Ember Desktop User Guide

Version 3.3 Build 1985

Table of Contents

Table of Contents

Table of Contents

Table of Contents

1 About the Ember Debug Environment™

Document Number: 120-4036-000

The Ember integrated network debug environment, ties together firmware and hardware components of Ember breakout boards with the Simplicity Studio software.

- **Ember breakout boards** are testbeds for developing and deploying a ZigBee application. The breakout board provides a direct attachment to the radio communication module (RCM). It also may contain components for prototyping and testing an embedded wireless application (such as a temperature sensor), two buttons, a piezo buzzer, two LEDs, and a 2" x 2" through-hole prototyping area.
- **The Ember radio communication module** (RCM) offers a complete ZigBee wireless solution for development and deployment of a low-data-rate, low-power ZigBee application. The four-layer (FR4) RCM features one of the family of Ember radio chips.
- **Debug Adapter** connects the RCM to the Ethernet. It collects network data over the Packet Trace Port cable connection and conveys it over its Ethernet connection to Simplicity Studio. It also delivers messages and new software that are addressed to the RCM or its breakout board.
- **Simplicity Studio's Network Analyzer** is a graphical tool that displays network and node activity in realtime. It also provides an interface for the loading of new applications on the RCM through the Debug Adapter.

For detailed information about the Ember breakout boards, Debug Adapter, and Packet Trace Port, refer to their associated technical specifications.

2 Registering Simplicity Studio

Each installation of Simplicity Studio should be registered with Silicon Labs prior to use. While you may use Simplicity Studio without registering it with Silicon Labs, you will be prompted at startup until registration is completed. In addition, you will not be able to update Simplicity Studio via the customer support portal without first registering your installation of Simplicity Studio.

To register your version of Simplicity Studio, you must first have an account with the Customer Support Portal. The Customer Support Portal provides Silicon Labs' customers with access to the latest releases of Ember software and other information related to Silicon Lab's products. To get an account for the Customer Support Portal, visit this link:

http://www.silabs.com/zigbee-support

Once you have an account for the Customer Support Portal, you may register your version of Simplicity Studio at any time by choosing **Help | Register...**

Enter your username in the customer support registration dialog and click save. You can now use your registered username to access the Support Portal and to download the latest updates to the Simplicity Studio software.

3 Simplicity Studio Network Analyzer Features

The Network Analyzer provides a rich and flexible interface to Silicon Labs embedded networks that helps you develop and debug network applications. Simplicity Studio includes these features:

- Multiple editor panes that provide tiered displays of network activity, letting you drill down from a high-level map of node interactions to the details of each packet.
- Customizable filters that let you specify exactly which network activities to display.
- The ability to view the embedded code's function-call stack traces and observe core dumps and other low-level information.
- Log files that save captured data, so you can share and analyze transactions and events later.
- A file browser that lets you easily upload new applications to any connected node.
- An interface for automatic discovery of Ethernet-connected adapters and easy management of adapter applications.

3.1 Additional documentation

To obtain additional Silicon Labs documents mentioned in this guide, see http://www.silabs.com/support/pages/document-library.aspx.

If you are a current Silicon Labs customer, you may obtain a support site login here: http://www.silabs.com/zigbee-support which may be used to login to our Support Portal here: http://www.silabs.com/zigbee-support

The customer support portal provides downloads for all Silicon Labs documentation and software as well as Blogs, FAQs etc...

4 Network Analyzer Legal Information

4.1 Third Party Software

Silicon Laboratories' Network Analyzer includes several third party open source libraries which are covered by the following license agreements: Apache License v2.0, Eclipse Public License, Lesser Gnu Public License and the Berklee Software Distribution License. Silicon Laboratories Network Analyzer links to these libraries and in accordance with their license agreements includes them in an unmodified form. In some cases where covered by the Eclipse Public License, some libraries are included in a modified form. Source code to modified libraries is available upon request.

In accordance with the above mentioned license agreements, the providers of, and contributors to libraries included in Silicon Laboratories Network Analyzer are not responsible for any warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose. Furthermore, the providers and contributors of the libraries listed below are not responsible for liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits incurred through the use of this software or any of its components.

Also in accordance with the above mentioned license agreements, any user who wishes to receive a copy of any of these libraries may contact Silicon Laboratories http://www.silabs.com or download any of the libraries from one of the links provided below. In accordance with the Eclipse Public License agreement (EPL) the source code to any libraries included in a modified form will also be made available upon request.

Software libraries to which Network Analyzer links include the following:

- **The Eclipse SDK** which is covered by the Eclipse Public License and is available for download here: http://www.eclipse.org/downloads/

- **The Eclipse Graphical Editor Framework (GEF)** which is covered by the Eclipse Public License and is available for download here: http://download.eclipse.org/tools/gef/downloads/

- **BeanShell** which is covered by the Lesser GNU Public License and is available for download here: http://www.beanshell.org

- **Glazed Lists** which is covered by the Lesser GNU Public License and is available for download here: http://www.publicobject.com/glazedlists/

- **XStream** which is covered by the Berklee Software Distribution License and is available for download here: http://xstream.codehaus.org/download.html

- **Tomcat** which is covered by the Apache License v2.0 and is available for download here: http://tomcat.apache.org/

5 Navigating Simplicity Studio Network Analyzer

The Simplicity Studio Network Analyzer environment contains the following work areas at startup:

- **The menu bar** includes commands that let you manage file access and Simplicity Studio preferences, editing tasks and display settings, filter management for data display, window management for view organization, and help information.
- **The Toolbar** includes icons you can click to quickly access common tasks.
- **Adapters view** lists all adapters and their nodes that are accessible to Simplicity Studio. See Interacting with Debug Adapters.
- **Editor panes** display the data of transactions and event capture sessions from one or more nodes. See Capturing Data and Managing a Session and Viewing Captured Data.

6 Simplicity Studio Network Analyzer Toolbar

The Network Analyzer provides shortcuts to often used features. These are contained in a row of iconed buttons at the top of the Simplicity Studio window just below the MenuBar. The Tool Bar buttons are either enabled or disabled depending on what the user is doing.

The following buttons are available in Simplicity Studio's Network Analyzer toolbar:

Open File

Opens a file dialog from which the user may choose a capture file (.isd), a raw log file (.log) or an application configuration file (.isc). The toolbar icon is functionally the same as using File > Open File ...

For more information on opening a file check out the open and close options help section located here: Open and close options

Save (Ctrl+S)

Saves any edits to the currently opened file to disk. If the file has never been saved before, the user has the opportunity to say where they want it saved and what extension they want to give it. If the file was previously saved, Simplicity Studio simply saves over the old file.

Save All (Ctrl+Shift+S)

Saves all files which have been edited. As with Save, the user has the option to indicate where and how to save any files which have never been saved before.

Save As

Allows user to save the currently open file with a new name, location and format.

Refresh

Closes and reopens the editor. This is useful in the case where you have changed or added new decryption keys to the preferences. It is useful to be able to reopen the file and run it through the decryptors and decoders again.

Clear Events

Clears the editor of all events. This is useful in the case where you are capturing on a network but are waiting for some set of events to occur but don't want to keep everything else around.

Pause Stream

Pauses a capture without stopping it. This is useful in the case where a user wants to stop capturing for period of time but wants the capture to continue as if nothing happened in the future. Live events are simply dumped on the floor during a pause and are not retrievable.

Live Capture Options

Opens the Capture Options dialog. This dialog is used to tailor the condition under which events will be captured. For more information check out Capturing with options.

View and Modify Security Keys

Opens the Active live capture dialog. This dialog allows the user to add additional security keys to be used during the current capture in progress. Any keys which are added will also be added to the list of security keys in the user preferences.

Show Short Id

Displays a nodes shortId on the map, if one is known for the currently selected time.

Show Long Id

Displays the long Id (64 bit identifier) on the map for a given node if one is known.

Show Pan Id

Displays the pan Id on the map, if one is know for the currently selected time.

Show Node Label

Displays the node's label. If the node is connected to Simplicity Studio over the backchannel, this value will be the host name of the adapter connected to the node.

Show LQI

Displays the Link Quality Indicator (LQI) value in the map as reported by a node for a given time.

Show All Connectivity

Shows the quality of connections between nodes on the map.

Show all Simultaneous Events on Map

During periods of heavy traffic, several simultaneous transactions may overlap. When this button is enabled, the map will show all traffic. When it is disabled (default) it will show only the traffic related to the currently selected transaction.

Show All Transactions on Map

When this button is enabled, all transactions which overlap with the currently selected one are shown on the map.

Map Zoom In

Increases the size of the map in the map pane. This can be useful if you are looking at a very large network where a large number of nodes are positioned very close together.

Map Zoom Out

Decreases the size of the map in the map pane.

Edit Trace File Description

Each saved Simplicity Studio trace file contains a description which can be used to store information which is important to the trace but may not be included by default. Generally the description is used to provide context for the trace and any other information that may be of help to the viewer.

Go To Line

Moves the event cursor directly to the event number entered. This button is only enabled if the Stream preference "Show event numbers" is selected. To turn this feature on go to: Edit > Preferences > Stream > Show event numbers. For more information see Stream preferences.

Go To Bookmark

Moves the event cursor to the bookmark selected in the bookmark dialog.

Decrease Font Size

Decreases the size of the font used in the Event, Transaction and Detail panes.

Increase Font Size

Increases the size of the font used in the Event, Transaction and Detail panes.

Apply Row Coloring

Turns coloring on and off in the Transaction and Event panes. Row colors are applied based on the pre-defined filters included in the Filter Manager.

Lock To Bottom

Locks the event cursor to the bottom of the Event pane. During a live capture, this will cause the Map and Details panes to always show the latest event.

Start Replay

Begins scrolling forward through events from the current event selected. If no event is selected, the scrolling will begin from the start of the current trace file.

Toggle Event Difference View

Toggles the Event Difference View. Click here for more information on the Event Difference View.

Toggle Timeline Bar

Toggles the timeline shown at the top of the currently opened Stream Editor. Click here for more information on the Timeline.

Toggle Filter Bar

Toggles the Quick Filter Bar at the top of the currently opened Stream Editor. Click here for more information on the Quick Filter Bar.

Show Filter Manager View

Toggles the Filter Manager View. Click here for more information on the Filter Manager View.

CH:11 Sniffer Settings

Opens the Sniffer Settings Dialog. Click here for more information on the Sniffer Settings Dialog.

Discover Adapters

Discover's adapters located on the backchannel as well as other devices connected to Simplicity Studio. For more information on discovery, check out the Adapters View.

Discovery Preferences

Opens the Discovery Preferences dialog. This dialog can be used to control how Simplicity Studio discovers adapters over the backchannel. Click here for more information on Discovery Preferences.

7 Updating Simplicity Studio

Simplicity Studio can update itself from the Silicon Labs update site. You can run the Simplicity Studio update mechanism manually or set it to run each time Simplicity Studio starts up.

For more information, see:

- Updating to the latest release
- Managing your update configuration
- Setting auto-update preferences

8 Manually Updating to the Latest Version

You can manually update your Simplicity Studio software from within Simplicity Studio itself. This approach spares you from having to download a large installer and run through a complete, fresh install to get the latest features or bug fixes.

8.1 Register First

Before you can update your version of Simplicity Studio, you must first register with the Silicon Labs Support Portal and receive a username for logging into the support and update sites. For more information, see Registration.

8.1.1 Login to the Support Portal

To run the software update manually, choose **Help | Update now...**. Simplicity Studio will use your registered username to login to the support portal. Once Simplicity Studio has logged in, it will download the latest updates of the Simplicity Studio software.

If you are currently using the latest version Simplicity Studio, you will see a dialog indicating that there are no updates available for your version of Simplicity Studio.

If updates are available to your installation, a dialog will pop up to guide you through the update process.

8.1.2 Complete the Update Dialog

In the update dialog, you will see a data tree indicating which features of Simplicity Studio are available for update. To proceed with the update, click **Next**.

In order to update, you must accept the end-user license agreement. Click the radio button marked "I accept the terms in the license agreements" and then click **Next**.

The install dialog will now show you which new features will be installed for you. Click **Finish** to proceed with the installation.

The Update Manager will now download the selected features.

Once the selected features are downloaded, the feature verification dialog will warn you that the features are unsigned. This is expected. Select the **Install** button to proceed.

8.1.3 Restart Simplicity Studio

Once the install is complete, you will need to restart Simplicity Studio for the changes to take effect. When Simplicity Studio restarts, you will be running the latest version of the Simplicity Studio software.

9 Managing Simplicity Studio Installation

Warning: This functionality should be used by the expert user or customer support personnel only. Standard users do not need it.

Simplicity Studio provides an interface for managing the software updates provided by Silicon Labs. To access the Update Configuration Manager, choose **Help | Manage configuration**.

From the Manage Configuration dialog, you can Disable, Add Extensions or Show the Properties associated with your Simplicity Studio installation.

10 Automatic Updates for Simplicity Studio

Simplicity Studio can check for updates to its own software each time that it starts up. For this process to be successful, your computer must be connected to the internet.

To turn on the automatic-update feature, choose **File | Preferences | Install/Update | Automatic Updates**.

Select the check box "Automatically find new updates and notify me." When you click **Apply**, Simplicity Studio will check for updates on Ember's update server. Simplicity Studio will also check for updates each time you start it. If any new updates are available, Simplicity Studio will take you through the update process described in Manually Updating to Latest Version.

11 Interacting with Debug Adapters

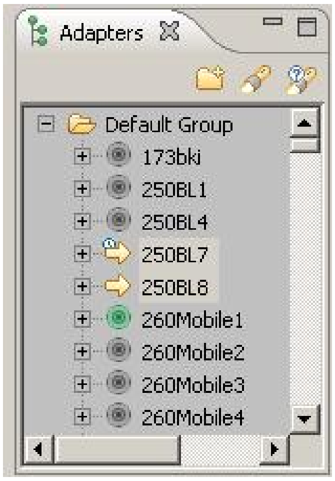
The following sections discuss the adapter actions that you can initiate and control through Simplicity Studio. These actions are accessible through the pop-up menu that you invoke by right-clicking selected Debug Adapters or an adapter group. Adapter features include:

- Adapters view
- Connecting to adapters
- Uploading an application
- Making an adapter a sniffer
- Commandline interaction with an adapter
- Node/adapter management

Capturing network data from an adapter's node is discussed in Capturing Data and Managing a Session.

11.1 Adapters view

On startup, Simplicity Studio automatically discovers and lists all Debug Adapters that are connected to the local subnet. You can also configure Simplicity Studio to discover Debug Adapters on other subnets. To force Simplicity Studio to discover adapters, click the Discovery icon.



11.1.1 Default and custom groups

When you first start Simplicity Studio, it displays in a single default group all Debug Adapters that are discovered by Simplicity Studio. You can create custom groups and move Debug Adapters to them.

Note: Custom groups simply provide a way to organize adapters and have no functional impact on adapters or their interaction.

11.1.2 Adapter details

You can expand each Debug Adapter to display the following details about it:

- **Node type** identifies the Ember radio chip. (Note: The adapter requires that the node have been reset for it to know what type of node it is working with. In some cases, when the node has not been reset, the node type will say: *Not set*.)

- **Last bootload** identifies the last bootload to this adapter's node from your machine. If another user uploaded an application to the node, this information may not actually reflect the application that is currently loaded on the node itself. If Simplicity Studio is unable to identify the application for some other reason, this field displays *Unknown*.
- **Node EUI** identifies the Extended Unique Identifier.
- **Hostname** and **IP address** identify this Debug Adapter.
- **Connection status**, **Connection address**, and **Connection time** show the current connection information about this Debug Adapter.
- **Board type** and **Board name** identify this adapter's platform.
- **Hardware type** and **Hardware version** identify the adapter hardware.
- **Software version** shows which version of Debug Adapter is running on this adapter.

11.1.3 Using multiple adapters

Most commands can be issued to multiple adapters simultaneously. You can select multiple adapters in one of two ways:

- Select individual adapters:
    - ♦ For contiguous selections, Shift+Click
    - ♦ For non-contiguous selections, Ctrl+Click
- Select one or more adapter groups to select all adapters within those groups.

11.1.4 Adapter Properties

Adapter properties can be changed from the view interface by right clicking on the adapter and selecting properties.

11.1.4.1 Adapter name:

The name of the adapter shown in the Adapters View

11.1.4.2 Adapter type:

The type of adapter connected.

11.1.4.3 Adapter version:

The version of the Adapter firmware loaded on the adapter.

11.1.4.4 Radio processor:

The Ember radio processor to which the adapter is connected.

11.1.4.5 Radio processor application:

The firmware running on the radio process.

11.1.4.6 Host processor:

The host processor to which the adapter is connected (in the case of the radio processor being an NCP above.)

11.1.4.7 Host processor application:

The firmware running on the host processor.

11.1.4.8 Show raw discovery data:

Pops up a dialog showing what data was returned to the discovery command sent out from Simplicity Studio in order to discover this adapter on the network.

11.1.4.9 Protect from automatic discovery changes:

Every time the adapter discovery process is performed, all adapters in the adapter search parameters will be updated with the information they return to the discovery query. You may use this checkbox to keep an adapter entry from being updated with the information returned during discovery. This is useful in the case where the adapter discovery is not able to tell you enough information about the adapter's settings. For instance, if you are using an EM357 with EZSP and a host micro, there is currently no way for the adapter to know that you have your EM357 configured as an NCP or which type of Host micro it is attached to. You can configure this information in the adapter preferences dialog by entering the information above. You may also persist the information you enter by selecting this check box. When this box is checked, a "[p]" symbol will be shown next to the adapter name in the view to remind you that this adapter is being protected from discovery changes.

11.2 Connecting to adapters

To interact with any adapter, you must establish a connection between it and Simplicity Studio. When you connect to an adapter, Simplicity Studio opens the TCP/IP socket connections that it requires in order to communicate with Debug Adapter.

Note: Before you connect, configure your firewall to allow connections to ports 4900-5000. If you are using Windows Vista, you must allow Simplicity Studio access through Vista's firewall. To give Simplicity Studio access through Vista's firewall do the following:

1. From the **Start** button, choose **Control Panel**
2. In the Control Panel Home view, you will see a Windows Security icon with a link that says "Allow a program through Windows Firewall." Click on this link. When Vista asks if you want to run this program, click **Continue**.
3. In the **Exceptions** tab, click on the **Add Program** button.
4. Select **Simplicity Studio** from the list.
5. Click **OK**,
6. You will see Simplicity Studio added to the list of exceptions with a check mark next to it.
7. Click **OK**
8. Close the **Control Panel**

To connect an adapter:

Right-click the adapter from the Adapters view and select **Connect** from the pop-up menu.

To disconnect an adapter:

Right-click the adapter from the Adapters view and select **Disconnect** from the pop-up menu.

When you disconnect Simplicity Studio from an adapter, it closes the TCP/IP socket connections. If any capture is in progress from that adapter's node, it stops the capture first.

11.3 Uploading an application

You can upload an application to the selected adapter's node. If uploading to an System On a Chip (SOC) node, Simplicity Studio checks whether any application upload preferences are set.

To upload an application:

1. Ensure the adapter is connected as described in Connecting to adapters.
2. Right-click the connected adapter from the Adapter view and select **Upload Application** from the pop-up menu.
3. Select an application image from the dialog's list, or browse for a binary image. The Browse list maintains previously loaded application images. Standard applications supplied with Simplicity Studio include:
    - ♦ Sniffer. See "Making an adapter a sniffer".
    - ♦ When uploading an application you have the option of, also uploading a bootloader image to go along with the application. The application must be compiled specifically for the bootloader image that you are uploading. In order to upload a bootloader image, click on the check box entitled: Bootloader. Then either select the bootloader you wish to include from a drop down list or navigate to the bootloader image you wish to include. Both application and standalone bootloader images are provided for each stack in the stack install directory.

Caution! Be sure that the application you upload is compatible with the architecture of the target node.

If you upload to an Network Co-Processor (NCP) node, Simplicity Studio displays a dialog that asks you to choose between two possible upload targets:

- **Upload Host Processor** lets you upload an application image to the host microcontroller.
- **Upload Network Co-processor** lets you upload a new image to the EM260 network co-processor.

To reload an application:

1. Ensure the adapter is connected as described in Connecting to adapters, and then do one of the following.
2. Right-click the connected adapter from the Adapter view and select **Redo last upload** from the pop-up menu.
3. Use the **Ctrl-Y** shortcut on your keyboard to redo the last action taken on an Adapter.

11.4 Making an adapter a sniffer

Any adapter's node that connects to Simplicity Studio can be designated as a sniffer. A sniffer node is capable of capturing all data that is transmitted among nodes over the designated channel.

A sniffer node must have a sniffer application loaded on its RCM. If no sniffer application is currently loaded on the RCM, you may load one in either of the following ways:

1. Once connected to an Adapter, choose the Upload application action from the right click menu and choose **Sniffer** as the application you wish to upload.
2. Once connected to an adapter, choose the **Make Sniffer** action from the right click menu. This will cause Simplicity Studio to first check whether the node has a Sniffer image on it before attempting to upload the new Sniffer image.

After a node is loaded with the sniffer application, you can start using it to capture network data.

11.5 Commandline interaction with an adapter

Simplicity Studio provides a commandline console for interacting with an adapter and its associated application.

1. Right-click an adapter and select **Launch Console** from the pop-up menu.
   The Console Editor will pop-up to the right of the Adapters View. In the editor, you will see a tab for each serial connection Simplicity Studio has to the adapter. The typical Ember adapters provide the following console interfaces:
   - ♦ **ADMIN (port 4902)**
   - ♦ **SERIAL0 (port 4900)**
   - ♦ **SERIAL1 (port 4901)**
   - ♦ **DEBUG (port 4905)**
2. Click in the console window of the tab connected to the serial port you wish to interact with. Enter a command appropriate for the application console and press **Enter**.
   Valid commands depend on what application is loaded into the adapter. For example, a version command might display the application version.

```
version

version

v. 1.2b14 RCM3400, Jan 19 2006 11:22:51

ember
```

11.6 Node/adapter management

11.6.1 Reset node

If a node is in an inconsistent state, you can reset its RCM. A reset of the RCM pulls down its reset line for a millisecond.

1. Ensure the adapter is connected as described in Connecting to adapters.
2. Right-click the connected adapter and select **Reset node** from the pop-up menu.

11.6.2 Power Node On/Off

You can power on or off any connected node from Simplicity Studio. If you are capturing data from a node, power-off suspends the capture, and power-on resumes the capture.

1. Ensure the adapter is connected as described in Connecting to adapters.
2. To power-off a node, right-click it and select **Power off node** from the pop-up menu.
3. To power-on a node, right-click it and select **Power on node** from the pop-up menu.

11.6.3 Reset Debug Adapter

To reset a Debug Adapter, right-click the adapter and select **Reset adapter** from the pop-up menu.

11.6.4 Upload adapter firmware

Use this option to upgrade Debug Adapter firmware.

Caution! Upgrading incompatible firmware to a Debug Adapter can damage the adapter. Ensure that the adapter firmware you upload is compatible with the architecture of the target node.

1. Ensure the adapter is connected as described in Connecting to adapters.
2. Right-click the connected adapter and select **Upload adapter firmware** from the pop-up menu.

3. Select an image from the dialog's list, or browse for a binary image.

12 Using USB Link Devices

Simplicity Studio can discover and interact with USB Link devices that are connected to your computer. USB Link devices are shown in the USB Link View. To open the USBLink View, choose **Window | Show View | USB Link View**.

Simplicity Studio discovers USB Link devices connected to your computer automatically. If you do not see your USB Link device in the USB Link View, you can manually run the discovery process by clicking the **Discovery** icon in the top right corner of the USB Link View. Simplicity Studio will list each USB Link device discovered and the id and status for each device.

Loading software using the USB Link
To load a binary image onto your device using the USB Link, select the USB Link you wish to use and click the **Load** icon (). This will launch the Binary Image Chooser dialog. Choose or browse to the image you wish to load onto your device, and click **OK**. This will begin the upload process. The status of the upload process is shown beside the USB Link in the USB Link View.

Reading a binary image using the USB Link
To read a binary image off of a device connected to a USB Link, select the USB Link you wish to use for the read process and click the **Read** icon (). This will launch a File dialog that allows you to save the binary image as an ASCII .hex file.

Changing the ID of your USB Link
By default, every USB Link has the ID 0. If you want to use multiple USB Links, you should change their IDs to be unique. Click the **Change ID** icon () to launch a dialog that allows you to enter a new ID for the selected USB Link.

Identifying a USB Link
You may often wish to know which USB Link represented in Simplicity Studio is connected to which physical device. You can identify a USB Link by making it blink its LED. When you click the **Blink** icon (), the USB Link that is selected in the USB Link View will flash its LED amber for few seconds.

For more information on USB Link devices, see the USB Link documentation provided with your USB Link product.

13 Capturing Data and Managing a Session

Simplicity Studio can display one or more network capture sessions. Each capture session displays the transaction and event data captured from one or more nodes. The captured data is shown in editor panes.

Related topics for this section include:

- Capturing perfect trace sessions
- Capturing from a sniffer node
- Managing a session
- Saving a session
- Open and close options

Simplicity Studio supports two types of capture sessions:

- **Live sessions** display data as it is captured. The display is continuously refreshed as new data arrives. Simplicity Studio maintains one live session at a time. When it starts, the live session is unnamed, and its data is maintained in temporary storage until you save it to a named file.
- **Saved sessions** contain captured data that has been saved to permanent storage in a named `.isd` file. You can load a saved session and play it back at any time.

When you start Simplicity Studio, it opens a new live session and any saved sessions that were displayed when you last exited.

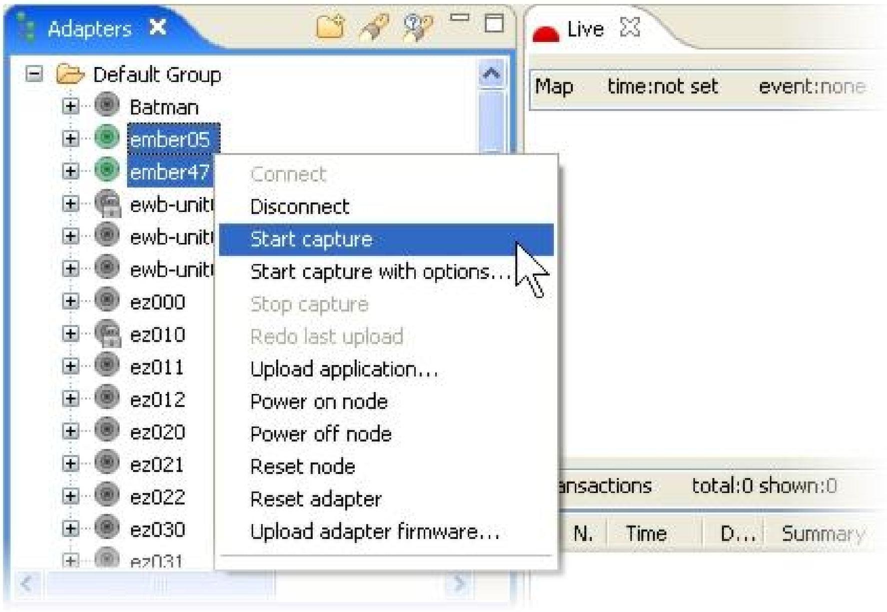
Simplicity Studio displays each session in an editor with its own tab. The tab of each saved session is labeled with the session's file name; the live session is labeled **Live**.

You can capture data from the node of any connected Debug Adapter, from one node at a time or multiple nodes simultaneously. You can also capture all network traffic over the current channel by capturing data from a connected sniffer node.

Note: The types of data that are captured from a node depend on the capabilities of the node's radio chip and Debug Adapter.

13.1 Starting a capture

1. Select one or more connected adapters.
2. Right-click the selected adapters.
3. From the pop-up menu, choose **Start Capture**.



13.1.1 Setting capture options

You can filter packets out of the stream during a capture. For example, you can choose to see only packets from a certain PAN ID (preferred area network identifier) and drop all other packets.

1. Select one or more connected adapters.
2. Right-click the selected adapters.
3. From the pop-up menu, choose **Start capture with options**.
4. In the **Capture options** dialog:
5. ♦ To filter for PANs, check the **Capture only PANs** checkbox and enter the hex value for the filter.
   ♦ (Optional) Check the **Remove non-packets** checkbox if desired.
   ♦ Choose a **Saved Defaults** setting.

When the Live Editor is active, you can modify capture options by clicking the *live capture options* icon.

13.1.2 Capturing from multiple nodes

You can capture data from multiple nodes simultaneously. This is typically done for nodes that belong to the same network.

1. Select the corresponding Debug Adapters from the Adapters view:
2. ♦ For contiguous adapters, Shift+Click
   ♦ For non-contiguous adapters, Ctrl+Click
3. Right-click the selected adapters.
4. From the pop-up menu, choose **Start Capture**.

Alternatively, you can start a capture from a group of Debug Adapters, provided all adapters in the group are connected: Right-click the group folder and, from the pop-up menu, choose **Start Capture**.

13.2 Capturing perfect trace sessions

Simplicity Studio captures all incoming and outgoing packet data via the selected adapters, regardless of whether the host nodes have sniffer applications. The captured data includes failed transmissions, as well as debug messages from node applications that are compiled in debug mode.

This kind of capture is called a *perfect trace session*. The capture nodes of a perfect trace session are not sniffers but nodes that might be running your own application that you are trying to debug. The perfect trace session compiles all incoming and outgoing data from each node in chronological realtime, providing a richly layered display of all activity within a network.

A perfect trace session can be especially useful for debugging a network in development as it allows you to see every single packet on the network.

13.3 Capturing from a sniffer node

A sniffer node has a sniffer application loaded on its RCM. The sniffer appication enables the node to capture over-the-air transmissions between nodes over the designated channel. When you start capturing from a sniffer node, the sniffer node captures all packets that are exchanged by the nodes on the designated channel.

A sniffer can capture only those packets transmitted within range of its radio. If you are working with a very large network, you may wish to use multiple sniffers positioned around the network to be sure that you will hear all transmissions within the network.

13.3.1 Changing the channel

You can change the sniffer capture channel by clicking the **Channel** button CH:11.

13.3.2 Checking sniffer status

To view a sniffer's status, right-click the sniffer node and select **Sniffer status**. Status items include the channel the sniffer is using, the PAN ID, and the power setting.

13.3.3 Discovering available networks

1. From the Simplicity Studio toolbar, click the **Channel/PAN** button CH:11.
2. Select a sniffer node from the drop-down list and click **Scan**.

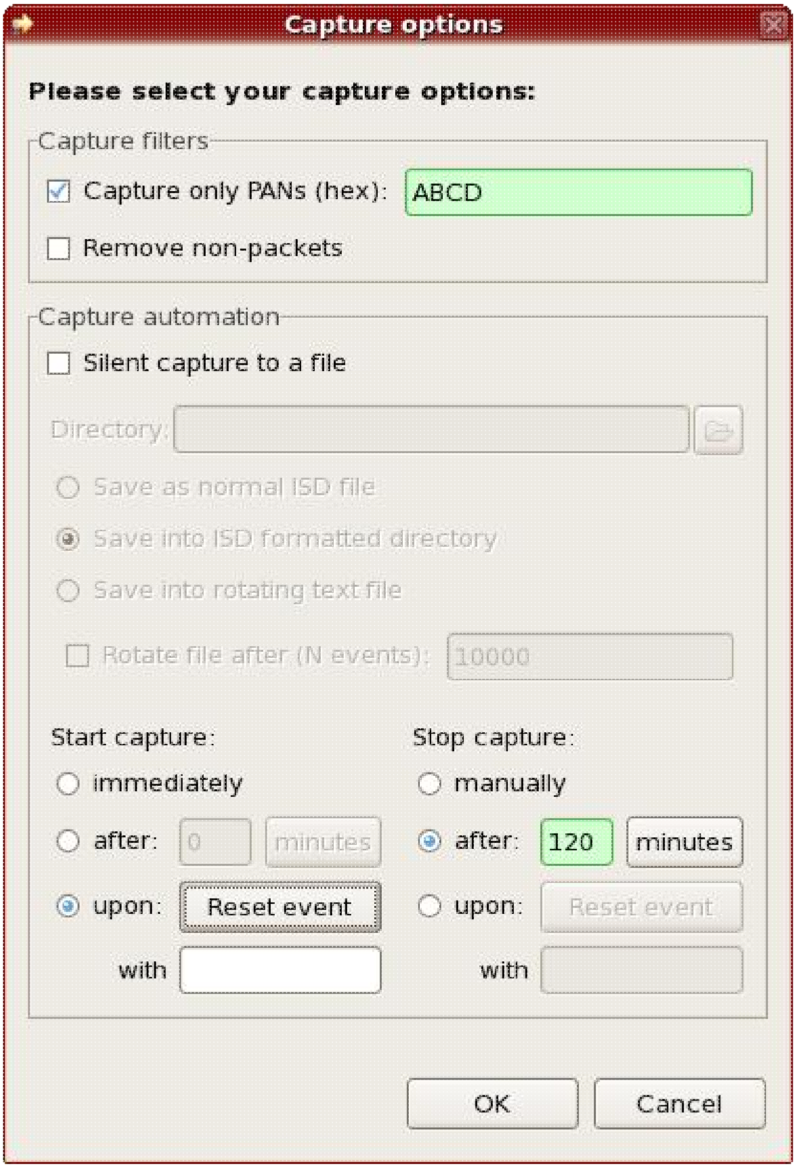
   Note: If you select a node that is not a sniffer, Simplicity Studio asks whether you wish to upload a sniffer application to it.

   The selected sniffer scans all channels for networks that are within range, and lists their PAN ID and channel.
3. Choose a network from the list and click **OK**.
4. From the Adapters view, right-click a sniffer adapter, and select **Start Capture**.

   Note: A single sniffer is not guaranteed to capture all over-the-air transmissions. For instance, a sniffer cannot capture transmissions that are outside its reception area. When monitoring a wide area, you should capture data from multiple sniffers that are geographically dispersed so that you can intercept all transmissions.

14 Capturing with options

You can invoke capture options by right-clicking an adapter and selecting **Capture with Options**.



Capture options

Please select your capture options:

Capture filters

☑ Capture only PANs (hex): ABCD

☐ Remove non-packets

Capture automation

☐ Silent capture to a file

Directory:

○ Save as normal ISD file

● Save into ISD formatted directory

○ Save into rotating text file

☐ Rotate file after (N events): 10000

Start capture:

○ immediately

○ after: 0 minutes

● upon: Reset event

with

Stop capture:

○ manually

● after: 120 minutes

○ upon: Reset event

with

In the capture options dialog, you can set the following:

- **Capture only PANs:** This filter, based on PAN ID, allows a comma-separated list of hexadecimal values.
- **Remove non-packets:** Filters out all the events that are not packets (Printf's, EZSP messages, Asserts, etc.).
- **Silent capture to file:** Select this option if you wish to do a lengthy capture as a background task. Traffic will not show in the GUI until the capture is stopped and the file is opened. The silent capture enables you to run a capture over several days, as it does not consume memory resources, only disk space.
- **Start capture:** Specifies triggers for delayed start of capture.
    - ♦ **immediately:** No automation, capture starts right away.
    - ♦ **after:** Capture starts after a certain time or after a certain number of events.
    - ♦ **upon:** Capture starts upon Node reset, or upon an event containing a specified ASCII or byte pattern.
- **Stop capture:** Specifies triggers for the automatic stop of a capture.

- **manually:**No automation, capture will stop only when user stops it.
- **after:** Capture starts after a certain time or after a certain number of events.
- **upon:** Capture starts upon Node reset, or upon an event containing a specified ASCII or byte pattern.

You can also change some capture options by clicking the **Capture options** button in the toolbar.

14.1 Start/Stop capture on node reset

The triggers for starting and stopping capture work only if you have DEBUG level NORMAL turned on for the node that you are capturing from. With the debug level set to Normal, the chip will send debug information, including node resets, over the back channel to Simplicity Studio. Simplicity Studio uses these reset and other commands to trigger the start-and-stop capture process. If an image does not have debug turned on, it will not send reset information to Simplicity Studio, and Simplicity Studio has no basis for triggering start or stop capture.

14.2 Managing a Session

14.2.1 Live sessions

Live network data is captured into a *live session*, which is different than a session opened from a session file. Simplicity Studio can maintain multiple live sessions, capturing from multiple sets of adapters into the same session. You cannot, however, capture from one adapter into different sessions at the same time.

You can create a new live session in a following ways:

- Perform start capture on an adapter while no live sessions are active on top of editor list. This will create a new live session and start capturing data into it.
- Choose **File | New | New Live Capture Session**. This will create new live capture session and put it on top of editor list.

When you start a capture on an adapter, or on multiple adapters at the same time, the live session used for the capture is assigned in one of the following ways:

- If no live sessions are currently active, a new live session is created and used.
- If live sessions are active, but they are not on top of the editor stack (for example, another file is opened and currently on top), a new live session is created and used.
- If a live session is active and it is also *on top of the editor stack*, then this session will be used for capture.

14.2.2 Stopping a capture

1. Select one or more connected adapters.
2. Right-click the selected adapter(s).
3. From the pop-up menu, choose **Stop Capture**.

The **Start Capture** command resumes capturing to the live session.

14.2.3 Pausing a capture

You can pause a capture at any time by choosing **Edit | Pause**, or clicking the **Pause** button ⏸ in the toolbar. This is a convenient way to stop capturing from a device or devices without having to start a new capture at a later time.

14.2.4 Replaying a session

You can replay events of the current session, whether live or saved, by choosing **Edit | Start Replay** ▶. Simplicity Studio replays the session from the selected event at a constant speed. Replaying events in a live session has no effect on the capture in progress.

To stop the session replay, choose **Edit | Stop Replay** 🟥.

14.2.5 Clearing session events

Click the **Clear Events** icon 🗑 on the toolbar to purge all events and their associated transactions from the current session.

Caution! You cannot retrieve cleared events.

14.3 Saving a session

When you start a capture, it is initially written to an unnamed live session. At any point during a live session, you can save the data thus far captured to a file by choosing **File | Save** (💾). After you save a session file, Simplicity Studio continues to append capture data to it; however, you must save again in order to retain this data in the session file.

Simplicity Studio saves session data to an `.isd` file, which is a compressed file that stores session data and the network state. Network state includes display settings such as map modifications, which Simplicity Studio restores when you reload the session file.

Simplicity Studio closes a saved session from further captures after you explicitly stop the capture, or when you start another live session. After a saved session is closed, it cannot be reopened to capture more data.

If you modify a saved session file--for example, set bookmarks or reposition icons in the Map pane--Simplicity Studio asks whether to save or discard those changes before you close the session.

Note: For security reasons, Security Keys that you may use to decrypt captured data are not included in saved `.isd` files by default. If you wish to share security keys with your files, you should turn on the option "Save decryption keys in Simplicity Studio files" on the Security Keys preference page, which you can access by choosing **File | Preferences | Decoding | Security Keys**

14.3.1 Saving multiple sessions

If multiple open sessions have unsaved data, you can save all of them at once by choosing **File | Save All** (💾).

14.3.2 Saving to text files

You can save transactions to a flat ASCII file.

To save a session as a text file:

1. Choose **File | Export as | Simple text event log**.
2. Name the text file.
3. Click **Save**.

You can also extract specific events from the Transactions pane, Events pane, or Hex Dump pane, and save them into a separate text log file.

To extract specific events to a text file:

1. Right-click the event in the Transactions, Events, or Hex Dump panes that you want to extract.
2. Select **Extract to** from the pop-up menu.
3. Name the text file.
4. Click **Save**.

Once you specify a file, you can append additional events to it by right-clicking the event that you want to append and selecting **Append to file**.

14.4 Open and close options

You can open any session file and review its contents by choosing **File | Open**, or by clicking 📂, and browsing to the desired file. For fast access, choose a recently viewed file from the File menu.

Simplicity Studio provides several options for closing sessions:

To close a single session: Click the session tab's close control ☒.

To close all sessions: Right-click any session tab and choose **Close All** from the pop-up menu. Alternatively, choose the menu bar option **File | Close All**.

To close all but the current session: Right-click its session tab and choose **Close Others** from the pop-up menu.

15 Viewing Data in Editors

Choose **File | Open File...** to open your data in an editor. If the file is smaller than the size set in Preferences for a file to be considered large, Simplicity Studio opens it in the Stream Editor. If your file contains more events than this preference allows, Simplicity Studio opens it in the Large File editor.

15.1 The Stream Editor

The Stream Editor decrypts, decodes, and displays details of individual events.

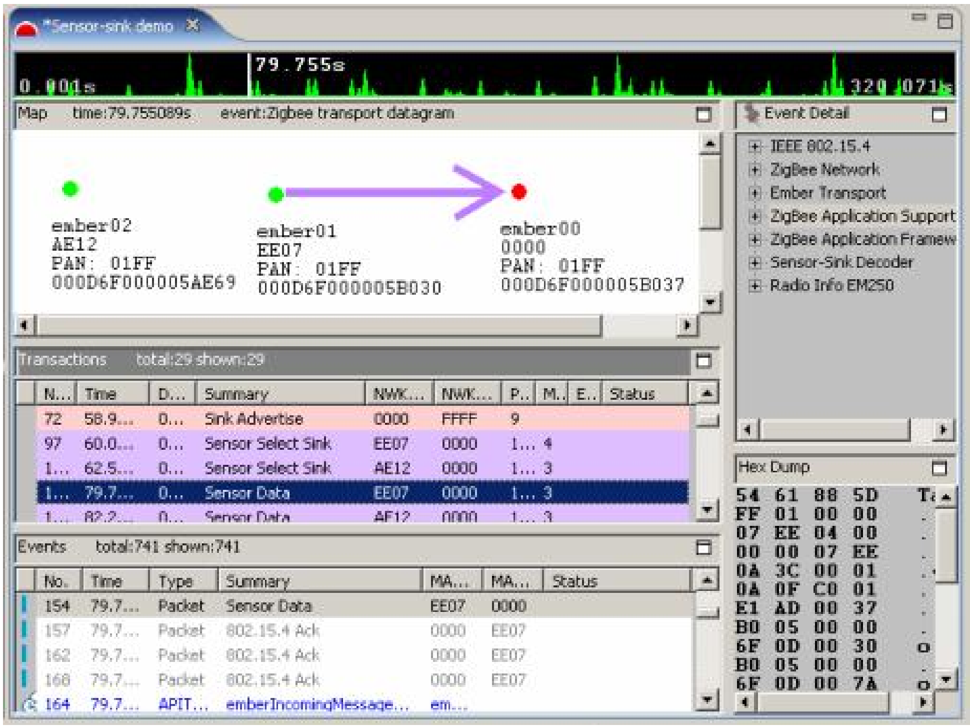
15.1.1 Large File Editor

The initial releases of Simplicity Studio had only the Stream Editor. Our users very quickly ran into difficulty, though, when dealing with very large files (>100K events). While they liked the level of detail shown by the Stream Editor, the amount of processing required to present that information with files of large size caused Simplicity Studio to run very slowly. To resolve this issue, we created the Large File Editor. The Large File Editor does not offer any detailed decoding and presentation of events. Instead, it provides a high-level overview of a file and allows users to open their points of interest in the Stream Editor.

The Large File Editor allows you to search over very large files by showing an overall timeline and node statistics. It provides an overview of all the traffic captured in a trace, but does not itself decode events. Instead, you use the Large File Editor to narrow down the areas in a trace that interest you. You can then open the points of interest in the Stream Editor for further examination.

16 Stream Editor

You can view captured data in the Stream Editor.



Transactions total:29 shown:29

N...	Time	D...	Summary	NWK...	NWK...	P...	M...	E...	Status
72	58.9...	0...	Sink Advertise	0000	FFFF	9			
97	60.0...	0...	Sensor Select Sink	EE07	0000	1...	4		
1...	62.5...	0...	Sensor Select Sink	AE12	0000	1...	3		
1...	79.7...	0...	Sensor Data	EE07	0000	1...	3		
1...	82.2...	0...	Sensor Data	AE12	0000	1...	3		

Events total:741 shown:741

No.	Time	Type	Summary	MA...	MA...	Status
154	79.7...	Packet	Sensor Data	EE07	0000	
157	79.7...	Packet	802.15.4 Ack	0000	EE07	
162	79.7...	Packet	802.15.4 Ack	0000	EE07	
168	79.7...	Packet	802.15.4 Ack	0000	EE07	
164	79.7...	APIT...	emberIncomingMessage...	em...		

Event Detail

- IEEE 802.15.4
- ZigBee Network
- Ember Transport
- ZigBee Application Support
- ZigBee Application Framew
- Sensor-Sink Decoder
- Radio Info EM250

Hex Dump

```
54 61 88 5D    Ta
FF 01 00 00    .
07 EE 04 00    .
00 00 07 EE    .
0A 3C 00 01    .
0A 0F C0 01    .
E1 AD 00 37    .
B0 05 00 00    .
6F 0D 00 30    o
B0 05 00 00    .
6F 0D 00 7A    o
```

The Stream Editor contains five editor panes, each of which provides a different view of the captured session data:

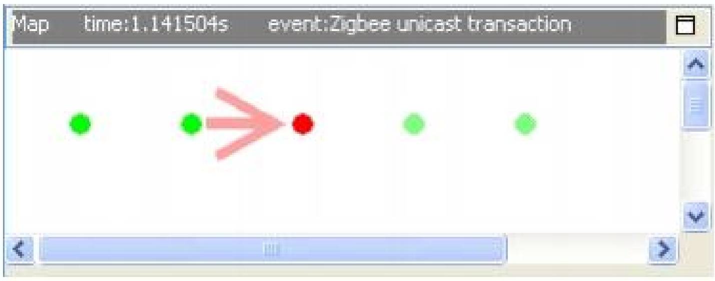
- Map pane: Provides a map of the network, with nodes displayed with their network identifiers. The map also displays network activity.
- Transactions pane: Displays high-level node interactions that might comprise multiple events.
- Events pane: Displays information about all packets transmitted and received during a capture session.
- Event Detail pane: Displays the decoded contents of the packet that is currently selected in the Events pane.
- Hex Dump pane: Displays the data of the selected event in raw bytes. Simplicity Studio highlights bytes that map to the data currently selected in the Event Detail pane.

All five editor panes may be open at once. Live captured data is continuously updated and displayed in the editor panes.

For more information about using the Stream Editor panes, see Editor navigation tools.

16.1 Map pane

The Map pane shows all interaction between nodes at a high level. As events occur or are replayed, the Map pane refreshes to show the pattern of network communication. Debug messages issued from a node also display next to the node.



Each node is represented by an icon. To view a node's current information, hover the cursor over the node.

16.1.1 Node Colors

Each node in the map pane is given a different color depending on its capabilities within the network as they are understood by Simplicity Studio based on captured data.

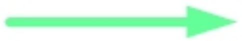
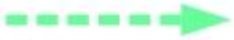
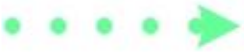
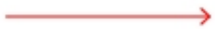
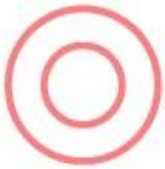
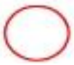
RED: Signifies that this node is a network coordinator.

BLACK: Signifies that this node is a router.

GREEN: Default color for network nodes.

Table 1 shows the graphical elements that appear in the Map pane to depict network activity. Thick lines depict transactions, while thin lines depict single packets.

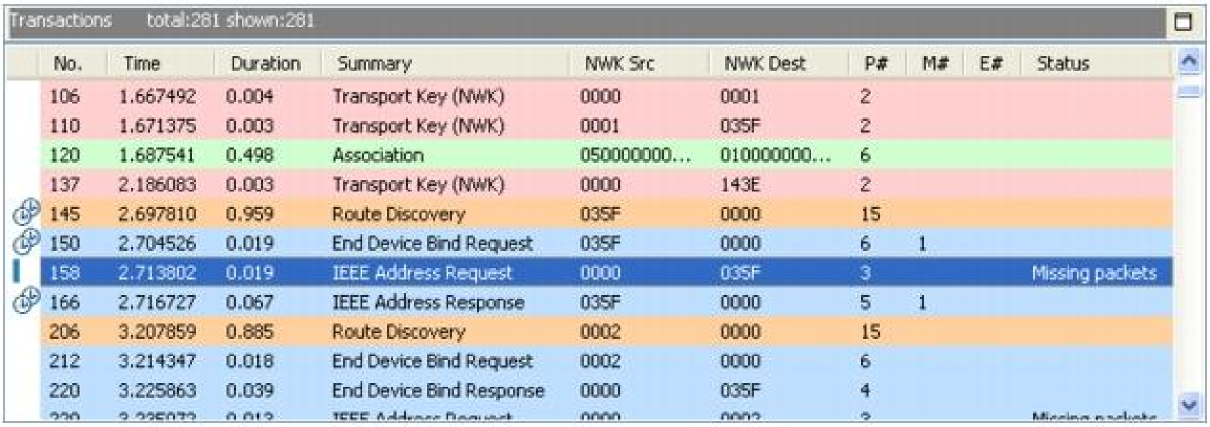
Table 1: Graphical Elements in Map Pane

Unicast transmissions	
	Unicast transaction: completed after a single try.
	Unicast transaction: completed after multiple retries.
	Failed transaction: no end-to-end ACK
	Transmission of a unicast packet.
Multicast transmissions	
	Multicast transaction transmitted from initiating node.
	Transmission of multicast packet.

Note: The colors shown vary according to the transaction or event type, and can be configured through the Filter Manager.
For details on how to modify the Map pane, see Customizing editor panes.

17 Transactions Pane

The Transactions Pane displays higher-layer protocol events that consist of multiple packet transmissions. For example, a ZigBee broadcast is retransmitted by every node in the network. By analyzing packet headers, Simplicity Studio determines which packets belong to the same transaction and groups them accordingly.



Transactions total:281 shown:281

No.	Time	Duration	Summary	NWK Src	NWK Dest	P#	M#	E#	Status
106	1.667492	0.004	Transport Key (NWK)	0000	0001	2			
110	1.671375	0.003	Transport Key (NWK)	0001	035F	2			
120	1.687541	0.498	Association	050000000...	010000000...	6			
137	2.186083	0.003	Transport Key (NWK)	0000	143E	2			
145	2.697810	0.959	Route Discovery	035F	0000	15			
150	2.704526	0.019	End Device Bind Request	035F	0000	6	1		
158	2.713802	0.019	IEEE Address Request	0000	035F	3			Missing packets
166	2.716727	0.067	IEEE Address Response	035F	0000	5	1		
206	3.207859	0.885	Route Discovery	0002	0000	15			
212	3.214347	0.018	End Device Bind Request	0002	0000	6			
220	3.225863	0.039	End Device Bind Response	0000	035F	4			
[illegible]	[illegible]	[illegible]	IEEE Address Request	0000	0002	3			Missing packets

Typical transactions include:

- **802.15.4 association**: Involves a request-response protocol that consists of at least 6 packet transmissions.
- **APS unicast**: Can contain the following events:
    - ♦ A MAC layer unicast packet and its MAC retries
    - ♦ Acknowledgements for each hop along the route
    - ♦ An end-to-end APS acknowledgement message, which itself consists of multiple MAC unicast packets
    - ♦ Multiple end-to-end APS retries
- **ZigBee route discovery**: Involves a broadcast route request followed by unicast route-reply packets across multiple hops.

Simplicity Studio understands the protocol semantics for many transaction types. Therefore it can group multiple packets in real time to facilitate high-level analysis.

All transactions are listed in chronological order, using transaction start times. Each selection maps to one or more events in the Events Pane, which are marked accordingly. **Clock** icons 🕒 indicate concurrent transactions with the current selection.

All transactions and their events are uniquely numbered. However, the transaction numbers may not be in sequence, and various factors will result in number gaps. For example, Ember Desktop shows only top-level transactions and the lowest-level packets. Intermediate transactions are not shown. Also, number gaps are likely to occur if filters are turned on.

17.1 Transaction Display

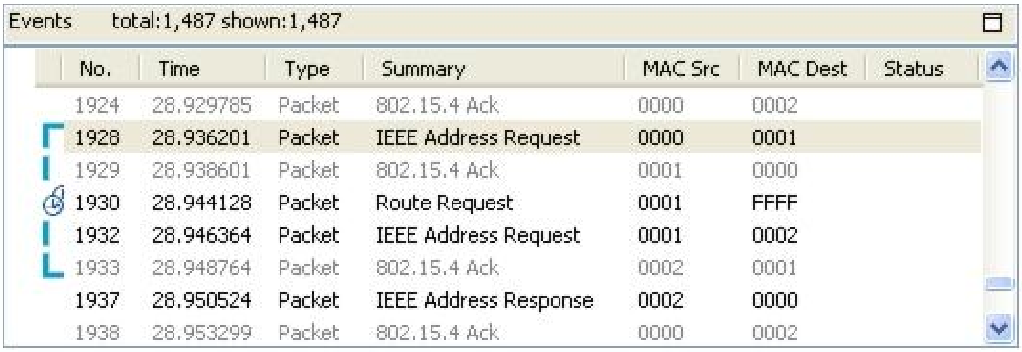
When you click on a transaction, the information shown in the Event Detail Pane and the Hex Dump Pane corresponds to the first packet in the transaction. However, if you have filters turned on, it is possible that the first transaction might not be shown in the Event Pane. In that case, the event detail information in the transaction display will not be consistent with the first packet shown in the Event Pane. In fact, with a filter expression such as

```
show(transaction.summary != null, SELF)
```

you will see only transactions and the Event Pane will be blank. In that case, click on the transaction to see the first events in the transaction in the Event Detail and Hex Dump Panes.

17.2 Events pane

The Events pane displays information about packets received by the current session. All events are displayed in chronological order.



Events total:1,487 shown:1,487

	No.	Time	Type	Summary	MAC Src	MAC Dest	Status
	1924	28.929785	Packet	802.15.4 Ack	0000	0002	
┌	1928	28.936201	Packet	IEEE Address Request	0000	0001	
|	1929	28.938601	Packet	802.15.4 Ack	0001	0000	
🕓	1930	28.944128	Packet	Route Request	0001	FFFF	
|	1932	28.946364	Packet	IEEE Address Request	0001	0002	
└	1933	28.948764	Packet	802.15.4 Ack	0002	0001	
	1937	28.950524	Packet	IEEE Address Response	0002	0000	
	1938	28.953299	Packet	802.15.4 Ack	0000	0002	

Events that belong to the currently selected transaction in the Transactions pane are marked by one of the following icons:

Table 2: Transaction/Event Icons

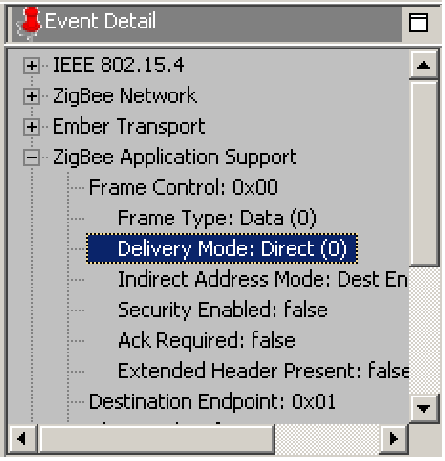
Transaction icon	Meaning
┌	Start of transaction
|	Intermediate event
└	End of transaction
⊏	Single-event transaction

Clock icons,🕓, mark unrelated events that are concurrent with the selected transaction in the Transactions pane.

17.3 Event Detail pane

The Event Detail pane displays the decoded contents of the packet that is currently selected in the Events pane. The content of this pane varies according to the packet type. If a transaction is selected on the Transactions pane, the Event Detail pane shows the details of the *first* event in the transaction.

When capturing from multiple devices, Simplicity Studio may capture the same packet as heard by several different sources. In order to reduce confusion, Simplicity Studio automatically performs duplicate detection on all packets captured. If the transmission is captured over the backchannel, only the transmitted packet will be kept by Simplicity Studio, otherwise the first receive packet is kept. All duplicate packets are dropped after extracting their RadioInfo data. Only the radio info frame for each duplicate packet is kept. The radio info for each individual instance of a packet captured by Simplicity Studio is visible in the Radio Info View



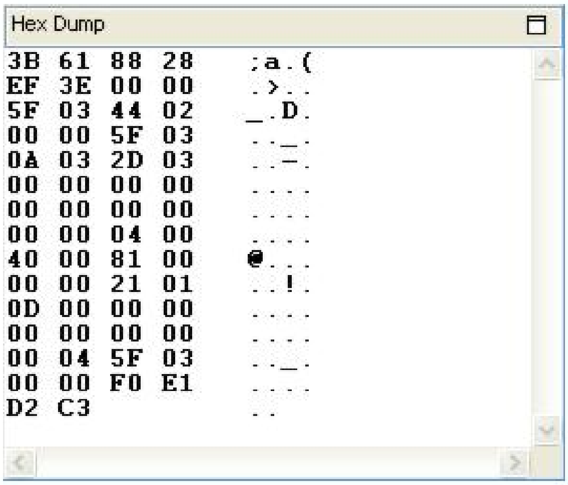
Event Detail

```
⊞ IEEE 802.15.4
⊞ ZigBee Network
⊞ Ember Transport
⊟ ZigBee Application Support
    Frame Control: 0x00
        Frame Type: Data (0)
        Delivery Mode: Direct (0)
        Indirect Address Mode: Dest En
        Security Enabled: false
        Ack Required: false
        Extended Header Present: false
    Destination Endpoint: 0x01
```

Pinning a field:The Event Detail pane has the ability to "pin" a field into view. When you double-click on a specific field, the **Pin** icon in the top left of the pane will turn bright red indicating that it is active. This makes it so that, as you move through events, this field will always be visible when it is present in the currently selected packet. This is useful if you are interested in a specific field across multiple events in a trace file. In the image to the left, the ZigBee Application Support Delivery Mode is "pinned" into view. The pin can be deactivated at any time by either double-clicking on the pinned field, or by clicking the **Pin** icon itself.

17.4 Hex Dump pane

The Hex Dump pane displays data in raw bytes of a selected event in the Events pane. Clicking on bytes in the Hex Dump pane selects the corresponding field in the Event Detail pane. Alternatively, selecting a field or a frame in the Event Detail pane highlights the corresponding bytes in the Hex Dump pane.

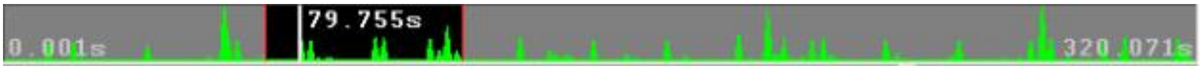


Hex Dump

```
3B 61 88 28    ;a.(
EF 3E 00 00    .>..
5F 03 44 02    _.D.
00 00 5F 03    .._.
0A 03 2D 03    ..-.
00 00 00 00    ....
00 00 00 00    ....
00 00 04 00    ....
40 00 81 00    @...
00 00 21 01    ..!.
0D 00 00 00    ....
00 00 00 00    ....
00 04 5F 03    .._.
00 00 F0 E1    ....
D2 C3          ..
```

17.5 The Timeline bar

The Timeline bar displays the statistics of the traffic over time.



Available actions are:

- Click on the Timeline bar to move the cursor to the event closest to the time selected.
- Click and drag on the timeline to filter the display to only the time within the selected area.
- Right-click to display a timeline menu.

The Timeline bar also shows the bookmarks as yellow flags. You can click a bookmark to jump to it in the Transaction and Event panes.

17.6 Editor navigation tools

Simplicity Studio provides additional **Edit** menu options for navigating editor panes:

- **Edit | Lock to Bottom** (⤓): locks the cursor on the latest event during a live session. To remove the lock, select any event or transaction during a live session, which causes a view to scroll as the events are captured.
- **Edit | Go to Line** (➜☰): moves the cursor to the event or transaction having the specified event number.
- **Edit | Go to Time** (➜🕓): moves the cursor to the transaction and event that match or immediately follow the specified time.
- **Edit | Go to Bookmark** (➜🔖): moves the cursor to the selected bookmark.
- **Edit | Timeline** (📈): displays a graph above the Map pane of network activity over the time of capture. Select any time on the timeline to display the related information in the other panes. You can also Configure the Timeline display.

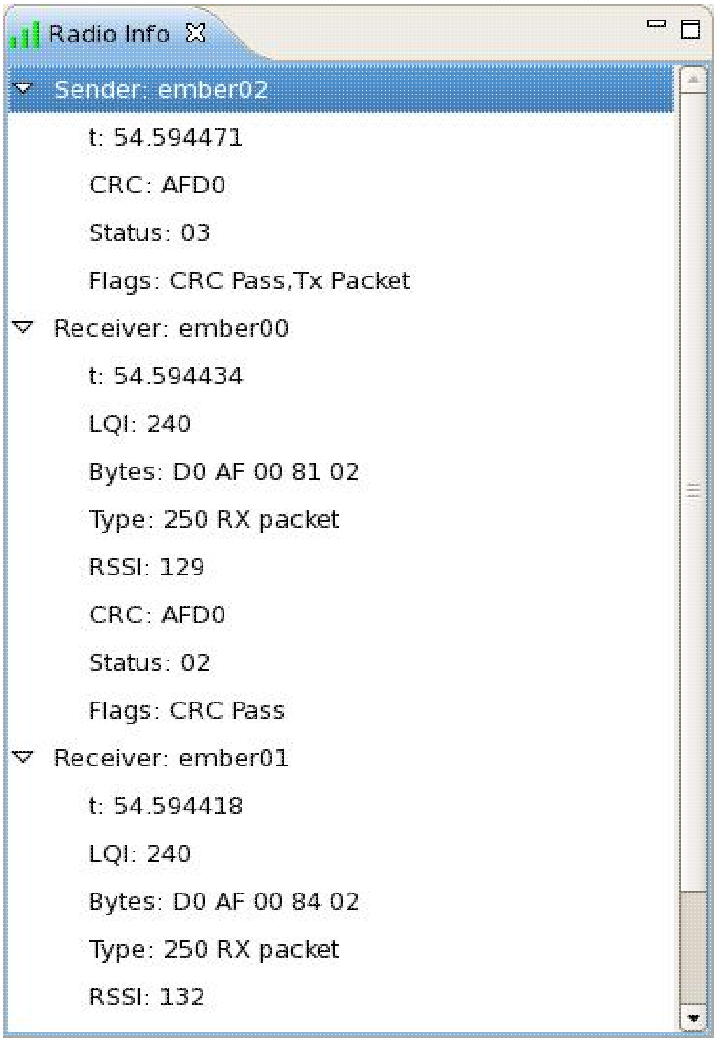
17.7 Other features

- **Edit | Edit description of trace file** (📄): opens a simple dialog which allows you to view and edit overall description of the captured data. This is helpful if you need to pass on some information to either the Silicon Labs support team, or to your other team members for analyzing the contents of the trace file.

18 Radio Info View

The Radio Info view is a helper view that shows the information from the radio of all the receivers in the network that have heard the currently selected event.

To open the Radio Info view, choose **Window | Show View | Radio Info**. The view is positioned by default in the bottom-left corner of the screen and will display in a tree all the information that has been gathered from the receiver nodes. Displayed information will include LQI value, CRC value, and the status bits that show several states of the radio.

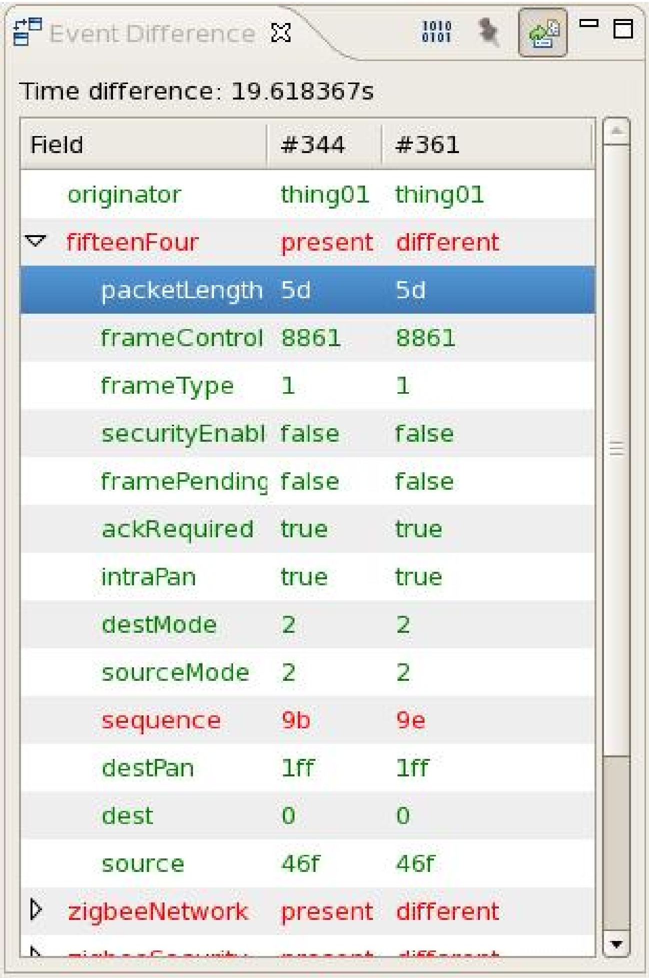


```
▽ Sender: ember02
      t: 54.594471
      CRC: AFD0
      Status: 03
      Flags: CRC Pass,Tx Packet
▽ Receiver: ember00
      t: 54.594434
      LQI: 240
      Bytes: D0 AF 00 81 02
      Type: 250 RX packet
      RSSI: 129
      CRC: AFD0
      Status: 02
      Flags: CRC Pass
▽ Receiver: ember01
      t: 54.594418
      LQI: 240
      Bytes: D0 AF 00 84 02
      Type: 250 RX packet
      RSSI: 132
```

The event that supplied radio information for the preceding pane was captured from both the sending and receiving nodes. This is possible because the trace that contains this event was created by capturing from both the sending and receiving nodes simultaneously using Simplicity Studio's Perfect Trace capability. While the original events were merged into a single event by Simplicity Studio's duplicate detection mechanism, the radio information was retained for each event and is shown in the Radio Info view with the time that the event was captured by Simplicity Studio.

19 Event Difference View

Event Difference view is a helper view that displays the specific differences between two packets.



Time difference: 19.618367s

Field	#344	#361
originator	thing01	thing01
▽ fifteenFour	present	different
packetLength	5d	5d
frameControl	8861	8861
frameType	1	1
securityEnabl	false	false
framePendinc	false	false
ackRequired	true	true
intraPan	true	true
destMode	2	2
sourceMode	2	2
sequence	9b	9e
destPan	1ff	1ff
dest	0	0
source	46f	46f
▷ zigbeeNetwork	present	different
▷ zigbeeSecurity	present	different

To open the Event Difference view, click the **diff view** toolbar button ( ), or choose **Window | Show view | Event difference**.

Once diff view is shown, it tracks the selected events. Diff view will by default show the difference between the last two events selected. If you select event 1, and then click on event 2, diff view will show the diff between those two events. If later you select event 3, view will show the differences between event 2 and 3.

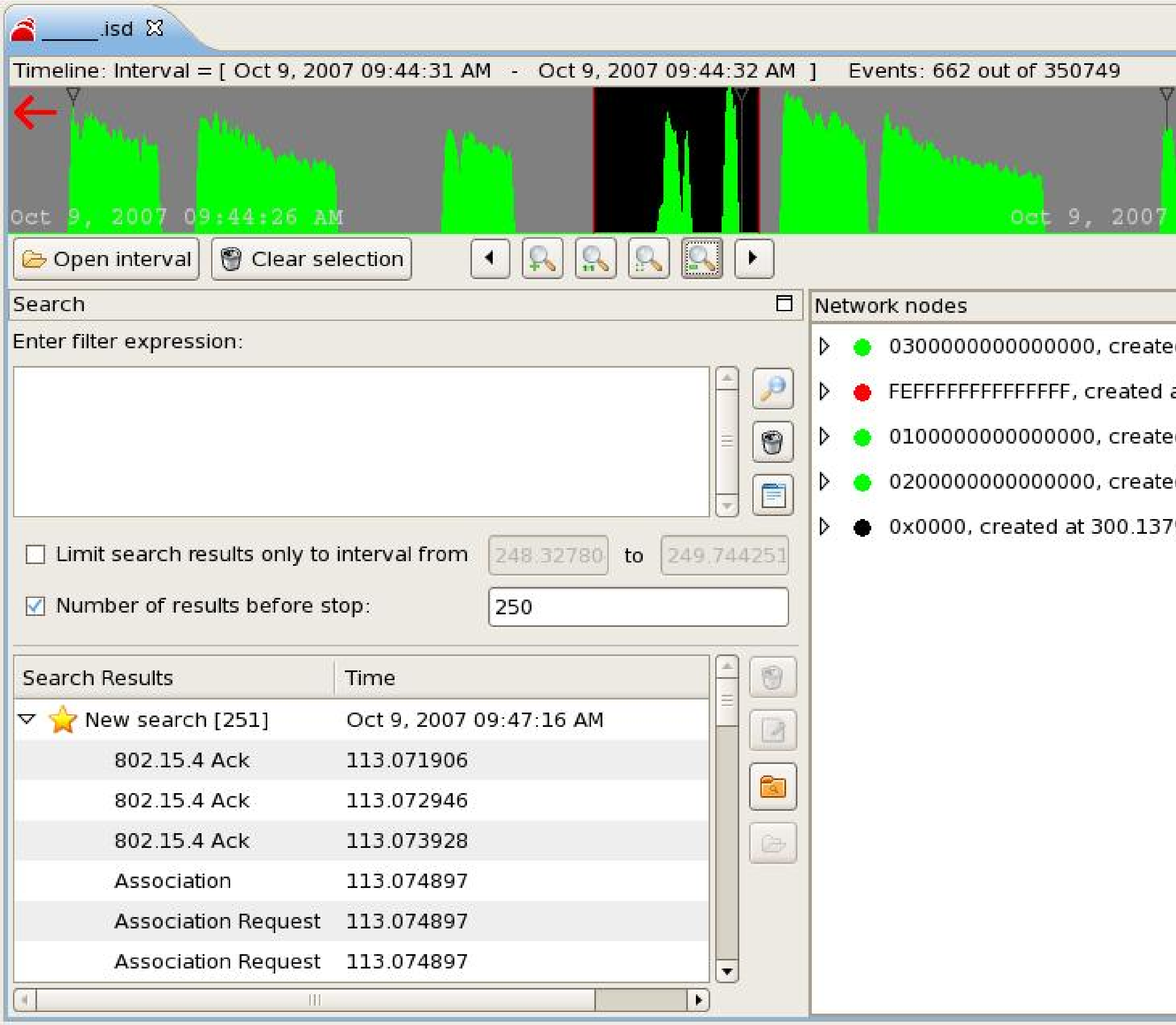
Packet frames that do not have any differences are shown in green. Frames that contain differences are shown in red. To see which portions of the frames are different, expand the tree by clicking on the plus marker to the left of the frame's name.

There are few additional controls on top:

- **Show byte differences** ( ) enables viewing of individual bytes in diff view.
- **Pin last selected event** ( ) changes the way events are tracked. If this is enabled, then the first event for diffing stays the same, and only the second event changes. You can use this if you wish to always differentiate events against a certain static event, rather than always viewing last two selected events.
- **Show fields that are same** ( ) enables filtering out fields that are same in both events.

20 Large File Editor

The Stream Editor provides detailed information about all the events in a trace. Providing such detailed information, however, is costly in terms of memory consumption and software performance. Captured traces can be too large for easy viewing in the Stream Editor. To view files that are too large for the Stream Editor, use the Large File Editor. The Large File Editor allows you to find and select a region of interest, which you can then open and analyze with the Stream Editor.



Timeline: Interval = [Oct 9, 2007 09:44:31 AM - Oct 9, 2007 09:44:32 AM] Events: 662 out of 350749

☐ Limit search results only to interval from 248.32780 to 249.744251

☑ Number of results before stop: 250

Search Results	Time
New search [251]	Oct 9, 2007 09:47:16 AM
802.15.4 Ack	113.071906
802.15.4 Ack	113.072946
802.15.4 Ack	113.073928
Association	113.074897
Association Request	113.074897
Association Request	113.074897

- ● 0300000000000000, create
- ● FEFFFFFFFFFFFFFF, created a
- ● 0100000000000000, create
- ● 0200000000000000, create
- ● 0x0000, created at 300.137

You can determine what size files will use the Large File Editor by choosing **File | Preferences | File | Number of events for file to be large**. Any file larger than the value you set will be opened in the Large File Editor.

The Large File Editor consists of three component panes:

- Large File Timeline
- Large Search pane
- Large File Network Nodes pane

21 Large File Timeline

The Large File Timeline shows a high-level view of a large file's traffic over time. It works similarly to the Stream Editor's timeline. In fact, the Large File and the Stream Editor timelines use the same widget.

Large File Timeline Segments
The entire set of events shown in the Large File Timeline is broken into segments. By default, each segment includes up to 5,000 events. Segments boundaries are shown in the Large File Timeline by horizontal grey lines.

Large File Timeline Time Markers
The Large File Timeline shows the actual time during which a file was captured. The capture start-time appears in the bottom left corner of the Timeline. The capture end-time appears in the bottom right corner.

Moving your cursor to any point on the timeline displays the time for that point.

Large File Intervals
A Large File Interval is a subset of an entire trace. You can create an interval by clicking and dragging. Click on the Large File Timeline at your desired start-point, and then drag your cursor along the timeline to your desired end-point.

The click and drag operation will create the interval in the timeline that you defined and zoom the timeline view into that interval. To clear the interval, right-click on the timeline and choose **Clear interval**.

Once you have created an interval, you can open that interval in the Stream Editor by clicking the **Open Interval** button.

Timeline Flags
Search results and errors are displayed in the Large File Timeline by flags. Search results are displayed as a yellow flag. Errors are displayed as a red flag. When you move the mouse over the flag, the Timeline will display the Summary of the Event or Transaction that is associated with the flag.

22 Large File Search

Use the Large File Search mechanism to search for events across very large files. The entire filter language is supported in the Large File Search. For more information, see Filter Language.

To run a search in the Large File Editor, enter your filter expression into the Filter Expression text box and click the **Search** icon. The search progress is shown in the Large File Timeline. Options allow you to limit the search within a time interval or to limit the number of search results. It is useful to limit the number of search results, as the system can become slow if your search expression matches an extremely large number of events.

Filter results are shown in the Search Results table. Search results are grouped into results trees and labeled with the time and date that the search was performed.

To view the details of your search results, double-click an individual search result. This will open three segments in a Stream Editor: one before your selected event, one that contains your event, and one after your event.

Note: The expresssions and search results are saved into the Simplicity Studio file. Thus they will be seen by other users who open the same Simplicity Studio file.

Controls to the right of the search results allow you to delete, tag, open, and assign decorative icons to the searches and search results.

23 Large File Network Nodes

The Large File Network Node pane shows all of the network devices included in a large file. The information provided about each node in a trace includes:

- EUI64 address
- short address
- PAN ID
- node type

Since each of these values is subject to change over time, the summary also includes the time at which each value was discovered.

24 Filtering Captured Data

By default, the Events Pane displays all session events. You can build and apply filters that constrain Simplicity Studio to show only those events that interest you. By including only events of interest, you can analyze results more efficiently.

You apply filters to each session individually. When you change a session's filters, Simplicity Studio immediately refreshes the display. When you exit Simplicity Studio, all session filters are cleared and must be reapplied when you restart the next session.

Simplicity Studio provides the following ways to create and manage filters:

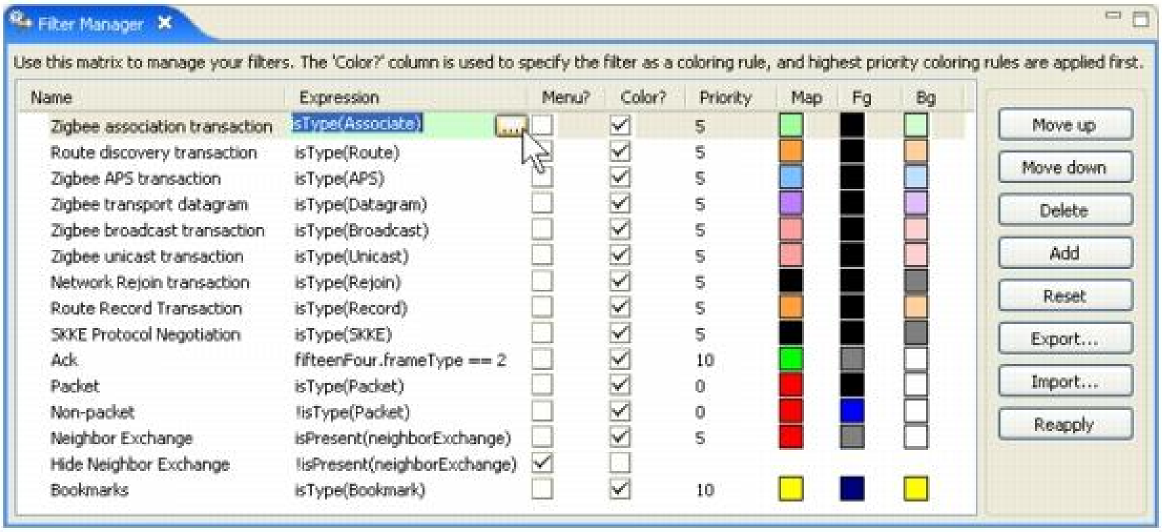
- **The Filter Manager** maintains a set of saved filters that you can review and edit. You can also add new filters. You specify any of the saved filters for display on the Filters menu, where they are accessible for use in one or more sessions.
- **The Filter Bar** is a drop-down pane that attaches to a given session in which you can enter one or more filter expressions on the fly. When you exit Simplicity Studio, it discards filter bar expressions for all sessions.
- **Quick filters** are available when you right-click on events and transactions. They provide an easy way to create common expressions for the filter bar.
- **Filter language** is a powerful syntax used to create filters.

24.1 Filter Manager

The Filter Manager lets you perform the following tasks:

- Compose filter expressions with the Expression Builder.
- Use filter expressions to customize display of entries in the Events pane.
- Specify which filters appear in the **Filters** menu.

To show the Filter Manager: Choose the menu option **Filters | Filter Manager**.



Use this matrix to manage your filters. The 'Color?' column is used to specify the filter as a coloring rule, and highest priority coloring rules are applied first.

Name	Expression	Menu?	Color?	Priority	Map	Fg	Bg
Zigbee association transaction	isType(Associate)	□	☑	5	🟩	⬛	🟩
Route discovery transaction	isType(Route)	□	☑	5	🟧	⬛	🟧
Zigbee APS transaction	isType(APS)	□	☑	5	🟦	⬛	🟦
Zigbee transport datagram	isType(Datagram)	□	☑	5	🟪	⬛	🟪
Zigbee broadcast transaction	isType(Broadcast)	□	☑	5	🟥	⬛	🟥
Zigbee unicast transaction	isType(Unicast)	□	☑	5	🟥	⬛	🟥
Network Rejoin transaction	isType(Rejoin)	□	☑	5	⬛	⬛	▩
Route Record Transaction	isType(Record)	□	☑	5	🟧	⬛	🟧
SKKE Protocol Negotiation	isType(SKKE)	□	☑	5	⬛	⬛	▩
Ack	fifteenFour.frameType == 2	□	☑	10	🟩	▩	⬜
Packet	isType(Packet)	□	☑	0	🟥	⬛	⬜
Non-packet	!isType(Packet)	□	☑	0	🟥	🟦	⬜
Neighbor Exchange	isPresent(neighborExchange)	□	☑	5	🟥	▩	⬜
Hide Neighbor Exchange	!isPresent(neighborExchange)	☑	□				
Bookmarks	isType(Bookmark)	□	☑	10	🟨	🟦	🟨

24.1.1 Building filter expressions

The Filter Manager provides an Expression Builder that provides a built-in set of valid operands and operators that you can use to compose filter expressions. You can also enter expressions manually. In both cases, Expression Builder validates entries as you work and prevents you from setting invalid expressions.

To use Expression Builder:

1. Select an **Expression** from the Filter Manager list.
2. Click the Expression Builder control, … to display the Expression Builder.
3. Choose the required operands, operators, and values. You can use logical operators such as && and || to specify multiple conditions, and parentheses ( ) to group expressions and set precedence.
4. When your expression is complete, click **OK** from the Expression Builder.

24.1.2 Maintaining filters

To add a filter to the Filters menu: Set the **Menu** checkbox. This makes the filter available to the current sessions.

To restore filters to installation settings: Click **Reset** on the Filter Manager.

You can also add, delete, and rename saved filters.

24.1.3 Setting filter color schemes

You can associate a color scheme with each filter. If an event evaluates as true for a given filter expression, Simplicity Studio applies the filter's color scheme to that event. An event can be configured with a color scheme for two display levels:

- For the Map pane, the color used by the graphic representing the event type.
- For the Events pane, the foreground and background colors used by event type instances.

> **Note:** Setting a color scheme on a filter expression does not specify whether to display events; it only determines how to display certain event types.

> **Note:** If a filter's foreground color is the same as its background color you will not be able to read the text in the event and it will effectively disappear from view.

To set a filter color scheme:

1. From the Filter Manager's **Map**, **Fg**, or **Bg** columns, click the filter's checkbox.
2. Set the filter's color of the Map pane graphic (**Map**), and foreground (**Fg**) and background (**Bg**) colors of event instances, by clicking the control. …
3. To set the filter's **Color** checkbox to determine whether to use this color scheme.
4. For each filter, set the color scheme's **Priority** level by assigning a positive or negative integer value. If an event evaluates as true for multiple filters, Simplicity Studio uses the color scheme with the highest precedence. In the case where there is no clear precedence, Simplicity Studio randomly chooses one of the matching color schemes.
5. To refresh the Map pane and Events pane with the new color scheme, click **Reapply** on the Filter Manager.
6. Filters whose menu checkbox is selected will be shown in the top filter navigation. The order in which the filters are displayed in the Filters menu is determined by their order in the filter manager table. Consequently, you may move filters up and down the list as you wish them to be displayed in the

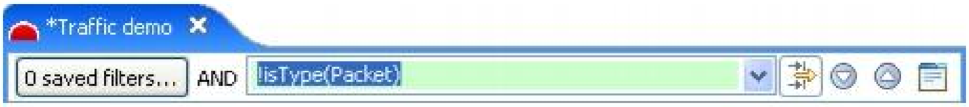
filter menu.

24.2 Filter Bar

A session's Filter Bar provides the same level of functionality as the Filter Manager for building expressions. Any filter expression that appears in a session's Filter Bar is combined with the saved filter expressions that are already in effect for that session.

To show the Filter Bar:

Choose **Filters | Filter Bar** or click the **Filter Bar** icon (⇉).



!isType(Packet)

You can compose a Filter Bar expression in the following ways:

- Enter the expression directly in the filter edit field, and press **Enter**.
- Click the Expression Builder icon in the Filter Bar to use the Expression Builder (▤).
- Create a Quick filter.

To apply a Filter Bar filter: Enter an expression into the filter edit field and press **Enter**.

To find an event that matches the filter: Click the **Find** icons (▲ ▼) on the Filter Bar.

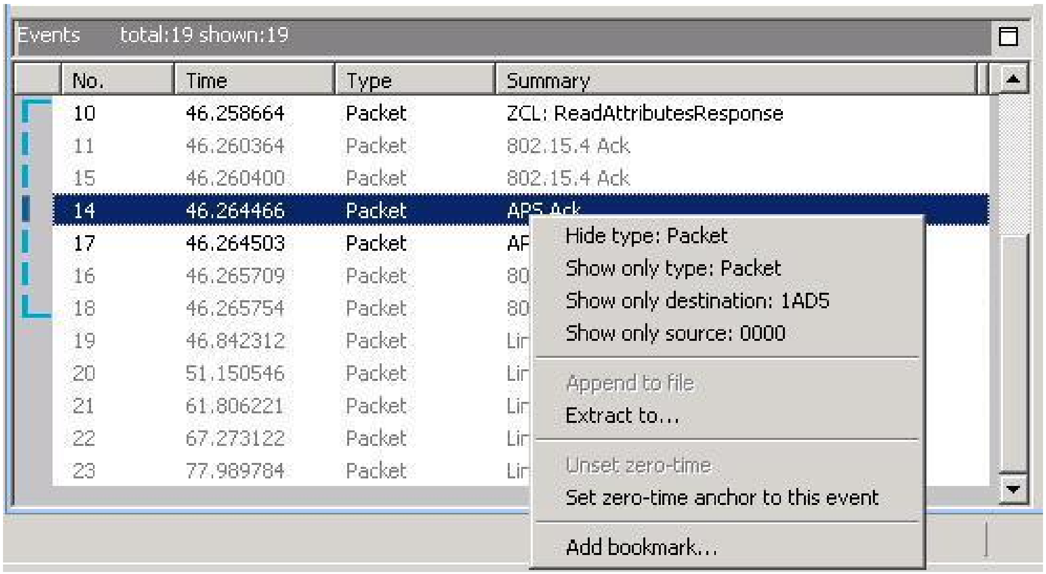
To remove a filter: Select it and press **Enter**.

Note: A history of filter expressions is maintained in the drop-down list.

24.3 Quick filters

Simplicity Studio provides several filters that are available from the Transactions pane, Events pane, Event Detail pane, and Hex Dump pane.

To access a quick filter: Right-click an item and choose a quick filter pop-up menu option.



Events total:19 shown:19

No.	Time	Type	Summary
10	46.258664	Packet	ZCL: ReadAttributesResponse
11	46.260364	Packet	802.15.4 Ack
15	46.260400	Packet	802.15.4 Ack
14	46.264466	Packet	APS Ack
17	46.264503	Packet	AF
16	46.265709	Packet	80
18	46.265754	Packet	80
19	46.842312	Packet	Lir
20	51.150546	Packet	Lir
21	61.806221	Packet	Lir
22	67.273122	Packet	Lir
23	77.989784	Packet	Lir

24.3.1 Hide or show events/transactions

You access these filters by right-clicking an event (Events pane) or transaction (Transactions pane) and choosing a pop-up menu option to hide or show specific information. The filter options that are available depend on the transaction or event that you select. You can specify to hide all events/transactions of the selected type, or to show only that type. In addition if you select an event of type APITrace, the pop-up menu displays two filter options:

- Hide type: APITrace
- Show only type: APITrace

Further, if you select an event such as a neighbor exchange that has a source and/or destination address, the pop-up menu also contains these two filter options:

- Show only destination: short-ID
- Show only source: short-ID

In all cases, Simplicity Studio enters the corresponding filter expression in the session's Filter Bar. For example, if you specify to show only route discovery transactions, this expression is set in the filter bar:

```
isType(Route)
```

Note: When you choose a quick filter, its expression is added to the filter bar expression editor. View these changes to better understand the filter language.

24.3.2 Frame byte pattern filtering

A frame pattern filter matches a specific byte array pattern. For example, you could filter for packets in a payload whose frame has the third byte equal to 0x33. (Many more complex combinations are possible.)

To create a frame pattern filter:

1. Right-click a frame in the Event Detail pane or Hex Dump pane.
2. Select **Filter by frame pattern** from the pop-up menu.
3. In the **Byte Pattern** dialog, check the byte pattern match desired and click **OK**.

The filter is added to the Filter Bar for this session.

25 Filter Language

Filter language enables you to construct logical expressions, based on decoded fields in events. The following are some examples:

- **fifteenFour.sequence == 0x52**: Matches events where 15.4 sequence number equals hex 0x52.
- **fifteenFour.ackRequired == true && fifteenFour.source == 0x035f**: Matches events where 15.4 ack required flag is set, and source shortId is 0x035f.
- **isPresent(zigbeeSecurity.frameCounter)**: Matches events that contain the ZigBee security frame, and the frameCounter field within it.
- **event.summary | "string"**: Matches events where a *string* is a substring of the summary.
- **isType(Packet)**: Matches events that are packets.
- **frameMatch(fifteenFour,"**88**EF**********")**: Matches events where 15.4 frame contains second byte equal to 0x88 and fourth byte equal to 0xEF.

A good way to learn the filter language is by first using the **Add to filter** right-click option in the Event Detail Pane. This option will add a filter expression for the chosen field.

You can use most standard logical operators (&&, ||) and standard comparison operators (==, !=, |, <, >, <=, >=, etc.) in filter expressions.

25.1 Event and Transaction Filter extensions

In addition to filtering on decoded packet fields, you can filter on several other Event and Transaction values.

25.1.1 Event Extensions

1. **event.summary**: A String value of the summary shown in the Event Pane.
    - ♦ Example: event.summary == "APS Ack"
2. **event.linkStatus**: True if the packet is a Link Status packet.
    - ♦ Example: event.linkStatus == true
3. **event.ack**: True if the packet is an 802.15.4 ack.
    - ♦ Example: event.ack == true
4. **event.time**: The time that the event was transmitted (tx) or received (rx).
    - ♦ Example: event.time >= 75.78
5. **event.originator**: The adapter that saw and reported the event.
    - ♦ Example: event.originator == "ewb-unit04"
6. **event.status**: The event status, listed in the righthand column of the event status window
    - ♦ Example: event.status == "ZCL: ReportEventStatus"
7. **event.type**: The type of event, shown in the Type column of the Event and Transaction Panes.
    - ♦ Example: event.type == "Packet"
8. **event.corrupt**: The event corruption string, empty if event is not corrupt
    - ♦ Example: event.corrupt < "crypt"

25.1.2 Transaction Extensions

1. **transaction.summary**: Filters on the transaction summary field shown in the Summary column of the Transaction Pane.
    - ♦ Example: transaction.summary == "ZCL: LoadControlEvent"
2. **transaction.packetCount**: Filters on the number of packets in the transaction shown in the P# column of the Transaction Pane
    - ♦ Example: transaction.packetCount == 4
3. **transaction.macRetries**: Filters on the number of MAC retries in the transaction shown in the M# column in the Transaction Pane.
    - ♦ Example: transaction.macRetries == 2

4. **transaction.endToEndRetries**: Filters on the number of end to end retries shown in the E# column in the Transaction Pane.
    - Example: transaction.endToEndRetries == 3
5. **transaction.status**: Filters on the status of the transaction shown in the Status column of the Transaction Pane.
    - Example: transaction.status == "CRC failed"
6. **transaction.dest**: Filters on the network destination of the transaction shown in the NWK Dest column of the Transaction Pane.
    - Example: transaction.dest == 0x05c7
7. **transaction.source**: Filters on the network source of the transaction shown in the NWK Src column of the Transaction Pane.
    - Example: transaction.source == 0x0000

25.2 How Simplicity Studio applies a filter

When Simplicity Studio captures an over-the-air message, it runs the message through a processing stream. The processing stream is made up primarily of Decoders and Groupers.

Decoders: Decoders are responsible for making sense of the message based on its format so that it may be displayed to the user. Each over-the-air message captured becomes a single Event of type Packet. This Event is displayed to the user in the Event view.

Groupers: The groupers are responsible for making sense of a series of packets and grouping them into a hierarchy under a single Transaction. The Transaction is displayed in the Transaction view.

25.3 Using *show(expression, SELF|PARENT|CHILD|SIBLING)*

Events exist within a hierarchical structure where Transactions represent the top of the hierarchy and Events are at the bottom. The hierarchical nature of trace data creates something of a problem for filtering. In most cases, you wish to see Transactions associated with filter-matching Events, and vice versa.

For instance, if you use a filter like: "transaction.summary == Association", you probably do not want to see only the transactions in the Transaction Pane. You probably also want to see the events contained within the Association displayed in the Event Pane.

You can solve this problem by using the optional *show(expr, args)* syntax in your filters.

The show syntax allows you to explicitly indicate the conditions under which an event or transaction should match your filter. The arguments for the show syntax are as follows:

SELF

The Event or Transaction matches if it contains data that matches the expression provided.

PARENT

A Transaction should be shown if any one of its child Events matches the filter.

CHILD

An Event should be shown if the Transaction to which it belongs matches the filter.

SIBLING

An Event matches if its PARENT transaction contains another Event which itself matches the filter.

25.4 Filter display defaults

Filter expressions that do not explicitly contain the optional *show(expr, args)* syntax are implemented as though they contain one of two default syntaxes. Which default syntax is used depends on where the filter is executed, in the Stream Editor or in the Large File Editor.

25.4.1 Stream Editor default: show(*expression*, SELF|PARENT|CHILD)

By default, filter expressions that do not explicitly contain the *show(expr, args)* syntax are implemented as though they were wrapped in the following syntax:

show(*expression*, SELF|PARENT|CHILD)

The SELF|PARENT|CHILD arguments provide what Ember believes a user expects to see when a filter is run in the Stream Editor. The filter display includes the events and transactions that match the filter itself. If an Event matches the filter, you also see an associated Transaction (PARENT) regardless of whether that Transaction matches the filter. Likewise, if a Transaction matches the filter, you also see its associated Events (CHILD) regardless of whether those Events match the filter.

25.4.2 Large File Editor default: show(*expression*, SELF)

By default, the search mechanism in the Large File Editor returns only those Events and Transactions which themselves match the filter expression provided. This behavior makes it very easy to run a filter to search for all the Transactions with a given summary without having the search results bogged down with hits for their associated Events.

25.5 Expression Validation

25.5.1 Lexical validation

When you enter a filter expression, the filter engine validates whether the expression is lexically correct. If an expression is not lexically correct, Simplicity Studio gives you an error message ith a suggestion about where there may be a problem in the expression.

25.5.2 Event Key validation

When you enter a lexically correct expression, Simplicity Studio also runs an event key validation. It checks that any identifier provided represents a real entity within an Event or Transaction. If the filter engine is not able to find any associated data for an event key within the expression, Simplicity Studio will warn you that you are using an unverified identifier.

For example, in the expression

fifteenFour.dest == 0xffff

fifteenFour.dest is a verifiable event key in that Simplicity Studio knows that it represents real data in an event.

Here are two examples to illustrate the validation of expressions in Simplicity Studio. Consider this expression:

foo == bar

The filter mechanism has no way of knowing what **foo** and **bar** represent, or that they even represent any type of data within an Event or Transaction. While this expression is lexically correct, Simplicity Studio will warn the user that foo and bar could not be verified, and that the expression may provide unexpected results. In fact, this expression will not show anything, since foo may very well equal bar, but the filter engine has no way of knowing that.

Consider also this expression:

foo == foo

This expression also displays a warning. However, when run, it will return ALL events, because while Simplicity Studio does not know what foo is, it knows it definitely equals foo.

25.5.3 Special identifiers

There are several special identifiers, which are not mapped to event key.

- **payload.xxx** which evaluates into the payload bytes for a given layer xxx, for example **payload.raw** or **payload.tcp_stream**. You can form expressions like: **payload.raw == {001122aabbcc}** to match payloads
- **flag.xxx** which evaluates into the value of the event flag for a given event, for example: **flag.neighbor_exchange** or **flag.fragment**

26 Managing Events

Simplicity Studio provides a set of facilities for managing events. You can extract data from captured traces and import data from other sources.

For more information, see:

- Creating a file
- Creating or changing a file's description
- Importing events
- Exporting bulk events
- Appending and extracting individual events
- Converting a file
- Reloading a file

27 Creating a File in Simplicity Studio

You can create two types of files in Simplicity Studio:

- capture files, which have the .isd extension
- app configuration files, which are used by the Ember AppBuilder

27.1 Creating a capture file

The process of capturing data from a network results in the creation of a capture file. Creating a capture file this way is described in detail in the section: Capturing Data and Managing a Session.

You can also create a capture file by choosing **File | New | New Live Capture Session**.

27.2 Creating an AppBuilder configuration file

The Ember AppBuilder is integrated into Simplicity Studio. You can access the AppBuilder by choosing **File | New | ZigBee Application Configuration**.

This will open a new application configuration file inside Simplicity Studio. For complete information on how to use the Ember AppBuilder, consult the Ember AppBuilder documentation.

28 Creating or Changing a File Description

Each capture file created by Simplicity Studio includes a short file description. You can create or change a loaded file's description by clicking the **Description** toolbar button 📄 or choosing **Edit | Edit description of trace file**.

This will open a dialog that has a text box where you can enter the information you wish to include with your file.

A preference at the bottom of the description dialog allows the file description to be shown whenever the file is loaded. If this preference is selected, the file description will pop up in the description dialog when the file is loaded into Simplicity Studio.

28.1 Importing events

You can import events only into a live window, not into the window opened from a file.

To import events, choose **File | Import**. Then choose either:

- **Simplicity Studio file**: Imports events from an existing Simplicity Studio file, ignoring any metadata present in the file besides the events.
- **Plain text file**: Imports events from plain text log file.

28.2 Exporting events

You can export events for processing with a different tool using one of three options.

To export an event, choose **File | Export**. Then select either:

- **Simple text event log**: Exports the events into a simple text log, which shows the binary data received from the backchannel and the timestamps with each event. This file can be read back into Simplicity Studio.
- **Transactions comma-separated file**: Exports the data as shown in the Transaction pane. All the columns will be exported as a comma-separated text file. You can import this file into any statistical program or spreadsheet for analysis.
- **Events comma-separated file**: Exports the data as shown in the Events pane. All the columns will be exported as a comma-separated text file. You can import this file into any statistical program or spreadsheet for analysis.

28.3 Appending events

To extract specific events to a text file:

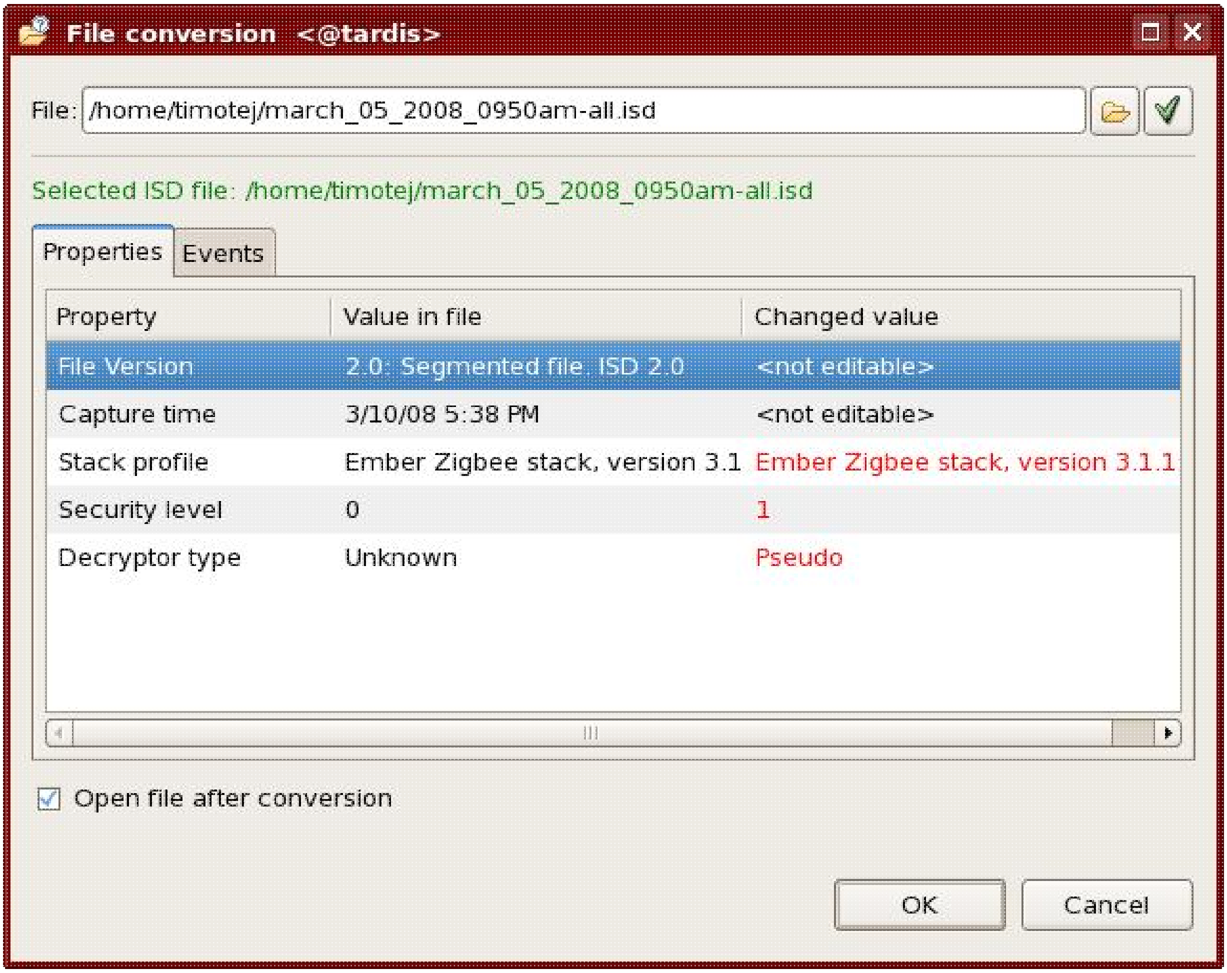
1. Right-click the event that you want to extract from the Transactions, Events, or Hex Dump pane .
2. Select **Extract to** from the pop-up menu.
3. Name the text file.
4. Click **Save**.

Once you specify a file, you can append additional events to it by right-clicking an event and selecting **Append to file**.

28.4 File Conversion

File Conversion allows you to modify metadata inside the Ember Desktop file. Metadata includes the stack version used during file capture, security level, and similar data.

Additionally, file conversion can be used to convert files from Simplicity Studio versions prior to 2.0 into the 2.0 file format. This is not a requirement, as 2.0 will be able to read older formats as well, but in that case the new large-file features can not be used.



File: /home/timotej/march_05_2008_0950am-all.isd

Selected ISD file: /home/timotej/march_05_2008_0950am-all.isd

Property	Value in file	Changed value
File Version	2.0: Segmented file. ISD 2.0	<not editable>
Capture time	3/10/08 5:38 PM	<not editable>
Stack profile	Ember Zigbee stack, version 3.1	Ember Zigbee stack, version 3.1.1
Security level	0	1
Decryptor type	Unknown	Pseudo

☑ Open file after conversion

To convert a file, choose **File | Convert File** and select the file to convert. Once you select the file, the bottom part of the dialog shows a table with the current values in the file on the left, and modified values on the right. You cannot edit all of the values, but the ones you can edit you can do so via the table.

Once you have modified the file, you should choose whether to set the **Open file after conversion** option. If you check this option, the new values will be used while opening the file. The values will not be saved, however, until you save the file. If you do not check this option, the file will be converted in the background and the original file will be modified.

29 Reloading a File

You can reload a file that is already loaded in a Stream Editor. This feature is useful, for example, when you have added a decryption key or a new decoder, and want to apply that newly added information to the decryption and decoding of the current file.

To reload a file, choose **File | Reload File**, or click the **Reload** button () in a toolbar. This will cause the entire trace to be re-run through the decryption and decoding mechanism, loaded into Simplicity Studio, and shown in the same Stream Editor previously displaying the file.

30 Simplicity Studio's Views

Simplicity Studio provides access to a wide range of functionality through the use of views. The views are all accessible in the menu **Window | Show View**. For a complete listing of all available Views, choose **Window | Show View | Other...**

30.1 Error Log View

The Error Log View shows in a tabular format all of the errors that Simplicity Studio has generated and captured. The navigation at the top of the view allows you to perform basic functions on the Error log itself, including **Export**, **Import**, **Clear**, **Delete**, **Open**, and **Restore**.

30.1.1 Help View

When you choose **Help | Help contents**, the help documentation opens in a popup browser outside of Simplicity Studio. Some users may prefer accessing help through a separate browser, while others may want help to be located within Simplicity Studio.

The Help View is accessible through Help | Dynamic Help. The Help View provides access to all of the Simplicity Studio help documentation in a View within Simplicity Studio, as opposed to a standalone popup browser. This view is useful if you wish to keep the help documentation open as a reference while working within Simplicity Studio.

30.1.2 Adapters View

The Adapters View is used to interact with Debug Adapters connected to Simplicity Studio over the backchannel. See Adapters View for more information.

30.1.3 Event Detail View

Event details are normally shown in the EventDetail Pane within the Stream Editor. If you want to see the event details in a separate window, you can open the Event Detail View. This view shows the details for the currently selected Event in the same format as the Event Detail Pane, but in a view that you can pull outside of Simplicity Studio and resize to your liking.

30.1.4 Hex Dump View

Similarly to event details, the hex dump information is shown the Hex Dump Pane within the Stream Editor. If you want to see the hex dump in a separate window, you can open the Hex Dump View. This view shows the hex dump of the currently selected Event in the same format as the Hex Dump Pane, but in a view that you can pull outside of Simplicity Studio and resize to your liking.

30.1.5 Event Difference View

Event Difference view is a helper view that displays the specific differences between two packets.

30.1.6 Filter Manager View

The Filter Manager View is used to compose and edit custom filtering expressions. See Filter Manager for more information.

30.1.7 Progress View

The Progress View shows the progress of user actions. Actions that take a long time to execute may be managed within the Progress View. For example, the application upload action can take several seconds. The

Progress View provides a user interface for managing this action. If you wish to stop an action, you may do so in the Progress View.

30.1.8 Radio Info View

The Radio Info View shows data captured for each event. See Radio Info View for more information.

30.1.9 Search View

Search data is displayed in the Search Pane within the Large File Editor. However, as with the Event Details and Hex Dump Panes, you may wish to see this data in a separate view. The Search View allows you to view search results in a window that you can pull outside of Simplicity Studio and resize to your liking.

30.1.10 USB Link View

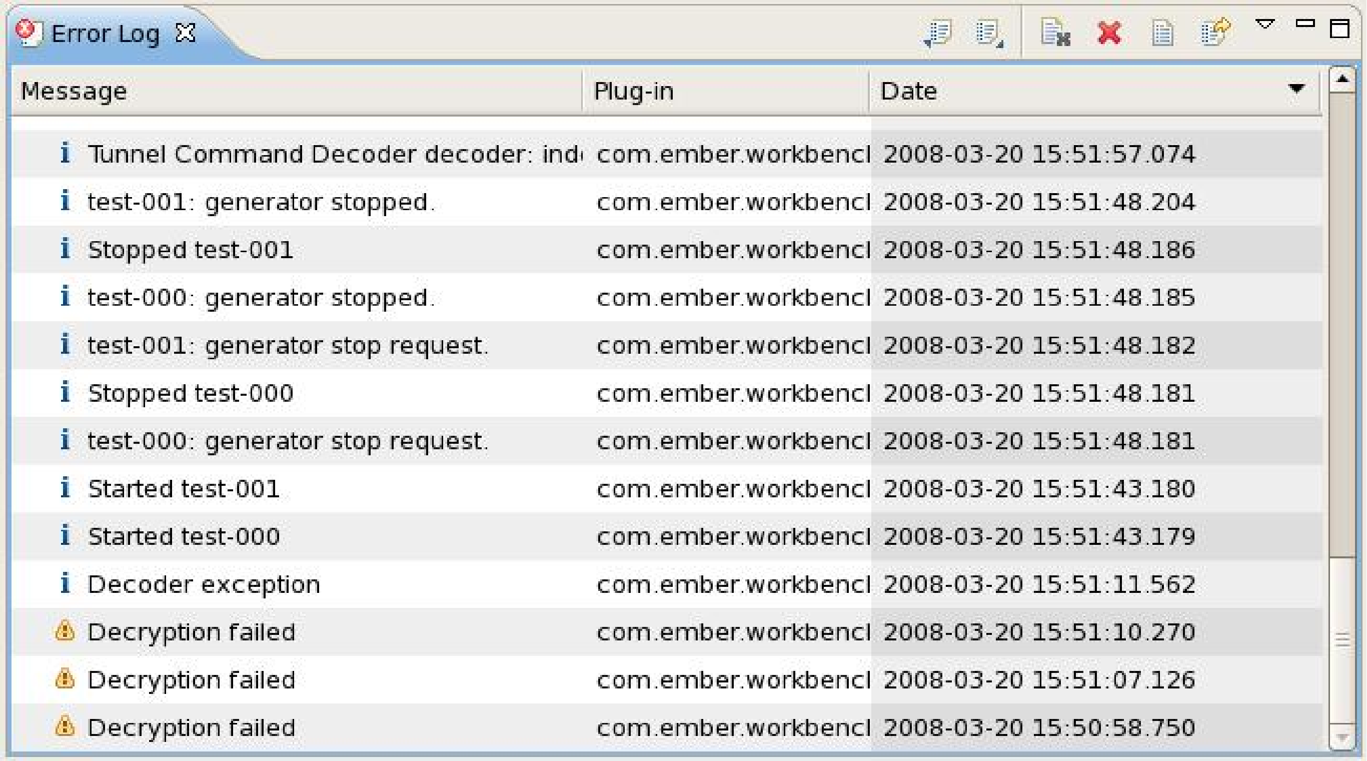
The USB Link View provides discovery and access to all USB Link devices connected to your computer

30.1.11 Shell View

The Shell View provides command line access to the scripting capabilities of Simplicity Studio. For a list of all commands available within the shell, simply enter **help**at the command line.

31 Error Logging in Simplicity Studio

Decoder, decryption, and other types of errors are displayed in the Error Log View. To open the Error Log View, choose **Window | Show View | Error Log**.

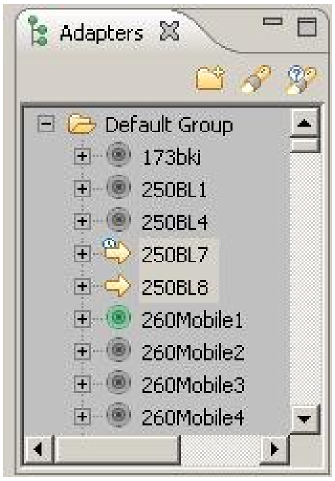


Message	Plug-in	Date
ℹ Tunnel Command Decoder decoder: ind	com.ember.workbencl	2008-03-20 15:51:57.074
ℹ test-001: generator stopped.	com.ember.workbencl	2008-03-20 15:51:48.204
ℹ Stopped test-001	com.ember.workbencl	2008-03-20 15:51:48.186
ℹ test-000: generator stopped.	com.ember.workbencl	2008-03-20 15:51:48.185
ℹ test-001: generator stop request.	com.ember.workbencl	2008-03-20 15:51:48.182
ℹ Stopped test-000	com.ember.workbencl	2008-03-20 15:51:48.181
ℹ test-000: generator stop request.	com.ember.workbencl	2008-03-20 15:51:48.181
ℹ Started test-001	com.ember.workbencl	2008-03-20 15:51:43.180
ℹ Started test-000	com.ember.workbencl	2008-03-20 15:51:43.179
ℹ Decoder exception	com.ember.workbencl	2008-03-20 15:51:11.562
⚠ Decryption failed	com.ember.workbencl	2008-03-20 15:51:10.270
⚠ Decryption failed	com.ember.workbencl	2008-03-20 15:51:07.126
⚠ Decryption failed	com.ember.workbencl	2008-03-20 15:50:58.750

The Error Log consists primarily of a table of errors reported by Simplicity Studio. Each error is shown in a single row with its summary message, the plugin that reported the error, and the date and time the error was encountered. To view detailed information about the error, double-click it. The error will be displayed in an Event Details dialog that includes the date, severity, message, and, if available, stack trace.

31.1 Adapters view

On startup, Simplicity Studio automatically discovers and lists all Debug Adapters that are connected to the local subnet. You can also configure Simplicity Studio to discover Debug Adapters on other subnets. To force Simplicity Studio to discover adapters, click the Discovery icon.



31.1.1 Default and custom groups

When you first start Simplicity Studio, it displays in a single default group all Debug Adapters that are discovered by Simplicity Studio. You can create custom groups and move Debug Adapters to them.

Note: Custom groups simply provide a way to organize adapters and have no functional impact on adapters or their interaction.

31.1.2 Adapter details

You can expand each Debug Adapter to display the following details about it:

- **Node type** identifies the Ember radio chip. (Note: The adapter requires that the node have been reset for it to know what type of node it is working with. In some cases, when the node has not been reset, the node type will say: *Not set*.)
- **Last bootload** identifies the last bootload to this adapter's node from your machine. If another user uploaded an application to the node, this information may not actually reflect the application that is currently loaded on the node itself. If Simplicity Studio is unable to identify the application for some other reason, this field displays *Unknown*.
- **Node EUI** identifies the Extended Unique Identifier.
- **Hostname** and **IP address** identify this Debug Adapter.
- **Connection status**, **Connection address**, and **Connection time** show the current connection information about this Debug Adapter.
- **Board type** and **Board name** identify this adapter's platform.
- **Hardware type** and **Hardware version** identify the adapter hardware.
- **Software version** shows which version of Debug Adapter is running on this adapter.

31.1.3 Using multiple adapters

Most commands can be issued to multiple adapters simultaneously. You can select multiple adapters in one of two ways:

- Select individual adapters:
    - For contiguous selections, Shift+Click
    - For non-contiguous selections, Ctrl+Click
- Select one or more adapter groups to select all adapters within those groups.

31.1.4 Adapter Properties

Adapter properties can be changed from the view interface by right clicking on the adapter and selecting properties.

31.1.4.1 Adapter name:

The name of the adapter shown in the Adapters View

31.1.4.2 Adapter type:

The type of adapter connected.

31.1.4.3 Adapter version:

The version of the Adapter firmware loaded on the adapter.

31.1.4.4 Radio processor:

The Ember radio processor to which the adapter is connected.

31.1.4.5 Radio processor application:

The firmware running on the radio process.

31.1.4.6 Host processor:

The host processor to which the adapter is connected (in the case of the radio processor being an NCP above.)

31.1.4.7 Host processor application:

The firmware running on the host processor.

31.1.4.8 Show raw discovery data:

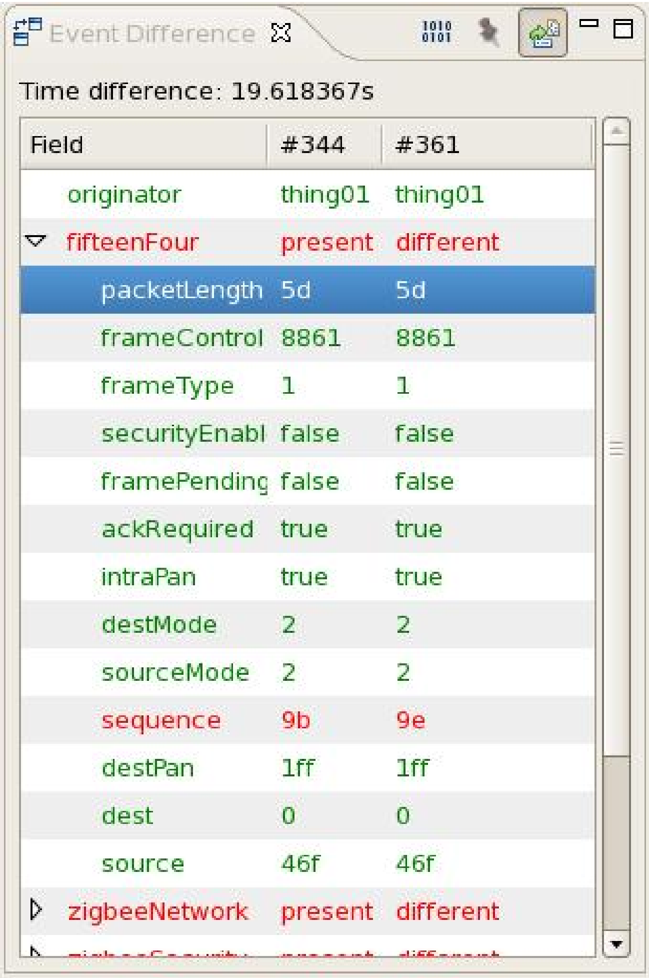
Pops up a dialog showing what data was returned to the discovery command sent out from Simplicity Studio in order to discover this adapter on the network.

31.1.4.9 Protect from automatic discovery changes:

Every time the adapter discovery process is performed, all adapters in the adapter search parameters will be updated with the information they return to the discovery query. You may use this checkbox to keep an adapter entry from being updated with the information returned during discovery. This is useful in the case where the adapter discovery is not able to tell you enough information about the adapter's settings. For instance, if you are using an EM357 with EZSP and a host micro, there is currently no way for the adapter to know that you have your EM357 configured as an NCP or which type of Host micro it is attached to. You can configure this information in the adapter preferences dialog by entering the information above. You may also persist the information you enter by selecting this check box. When this box is checked, a "[p]" symbol will be shown next to the adapter name in the view to remind you that this adapter is being protected from discovery changes.

32 Event Difference View

Event Difference view is a helper view that displays the specific differences between two packets.



Time difference: 19.618367s

Field	#344	#361
originator	thing01	thing01
fifteenFour	present	different
packetLength	5d	5d
frameControl	8861	8861
frameType	1	1
securityEnabl	false	false
framePending	false	false
ackRequired	true	true
intraPan	true	true
destMode	2	2
sourceMode	2	2
sequence	9b	9e
destPan	1ff	1ff
dest	0	0
source	46f	46f
zigbeeNetwork	present	different
zigbeeSecurity	present	different

To open the Event Difference view, click the **diff view** toolbar button (), or choose **Window | Show view | Event difference**.

Once diff view is shown, it tracks the selected events. Diff view will by default show the difference between the last two events selected. If you select event 1, and then click on event 2, diff view will show the diff between those two events. If later you select event 3, view will show the differences between event 2 and 3.

Packet frames that do not have any differences are shown in green. Frames that contain differences are shown in red. To see which portions of the frames are different, expand the tree by clicking on the plus marker to the left of the frame's name.

There are few additional controls on top:

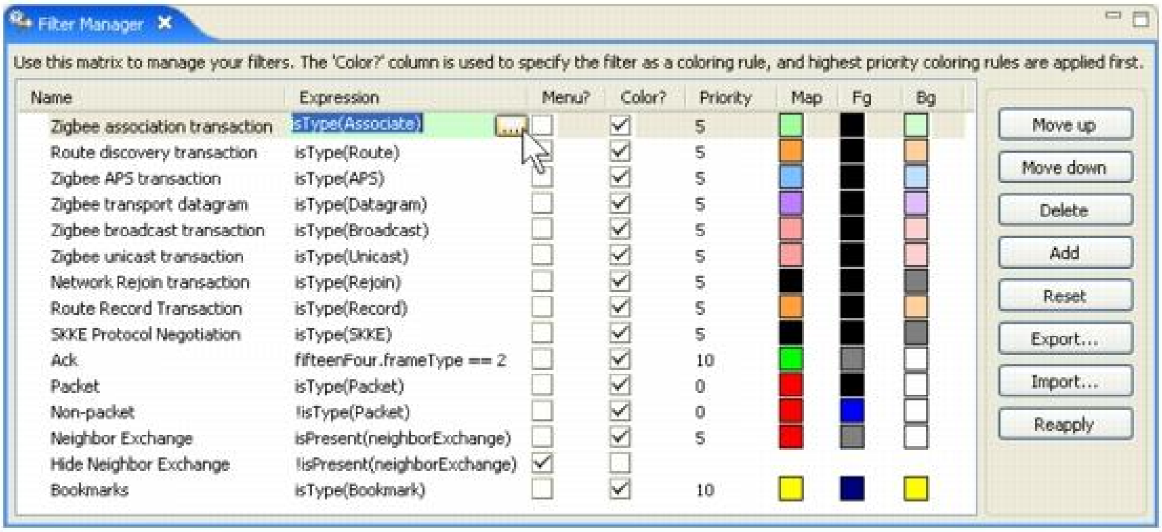
- **Show byte differences** () enables viewing of individual bytes in diff view.
- **Pin last selected event** () changes the way events are tracked. If this is enabled, then the first event for diffing stays the same, and only the second event changes. You can use this if you wish to always differentiate events against a certain static event, rather than always viewing last two selected events.
- **Show fields that are same** () enables filtering out fields that are same in both events.

32.1 Filter Manager

The Filter Manager lets you perform the following tasks:

- Compose filter expressions with the Expression Builder.
- Use filter expressions to customize display of entries in the Events pane.
- Specify which filters appear in the **Filters** menu.

To show the Filter Manager: Choose the menu option **Filters | Filter Manager**.



Use this matrix to manage your filters. The 'Color?' column is used to specify the filter as a coloring rule, and highest priority coloring rules are applied first.

Name	Expression	Menu?	Color?	Priority	Map	Fg	Bg
Zigbee association transaction	isType(Associate)	□	☑	5	🟩	⬛	🟩
Route discovery transaction	isType(Route)	□	☑	5	🟧	⬛	🟧
Zigbee APS transaction	isType(APS)	□	☑	5	🟦	⬛	🟦
Zigbee transport datagram	isType(Datagram)	□	☑	5	🟪	⬛	🟪
Zigbee broadcast transaction	isType(Broadcast)	□	☑	5	🟥	⬛	🟥
Zigbee unicast transaction	isType(Unicast)	□	☑	5	🟥	⬛	🟥
Network Rejoin transaction	isType(Rejoin)	□	☑	5	⬛	⬛	◼
Route Record Transaction	isType(Record)	□	☑	5	🟧	⬛	🟧
SKKE Protocol Negotiation	isType(SKKE)	□	☑	5	⬛	⬛	◼
Ack	fifteenFour.frameType == 2	□	☑	10	🟩	◼	⬜
Packet	isType(Packet)	□	☑	0	🟥	⬛	⬜
Non-packet	!isType(Packet)	□	☑	0	🟥	🟦	⬜
Neighbor Exchange	isPresent(neighborExchange)	□	☑	5	🟥	◼	⬜
Hide Neighbor Exchange	!isPresent(neighborExchange)	☑	□				
Bookmarks	isType(Bookmark)	□	☑	10	🟨	🟦	🟨

32.1.1 Building filter expressions

The Filter Manager provides an Expression Builder that provides a built-in set of valid operands and operators that you can use to compose filter expressions. You can also enter expressions manually. In both cases, Expression Builder validates entries as you work and prevents you from setting invalid expressions.

To use Expression Builder:

1. Select an **Expression** from the Filter Manager list.
2. Click the Expression Builder control, ... to display the Expression Builder.
3. Choose the required operands, operators, and values. You can use logical operators such as && and || to specify multiple conditions, and parentheses ( ) to group expressions and set precedence.
4. When your expression is complete, click **OK** from the Expression Builder.

32.1.2 Maintaining filters

To add a filter to the Filters menu: Set the **Menu** checkbox. This makes the filter available to the current sessions.

To restore filters to installation settings: Click **Reset** on the Filter Manager.

You can also add, delete, and rename saved filters.

32.1.3 Setting filter color schemes

You can associate a color scheme with each filter. If an event evaluates as true for a given filter expression, Simplicity Studio applies the filter's color scheme to that event. An event can be configured with a color scheme for two display levels:

- For the Map pane, the color used by the graphic representing the event type.
- For the Events pane, the foreground and background colors used by event type instances.

> **Note:** Setting a color scheme on a filter expression does not specify whether to display events; it only determines how to display certain event types.

> **Note:** If a filter's foreground color is the same as its background color you will not be able to read the text in the event and it will effectively disappear from view.

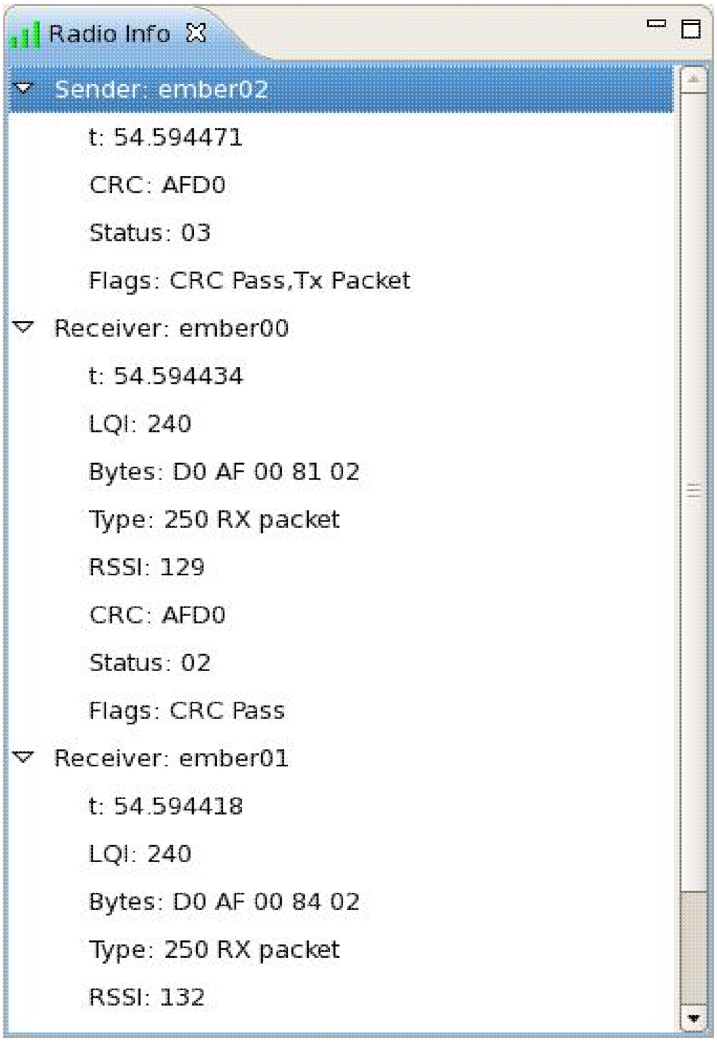
To set a filter color scheme:

1. From the Filter Manager's **Map**, **Fg**, or **Bg** columns, click the filter's checkbox.
2. Set the filter's color of the Map pane graphic (**Map**), and foreground (**Fg**) and background (**Bg**) colors of event instances, by clicking the control. ...
3. To set the filter's **Color** checkbox to determine whether to use this color scheme.
4. For each filter, set the color scheme's **Priority** level by assigning a positive or negative integer value. If an event evaluates as true for multiple filters, Simplicity Studio uses the color scheme with the highest precedence. In the case where there is no clear precedence, Simplicity Studio randomly chooses one of the matching color schemes.
5. To refresh the Map pane and Events pane with the new color scheme, click **Reapply** on the Filter Manager.
6. Filters whose menu checkbox is selected will be shown in the top filter navigation. The order in which the filters are displayed in the Filters menu is determined by their order in the filter manager table. Consequently, you may move filters up and down the list as you wish them to be displayed in the filter menu.

33 Radio Info View

The Radio Info view is a helper view that shows the information from the radio of all the receivers in the network that have heard the currently selected event.

To open the Radio Info view, choose **Window | Show View | Radio Info**. The view is positioned by default in the bottom-left corner of the screen and will display in a tree all the information that has been gathered from the receiver nodes. Displayed information will include LQI value, CRC value, and the status bits that show several states of the radio.



```
▽ Sender: ember02
        t: 54.594471
        CRC: AFD0
        Status: 03
        Flags: CRC Pass,Tx Packet
▽ Receiver: ember00
        t: 54.594434
        LQI: 240
        Bytes: D0 AF 00 81 02
        Type: 250 RX packet
        RSSI: 129
        CRC: AFD0
        Status: 02
        Flags: CRC Pass
▽ Receiver: ember01
        t: 54.594418
        LQI: 240
        Bytes: D0 AF 00 84 02
        Type: 250 RX packet
        RSSI: 132
```

The event that supplied radio information for the preceding pane was captured from both the sending and receiving nodes. This is possible because the trace that contains this event was created by capturing from both the sending and receiving nodes simultaneously using Simplicity Studio's Perfect Trace capability. While the original events were merged into a single event by Simplicity Studio's duplicate detection mechanism, the radio information was retained for each event and is shown in the Radio Info view with the time that the event was captured by Simplicity Studio.

34 Using USB Link Devices

Simplicity Studio can discover and interact with USB Link devices that are connected to your computer. USB Link devices are shown in the USB Link View. To open the USBLink View, choose **Window | Show View | USB Link View**.

Simplicity Studio discovers USB Link devices connected to your computer automatically. If you do not see your USB Link device in the USB Link View, you can manually run the discovery process by clicking the **Discovery** icon in the top right corner of the USB Link View. Simplicity Studio will list each USB Link device discovered and the id and status for each device.

Loading software using the USB Link
To load a binary image onto your device using the USB Link, select the USB Link you wish to use and click the **Load** icon (). This will launch the Binary Image Chooser dialog. Choose or browse to the image you wish to load onto your device, and click **OK**. This will begin the upload process. The status of the upload process is shown beside the USB Link in the USB Link View.

Reading a binary image using the USB Link
To read a binary image off of a device connected to a USB Link, select the USB Link you wish to use for the read process and click the **Read** icon (). This will launch a File dialog that allows you to save the binary image as an ASCII .hex file.

Changing the ID of your USB Link
By default, every USB Link has the ID 0. If you want to use multiple USB Links, you should change their IDs to be unique. Click the **Change ID** icon () to launch a dialog that allows you to enter a new ID for the selected USB Link.

Identifying a USB Link
You may often wish to know which USB Link represented in Simplicity Studio is connected to which physical device. You can identify a USB Link by making it blink its LED. When you click the **Blink** icon (), the USB Link that is selected in the USB Link View will flash its LED amber for few seconds.

For more information on USB Link devices, see the USB Link documentation provided with your USB Link product.

35 Customizing the Simplicity Studio Interface

You can customize the look and feel of Simplicity Studio to suit your work habits and preferences. Customization features can also help you work more effectively. For example, you can reposition the Adapters view, or resize editor panes.

For more information, see:

- Creating Debug Adapter groups
- Reconfiguring views
- Customizing editor panes

To restore the startup configuration, choose **Window | Reset perspective**.

35.1 Creating Debug Adapter groups

When newly installed, Simplicity Studio shows all Debug Adapters within a single default group. You can group adapters logically under folders that you create. For example, your application might have different types of sensor nodes, and various sink nodes for each type. In that case, it makes sense to group the adapters of similar nodes under a folder that you name. This can help you remember which applications are running on those nodes.

You can also use custom groups to issue commands and upload applications to multiple adapters and their nodes. Any command to the group is broadcast to all adapters within it. Similarly, an application that you upload to a group is uploaded to all its adapters.

35.1.1 Creating a custom group

1. From the Adapters toolbar, click the Add Group icon.

   Simplicity Studio creates a new folder named *New Group*.
2. If desired, rename the **New Group** folder:

   Select the folder name, type the folder's new name, then press **Enter**.
3. Click a node to be moved and drag it to the new folder. For multiple selections, use Shift-Click and Ctrl-Click for contiguous and non-contiguous selections, respectively.

35.1.2 Removing a group

1. In the Adapters view, right-click the group folder you want to remove.
2. From the pop-up menu, choose **Delete**.

Simplicity Studio removes the group folder and moves its adapters to the default group.

35.2 Reconfiguring views

The Simplicity Studio work environment has several views. The key views include:

- Adapters
- Filter Manager
- Progress
- RF Evaluation Tests

When you first open Simplicity Studio, it positions these views as follows:

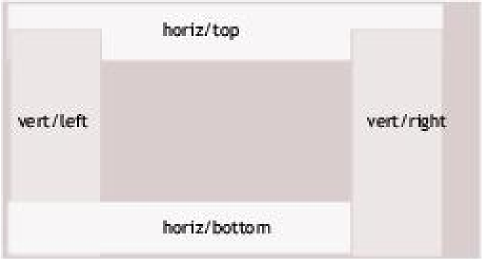
- Adapters view is on the left with Progress view behind it.

- Filter Manager is hidden. When Filter Manager is invoked, Simplicity Studio places it on the bottom right.

35.2.1 Repositioning a view

To reposition a view, click its title bar and drag it to the desired position.

You can reposition a view anywhere on the working area's perimeter that is not occupied by another view. Four positions are possible, as shown in the following figure:



35.2.2 Resizing and hiding views

Each view has the following controls:

- **Maximize** expands the view to encompass the entire working area. You can also maximize the view by double-clicking the view's titlebar.
- **Minimize** shrinks the view to a narrow strip.
- **Close** removes the view from the working area. You can restore a closed view by choosing **Window | Show view**.
- **Restore** resets the view to its dimensions before it was minimized or maximized.

To manually resize a view: Drag any inside border in or out.

35.2.3 Using fast views

You can designate the Adapters and Filter Manager views as fast views. When you do so, Simplicity Studio anchors the view to a **Fast View** icon in the lower right that can be toggled to display or minimize the associated view.

Note: While the Adapters and Filter Manager views can both be designated as fast views, only one can be active as a fast view at any given time.

To create a fast view:

1. Right-click the view's title bar.
2. From the pop-up menu, choose **Fast View**. Simplicity Studio minimizes the view and associates it with a fast-view icon that displays on the Fast View bar in the lower right.

Alternatively, drag the view to the Fast View bar.

When you restore a fast view, it restores its previous orientation--vertical or horizontal. To change the current orientation, right-click the view's title bar and choose an **Orientation** option.

35.2.4 Using tear-off views

Any view can be removed from the main Simplicity Studio window so that it floats freely.

Note: A view can be either a tear-off or a fast view, but not both.
To create a tear-off view:

1. Right-click the view's title bar.
2. From the pop-up menu, choose **Detach**. Simplicity Studio detaches the view from the main window.

35.3 Customizing editor panes

The relative positions of editor panes are fixed. However, you can change their dimensions in two ways:

- Resize an editor pane by dragging an inside border.
- Click the editor pane's resize control in the upper left corner. Depending on the pane's current state, this control toggles between **Maximize** and **Restore**.

Simplicity Studio also provides options that let you change the appearance of the editor panes:

- Map pane options
- Node display options
- Transactions and Events pane options
- Bookmarks

35.3.1 Map pane options

The following Edit menu options let you customize the Map pane appearance and behavior. Some of the options toggle identifying features of nodes.

- **Edit | Show Short ID** () toggles display of the node's 16-bit address that is unique within the personal area network (PAN).
- **Edit | Show EUI64** () toggles display of the node's unique 64-bit IEEE address.
- **Edit | Show PAN ID** () toggles display of the PAN identifier of the node's network. This label can be useful when the map displays multiple networks.
- **Edit | Show Node Label** () displays the custom label that you create for map display only.
- **Edit | Show LQI** () toggles display of link quality data that pertains to the quality of connection between nodes. This is available with perfect trace captures, but not with sniffer captures.
- **Edit | Show Connectivity** () shows the neighbor relationships between nodes in the network.
- **Edit | Simultaneous Events** () displays on the Map pane all events that occurred at the same time as the transaction or event that is currently selected. The currently selected event is in color and any other events display in gray.
- **Edit | Zoom Map In** () and **Edit | Zoom Map Out** () enlarge and shrink the space that the map uses to display nodes. Zoom options have no effect on the size of node icons.
- **Edit | Organize Map** establishes the layout of all nodes on a map. You can also modify individual node positions as needed. The following layouts are available:
  - ♦ **Default Placement** aligns nodes in a linear pattern.
  - ♦ **Random Placement** scatters the nodes across the map randomly.
  - ♦ **Concentric Circles** aligns the nodes in arcs.
  - ♦ **Square Grid** aligns the nodes in a grid.
  - ♦ **Hexagonal Grid** aligns the nodes in a hexagonal, offset pattern.
- **Edit | Load Background Image** and **Edit | Clear Background Image** manage the display of a background image in the Map pane.

35.3.2 Node display options

You can move node icons within the Map pane display. This has no effect on network functionality. However, it can help to highlight certain node interactions and relationships. When you move node icons in a session, Simplicity Studio asks whether to save those changes before you close the session.

When you right-click a node in the Map editor pane, the following display options are available from its pop-up menu:

- **Organize Map** establishes the layout of all nodes on a map as described above.
- **Assign EUI64** lets you assign a EUI64 to a node. Ember Desktop obtains a node's EUI64 only when that node associates with a network. If the node already belongs to a network when a session begins, its EUI64 is unknown to Simplicity Studio. This option lets you display a known EUI64 for a node; the node's actual EUI64 is unaffected by this label.

  If Simplicity Studio obtains a node's EUI64, this option is unavailable.
- **Label** lets you customize the node's adapter (device) label with any string up to 25 characters long. This string appears in brackets after the node's device name. (By default, the Map pane labels each node that is undergoing capture with its device name.)

  You can also make the labels time-dependent by entering a start time. This lets you supply multiple names for the same node. This can be useful while debugging applications, by indicating the node's current state.
- **Icon** provides a menu of defined icons to represent the selected node.
- **More Icons** invokes the Node Icons dialog, which allows you to add custom icons for nodes.

35.3.3 Transactions and Events pane options

The following Edit menu options let you customize the appearance and behavior of the Transactions pane and Events pane:

- **Decrease Font Size** (T) and **Increase Font Size** (T) change the font size for the Transactions, Events, and Event Detail panes.
- **Apply Row Coloring** (🌈) toggles the display on or off of filter color definitions for the Transactions and Events panes.

35.3.4 Bookmarks

You can set a bookmark on any transaction or event as follows:

1. Right-click the event.
2. From the pop-up menu, choose **Add bookmark**.
3. In the Add Bookmark dialog, enter a name and time for the bookmark:
   - ♦ The bookmark name can be any string.
   - ♦ The bookmark's time initially is set automatically just before the selected event. You can change this to any time that precedes or follows existing events or bookmarks.

36 Setting Simplicity Studio Preferences

To access Simplicity Studio's preferences, choose **File | Preferences**. The preferences you set apply to all capture sessions. Simplicity Studio saves your preferences and uses them each time it restarts. Preference categories include:

- AppBuilder
- Application Upload
- Decoding
- Discovery
- File
- Help
- Auto Update
- Node Icons
- Optional Dialogs
- Serial Adapters
- Stream
- Timeline
- USB Adapters
- Web Browser

36.1 AppBuilder preferences

Choose **File | Preferences | AppBuilder**.

You use the Ember AppBuilder to build ZigBee certifiable sample applications leveraging the Silicon Labs implementation of the ZigBee Cluster Library. The Ember AppBuilder is documented separately. Ember AppBuilder's preference documentation is included with the general Ember AppBuilder documentation.

36.1.1 Application Management

Choose **File | Preferences | Application Management**.

The Ember installation sets the Windows registry with default paths to the application loader and bootloader image. Simplicity Studio uses these paths when it uploads applications to EM2xx nodes. Use this Preferences dialog to specify different paths to the application loader and bootloader image.

The application loader path specifies the loader utility that is used to upload bootloader or application images to EM2xx nodes. Set this field by browsing for another application loader.

The bootloader image path specifies the bootloader application that is uploaded to an EM2xx node when you upload an application image. You can upload the installed bootloader image or you can browse for a different one.

All of the binary images that are known to Simplicity Studio are displayed in the binary images table. The table also displays the chip, board, and type of the image along with its physical location on disk. Sample images that come bundled with Simplicity Studio are marked as <internal>.

36.1.2 Decoding preferences

Choose **File | Preferences | Decoding**.

Note: Changes in decoding preferences will apply to new capture sessions, not to current sessions.

Selected stack: The stack selected for decoding newly captured data.

Note: This setting has no effect on a previously captured trace. The decoding stack for a trace file is stored in the file itself. If you wish to open a previously captured trace file with a stack version that is different from the one it was captured with, use **File | Convert File**. See File Conversion for more information.

Security level: The security level set for the MAC and network layers. Valid values include 1 through 5.

Custom decoders: Specifies which custom decoders to use. If enabled, custom decoders provide an extra layer of decoding for application-level data.

Security keys: Specifies the security decryption keys. To enable a key, select the **Decryption key** checkbox. To edit a key, select its name or key values. All enabled decryption keys are run against each incoming packet until one is successful. Successful keys are automatically moved to the top of the key list to improve performance.

The buttons provide the following features:

- **New** creates a new key for editing.
- **Clone** creates a copy of the currently selected key.
- **Delete** removes the currently selected key.
- **Invert** swaps the order of the key values.
- **Clear All** removes all security keys.
- **Run HMAC** opens up a dialog that allows you to manually calculate the HMAC authenticated key from trust key and IEEE EUI64 address.

Decryption keys can also be obtained from network traffic for future use.

Save decryption keys in Simplicity Studio file: Saves the security keys that you have specified into the capture file along with your traffic.

Note: If you are sharing the Simplicity Studio file with users who have a right to know your key, this may make the opening of the file easier for them, as they will not have to separately enter the key. However, by doing this, you create a security risk. **Never enable this option, if your keys must remain secure and known to you only!**

Stack Versions: A list of profiles representing Ember stack versions. Check the Ember stack that is deployed on the network you are using.

Transaction groupers: A table of all the groupers loaded into Simplicity Studio. Transaction groupers are responsible for making transactional sense out of a trace of network data. Groupers watch for batches of events and record them as a transaction.

36.1.3 Discovery preferences

Choose **File | Preferences | Discovery**, or click the **Discovery Preferences** icon.

Simplicity Studio searches for all Debug Adapters to display according to the settings in the Discovery Preferences dialog. Discovery options you can select include:

- **Scan for adapters on startup of Simplicity Studio**
- **Scan for devices at a specified time interval**
- **Scan local subnet for adapters**
- **Use TCP discovery for complete addresses**

By default, Simplicity Studio scans the local subnet for adapters. However, you can also configure Simplicity Studio to find adapters on other subnets.

Additional discovery subnets: This table can be used to add and configure subnets you wish to scan. Checkboxes allow you to temporarily disable a subnet without deleting it from preferences. You may type a DNS hostname into the address column, and application willresolve the hostname into the IP address.

TCP Discovery: In addition you can configure Simplicity Studio to use a TCP connection for discovery. When this configuration option is set, Simplicity Studio will attempt to form a connection to any discovery subnets that contain a complete address on port 4902. This discovery format should only be used on adapters that you are sure are not being used by any other users. If another user is connected to the adapter on port 4902 they will be kicked off their connection by your discovery process.

36.1.4 File preferences

Choose **File | Preferences | File**.

Monitor files for appending: Controls whether loader should stop at the end of the file, or continue monitoring.

Monitor the file timestamp: Monitors whether the file has changed on the disk, and prompts to reload if so.

Enable large file handling: Enables Simplicity Studio to handle large files in the Large File Editor. Files are considered large if they contain more events than the number entered in the preference titled: "Number of events for a file to be large."

Preserve open files across sessions: Instructs Simplicity Studio to automatically reopen files from previous session

Monitor files for appending after opening: Enables Ember Desktop to detect new packet data appended by an external program to an open session file, and read it in for display.

36.1.5 Help preferences

Choose **File | Preferences | Help** to modify the display options for Simplicity Studio's help.

Specify how help information is displayed: By default help opens in an Simplicity Studio window. Select **Use external browser** to have help open in your system's default external HTML browser.

Open window context help:. Select either **in a dynamic help view** (default) or **in an infopop**.

Open dialog context help: Select either **in a dynamic tray** (default) or **in an infopop**.

Search: Select either **Show all potential hits** (default) or **Show only actual hits**.

Help Server: The Help Server is dynamically computed by the help system. If you need to specify a different help server, enter the host name or IP address and server port here.

36.1.6 Node Icon preferences

Choose **File | Preferences | Node Icons**.

Simplicity Studio provides several predefined icons. The Node Icons dialog displays all icons that are available for customizing display of nodes in the map pane. You can also add icons of your own.

36.1.7 Optional Dialog preferences

Choose **File | Preferences | Optional Dialogs**.

Some dialogs can be turned off if you prefer not to interact with them. You can select the following options:

- **Do not ask me when I want to quit**.
- **Do not alert me of decryption failures**.
- **Do not alert me if I stop capture on time server only**.

You can also disble these dialogs when they display.

36.1.8 Serial Adapter preferences

Choose **File | Preferences | Serial Adapters**.

To use adapters that are connected via a serial cable, you must select them from the Serial Adapters preferences window. For example, Development Kit hardware that has been loaded with the RF Evaluation application and connected via a serial cable must be selected before tests can run.

1. From the list of **Available serial ports**, select a port of a connected development adapter.

   Warning! Before you click **Probe port** in the next step, be sure that your system does not have hardware connected that may be damaged by communication over the serial port.
2. Click **Probe port**. For RF Evaluation, a correct port indicates an application of type *RF Evaluation App*.
3. To use the device, click **Yes** in the pop-up dialog.

   The port's **Status** should now read *Detected: RF Evaluation App*.
4. (Optional) To have Simplicity Studio always recognize the device when it runs, check **Remember recognized devices for next session**.
5. Click **Apply**.
6. Click **OK** to close the **Preferences** window. The **Adapters** view displays the serial port adapter.

36.1.9 Stream preferences

Choose **File | Preferences | Stream**.

Note: Stream preferences will apply only to the new streams opened, not the already opened ones.

You can configure how Simplicity Studio handles incoming data for capture sessions with these options:

Maximum number of cached events: Specifies how many events can be displayed in a live session. When you exceed this limit, Simplicity Studio removes the earliest events from the cache and no longer displays them. Use this setting for very long captures that may exceed the memory capacity of your computer.

Allow drift for auto-correction: Specifies the amount of time-drift that Simplicity Studio will tolerate between the absolute PC clock and the adapter clocks before it resets its "time-correction" factor.

In most situations you should not change this value. In a typical situation, adapter clocks are more precise than PC clocks, but they do not provide absolute time, just relative time against an arbitrary reference.

Sorting and duplicate match window: Specifies in microseconds a time span in which duplicate packets can be detected. If identical packets arrive within the specified time span, Simplicity Studio detects the duplication and allows only one to display.

Show top-level transactions only: By default, Simplicity Studio will show top-level transactions only, and hide the transactions that belong to another super-transaction. Uncheck this option to allow showing of intermediate transactions. Warning: the transaction pane will be harder to read with more intermediate transactions.

Hide 15.4 MAC transactions: By default, Simplicity Studio hides the 15.4 MAC transactions, even if they are not grouped into higher-level transactions. Uncheck this option to change that behavior.

36.1.10 Timeline preferences

Choose **File | Preferences | Timeline**.

You can choose the colors and fonts for features of the Timeline including:

- The mouse annotation color
- Current time label color
- Timeline background color
- Timeline foreground color
- Label font

36.1.11 USB Adapter preferences

Choose **File | Preferences | USB Adapters**.

USB adapters are normally discovered and displayed automatically by Simplicity Studio. For example, the RF Evaluation Kit hardware uses a USB connection to run RF Evaluation tests in Simplicity Studio.

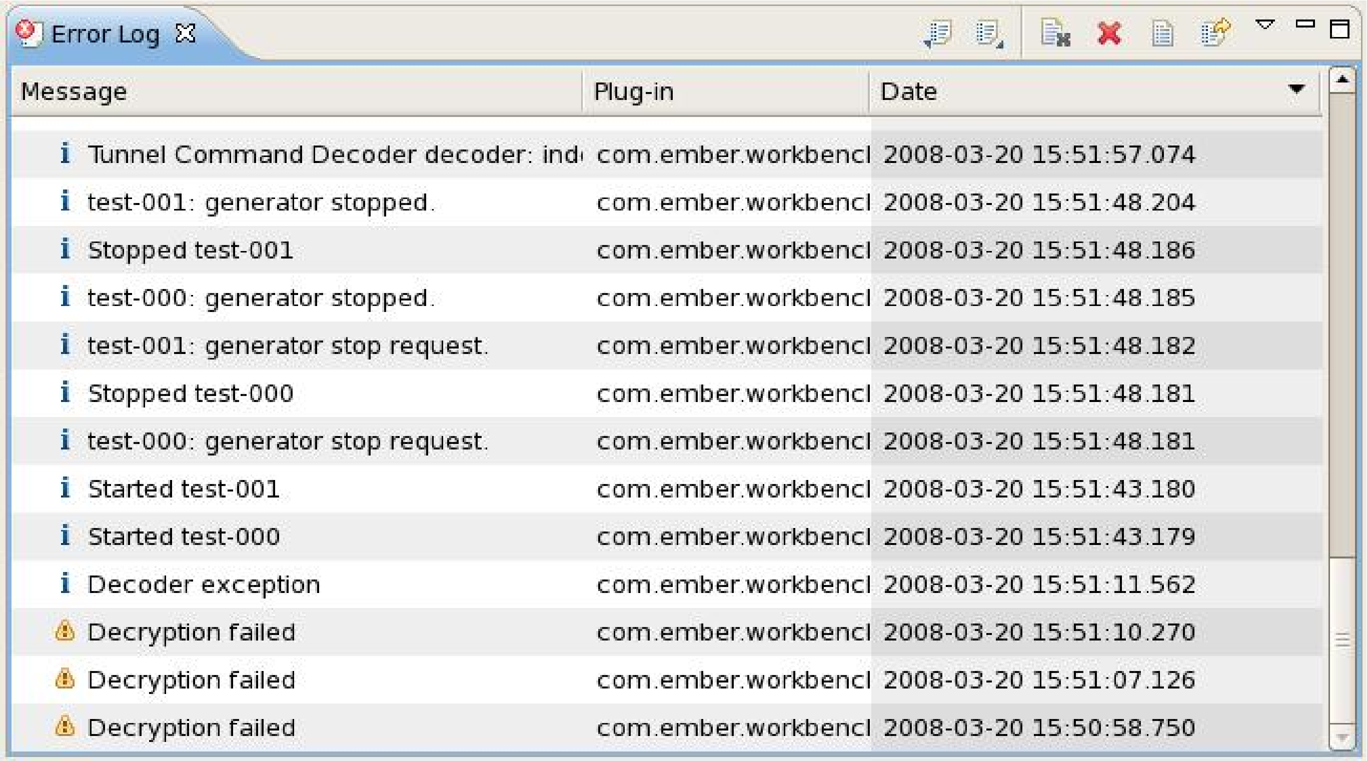
Note: If an RF Evaluation Kit is not automatically recognized by Simplicity Studio, there may be a USB driver conflict. In this case, please contact a Silicon Labs support representative through http://www.silabs.com/zigbee-support.

36.1.12 Web Browser preferences

Use this preference page to configure your web browser for viewing help documentation.

37 Error Logging in Simplicity Studio

Decoder, decryption, and other types of errors are displayed in the Error Log View. To open the Error Log View, choose **Window | Show View | Error Log**.



Message	Plug-in	Date
i Tunnel Command Decoder decoder: ind	com.ember.workbencl	2008-03-20 15:51:57.074
i test-001: generator stopped.	com.ember.workbencl	2008-03-20 15:51:48.204
i Stopped test-001	com.ember.workbencl	2008-03-20 15:51:48.186
i test-000: generator stopped.	com.ember.workbencl	2008-03-20 15:51:48.185
i test-001: generator stop request.	com.ember.workbencl	2008-03-20 15:51:48.182
i Stopped test-000	com.ember.workbencl	2008-03-20 15:51:48.181
i test-000: generator stop request.	com.ember.workbencl	2008-03-20 15:51:48.181
i Started test-001	com.ember.workbencl	2008-03-20 15:51:43.180
i Started test-000	com.ember.workbencl	2008-03-20 15:51:43.179
i Decoder exception	com.ember.workbencl	2008-03-20 15:51:11.562
⚠ Decryption failed	com.ember.workbencl	2008-03-20 15:51:10.270
⚠ Decryption failed	com.ember.workbencl	2008-03-20 15:51:07.126
⚠ Decryption failed	com.ember.workbencl	2008-03-20 15:50:58.750

The Error Log consists primarily of a table of errors reported by Simplicity Studio. Each error is shown in a single row with its summary message, the plugin that reported the error, and the date and time the error was encountered. To view detailed information about the error, double-click it. The error will be displayed in an Event Details dialog that includes the date, severity, message, and, if available, stack trace.