those nodes that are currently connected to the network, the incidence to the nodes to which they were connected will also be displayed.
If a node connects multiple times to the network, will also be reflected in the topology, drawing each one of the times that this node joined the network during the capture in the 'Graphic View'. To be able to see this behavior click the 'Layers' button in the toolbar of the 'Graphic View', then uncheck the 'Hide ghost nodes' option in the context menu, if you want those nodes not visible, keep the 'Hide ghost nodes' option checked.


Graphic View
Light
David Light
Light
C
Coordinator
Light
Wemo Bulb Entrance
Pedro Light
Light
R
Wemo Bulb Stairs
Light

Chapter 9: Properties View

The Properties View shows you detailed information of every node of the network, including Channel, Protocol, Status and so on. Analyzing the data in this window in conjunction with the Graphic View or the Network Explorer gives us both graphical and the technical information needed to understand the structure and interaction between the nodes of the network topology representation.

Information

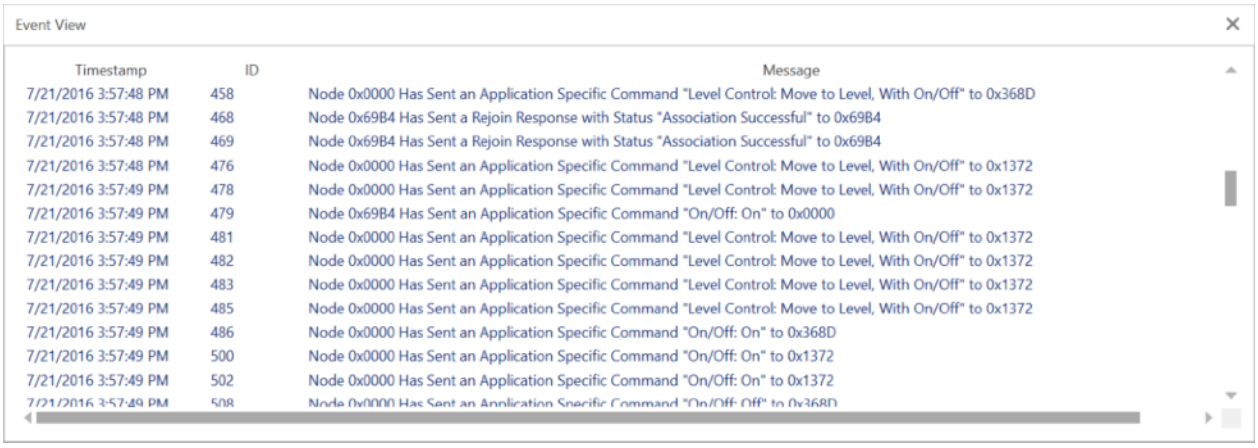
Every time you select a network node from the Graphic View or Network Explorer, its correspondent information will be automatically loaded in the Properties View, such as the protocol used to detect the node, the channel on which it is located, its address, status, among others.

Visualization

In the Graphic View section there are 4 different node icons to represent different states, nevertheless in the visualization area of the Properties View is the Icon and the Label fields, both for you to customize this node properties, this to make it easier for you to identify them in the network representation of the Graphic View. In the Icon field you have to browse on your computer an icon or image file to replace each of the the default icons that you would like to change, making this images work as a new graphic representation of the nodes. You can change the node label as well, you just have to erase the default data and type the new name you want to give to the node in the Label field.

Chapter 10: Event View

The Event View highlights notable traffic events in a dedicated listing. By clicking on an entry in the Event View, the corresponding packet will be selected in the Packet View.



Supported Zigbee events

Node Join
Triggered when a new node joins the network.

Touchlink Node Join
Triggered when a node joins the network using the touchlink mechanism.

Node Leave
Triggered when a node leaves the network.

Node Rejoin
Triggered when a node rejoins the network.

Application Command
Generic log of an application-layer command.

APS Transport Key
Triggered when Ubiqua registers a new Zigbee key.

Supported Thread events

Child Attach
Triggered when a node joins the network.

Network Data
Triggered when Ubiqua receives a Type Length Value payload.

Child Becomes Router
Triggered when a network node assumes the Router role.

Chapter 11: Output View

The Output View is an 'Active Subscription' feature that has the purpose to display the status messages generated for every petition response made to the Ubiqua Remote Services. These include the visualization of complete data captures, error responses, and the request/response output which can be made in external applications to the remote services will be seen in this window.

The toolbar for this feature includes from start to end, the Autoscroll, Toggle Word Wrap and Clear buttons.

- 'Autoscroll' button allows the user to take a shortcut through the window output, putting the cursor at the end of the last line of the content.

- 'Toggle Word Wrap' button in its active state breaks long text strings onto multiple lines, if this option is inactive the text is wrapped and an horizontal scroll is activated in case the content wider than the window to avoid the content overflow.

- 'Clear' button automatically scrolls to the top-left of the window and erases all the data displayed.

Chapter 12: Ubiqua Services

Ubiqua Protocol Analyzer provides a number of services exposing a limited set of the functionality available throughout the application as Web resources. The Remote Access Service is one of such services. In this chapter you will find the specification of its interface containing the provided resources, what HTTP methods are allowed, what the data representation formats are, and some examples of what you can expect as service outputs.

Remote Access Service

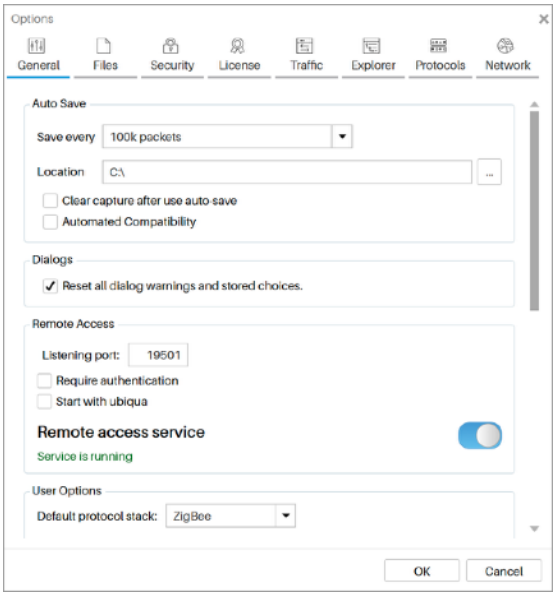
Enabling the Service

At the bottom right of Ubiqua there is a server icon that represents the status of your RESTful service, this can have 2 different states as shown in the image below, the left icon which is the default status appears when the service is not enabled and the icon with the green circle appears when the service is running.

Service is not running

Service is running

To enable the RESTful service, you can press the previously mentioned icon, or go to Tools > Options... in the main menu (see the figure below) and check the "Enable Remote Access Service" box; this box does not denote the service status, it only enables the service.



A label showing the service status is next to the right of the "Listening port:" box. The default port is 19501, to specify another one use the "Listening port:" option. The RESTful service will start after you click on the OK button. If you keep the enable option checked, the service will start as soon as the application starts. Once the application has started the default icon will be updated with the server icon with the green circle.

> **Service**
> As of Ubiqua 2.0, for better security and user protection, the Remote Services functionality requires an administrator to authorize the use of the network port assigned to Ubiqua. You can authorize the port by running the following command with a privileged account:
>
> **netsh http add urlacl url=http://*:19501/ user=Everyone listen=yes**
>
> The parameter user may vary depending on the language in which your operating system is, for example in English as shown in the command above user=everyone, in Spanish and Portuguese user=todos, Swedish user=alla, German user=Jeder, French user="Tout le monde", Dutch user=iedereen, Russian user=все.

If you are having problems connecting to the Remote Access Service, your Operating System may be blocking access to the configured port, refer to this link for more information.

If you want to secure the access, the service supports HTTP Basic Authentication. Just check the "Require authentication" box and provide a username and a password.

The service is still enabled if an error occurs and the status label will indicate the current status of the service.

Available Resources

The Remote Access Service provides 5 resources: `/capture`, `/sniffers`, `/filters`, `/keys` and `/addresses`. The `/capture` resource represents the current capture in Ubiqua, and it can be used to retrieve the current number of packets and a specific range of packets, to export the capture file to the local file system, or to clear all the packets. The `/sniffers` resource represents the sniffer devices currently attached and available in Ubiqua, it can be used to retrieve the list of all devices, the status of a specific device, and to start or stop a sniffer. The `/filters` resource represents the current filters in Ubiqua, and it can be used to retrieve the list of filters, to enable or disable a filter. The `/keys` resource represents the current security keys in Ubiqua, and it can be used to retrieve the list of

security keys, and to insert a new security key. The `/addresses` resource represents the current addresses in Ubiqua, and it can be used to retrieve the list of addresses, and to insert a new relationship of long address and short address.
The schema document for this service responses is located at

https://www.ubilogix.com/resources/ubiqua/services.xsd

Current Capture

GET / `capture`

Returns information about the current capture.

Request Parameters

- **offset** – An integer accepting positive numbers and zero 0. The offset of the packets to retrieve.

- **limit** – An integer accepting positive numbers and zero 0. The number of packets to retrieve.

Response Code 200 (OK)

An XML document with the information about the current capture.

Sample Request

A GET request to the URL `http://localhost:19501/capture?offset=10&limit=2`, produces a response with code 200, and the following document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Capture xmlns="urn:ubilogix:services">
  <Packets Count="48454" Offset="10" Limit="2">
   <Packet Id="11">
   <Info>Beacon</Info>
   <Source>File</Source>
     <Status>Normal</Status>
   <Raw>008038AA1A0000FFCF800000228411111111111111FFFFFF00FFFF</Raw>
  </Packet>
  <Packet Id="12">
   <Info>Beacon Request</Info>
   <Source>File</Source>
     <Status>Normal</Status>
   <Raw>03087DFFFFFFFF07FFFF</Raw>
  </Packet>
```

```
 </Packets>
</Capture>
```

PUT / `capture`

Performs the specified action on the current capture.

Request Parameters

- **action** – A string with one of the following values:
  - **clear**, to reset the capture.
  - **save**, to store the capture log in Ubiqua file format on the specified file location.
  - **export**, to store the capture log in other file format on the specified file location. Use one of the valid file extensions: `.txt`, `.xls`, `.csv`, `.opml`, and `.cubx`.
  - **load**, to open a file capture with any supported file format of Ubiqua.
- **filename** – A string with the file name to export the capture to or, the file name to open capture.

Response Code 200 (Created)

- The capture was successfully exported.
- The file is created with no packet information if there is no data to export.

Response Code 202 (Accepted)

- The capture was successfully cleared or, it is being exported but the process is not yet completed.

Response Code 400 (Bad Request)

- The file type of the file specified in the `filename` parameter is not supported by Ubiqua.

Response Code 500 (Internal Server Error)

- The export process failed.

Sample Requests

Save a capture by sending a PUT request to the URL `http://localhost:19501/capture`, with the following request body: `action=save&filename=C:\test.cubx` and a content-type request header containing `application/x-www-form-urlencoded`, produces a response with code 200 and an empty body.

Export a capture by sending a PUT request to the URL `http://localhost:19501/capture`, with the following request body: `action=export&filename=C:\test.txt` and a content-type request header containing `application/x-www-form-urlencoded`, produces a response with code 200 and an empty body.

Sniffer Devices

GET / `sniffers`

Returns a list of the sniffer devices connected to Ubiqua.

Response Code 200 (OK)

An XML document with the list of sniffer devices.

Sample Request

A GET request to the URL `http://localhost:19501/sniffers`, produces a response with code 200, and the following document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Devices xmlns="urn:ubilogix:services">
  <Sniffers>
    <Sniffer Id="USB5255" Link="http://10.0.1.9:19501/sniffers/USB5255">
      <IsStarted>false</IsStarted>
      <IsPlugged>true</IsPlugged>
      <Channel>25</Channel>
      <Protocol>6</Protocol>
    </Sniffer>
    <Sniffer Id="COM35" Link="http://10.0.1.9:19501/sniffers/COM35">
      <IsStarted>false</IsStarted>
      <IsPlugged>true</IsPlugged>
      <Channel>25</Channel>
      <Protocol>11</Protocol>
    </Sniffer>
```

```
    <Sniffer Id="01112004" Link="http://10.0.1.9:19501/sniffers/01112004">
      <IsStarted>false</IsStarted>
      <IsPlugged>false</IsPlugged>
      <Channel>11</Channel>
      <Protocol>0</Protocol>
    </Sniffer>
  </Sniffers>
</Devices>
```

GET / sniffers/{id}

Returns the information of the sniffer device with the specified ID.

Response Code 200 (OK)

Gets an XML document with the status of the sniffer device with the specified ID.

Response Code 404 (Not Found)

The sniffer with the specified ID was not found on the list of available sniffer devices.

Sample Request

A GET request to the URL `http://127.0.0.1:19501/sniffers/USB5255`, produces a response with code 200, and the following document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Sniffer         Id="Texas         Instruments         CC2531         [USB5255]"
xmlns="urn:ubilogix:services">
  <IsStarted>false</IsStarted>
  <IsPlugged>true</IsPlugged>
  <Channel>25</Channel>
  <Protocol>0</Protocol>
</Sniffer>
```

PUT /sniffers/{id}

Starts or stops the sniffer device of the specified ID.

Request Parameters

- **action** – A string with one of the following values:
  - **start**, to start the sniffer.
  - **stop**, to stop the device.
- **channel** – A value between 11 and 26 representing the physical channel (optional).
- **protocol** – A string with one of the following supported values (optional): **IEEE 802.15.4-2003, IEEE 802.15.4-2006, Zigbee, IP, Synkro RF, Raw Data, PopNet, Zigbee RF4CE, Zigbee IP, JenNet IP, IETF 6LoWPAN.**

Response Code 200 (OK)

Starts or stops a sniffer device as instructed by the `action` parameter.

Response Code 400 (Bad Request)

The sniffer can not start: channel out of range or protocol not supported.

Response Code 404 (Not Found)

The sniffer with the specified ID was not found on the list of available sniffer devices.

Response Code 503 (Service Unavailable)

The sniffer with the specified ID is already in the requested state.

Sample Request

A PUT request to the URL `http://localhost:19501/sniffers/USB5255`, with the following request body: `action=start, channel=16`, and `modulation=OQPSK_250` of content-type `application/x-www-form-urlencoded`, produces a response with code 200 and an empty body. The device was started.

Filters

GET /filters

Returns a list of the filters in Ubiqua.

Response Code 200 (OK)

An XML document with the list of filters.

Sample Request

A GET request to the URL `http://localhost:19501/filters`, produces a response with code 200, and the following document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Filters xmlns="urn:ubilogix:services">
    <Filter Name="Time Delta is 0.000000" IsEnabled="true" Link="http://
10.0.1.8:19501/filters/0" />
   <Filter Name="channel 21" IsEnabled="false" Link="http://10.0.1.8:19501/
filters/1" />
</Filters>
```

GET /filters/{id}

Returns the status and conditions of the filter with the specified ID.

Response Code 200 (OK)

Gets an XML document with the status and conditions of the filter with the specified ID.

Response Code 404 (Not Found)

The filter with the specified ID was not found on the list of filters.

Sample Request

A GET request to the URL `http://127.0.0.1:19501/filters/2`, produces a response with code 200, and the following document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Filter>
  <Name>Sequence Number is 10</Name>
  <IsEnabled>true</IsEnabled>
  <Conditions Match="All">
    <Condition>
      <TargetName>SequenceNumber</TargetName>
      <TargetDisplayName>Sequence Number</TargetDisplayName>
      <Operator>Is</Operator>
      <TargetType>Integer</TargetType>
      <TargetLength>8</TargetLength>
      <TargetValue>10</TargetValue>
      <TargetAdditionalValue />
    </Condition>
  </Conditions>
</Filter>
```

PUT / `filters`

Enables or disables the filter.

Request Parameters

- **action** – A string with one of the following values:
    - **enable**, to enable the filter.
    - **disable**, to disable the filter.
- **filter** – A string with the filter name.

Response Code 200 (OK)

Enables or disables the filter as instructed by the exttt{action} parameter.

Response Code 400 (Bad Request)

The filter does not exist, or the action is unknown.

Response Code 503 (Service Unavailable)

The filter cannot be enabled or disabled when a task is running.

Sample Request

A PUT request to the URL `http://localhost:19501/filters`, with the following request body: `action=enable&filter=Frame%20Type%20is%20Command` of content-type `application/x-www-form-urlencoded`, produces a response with code 200 and an empty body. The filter was enabled.

Security Keys

GET / `keys`

Returns a list of the Security keys in Ubiqua.

Response Code 200 (OK)

An XML document with the list of security keys.

Sample Request

A GET request to the URL `http://localhost:19501/keys`, produces a response with code 200, and the following document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Keys xmlns="urn:ubilogix:services">
 <Key Type="NetworkKey">76:D9:74:54:AE:A1:C6:1B:24:A1:90:96:0E:D1:D2:39</Key>
 <Key Type="LinkKey">A8:53:6D:BC:99:11:5B:E1:31:B5:F4:FD:A5:35:60:42</Key>
 <Key Type="LinkKey">E2:D6:45:F2:8C:A8:B0:E8:EF:7E:CC:E0:4C:BF:8D:BE</Key>
</Keys>
```

POST / keys

Inserts a new security key in Ubiqua.

Request Parameters

- **type** – A string with one of the following supported values: **PopNetKey, RF4CEKey, NetworkKey, LinkKey, Trust-CenterMasterKey, IEEE-802.15.4-2006Key.**
- **key** – A string with the new security key to add. The security key length must be 128 bits.

Response Code 202 (Accepted)

New security key was added with the specified key type.

Response Code 400 (Bad Request)

The key type is invalid, or the key length is invalid.

Sample Request

A POST request to the URL `http://localhost:19501/keys`, with the following request body: `type=NetworkKey&key=64:B9:99:95:4D:B1:F9:0F:20:F6:80:02:FB:FE:6F:C4` of content-type `application/x-www-form-urlencoded`, produces a response with code 202 and an empty body. The security key was added.

Network Addresses

GET / addresses

Returns a list of the addresses in Ubiqua.

Response Code 200 (OK)

An XML document with the list of addresses.

Sample Request

A GET request to the URL `http://localhost:19501/addresses`, produces a response with code 200, and the following document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Addresses>
 <AddressRelation>
  <LongAddress>0000000000000000</LongAddress>
  <ShortAddressList>
   <ShortAddress>0000</ShortAddress>
   <ShortAddress>7C89</ShortAddress>
   <ShortAddress>ECAA</ShortAddress>
  </ShortAddressList>
 </AddressRelation>
 <AddressRelation>
  <LongAddress>001DB70000033D6D</LongAddress>
  <ShortAddressList>
   <ShortAddress>F02D</ShortAddress>
  </ShortAddressList>
 </AddressRelation>
</Addresses>
```

POST / `addresses`

Inserts a new relationship of long address and short address.

Request Parameters

- **longAddress** – The long address to add, with a length of 8 bytes.
- **shortAddress** – The short address to relate with the specified long address, with a length of 2 bytes.

Response Code 202 (Accepted)

New addresses relationship was added.

Response Code 400 (Bad Request)

Either the longAddress or shortAddress value is invalid.

Sample Request

A POST request to the URL `http://localhost:19501/addresses`, with the following request body: `longAddress=001DB70000033D6D&shortAddress=40EA` of content-type `application/x-www-form-urlencoded`, produces a response with code 202 and an empty body. The addresses relationship was added.

Using The Command Line

If you prefer to interact with Ubiqua services from the command line you can use cURL, a free command line tool to send HTTP requests. Download cURL from http://curl.haxx.se/. This section shows some examples on how to use cURL with Ubiqua.

Getting the Sniffers List

```
C:\>curl -v "http://localhost:19501/sniffers"
* About to connect() to localhost port 19501 (#0)
*   Trying 127.0.0.1... connected
* Connected to localhost (127.0.0.1) port 19501 (#0)
> GET /sniffers HTTP/1.1
> User-Agent: curl/7.21.6 (i386-pc-win32) libcurl/7.21.6 OpenSSL/0.9.8r zlib/
1.2.5
> Host: localhost:19501
> Accept: */*
>
< HTTP/1.1 200 OK
< Content-Length: 341
< Content-Type: application/xml
< Server: Ubiqua-RemoteControl/1.3.2183 Microsoft-HTTPAPI/2.0
< Date: Wed, 25 Sep 2013 00:18:04 GMT
<
<?xml version="1.0" encoding="UTF-8"?>
<Devices xmlns="urn:ubilogix:services">
  <Sniffers>
    <Sniffer Id="01TP2RLY" Link="http://10.0.1.8:19501/sniffers/01TP2RLY">
      <IsStarted>false</IsStarted>
      <IsPlugged>true</IsPlugged>
      <Channel>25</Channel>
      <Protocol>0</Protocol>
    </Sniffer>
  </Sniffers>
</Devices>
* Connection #0 to host localhost left intact
* Closing connection #0
```

Starting a Sniffer

```
C:\>curl -v -X PUT -d "action=start" "http://localhost:19501/sniffers/
01TP2RLY"
* About to connect() to localhost port 19501 (#0)
*   Trying 127.0.0.1... connected
* Connected to localhost (127.0.0.1) port 19501 (#0)
> PUT /sniffers/01TP2RLY HTTP/1.1
> User-Agent: curl/7.21.6 (i386-pc-win32) libcurl/7.21.6 OpenSSL/0.9.8r zlib/
1.2.5
> Host: localhost:19501
> Accept: */*
> Content-Length: 12
> Content-Type: application/x-www-form-urlencoded
>
< HTTP/1.1 200 OK
< Content-Length: 0
< Server: Ubiqua-RemoteControl/1.3.2183 Microsoft-HTTPAPI/2.0
< Date: Wed, 25 Sep 2013 00:12:34 GMT
<
* Connection #0 to host localhost left intact
* Closing connection #0
```

Saving the Capture Log

```
C:\>curl -v -X PUT -d "action=save&filename=c:\data.pcap" http://localhost:
19501/capture
* About to connect() to localhost port 19501 (#0)
*   Trying 127.0.0.1... connected
* Connected to localhost (127.0.0.1) port 19501 (#0)
> PUT /capture HTTP/1.1
> User-Agent: curl/7.21.6 (i386-pc-win32) libcurl/7.21.6 OpenSSL/0.9.8r zlib/
1.2.5
> Host: localhost:19501
> Accept: */*
> Content-Length: 35
> Content-Type: application/x-www-form-urlencoded
>
< HTTP/1.1 200 OK
< Content-Length: 0
< Server: Ubiqua-RemoteControl/1.3 Microsoft-HTTPAPI/2.0
< Date: Wed, 25 Apr 2012 01:20:03 GMT
<
* Connection #0 to host localhost left intact
* Closing connection #0
```

Sample Source Code

The code snippets listed in this chapter are written in the C# programming language and are intended to be used with the .NET Framework.

Getting the Sniffers List

```
using System;
using System.IO;
using System.Net;
using System.Text;

// Create the web request
HttpWebRequest request = WebRequest.Create(
    "http://localhost:19501/sniffers") as HttpWebRequest;

// Get response
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());

    // Console application output
    Console.WriteLine(reader.ReadToEnd());
}
```

Starting a Sniffer

```
using System.Web;

Uri address = new Uri("http://localhost:19501/sniffers/USB5255");

// Create the web request
HttpWebRequest request = WebRequest.Create(address) as HttpWebRequest;

// Set type to PUT
request.Method = "PUT";
request.ContentType = "application/x-www-form-urlencoded";

// Create the data we want to send
string action = "start";

StringBuilder data = new StringBuilder();
data.Append("action=" + HttpUtility.UrlEncode(action));

// Create a byte array of the data we want to send
byte[] byteData = UTF8Encoding.UTF8.GetBytes(data.ToString());

// Set the content length in the request headers
request.ContentLength = byteData.Length;
```

```
// Write data
using (Stream putStream = request.GetRequestStream())
{
    putStream.Write(byteData, 0, byteData.Length);
}

// Get response
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());

    // Console application output
        Console.WriteLine(reader.ReadToEnd());
}
```

Getting the Capture Status With Authentication

```
// Create the web request
HttpWebRequest request
    = WebRequest.Create("http://localhost:19501/capture") as HttpWebRequest;

// Add authentication to request
request.Credentials = new NetworkCredential("test", "test");

// Get response
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    // Get the response stream
    StreamReader reader = new StreamReader(response.GetResponseStream());

    // Console application output
    Console.WriteLine(reader.ReadToEnd());
}
```

Running Ubiqua As Server

To run Ubiqua as a server just add the `/server` argument to your Ubiqua shortcut. Or run Ubiqua from the command line as `Ubiqua.exe  /server`. Alternatively, if Ubiqua is already running, select the Window > Minimize To Tray menu item.

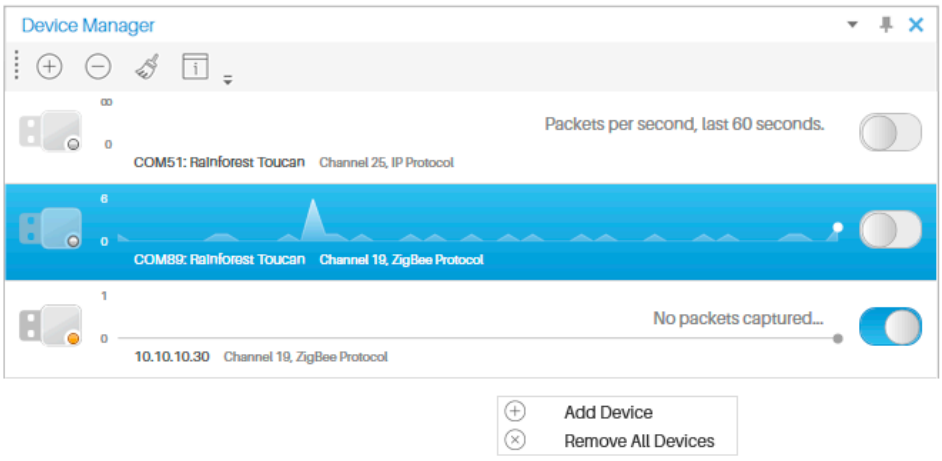
Postman Collection for Debug and Testing

To run your test of Ubiqua Services you can download this Postman Collection.

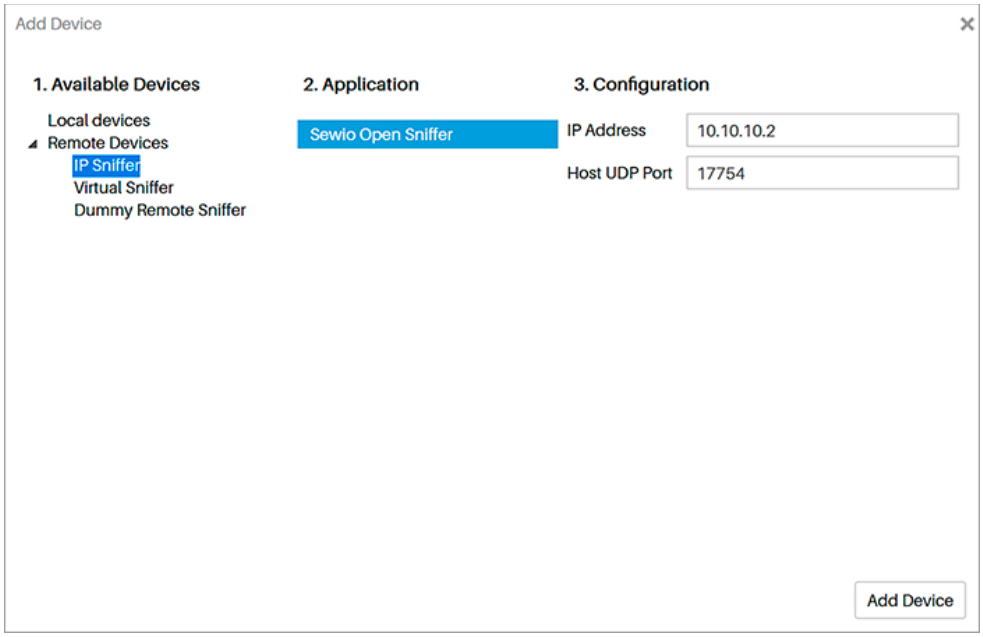
Chapter 13: Sewio Hardware

In order to use the Sewio Open Sniffer in Ubiqua, it must be connected and configured in the same subnetwork as the computer running Ubiqua. Please refer to "Part 1" of the sniffer installation http://www.sewio.net/open-sniffer/sniffer-installation/.

The process of configuring the sniffer in Ubiqua is simple. First, we add the device by right clicking on the Device Manager View or via the Device main menu.

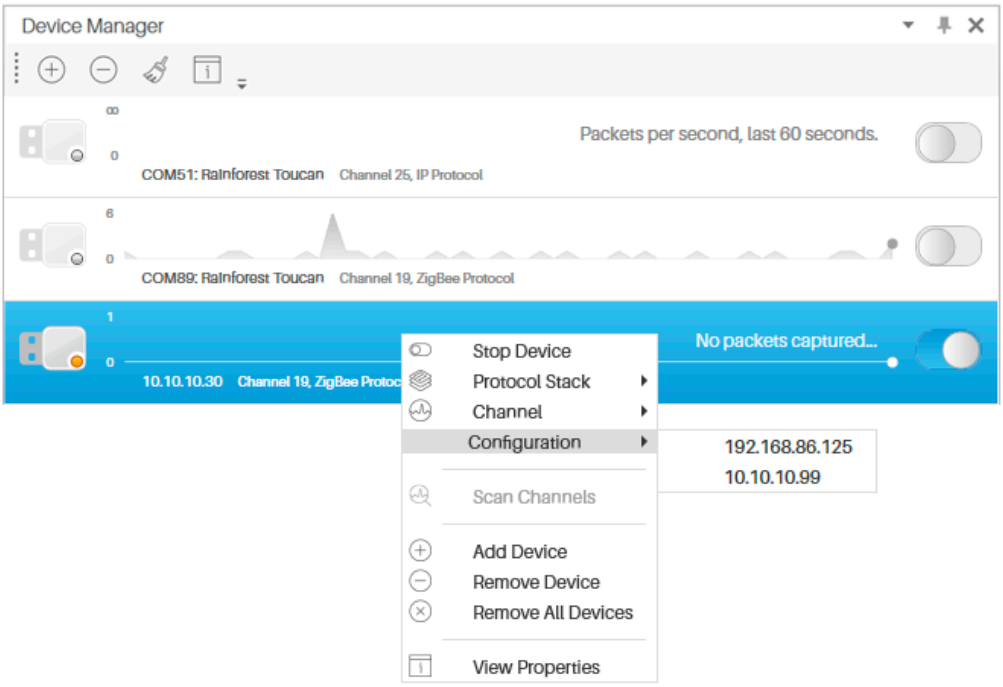


In the Add Device window select Sewio as the Vendor.



Then type the configuration parameters (IP Address and Port number) of the Sewio device, set the IP Address of the Sniffer and the UDP port that you want Ubiqua to listen for the packets. If you're configuring multiple sniffers, it is recommended to use a different UDP port for each sniffer.

At this point the Sewio device is added to Ubiqua. Before starting the device it's important to set the host, channel and protocol stack of the capture data using the context menu that appears by right clicking on top of the device. Usually the device is configured in the "10.10.10.XXX" IP address range. Lastly, select Start Device to begin capturing packets.



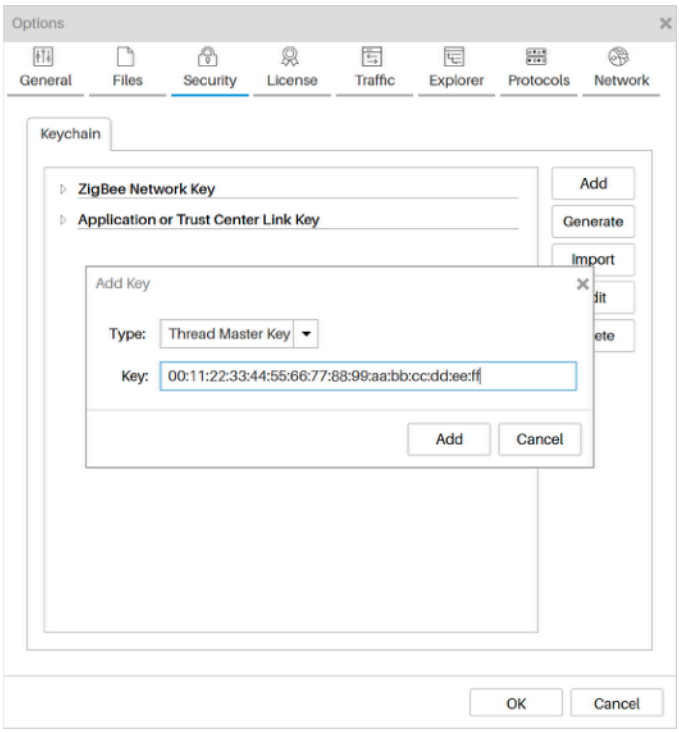
Note that if you change the device configuration from the Sewio web interface, the sniffer may stop sending packages to Ubiqua, just stop and restart the device from the Ubiqua Device Manager to correct the situation.

Chapter 14: Thread Support

Ubiqua decodes all protocols used by Thread such as IEEE 802.15.4-2006, 6LowPAN, UDP, DTLS, CoAP, DHCPv6 and MLE. To easily distinguish between protocols, each of them has a different color (see Traffic View).

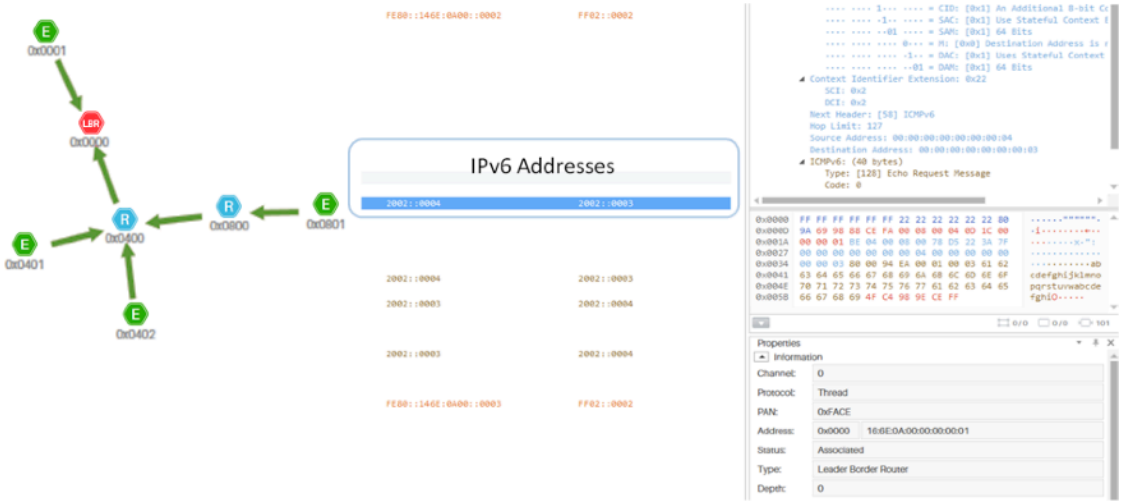
Ubiqua can decrypt Thread packets that use IEEE 802.15.4 2006 security (MAC Layer) and MLE encryption.

There are three types of keys used by Ubiqua, Thread Master Key, MLE key and MAC key to decode Thread packets. With the keychain used to storage all keys, a Thread Master Key can be used to derive a MLE or MAC key, and if the decrypting process is successful the MLE and MAC keys will also be stored on the keychain.



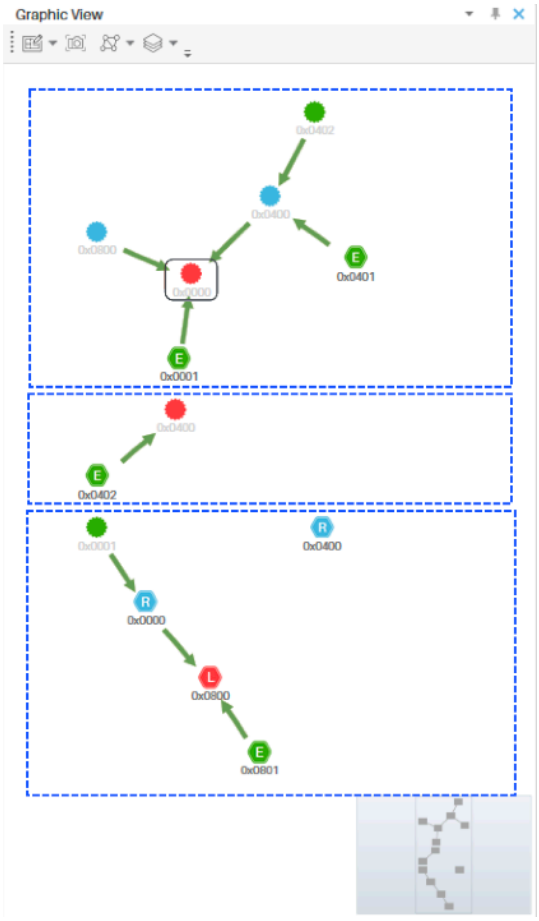
Network Data

Thread uses a lot of network data information to manage the generation of the ipv6 addresses and to identify the devices into the networks that work as border routers, commissioners or collapsed devices. Ubiqua collects this network data to generate IPv6 networks addresses (stateful using 6lowpan context id and stateless), etc. this information is displayed in the Graphic View.



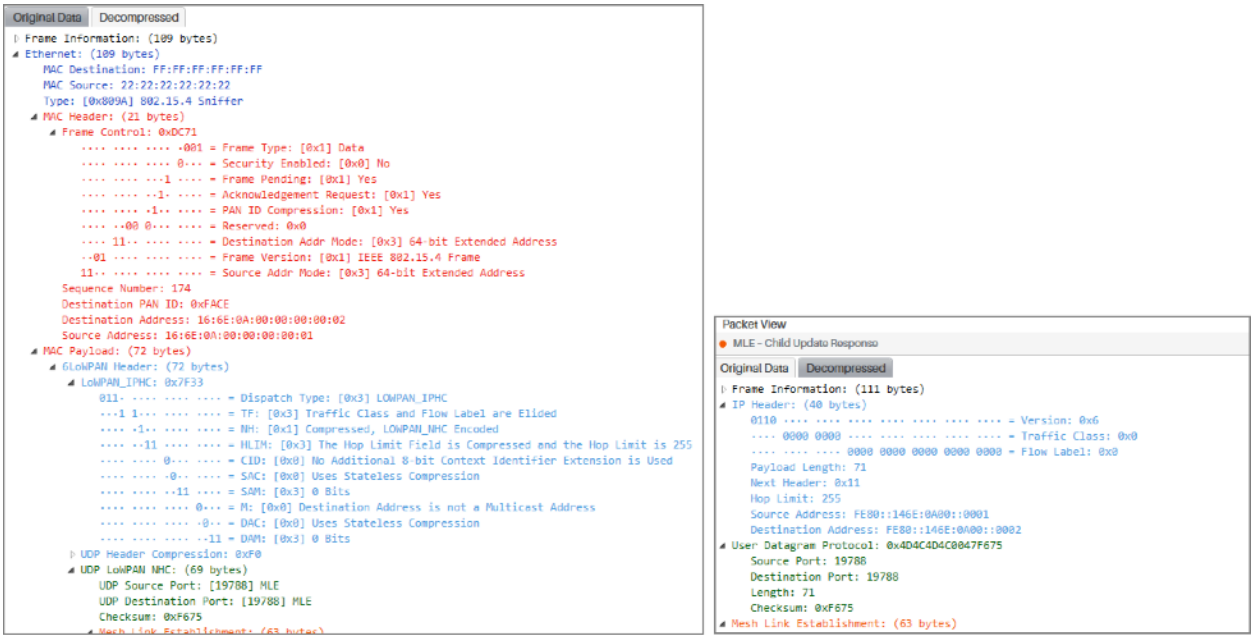
Graphic View

Ubiqua is capable to show the topology by partition ID used in Thread. If a capture has several partitions into the same PAN ID and channel, Ubiqua arranges the partitions by order of creation. The image below shows a topology formed by 3 different partitions.



Packet View

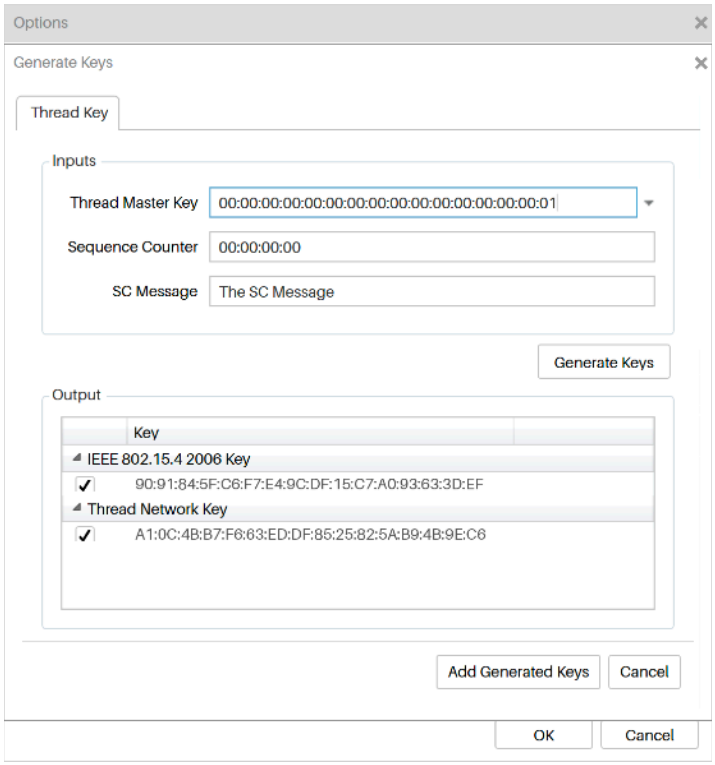
Ubiqua's Thread support adds a tab in the Packet View called "Decompressed". This tab shows the decompressed IP Header (from 6lowpan layer).



In Thread a specific node's behavior is very dynamic and the data of several packets needs to be analyzed to show the topology, addressing, security and behavior context. Ubiqua's Thread Packet View implements more than just a simple decoder by also showing information derived from previous packets to help engineers to easily analyze the capture.

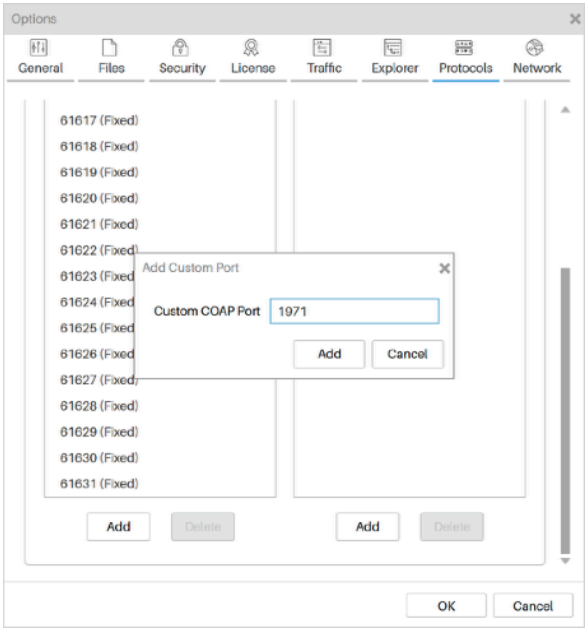
Generate Key

Some times you start a capture after the encryption Keys have been exchanged, and although Ubiqua performs a fair job in trying to get them for you, some times the traffic does no allow for it. In these cases you can make use of the "Generate" feature as follows: First click theTools > Options Menu, to show then Options window , select the "Security" Icon and then the "Keychain" tab. From the right side of the Keychain tab a column with 5 buttons will appear, click the "Generate" button to show the "Generate Keys" window. Input the known values for the "Thread Master Key", "Sequence Counter" and "SC Message" fields and finally click the "Generate Keys" button. You will see the 2 new keys in the output section ("IEEE 802.15.4 2006 Key" and the "Thread Network Key"). To save the Keys in the Keychain click the "Add Generated Keys" button or click "Cancel" to close the Generate Keys Window.



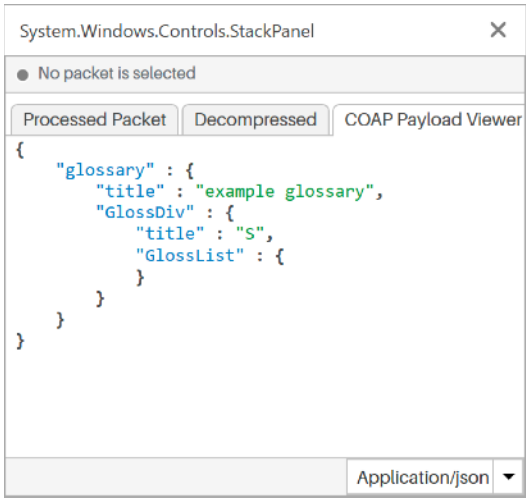
Define custom UDP ports of 6LowPan messages to decode payload as COAP or DTLS

6LowPan packets contain UDP ports that indicate how to decode the payload, Ubiqua knows how to decode a handful of UDP ports either as MLE, COAP, DTLS or another custom application protocol. However there could be some UDP ports that are application-defined to decode as COAP or DTLS and Ubiqua does not decode as such. For these cases, there is a COAP/DTLS custom UDP port feature in Options where the user can tell Ubiqua to treat a specific UDP of a 6LowPAN as either COAP or DTLS. To add the COAP/DTLS port number to your 'Custom COAP Ports' list. Click the Tools > Options menu item and then select the Protocols tab, in the tab body click the Thread expander, following this action you will be presented with the 'Custom COAP Ports' and the 'Custom DTLS Ports' columns, in both of them you will see a list of the default ports already handled by Ubiqua, these are composed by the port number followed by a '(Fixed)' postfix tag, the ports added by the user just show the port number. You can add and delete COAP and DTLS ports, note that the default ports cannot be deleted. Once you have added a port to one of the 2 lists, this won't be able to be added to the other list, to do so you have to delete it from the list where it is, and then add it to the other list. Please note that if a packet is already decoded in Traffic View, you will not see the changes until the packet is re-decoded again, clicking on the packet in Traffic View or reloading the entire capture is the easiest way to re-decode it.



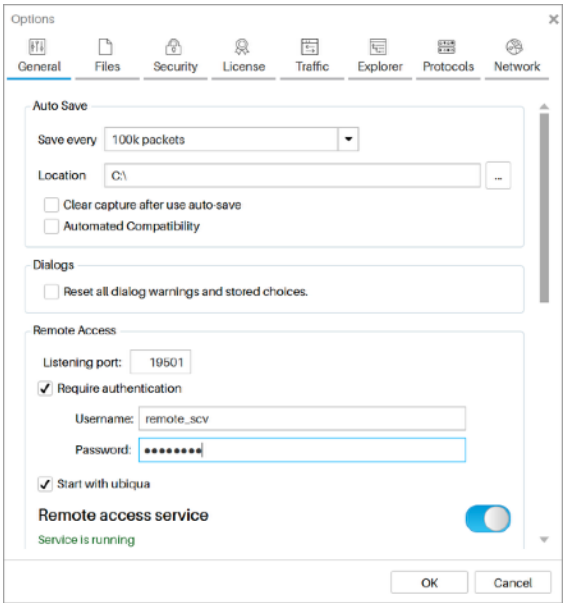
Display of COAP payload data in different content types

The COAP payload data can be decoded and displayed in the Packet View in CBOR, JSON, XML or TEXT format. After the packet is decoded and shown in the Traffic View, click on a Thread packet and see its information in the Packet View, a third tab with the 'COAP Payload Viewer' label will be added to the window, click on this tab and you will be able to see the Payload body message in formats like CBOR, JSON, XML or TEXT. At the bottom right of the window there is a dropdown menu with a list of formats in which you can display your payload, you can change the format of the content message selecting the different options of the dropdown menu. For CBOR data, if the message cannot be decoded in that format an error message will be displayed.



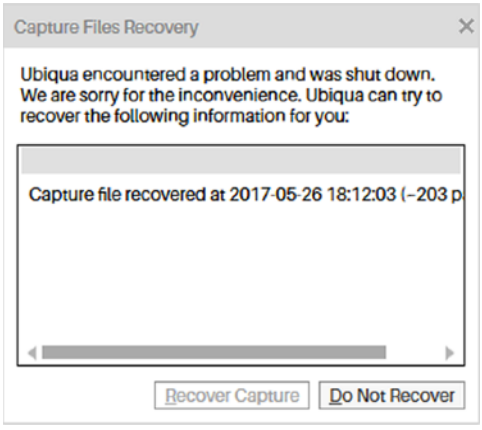
Chapter 15: Setting Preferences

To start setting your preferences open the Options dialog by selecting the Tools > Options... menu item on the main window of Ubiqua. You will find more instructions on the following sections.



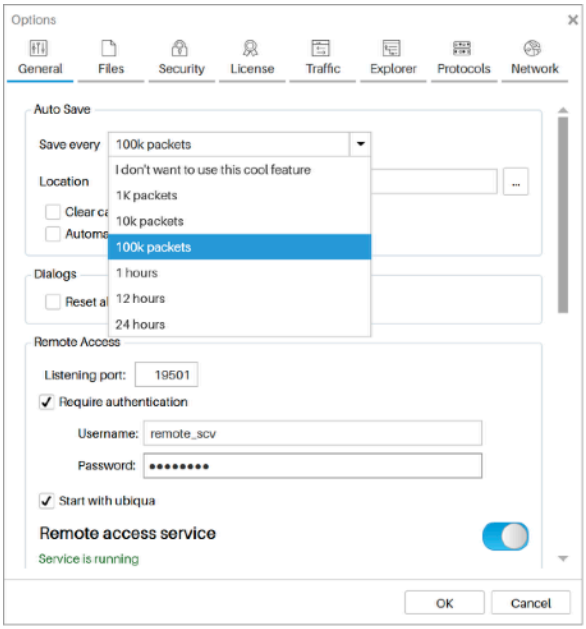
Capture Files Recovery

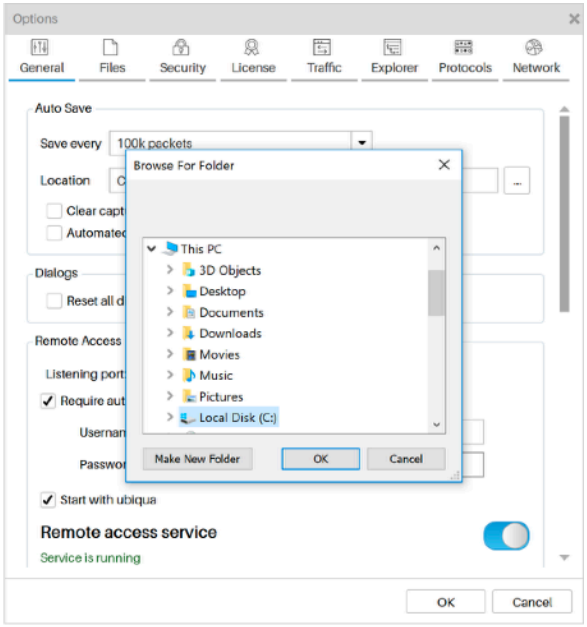
Ubiqua will help you recover the last capture if an error occurred recently and the application closed unexpectedly. When the application is started it checks for orphan files and presents a Capture Files Recovery dialog where you can select from a list of recoverable files with the corresponding date and time of when the problem occurred and the number of packets, select the file and click on the Recover Capture button and all the packets will be automatically loaded on your Traffic view.



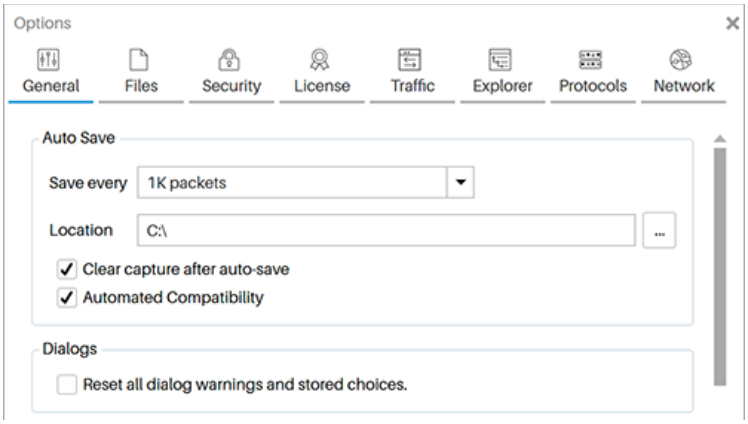
Auto Save

The ‘Auto Save’ feature automatically creates segmented files of live captures, allowing the user to set the option to save capture files every certain number of packets or determine a time interval to perform this task. This feature is helpful for cases where there will be long periods of capture time. When enabled, Ubiqua may generate multiple files in the path that was indicated, each file will contain a segment of the full trace. To enable this feature, open the ‘Tools > Options’ menu, go to the General tab and find the ‘Auto Save’ section.





Below the ‘Location’ field you can find the ‘Clear capture after use auto-save’ and ‘Automatic Compatibility’ options. When the ‘Clear capture ...’ option is checked, every time the 'Auto Save' task is performed, the ‘Traffic View’ will be automatically cleared, deleting all the capture packets that were saved from the window, and displaying only those packets that will be part for the next ‘Auto Save’ segment.



If the ‘Automation Compatibility’ option is selected by the time an unexpected close occurs, Ubiqua automatically will restart and will not show any of the dialogs (i.e. capture recovery, crash report) , which makes it a good choice for those who are running automated tests. Recovery dialogs are available throughout the main menu inside the application.

Dialogs

Ubiqua implements dialogs to remember the user's choices within the system. This option is used to remember the user preference and to suppress the same dialog in the future. The dialog appears when a capture file that does not contain protocol information is opened. The user can choose one of the supported protocols to decode it and selecting 'remember my choice' set the selected protocol as default to open the following captures. A dialog opens if the Frame Check-Sum validation fails, and the user can choose among to append 2 more bytes with the correct checksum, to replace the last 2 bytes with the correct checksum, or do nothing; the "remember my choice" will suppress the dialog when opening following captures. Finally, a dialog will open each time a new filter is created or the selected filter is edited; the "remember my choice" will suppress the dialog and it will apply the last user's choice. The stored dialog selections can be reset using the "Reset all dialog warnings and stored choices" box on the General tab.

Remote Access

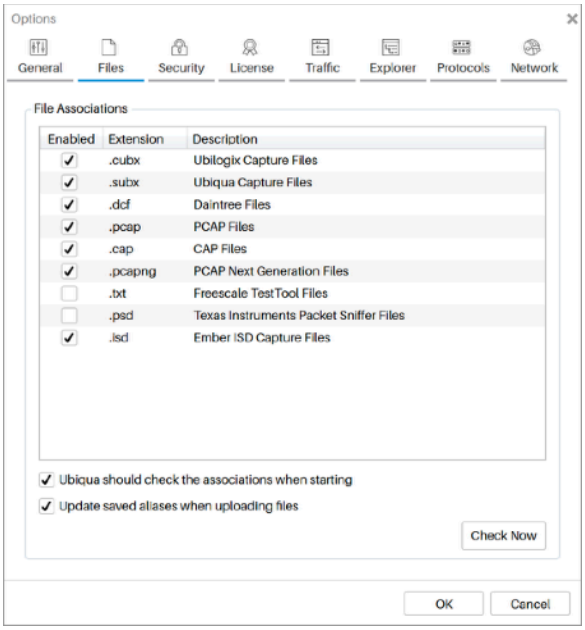
This option toggles on or off the Ubiqua Services which are web resources that expose certain Ubiqua functions for convenience. See here for more info.

Protocol Options

From this section you can change default protocol settings for unrecognized loaded capture files and customize other decoding options, such as the CoAP Custom Port that allows Ubiqua to listen on a designated UDP port and decode incoming traffic as CoAP.

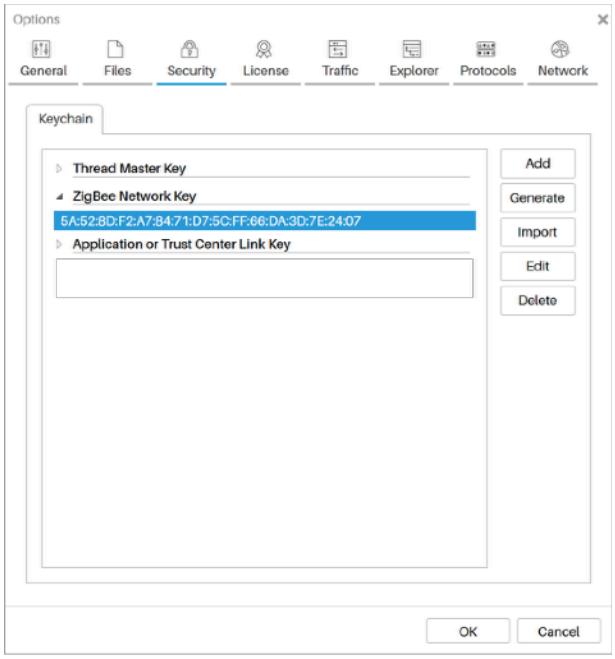
File Associations

You can set up Ubiqua to open capture files by double clicking on them in your system. To do so, open Options dialog, select the Files tab, and you will see a list of the supported capture files and extensions. Click the checkboxes on the Active column to associate Ubiqua to that file format. Use the checkbox at the bottom to make Ubiqua check if the associations are correctly set on your system each time it starts.



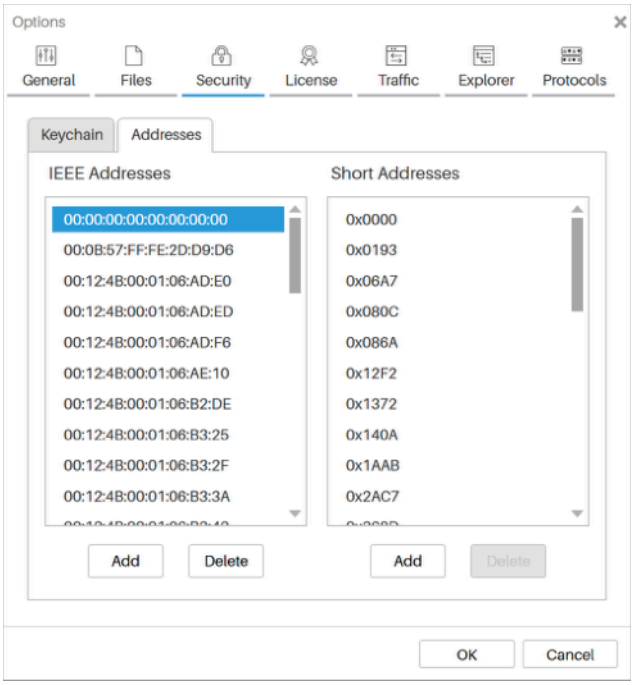
Security Keys

Security keys for decoding purposes are managed on the Security > Keychain tabs of the Options dialog. In this tab, you are presented with a list of all the security keys available in Ubiqua. This keys could have been discovered in the capture data or they could have been added manually. Use the buttons on the right to add, edit, delete, or import keys.



Addresses

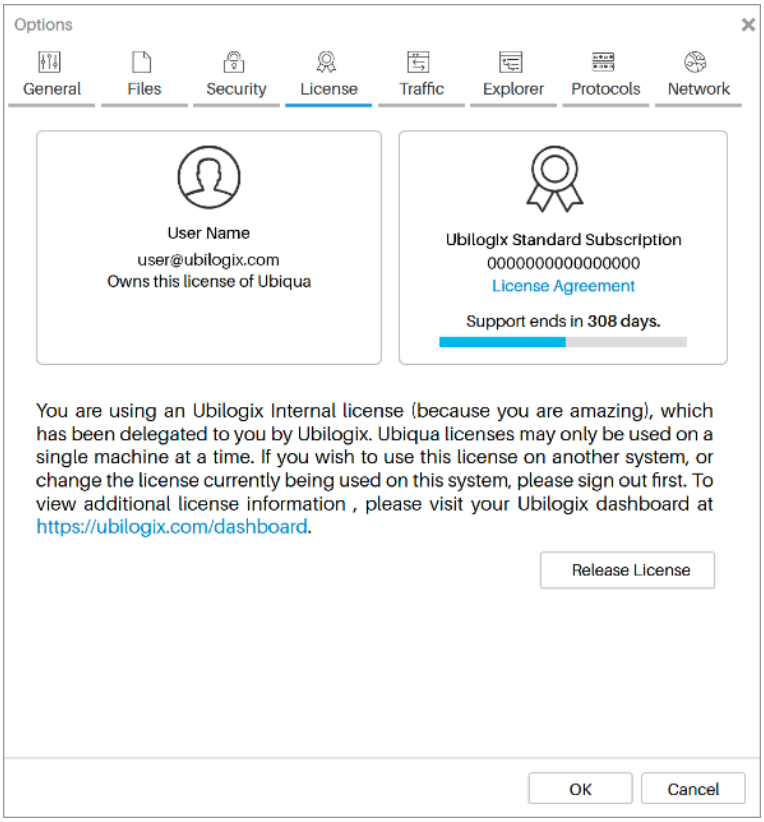
Addresses are the relationships between long address and short address used when decoding packets. Their management is similar to the security keys case. The Options dialog includes a list of all the available addresses on the Security > Addresses tab. Use the buttons on the right to add, or delete one of the addresses.



License Management

The License Management option is the most important part of the Options dialog. You will find it in the License tab. Here, you can see who the owner of the license is (name, email, and business). At the top right section, the status of the license is shown in terms of how many remaining days are left for the license to be valid (in case of the 21/days evaluation) or the number of days left of support (in case of a full license).

Your Computer ID is also displayed on this section. Ubilogix uses this ID to match your license with your computer. Many of the actions of your license (such as transferring), are linked to this number so be sure to include it when requesting changes to your license to support@ubilogix.com.
Use the buttons at the bottom to upgrade your current license to a more inclusive one, to start the license transfer process, or to regenerate your certificate file. Note that an active Internet connection is required for all these processes.



Check for Updates

The Check for Updates option is not included on the Options dialog. You will find it on the Help > Check for Updates... menu item. Use this option to know if a new version of Ubiqua is available, and if you are entitled to it (according to your license).

Environment Files

Environment files are loaded and saved from the File > Load/Save Environment menu items. These files include all the current selections and values of the Options dialog including address relationships, security keys, and others. Environment files do not store any license information.

Chapter 16: Supported Hardware

Ubiqua Protocol Analizer supports different sniffer hardware devices as specified in the table below. All the supported hardware drivers are located in `Program Files\Ubilogix\Drivers` folder in the drive where you installed the software. As provided from the vendor each USB dongle is already programmed with a sniffer firmware application. If you need more information about the sniffer firmware application or the sniffer drivers please refer to the Ubilogix support webpage.

	Vendor	Name	Part Number	Drivers Folder
Atmel	Atmel	RZ600 Evaluation Kit	ATZB-X-212-USB (SubG), ATZB-X-23x-USB, AT86RF212 (SubG) and the AT86RF23x families	`\Drivers\ATMEL`
NXP	NXP	MC1322x USB	1322X-USB	`\Drivers\FTDI`
NXP	NXP	JN5168 USB Dongle	DR1198	`\Drivers\FTDI`
NXP	NXP	JN5169 USB Dongle	OM15020	`\Drivers\FTDI`
NXP	NXP	JN5179 USB Dongle	OM15021	`\Drivers\FTDI`
NXP	NXP	KW2x: Kinetis KW2x Family	TWR-KW24D512 and KW24D512-USB	`\Drivers\FS KW2x`
NXP	NXP	Kinetis® KW41Z-2.4 GHz	KW41Z	
rainforest	Rainforest Automation	TOUCAN Wireless Sniffer	RFA-Z106-SN TOUCAn	`\Drivers\FTDI`
	Texas Instruments	CC2531 Ev. Module Kit	CC2531EMK	`\Drivers\TI`
sewio	Sewio	Open Sniffer	Open Sniffer for 802.15.4, Zigbee, 6LoWPAN	
ubisys	Ubisys	Zigbee USB Stick	Zigbee USB Stick U1	`http://www.ubisys.de/en/smarthome/download-software.html`

Chapter 17: Remote Sniffers

The Remote Sniffer Device system enables Ubiqua users to collect traffic data from stand-alone sniffer devices deployed in distant locations, with minimal effort required in terms of configuration and maintenance. Furthermore, it allows for the real-time sharing of captured data with multiple collaborators.

A supported Remote Sniffer Device (RSD) is configured to establish a connection with either our cloud-based Nexus service, or a private Nexus instance running on a controlled network. This service in turn proxies traffic and commands to and from authorized Ubiqua clients. The RSD owner may choose to share access to the device with other Ubiqua users.

Due to initiating rather than accepting connections, the RSD will normally not require manual configuration of firewalls in order to function.

Security and privacy

Traffic between the RSD and the Nexus service, as well as traffic between the Nexus service and Ubiqua, is sent over a TLS-encrypted websocket connection. The Nexus service does not store or process the transmitted information in any way, other than to forward it to the end-user client applications.

Support for end-to-end encryption between RSDs and Ubiqua clients is currently in development.

Mute Nexus sniffer

The traffic data between a Nexus service and Ubiqua can be simultaneously accessed by up to 10 users which can perform actions on this remote device such as change its channel or turn the service off. The actions executed on the device by one of the users will be reflected in all those who have access to this remote service. If one of the users needs to stop viewing the Nexus Sniffer without affecting the other users, it just have select the Nexus sniffer in the 'Device Manager', right click on the remote device and click the 'Mute Nexus Sniffer' option, following this action the real time transmission data service will stop to the current user. To resume the service right click on the remote device and press the 'Unmute Nexus Sniffer' toggle button and the data transmission will be reactivated in case the sniffer still on.

Chapter 18: Supported Protocols

IEEE 802.15.4

The 802.15.4 is a standard developed for Wireless Personal Area Networks (WPANs). WPANs convey information over short distances among the participants in the network. They enable small, power efficient, inexpensive solutions to be implemented for a wide range of applications and types of devices. Some key characteristics of an 802.15.4 network are:

- An over the air data rate of 250 Kbit/s in the 2.4 GHz band.
- 16 independent communication channels in the 2.4 GHz band.
- Large networks (up to 65534 devices).
- Devices use carrier sense multiple access with collision avoidance (CSMA-CA) to access the medium.
- Devices use Energy Detection (ED) for channel selection.
- Devices inform the application about the quality of the wireless link (Link Quality Indication).

The 802.15.4 Standard defines two network topologies, both using one and only one central device (the PAN coordinator). The PAN coordinator is the principal controller of the network.

- **Star Network Topology** – In a star network, all communication in the network is either to the PAN coordinator or from the PAN coordinator. That is, communication between non-PAN coordinator devices is not possible.
- **Peer-to-Peer Network Topology** – In a peer-to-peer network, communication can occur between any two devices in the network as long as they are within range of one another.

For more information visit the IEEE 802.15 Working Groups web sites for the latest specifications and other information: http://www.ieee802.org/15/pub/TG4.html

Zigbee

Zigbee is a specification for a suite of high-level communication protocols using small, low-power digital radios based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs). Zigbee is targeted at RF applications that require a low data rate, long battery life, and secure networking. Zigbee and IEEE 802.15.4 based networks consist of many devices working together supporting sensing and control applications. Ubiqua PA has support for Zigbee 2007 stack profile 1 and 2, application Zigbee profile HA, SE, HC, CBA and TP2:

- Zigbee Stack Profile 1 (also known as Zigbee).
- Zigbee Stack Profile 2 (also known as Zigbee Pro).
- Zigbee Device Profile: Zigbee Device Object Profile, Profile ID: `0x0000`.
- Zigbee Application Profile HA: Zigbee Home Automation Application Profile, Profile ID: `0x0104`
- Zigbee Application Profile HA: Zigbee Home Automation Application Profile, Profile ID: `0x0109`
- Zigbee Application Profile HC: Zigbee Health Care Profile, Profile ID: `0x0108`.
- Zigbee Application Profile CBA: Zigbee Commercial Building Automation, Profile ID: `0x0105`.
- Zigbee Application Profile TP2: Zigbee Profile Test Profile #2, Profile ID: 0x7F01.

For more information visit the Zigbee Alliance web site for the latest specifications and other information: http://www.zigbee.org

Thread

The Thread stack is an open standard for reliable, cost-effective, low-power, wireless D2D (device-to-device) communication. It is designed specifically for Connected Home applications where IP-based networking is desired and a variety of application layers can be used on the stack. General characteristics of the Thread stack and network:

- Simple network installation, start up and operation.

- Secure.
- Small and large networks.
- Better range.
- No single point of failure.
- Low power.

For more information visit http://threadgroup.org/.

PopNet™

PopNet is a full mesh networking protocol and operating environment designed to work with inexpensive IEEE 802.15.4 radios and microcontrollers for low-power sensor and control applications.

For more information visit the San Juan Software web site for the latest specifications and other information: http://www.sanjuansw.com/

Zigbee RF4CE

Zigbee RF4CE is a radio protocol standard for remote control of consumer electronics devices based on 2.4GHz PHY/MAC IEEE 802.15.4 radios. Some characteristics:

- It supports multiple STAR topology with inter-PAN communication.
- Has a simple security (uses AES 128 Engine) set-up configuration for all devices.
- Provides simple pairing mechanism between devices.
- Provides power saving mechanisms for all device classes.
- Support profile for Consumer Electronics products.

IETF 6LowPAN

6LowPAN is not a protocol, is the name of the working group in the Internet area of the IETF. The 6LowPAN group has defined encapsulation and header compression mechanisms to allow IPv6 packets to be sent and received over the IEEE 802.15.4 based networks. The base specification developed by the 6Low PAN IETF group is RFC 4944, the problem statement document is RFC 4919. Several functionalities have been obtained from this work:

- Adapting the packet sizes of the two networks IPv6 and IEEE 802.15.4.
- Address resolution.
- Differing devices designs.
- Differing focus on parameter optimization.
- Adaptation layer for interoperability and packet format.
- Addressing management mechanisms.
- Routing considerations and protocol for mesh topologies in 6LowPANs
- Device and service discovery.

Zigbee Light Link

Zigbee Light Link gives the lighting industry a global standard for interoperable and very easy-to-use consumer lighting and control products. It allows consumers to gain wireless control over all their LED fixtures, light bulbs, timers, remotes and switches. Products using this standard will let consumers change lighting remotely to reflect ambiance, task or season, all while managing energy use and making their homes greener. Some features:

- Long battery life mesh technology.
- AES 128 encryption used to protect lighting network against unauthorized use.
- Interoperable with other Zigbee profiles.

- Control lighting products via remote, sensors or smart phones, tablets and computers.
- Simple Touchlink mechanism to add components and manage your lighting network.
- Ensures power-efficient control solutions and low maintenance cost.

For more information visit the Zigbee Alliance web site for the latest specifications and other information: http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbee-light-link

Zigbee Green Power

Zigbee Green Power enables new capabilities available to the Zigbee and Zigbee PRO networks. When the Zigbee Green Power standard is made available to Alliance members at the end of 2009, only Zigbee will offer an established, competitive marketplace for deploying switches, sensors and controllers using harvested energy in residential, commercial and industrial environments. Its energy harvesting capabilities will give manufacturers greater flexibility when designing innovative Zigbee products and solutions. Because Zigbee Green Power will work seamlessly with Zigbee and Zigbee PRO networks, it will enjoy all of Zigbee's numerous strengths.

For more information visit the Zigbee Alliance web site for the latest specifications and other information: http://www.zigbee.org

JenNet-IP

JenNet-IP is an IPv6 based low power wireless networking solution enabling the "Internet of Things". It provides a wireless command and control solution optimised for building automation with the internet connectivity, fast response and ability to control groups and set scenes demanded by these applications.

JenNet IP uses the IETF 6LoWPAN standard with a "mesh-under" networking approach provided by NXP's industry proven JenNet network layer. It provides a self-healing, highly robust and scalable tree network solution, for networks of up to 500 nodes.

JenNet IP features an easy to use and powerful Management Information Base (MIB) API, "JIP" that provides a powerful application layer for interoperable device management and control, enabling developers to develop products to suit all applications. Some features are:

- Wireless IPv6 networking enabling the "Internet of Things".
- Gateway or non-gateway options.
- Optimised for lighting and building automation.
- Self-healing and re-shaping tree network.
- JIP" API based on SNMP.
- “Mesh-under".
- Highly secure.
- Over-Network Upgrade future proofs.
- Low memory footprint.
- Low cost of ownership.

For more information, the latest specifications and other information see http://www.nxp.com/products/wireless-connectivity/2.4-ghz-wireless-solutions/jennet-ip:JENNET-IP

Chapter 19: Custom Payloads

Starting Ubiqua version 1.4, users are now able to use custom decoders for the Mac payload and Application Zigbee layers. The custom decoding is described by the user in a XML file.

To integrate the custom payload and start enjoying this functionality follow these instructions:

1. Create the custom payload XML file. The XML file has to be named with a specific name as it is required for the proper functionality with Ubiqua; the file depending of the case may be named as APS-Custom.xml or MAC-Custom-Payload.xml.

2. With Ubiqua closed, add the APS-Custom.xml file and/or MAC-Custom-Payload.xml file in the Ubiqua Data Folder. Depending of your operating system the Ubiqua Data Folder can be located at,

   ○ Windows XP: `C:\Documents and Settings\All Users\Application Data\Ubilogix\Ubiqua\`

   ○ Windows Vista or later: `C:\ProgramData\Ubilogix\Ubiqua`.

3. Run Ubiqua and enable the custom decoding, you can access this option through the Tools option of the main menu, then choose Options, and the General tab will open, at the bottom of this window you can find both options, custom decoding of APS or MAC payloads.

Creation of the custom decoder

The custom decoders are constructed by following the XML schema `Decoders.xsd`. Next section briefly explains the different elements and attributes used by the XML custom decoder.

The "Field" element

The "Field" element, reads an amount of bits from the frame. The Field element includes important xml attributes used for identifying and assigning meta information to the field. Some of the important attributes of the Field element are the following:

- **Name** – It's an optional attribute that assigns an ID to the value that was read into the Field element.

- **IsGroup** – Hides a Field element from the Packet View.

- **Label** – This attribute assigns a label to display into the PacketView, if the Label is not present, the default Label is “reserved".

- **Offset** – It's an Attribute with an integer value that defines the amount of offset bits from where the reading begins.

- **Type** – Defines the data type that will be interpreted on the PacketView.

An example of Field:

```
<Field Name="CommandType" Label="Command Type" Offset="8" Length="16"
Type="Integer" />
```

In the example above, the field Contains an ID "CommandType" with the value that was read, has an offset of 8 bits, a length of 16 bits and the data type is integer. The label displayed into the PacketView is "Command Type”.

The "Binding" element

The "binding" element creates a link with a variable. Depending of the value of the element, the length of the field element will be determined.

Note: A variable is defined with the "name" attribute in the Field element.

Example:

```
<Field Label="Command Type" Name="CommandType" Length="8">
      <Options>
        <Option Value="0x00" Label="Start" />
        <Option Value="0x01" Label="Stop" />
        <Option Value="0x02" Label="Pause" />
        <Option Value="0x03" Label="Configure"/>
      </Options>
</Field>
 <!-- If the "CommandType" is 0x00 then the length of  "Command Payload"
field is 8 bits (0x01 -> 16,  0x02-> 24, 0x03 -> 32, any other value is 0
bits)  -->
```

```
<Field Label="Command Payload">
  <Field.Length>
    <Binding Source="CommandType">
      <Case Value="0x00" Result="8" />
      <Case Value="0x01" Result="16" />
      <Case Value="0x02" Result="24" />
      <Case Value="0x03" Result="32" />
      <Case Value="0x04-0xFF" Result="0" />
    </Binding>
  </Field.Length>
  <!--Payload-->
</Field>
```

Custom decoder Field data Types

Each Field element has a Type attribute which is used to indicate how to parse and display the field value. The possible data types of the Type attribute are the following:

- **Hexadecimal** – Value should be parsed as an unsigned integer and displayed as an hexadecimal string (default).
- **Bits** – Value should be parsed as an unsigned integer and displayed as an element of a bitmap.
- **Integer** – Value should be parsed and displayed as an unsigned integer.
- **SignedInteger** – Value should be parsed and displayed as a signed integer.
- **String** – Value should be parsed and displayed as an ASCII string.
- **Boolean** – Value should be parsed and displayed as a boolean.
- **DateTime** – Value should be parsed and displayed as a date time. The format must be "yyyy-MM-dd HH:mm:ss.SSSSSS", as defined by the LDML standard.
- **Time** - Value should be parsed and displayed as a time. The format must be "HH:mm:ss.SSSSSS", as defined by the LDML standard.
- **Seconds** – Value should be parsed and displayed as seconds. The format must be "ss.SSSSSS", as defined by the LDML standard.
- **SignalPower** – Value should be parsed as an integer and displayed with the "dBm" postfix.

- **Address** – Value should be parsed as an unsigned integer and displayed as an hexadecimal string with digits separated by a colon only if the length is 6 or 8 bytes.
- **Key** – Value should be parsed as an hexadecimal string and displayed as a security key.

Loop functionality

The "isRepeteable" attribute of the Field element allows creating a Field with a loop functionality. If the "isRepeteable" is sets to true, the field repeats the content until the length is equals to 0.

Example:

```
<Field Label="Commands" Length="80">
      <Field Label="Command" Length="8" IsRepeatable="true">
        <Field Label="Field 1" Length="2" />
        <Field Label="Field 2" Length="6" />
      </Field>
 </Field>
```

In the above example, the field with the label "Command" will be repeated 10 times because its length is 8 bytes and the length of the Commands field is 80 bytes.

Hiding unnecessary fields in the Ubiqua Packet View.

If there is a "Field" element that is not required to be visible in the Packet View, add the "isGroup" attribute.

Example:

```
<Field Label="Values" IsGroup="true">
 <Field Label="Value" Length="8" />
</Field>
```

In the above example only the Field with the label "Value" is shown in the Packet View.

Examples

Custom decoder complete examples

Custom MAC payload example

The MAC Custom Payload only works using the IEEE 802.15.4-2003 and IEEE 802.15.4-2006 protocol stacks. MAC-Custom-Payload.xml contains the MAC header variables needed to decode the custom payload.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    File: MAC-Custom.xml
    Abstract: The layer definition of MAC Custom Payload

    Disclaimer: IMPORTANT: This Ubilogix software is supplied to you by Ubilogix
    International Inc. ("Ubilogix") in consideration of your agreement to the
    following terms, and your use or modification of this Ubilogix software
    constitutes acceptance of these terms. If you do not agree to these terms,
    please do not use or modify this Ubilogix software.

    In consideration of your agreement to abide by the following terms, and subject
    to these terms, Ubilogix grants you a personal, non-exclusive license, under
    Ubilogix's copyrights in this original Ubilogix software ("the Ubilogix
    Software"), to use and modify the Ubilogix Software provided that you must
    retain this notice and the following text and disclaimers. Except as expressly
    stated in this notice, no other rights or licenses, express or implied, are
    granted by Ubilogix herein, including but not limited to any patent rights that
    my be infringed by your derivative works or by other works in which the Ubilogix
    Software may be incorporated.

    THE UBILOGIX SOFTWARE IS PROVIDED BY UBILOGIX ON AS "AS IS" BASIS. UBILOGIX
    MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE
    IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A
    PARTICULAR PURPOSE, REGARDING THE UBILOGIX SOFTWARE OR ITS USE AND OPERATION
    ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

    IN NO EVENT SHALL UBILOGIX BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR
    CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
    GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
    ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION
    OF THE UBILOGIX SOFTWARE, HOWEVER CAUSED AND WHETHER UNDER THEORY OF CONTRACT,
    TORT (INCLUDING NEGLICENCE), STRICT LIABILITY OR OTHERWISE, EVEN IF UBILOGIX HAS
    BEEN ADVICED OF THE POSSIBLITY OF SUCH DAMAGE.

    Copyright (C) 2015 Ubilogix International, Inc. All rights reserved.
-->
<Layer xmlns="urn:ubilogix:decoders"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:ubilogix:decoders ../../Schemas/Decoders.xsd"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    Name="MAC Custom Payload" ShortName="Custom" Language="en-US" Version="1.0.1">

  <Colors>
```

```
    <Color Name="NC001" Value="Black" />
    <Color Name="NC002" Value="Red" />
    <Color Name="RC001" Value="Pink" />
    <Color Name="RC002" Value="LightGreen" />
  </Colors>
  <Variables>
    <!--The Frame Type from MAC header-->
    <Variable Name="FrameType" Value="0xF" />
    <!--MAC security bit from MAC header -->
    <Variable Name="MacSecurityEnabled" Value="0xF" />
    <!--The destination PAN ID-->
    <Variable Name="DstPanId" Value="0xF" />
    <!--The destination Address-->
    <Variable Name="DstAddr" Value="0xF" />
    <!--The Source PAN ID-->
    <Variable Name="SrcPanId" Value="0xF" />
    <!--The Source Address-->
    <Variable Name="SrcAddr" Value="0xF" />
  </Variables>

  <Fields>
    <Field Name="DataPayload" Label="Data Payload">
      <Field Label="Payload" Length="Stretch">
        <!--Payload Here-->
      </Field>
    </Field>
  </Fields>
</Layer>
```

Custom APS decoder example

The APS Custom Layer allows decoding the payload for APS when the Profile ID is private or to decode a specific command from ZCL when the Cluster ID is private.

```
<?xml version="1.0" encoding="utf-8" ?>
<!--
    File: APS-Custom.xml
    Abstract: The layer definition of APS Custom Payload

    Disclaimer: IMPORTANT: This Ubilogix software is supplied to you by Ubilogix
    International Inc. ("Ubilogix") in consideration of your agreement to the
    following terms, and your use or modification of this Ubilogix software
    constitutes acceptance of these terms. If you do not agree to these terms,
    please do not use or modify this Ubilogix software.

    In consideration of your agreement to abide by the following terms, and subject
    to these terms, Ubilogix grants you a personal, non-exclusive license, under
    Ubilogix's copyrights in this original Ubilogix software ("the Ubilogix
    Software"), to use and modify the Ubilogix Software provided that you must
    retain this notice and the following text and disclaimers. Except as expressly
    stated in this notice, no other rights or licenses, express or implied, are
    granted by Ubilogix herein, including but not limited to any patent rights that
    my be infringed by your derivative works or by other works in which the Ubilogix
    Software may be incorporated.

    THE UBILOGIX SOFTWARE IS PROVIDED BY UBILOGIX ON AS "AS IS" BASIS. UBILOGIX
```

```
    MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE
    IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A
    PARTICULAR PURPOSE, REGARDING THE UBILOGIX SOFTWARE OR ITS USE AND OPERATION
    ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

    IN NO EVENT SHALL UBILOGIX BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR
    CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
    GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
    ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION
    OF THE UBILOGIX SOFTWARE, HOWEVER CAUSED AND WHETHER UNDER THEORY OF CONTRACT,
    TORT (INCLUDING NEGLICENCE), STRICT LIABILITY OR OTHERWISE, EVEN IF UBILOGIX HAS
    BEEN ADVICED OF THE POSSIBLITY OF SUCH DAMAGE.

    Copyright (C) 2015 Ubilogix International, Inc. All rights reserved.

-->
<Layer xmlns="urn:ubilogix:decoders"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:ubilogix:decoders ../../Schemas/Decoders.xsd"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    Name="APS Custom" ShortName="Custom" Language="en-US" Version="1.0.1">

  <Colors>
    <Color Name="NC001" Value="Black" />
    <Color Name="NC002" Value="Red" />
    <Color Name="RC001" Value="Pink" />
    <Color Name="RC002" Value="LightGreen" />
  </Colors>
  <Variables>
    <!--APS Header Variables-->
    <Variable Name="ApsZclClusterId" Value="0xFFFF" />
    <!--Cluster ID value from APS Header-->
    <Variable Name="ApsFrameType" Value="0xF" />
    <!--Frame Type value from APS Header-->
    <Variable Name="DeliveryMode" Value="0xF" />
    <!--Delivery Mode value from APS Header-->
    <Variable Name="AckFormat" Value="0xF" />
    <!--Acknowledgement Format value from APS Header-->
    <Variable Name="SecurityEnabled" Value="0xF" />
    <!--Security Enabled value from APS Header-->
    <Variable Name="DstEndpoint" Value="0xFF" />
    <!--Destination Endpoint value from APS Header-->
    <Variable Name="ProfileId" Value="0xFF" />
    <!--Profile ID value from APS Header-->
    <!--ZCL Header Variables-->
    <Variable Name="ZclGeneralCmdFrame" Value="0xFFF" />
    <!--General Command Frame value from ZCL Header-->
    <Variable Name="ZclDirection" Value="0xF" />
    <!--Direction value from ZCL Header-->
    <Variable Name="ZclFrameType" Value="0xF" />
    <!--Frame Type value from ZCL Header-->
    <Variable Name="ManufacturerSpecific" Value="0xF" />
    <!--Manufacturer Specific value from ZCL Header-->
    <Variable Name="ZclSpecificPrivatedCmd" Value="0xFF" />
    <!--Specific Privated Command value from ZCL Header-->
    <Variable Name="ZclTransSequenceNumber" Value="0xFF" />
    <!--Transaction Sequence Number value from ZCL Header-->
    <Variable Name="ZclSpecificCommand" Value="0xFF" />
  </Variables>
```

```
  <Fields>

    <!--For APS Custom Payload-->
    <Field Name="DataPayload" Label="DataPayload">
      <Field Label="APS Payload">
        <!--Payload Here-->
      </Field>
    </Field>

    <!--For ZCL Custom Specific Command-->
    <Field Name="SpecificCommandPayload" Label="SpecificCommandPayload">
      <Field Label="Specific Command Payload" Length="Stretch">
        <!--Payload  Here-->
      </Field>
    </Field>

    <!--For ZCL Custom Frame Type-->
    <Field Name="ZclCustomFrameType" Label="ZclCustomFrameType">
      <Field Label="Private Payload" Length="Stretch">
        <!--Payload  Here-->
      </Field>
    </Field>

  </Fields>
</Layer>
```

Specific functionality examples

The following examples shows some of the functionality used for defining your own custom decoder.

Example 1

In this example, depending of the value of the "Command Type" Field the payload will be decoded accordingly. The example uses a Field named "Payload Field", with this field the length of the payload is calculated.

```
  <Field Label="Custom Payload Enabled" Length="Stretch">
         <!--Reads a field with 8 bits of length , saves the value of the field in
"CommandType" and display the "Type" of command depending of the value-->
        <Field Name="CommandType" Label="Command Type" Length="8">
          <Options DefaultLabel="Unknown Command">
            <Option Value="0x00" Label="Type 1"/>
            <Option Value="0x01" Label="Type 2"/>
            <Option Value="0x02" Label="Type 3"/>
            <Option Value="0x03" Label="Type 4"/>
          </Options>
        </Field>
        <!--Reads a field with 16 bits of length, saves the value in "PayloadLength"--
>
        <Field Name="PayloadLength" Label="Payload Length" Length="16" Type="Integer">
          <Options DefaultLabel="Bits" />
```

```
        </Field>
        <!--Depending the value of "CommandType" appears the Command Type N-->
        <Field Label="Payload" GroupsSource="CommandType">
          <DefaultFieldsGroup>
            <Field Label="Unknown Command Payload" Length="Stretch" />
          </DefaultFieldsGroup>
          <!-- If the Command Type is 0x00-->
          <FieldsGroup Group="0x00" Label="Payload Command Type 1">
            <Field Label="Payload Command Type 1" NodeColor="NC001">
              <Field.Length>
                   <!-- The length of this field is based on the value of the field
PayloadLength -->
                <Binding Source="PayloadLength" DefaultResultMultiplier="1" />
              </Field.Length>
              <Field Label="Payload 1" Length="Auto">
                <Field Label="Number of Company" Length="8" Type="Integer" />
                <Field Label="Name of Company" Length="80" Type="String" />
                <Field Name="Status" Label="Status" Length="8">
                  <Options>
                    <Option Value="0x00-0xF0" Label="OK" />
                    <Option Value="0xF1-0xFF" Label="Bad"/>
                  </Options>
                </Field>

                <!-- Field "Condition" appear if Status is 0x00 to 0xF0-->
                <Field Label="Condition">
                  <Field.Length>
                    <Binding Source="Status">
                      <Case Value="0x00-0xF0" Result="Auto"/>
                    </Binding>
                  </Field.Length>
                  <Field Label="Payload" Length="Stretch" />
                </Field>

              </Field>
            </Field>
          </FieldsGroup>

          <!-- If the Command Type is 0x01-->
          <FieldsGroup Group="0x01" Label="Payload Command Type 2">
            <Field Label="Payload Command Type 2" NodeColor="NC001">
              <Field.Length>
                   <!-- The length of this field is based on the value of the field
PayloadLength -->
                <Binding Source="PayloadLength" DefaultResultMultiplier="1" />
              </Field.Length>
              <Field Label="Payload 2" Length="Stretch" />
            </Field>
          </FieldsGroup>

          <!-- If the Command Type is 0x02-->
          <FieldsGroup Group="0x02" Label="Payload Command Type 3">
            <Field Label="Payload Command Type 1" NodeColor="NC002">
              <Field.Length>
                   <!-- The length of this field is based on the value of the field
PayloadLength -->
                <Binding Source="PayloadLength" DefaultResultMultiplier="1" />
              </Field.Length>
              <Field Label="Payload 2" Length="Stretch" />
```

```
          </Field>
        </FieldsGroup>
       <!-- If the Command Type is 0x03-->
        <FieldsGroup Group="0x03" Label="Payload Command Type 4">
          <Field Label="Payload Command Type 1" NodeColor="NC002">
            <Field.Length>
                 <!-- The length of this field is based on the value of the field
PayloadLength -->
              <Binding Source="PayloadLength" DefaultResultMultiplier="1" />
            </Field.Length>
            <Field Label="Payload 2" Length="Stretch" />
          </Field>
        </FieldsGroup>
      </Field>
    </Field>
```

Example 2

This example illustrates the use of the ZCL Header.

```
<Field Label="Payload Example 2" Length="Stretch">

       <!--To include the ZCL Header decoding, is needed add the next line, if the
APS Payload custom do not include decoding for ZCL Header just omits the next line-->
      <Field Label="ZCL Header">
        <Break Target="Layer" LayerName="Zigbee ZCL" FieldName="DataPayload" />
      </Field>

      <Field Label="Command Type" IsGroup="true" GroupsSource="ZclFrameType">
        <FieldsGroup Group="0x1" Label="Specific Command">
                                        <Field   Label="Specific   Command   Payload"
GroupsSource="ZclSpecificPrivatedCmd">
          <FieldsGroup Group="0x00" Label="Command Type 0">
            <Field Label="Command Type 0" Length="Stretch" />
          </FieldsGroup>
          <FieldsGroup Group="0x01" Label="Command Type 1">
            <Field Label="Command Type 1" Length="Stretch" />
          </FieldsGroup>
          <!--More FieldsGroup with different "Group" value-->
        </Field>
        </FieldsGroup>
      </Field>
    </Field>
```

Chapter 20: Troubleshooting

Cannot Start a Device

Please check the port is not being used by another program in your system. If this does not solve the issue, unplug and then plug the device back in again.

Getting Further Help

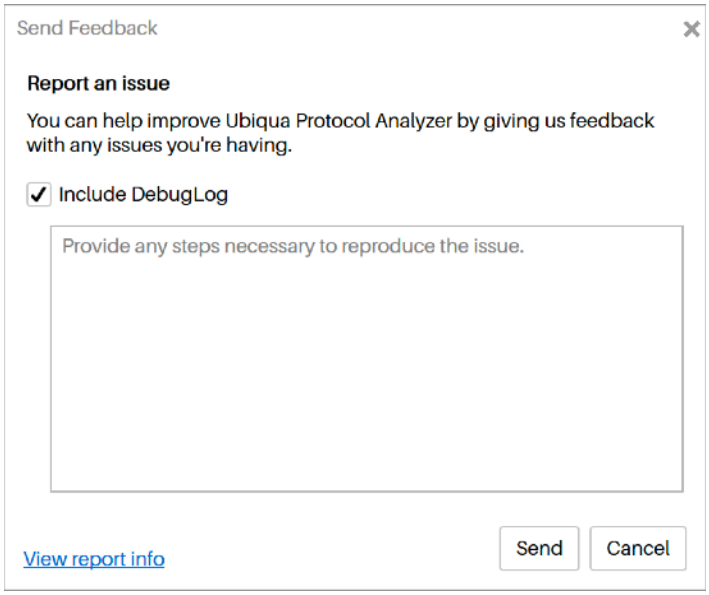
If you need further help, please open a support ticket via your Dashboard and we will be happy to assist you.

To help us to diagnose your issue, please include the `DebugLog.txt` file located in `C:\Users\USERNAME\AppData\Roaming\Ubilogix\Ubiqua`. You can reach this folder quickly by typing `%AppData%` in your File Explorer address bar.

Send Feedback

You can help improve Ubiqua Protocol Analyzer by giving us feedback with any issues you're having.

To report an issue through Ubiqua click the Help > Send Feedback menu item, following this action, a dialog window will appear on your screen with a text box where you will be



able to fully describe the issue you're having, we recommend you to write step by step the procedure you are following to replicate it. This will help us with the task to track it down and solve it.

At the bottom left of the window there is a link with the legend View report info, if you click this link, the dialog window will expand to display a box with your DegugLog, which is a historic record of the events that occur in Ubiqua, such as crash reports and thrown exceptions, to add this information to your report, click on the Include DebugLog checkbox.

Finally to send us your report click on the Send button located at the bottom right of the dialog window, otherwise click on the Cancel button.