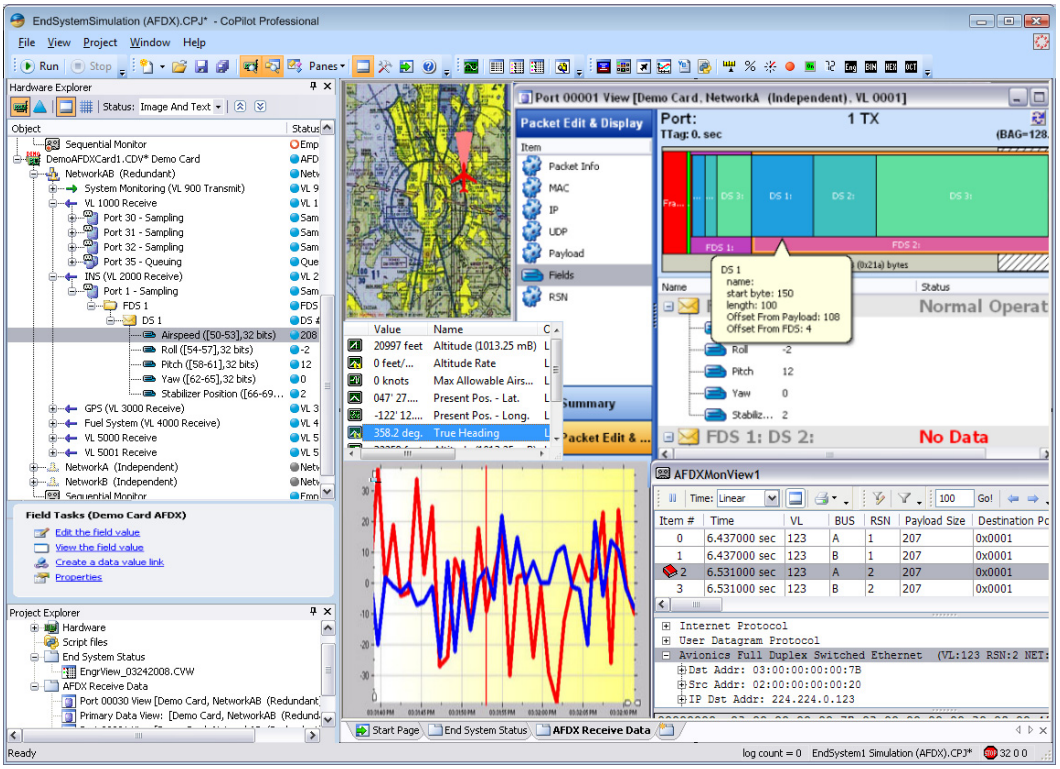
Getting Started Guide to

CoPilot AFDX

Version 5
Rev A



Ballard TECHNOLOGY

Phone: 1-800-829-1553 425-339-0281 Fax: 425-339-0915
E-mail: *support@ballardtech.com*
Web: *www.ballardtech.com*

MA154-20090115

Contents

THIS PAGE INTENTIONALLY BLANK

Introduction

CoPilot AFDX

The term "CoPilot AFDX" is used throughout this manual to identify the AFDX / ARINC 664 specific features available with CoPilot. CoPilot AFDX is part of the multi-protocol CoPilot System, a Windows®-based software program developed by Ballard Technology, Inc. to simplify the simulation and testing of MIL-STD-1553, ARINC 429, ARINC 708, AFDX/ARINC 664 and other avionics protocols. Using CoPilot AFDX, you can simulate End Systems by transmitting and receiving VL and Port data or record databus activity with a few clicks of the mouse. CoPilot automatically detects VL and port activity to configure receive parameters and can load and save configuration information using a database. You can also observe and change data in engineering units or other radices while the bus is running. The tools and filters built into CoPilot assist in locating and analyzing bus activity. Although CoPilot is capable of simultaneously supporting multiple protocols and multiple hardware interfaces, this getting started guide focuses on the operation of a single AFDX demonstration device (a.k.a. AFDX Demo Card).

Ballard Hardware

CoPilot AFDX operates in conjunction with the following AFDX boards: LP-AFDX-1 (PCI), LC-AFDX-1 (cPCI), and LM-AFDX-1 (PMC). These boards can be ordered with varying levels of capabilities, summarized it the table below.

	Level A	Level B
Transmit	✓	✓
Receive	✓	✓
Sequential Monitor (recording)	✓	✓
Filtering	✓	✓
Concurrent with Tx/Rx		✓
Error Injection		✓

CoPilot Standard and Professional Versions

The first two sections of this guide describe features available in the CoPilot Standard version. CoPilot Professional features are described in the third section of this document. CoPilot Standard software can be upgraded to CoPilot Professional. CoPilot Professional has all the capability of the standard version with additional features that include graphical displays, Python-powered ATE (Automated Test Environment), and VB (Visual Basic) scripting.

About This Guide

The *Getting Started Guide to CoPilot AFDX* is a tutorial designed to help you quickly learn the basics of CoPilot for AFDX and introduce you to the optional features available with CoPilot Professional. This guide is composed of fifteen brief lessons in four sections. Sections A and B of this guide describes the features available in CoPilot Standard. The last section of this document, Section C, describes CoPilot Professional features. Because each lesson builds logically to the next, new users are encouraged to work through this guide sequentially. Advanced users can easily jump ahead to a particular lesson using the built-in lesson projects.

Note: This guide assumes the reader is familiar with AFDX, the avionics databus protocol also known as ARINC 664 part 7. No effort has been made to define terms that are part of the AFDX specification. The Ballard Technology, Inc. AFDX Tutorial provides background information.

AFDX Lesson Projects

Template lesson projects for each of the lessons in this guide are built into CoPilot and are accessible from the Samples and Templates panel of the CoPilot Start Page. Because each lesson builds on the previous, opening the project for a specific lesson saves you the trouble of completing all the preceding lessons to obtain the data and configurations you need.

AFDX Demo Card

The lessons in this guide use the built-in AFDX demonstration card (see figure). Using this demonstration card, you can exactly duplicate on your screen what is described in the following lessons. This allows you to "try out" the features and capabilities of CoPilot AFDX without the need for a keyed Ballard AFDX card or access to an active AFDX databus. The AFDX Demo card emulates an AFDX Level B card (see table above for a summary of Level B capabilities). Features include transmit, receive, concurrent monitoring (recording), error injection and CoPilot AFDX Professional hardware license key.

Before You Begin

Install and Open CoPilot

To install CoPilot, insert the CoPilot CD into your drive. A menu screen will automatically open. Click the **Install CoPilot** button and follow the directions. If the menu screen does not automatically launch, run the **SETUP.EXE** program in the INSTALL folder on the CoPilot CD.

If you already have a version of CoPilot installed on your machine, the old version of CoPilot (version 4 or earlier) can remain on your computer and the new CoPilot version 5 can be installed along side this prior version.

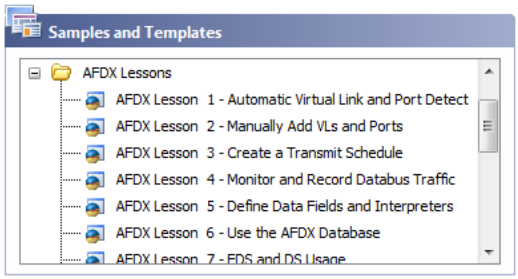
Note that files and projects saved in CoPilot version 5 will not be able to be opened by previous version of CoPilot. Be sure to save a copy of the files you still intend to open with the previous version.

Opening a Lesson

After installation is complete, open CoPilot from the Start | Programs menu or by double clicking the CoPilot icon on your desktop.

For all lessons except Lesson 1, opening a lesson project is the first step of the lesson. For example, to load the second Getting Started lesson project:

- ► Expand **AFDX Lessons** from the Samples and Templates on the Start Page (see figure below)
- ► Choose **AFDX Lesson 1**



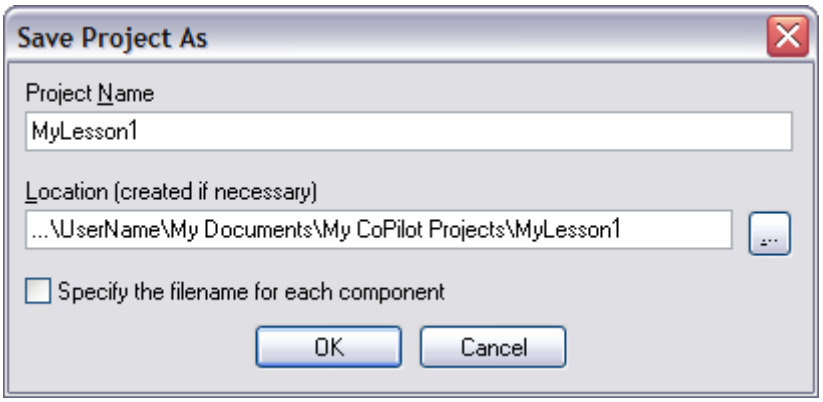
Create User Projects

You can open a lesson project using 'Samples and Templates' on the Start Page (as shown above), or you can save your work as you progress in a single project. To do this, you will use a template project to create your own working project.

To save a project for a template lesson:

- ► Open the lesson from the Start Page

► Select **File | Save Project As…** to open the Save Project As dialog
► Enter a unique name (e.g., MyLesson1) in the Project Name field
► (Optional) Type in or browse to a different file location (the dialog defaults to the CoPilot Projects folder)
► Click **OK** to save the new project (you will be prompted to confirm creation of a new folder)



The new project name will appear in CoPilot's title bar. To reopen this project later, choose **File | Open | Open Project…** and browse to your project. A list of recently used projects is available in the Start Page. Also, a list of recent projects is available in the file menu of CoPilot. As you progress through the lessons, you can continue to add to and save your project.

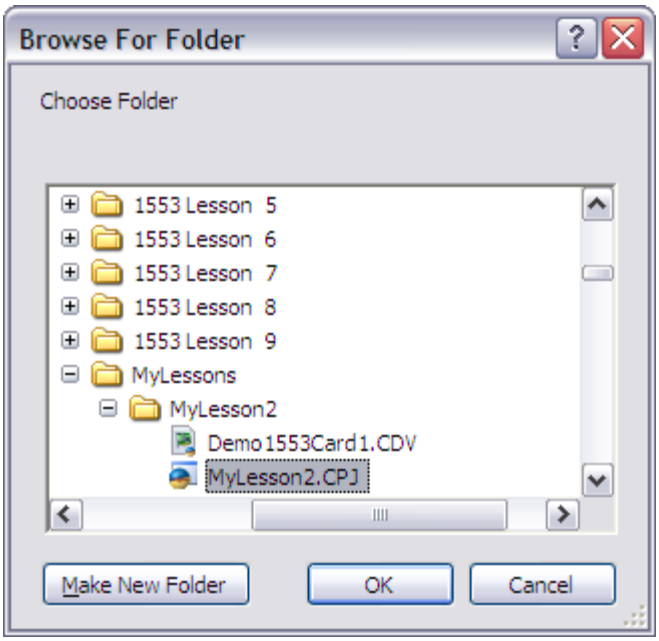
Begin the Lessons

Before you begin each lesson, be sure that you have read the section introductions. These introductions give an overview of larger concepts and provide background information for lessons in the section. Important configuration steps are placed in the introduction rather than duplicating them in each lesson.

Maintaining User Project

To update user project with your latest work, you will need to browse to and overwrite the previous versions of your project.

► Select **File | Save Project As…** to open the Save Project As dialog (see figure)
► Click the browse button to open the Browse for Folder dialog (see figure)
► Use the scroll bar to find your project folder, click the plus icon to expand the folder contents, and select your project file (for example, "MyLessons.CPJ")
► Click **OK** to close the Browse for Folder dialog and click **OK** again to close the Save Project As dialog (you will be prompted to confirm the overwriting of your project)



As you work through the lessons that follow, anytime you need to initialize a lesson from the lesson projects or save your work to your personal folder, you can refer back to this Introduction to review these procedures.

Other Resources

In order to be focused and brief, the lessons in this guide are not exhaustive. You are encouraged to refer to other resources for more detailed information. These include:

- The AFDX Tutorial
- The CoPilot User's Manual (on the CoPilot CD)
- Samples and Templates available from the Start Page
- The AFDX example projects and scripts (on the CoPilot CD)

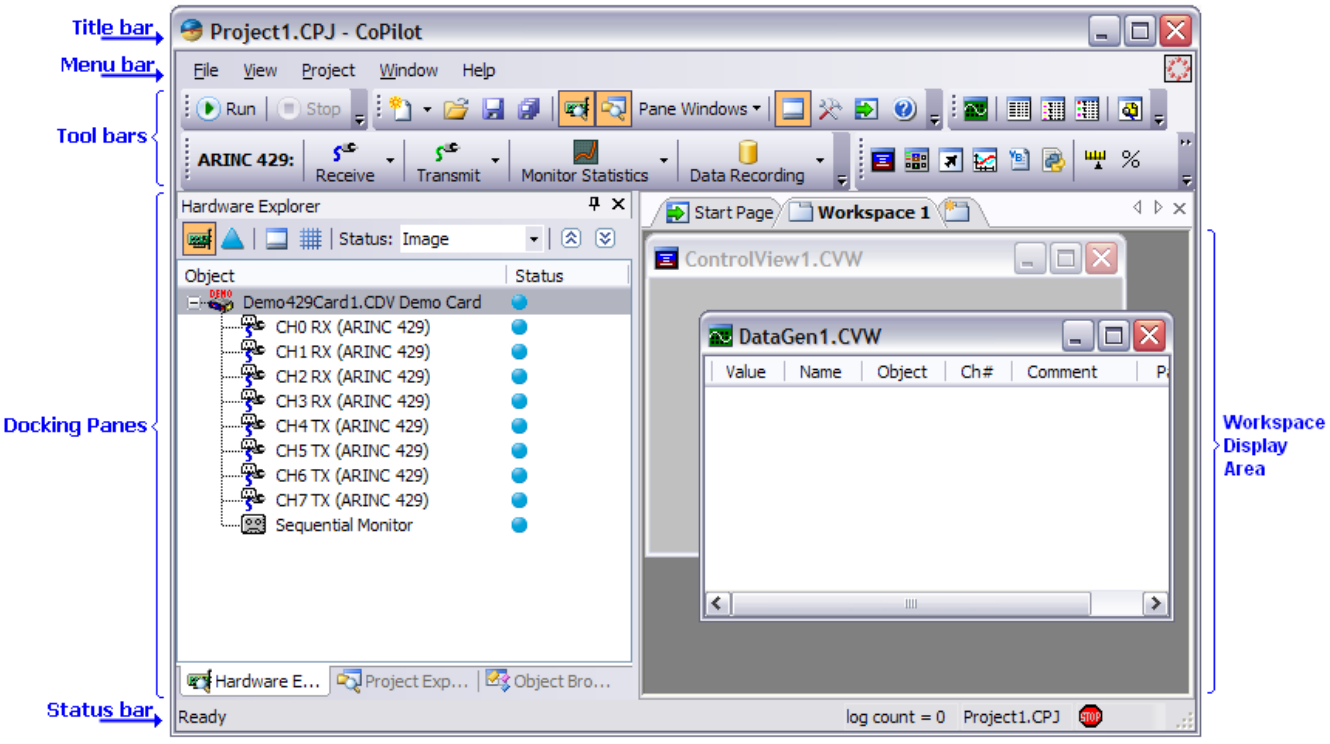
Section A: CoPilot AFDX Basics

The four lessons in Section A present the fundamental AFDX simulation capabilities of CoPilot AFDX. The topics covered are auto-detection of VLs and ports, manual VL and port creation, transmit schedules, and recording sequential monitor traffic. All of the functionality in this section is available with CoPilot Standard. The first three lessons could be run using Model A AFDX Ballard boards (see chart on page 1). From Lesson 4 forward, Model B AFDX Ballard boards are required based on the concurrent sequential monitor settings. Before you begin the lessons, take a moment to get acquainted with the CoPilot environment and learn the principles of CoPilot operation.

The CoPilot Environment

The CoPilot environment consists of an integrated set of windows, menus, toolbars, panes, workspaces, and other elements that allow you to create, edit, organize, and run a CoPilot project (see figure below).

The Hardware Explorer pane is used to configure one or more hardware devices. The Project Explorer pane manages the component files (hardware devices and view windows) that are part of the project. Panes are dockable windows used for display and configuration that may be moved around or undocked from the CoPilot application to change the display layout (see the CoPilot User's Manual for additional information). Various types of view windows, such as the Data Generator, are hosted in the Workspace Display area. Workspaces are used to group and sort display view windows in the display area as shown by 'Workspace 1' in the image below.



The menu bar across the top contains drop down menus. The project menus are File, View, Project, Window, and Help. Other menus appear when items in the Hardware Explorer or other displays are in focus (selected). Multiple toolbars sort the numerous shortcuts to commonly used commands. You can identify the toolbar buttons using tool tip displays (as described later in this section). The status bar along the bottom of the CoPilot desktop displays status information, project location, and run time state and duration. The toolbars and pane locations are fully customizable to suit your specific requirements.

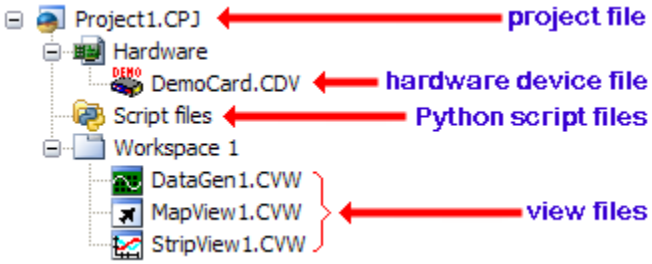
The CoPilot Project

Users interface with CoPilot through a project. The project records user actions and can be saved, closed, and reopened. Only one project at a time may be open in the CoPilot environment.

Project Files

The project is saved as several files stored within a single project folder (see figure below). This allows device and view files to be shared between projects. The file types in a CoPilot AFDX project include:

- A single **Project file (.CPJ)** organizes the project components and records project settings
- **Device files (.CDV)** are created for each Ballard hardware interface card (or demo card)
- Optional **View files (.CVW)** are created for each view window
- Optional **Python script files (.PY)** contain Python script code to perform the specified operations and extend the functionality of CoPilot



When a new CoPilot project is first saved, a project folder is created on the hard drive and the initial project file (.CPJ) is placed in that folder. As device and display components are defined and saved, they are added to the project folder.

Running a Project

CoPilot supports two basic operational modes: Edit mode and Simulation mode. When a CoPilot project is activated through the Run button, the configuration defined in the Hardware Explorer is loaded onto the Ballard hardware and CoPilot displays are activated. CoPilot reverts to Edit mode when the Stop button is pressed.

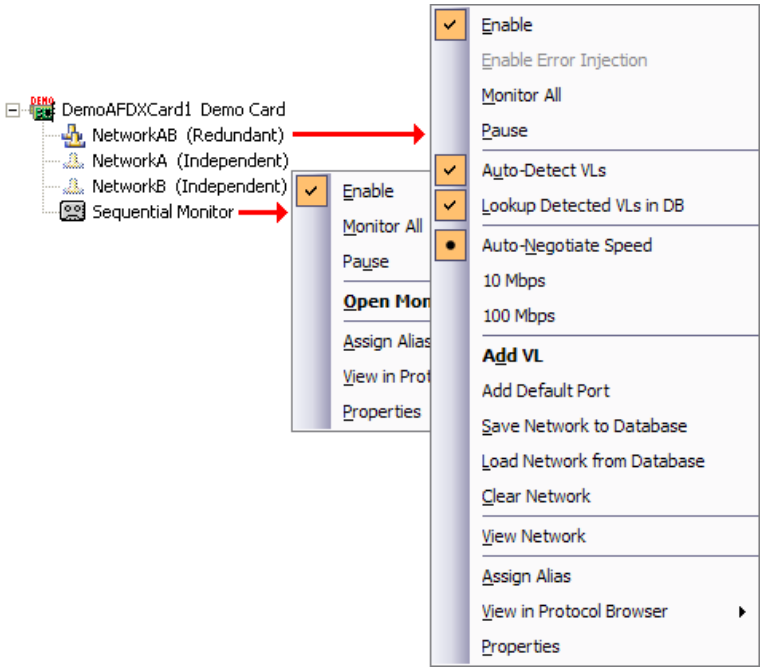
- **Edit Mode—**This mode is the default state for CoPilot during which most data initialization, configurations, and other settings are specified. No interaction with the databus or Ballard board takes place in Edit mode.
- **Simulation Mode—**When Simulation mode is initiated through the Run button, active objects (configured in Edit mode) are transferred to the Ballard avionics board(s). CoPilot actively transmits or receives on the databus and all displays, controls, and windows are animated during the simulation. While the project is running, you can modify data, pause and restart channels, add and delete items from view windows, add new views, and perform many other operations.

CoPilot Principles of Operation

There are certain consistent principles used throughout the CoPilot environment. Understanding these principles will aid you in using CoPilot effectively.

Context Menus

You can interact with objects in the Hardware Explorer and view windows in the workspace display area using context menus, or “shortcut menus.” Right click on items to access their context menu (see following figure).



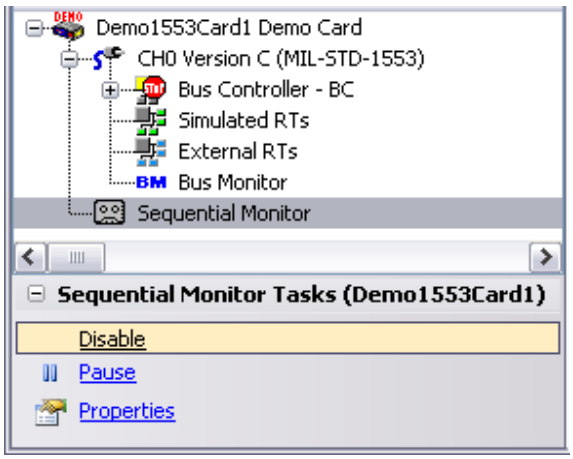
Context Menus are shown from a Right Mouse Click

Default Commands

The default command in each context menu is bolded (see figure on previous page). Simply double-clicking the item (without opening the context menu) will perform the bold entry from the context menu. The default commands for many items may change between simulation (running) and design (not running) to reflect the commands most likely to be used.

Hardware Explorer Tasks

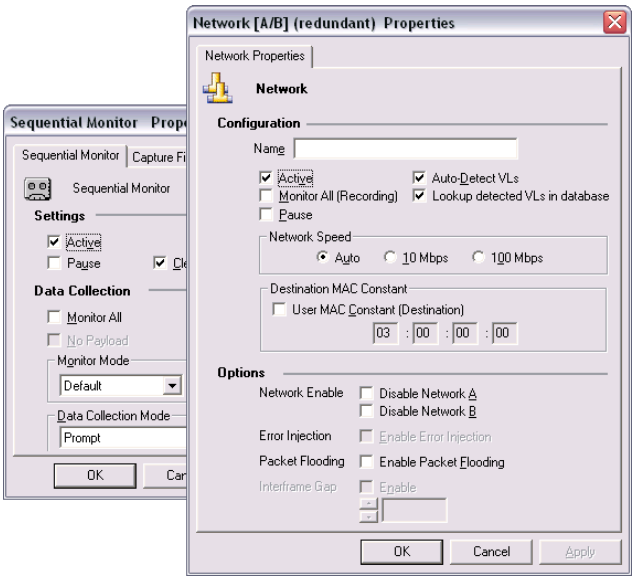
Selecting an item in the Hardware Explorer shows a list of available "quick tasks" for that item. These tasks lists contain the actions most commonly performed on that item. However, if more than one item is selected, then the task list will reflect those common tasks that can be performed by all the selected objects. The <Shift> and <Ctrl> keys are used for multi-selecting items in the Hardware Explorer. Objects can be deleted with the <Delete> key.



Hardware Explorer Task for a Sequential Monitor Object

Configuration using Property Pages

Objects in the Hardware Explorer and in many of the displays are configured and customized through property page dialogs. The figure below shows a few property page examples. The property page of an object is accessed from the Properties item in the object's context menu. Property pages often have multiple tabs used to logically group the properties.

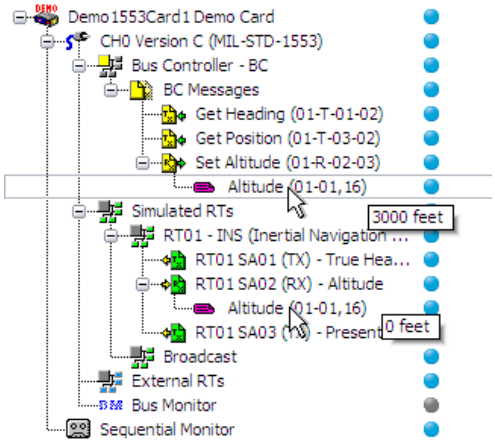


Drag and Drop

The "drag and drop" procedure is used throughout CoPilot to link objects, copy configurations, and automate functions. You can drag and drop objects from the Hardware Explorer into windows in the display pane (such as the Protocol Browser or Object Browser). VLs, Ports and fields can be copied to other locations in the Hardware Explorer by dragging and dropping. If you attempt to drag an item to an area that cannot accept it, a warning dialog will appear and the action will not be completed.

Tool Tips

Objects in the hardware tree pane use tool tips to display additional information. In addition, many properties pages, buttons, and other objects and windows use tool tip displays. To display the tool tip for an item, allow the mouse to hover over the item. A popup will appear with status, configuration, or data information. (see figure below).



THIS PAGE INTENTIONALLY BLANK

Lesson 1: Automatic Virtual Link and Port Detect

AFDX networks are comprised of Virtual Links (VLs) which define distinct unidirectional communication channels for transferring data between end systems. CoPilot can automatically detect VL and Port activity on the bus.

Note: If you have not already done so, please read the Introduction to this guide, especially the "Before You Begin" section (page 2), for instructions on opening AFDX lesson projects and maintaining a personal cumulative project.

Introduced in This Lesson

Network View, Network Statistics, Auto-Detection, VL View, VL statistics

Objective

Configure the redundant network to automatically detect VL and Port activity, and then view network and VL statistics.

Open the Lesson 1 Project

Be sure that the AFDX Lesson 1 project is open:

- ► Select **File | New | New Project**
- ► Click the **AFDX Lessons** tab
- ► Select **AFDX Lesson 1** and click **OK**

The demo card should look like the figure at right.

If you wish to save a cumulative project as you work through the lessons, see the Introduction for complete instructions.

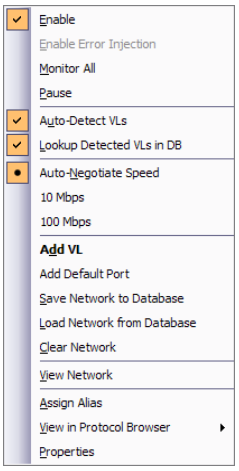
AFDX demo card in the hardware tree

Display Network View Statistics

The Network View contains statistics for network activity. To open a Network View:

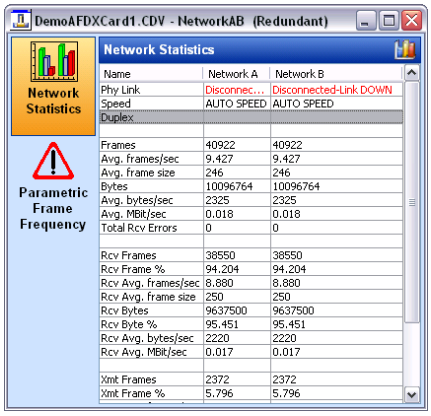
- ► Right click the redundant **NetworkAB** and then select **View Network** from the context menu

Note: The status of the physical link (Phy Link) depends on a physical connection to an actual card. The demo card always displays the link status: *Disconnected-Link DOWN*.



Open Network View from network context menu

The statistics available in this view update in real-time while the card is running. Keep this view open while the project is running to see these values update. Hovering the mouse over a statistic value displays a tooltip description.



Network View Statistics

Enable Auto-Detect VLs

The Auto-Detect VLs feature must be enabled to automatically detect VL activity on a network. To enable the Auto-Detect VLs option for the redundant network:

- ► Right click on the redundant **NetworkAB** then click **Auto-Detect VLs** to enable the VL auto-detection option (see image on next page)

Start the CoPilot Project

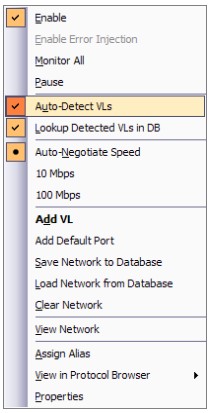
Running the demo card with auto-detection enabled detects VL123 and automatically inserts it into the hardware tree.

► Click the **Run** button to begin simulation

After the demo card is running and VL123 is detected, ports can also be auto-detected. Additionally, Lesson 6 describes how the database is utilized for auto-detection.

Note: Do not stop the project at this point. The project must be running to see the statistics updating.

The statistics in the Network View continue to update while the CoPilot project is running.



Set Auto-Detect VL option from the Network context menu

Analyze Detected VL Statistics

The VL View contains statistics for a VL similar to the statistics seen in a Network View. The number of frames, number of bytes, and other values are displayed for each half of a redundant network. To open a VL View:

► Right-click on **VL123** then select **View VL**

Resetting the statistics provides a starting point for easier analysis. To reset the statistics in either the Network View or the VL View:

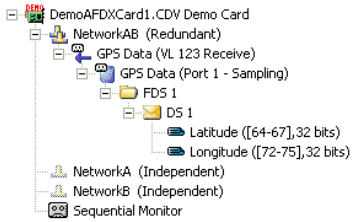
► Right click the **VL Statistics** to show a context menu

► Select **Reset All Counts**

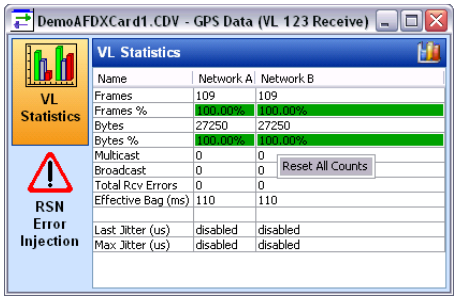
The counts are then reset to zero. As the simulation continues, the statistics are updated accordingly.

Note: Resetting the statistics does not erase any data. Closing and reopening the view undoes the reset.

Click the **Stop** button to finish this lesson.



Hardware Explorer with automatically detected VL



VL View statistics from VL context menu

Related Topics

- Ballard's AFDX devices either operate as two separate independent networks (see Lesson 11) or a single redundant network; NetworkAB is the redundant network that consists of a pair of physical network connections
- The Network properties dialog is used to configure additional network settings
- You can manually disable the auto-detection of VLs and ports (through their context menus)

Summary

In this lesson you learned…

- how to access context menus for hardware tree objects
- how to run a CoPilot project
- how to auto-detect VLs and Ports
- how to display statistics with a Network View and VL View

Building on these ideas, you will learn in Lesson 2 how to manually define transmit VLs and ports

Lesson 2: Manually Add VLs and Ports

The first lesson illustrated automatic detection of receive information. This lesson shows how to configure transmit VLs and ports. CoPilot uses port scheduling to transmit AFDX frames. Before configuring and running schedules, ports must be defined on transmit VLs.

Introduced in This Lesson

Add VL dialog, VL Properties, CoPilot Options Properties, Add Port dialog, Port Properties, Drag-Drop

Objective

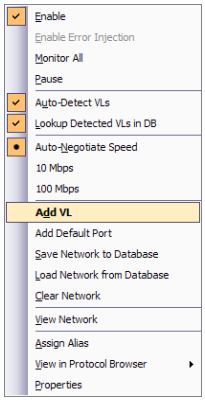
First, create two transmit VLs on the redundant network. Then configure the port data initialization option to incrementing values. Next, create two sampling ports and a queuing port on one of the newly created VLs. Finally, copy a port to another transmit VL.

Define Transmit VLs

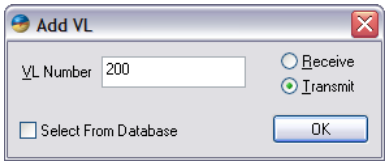
Note: Before you begin this lesson, open the AFDX Lesson 2 project (File | New | New Project | AFDX Lessons).

Begin by creating transmit VLs: 200 and 201.

- Right click the **NetworkAB** and select **Add VL**
- Enter **200** for the VL number, select **Transmit**, and click **OK**
- Enter **256** for the MTU (Maximum Transmission Units) and press **OK**
- Repeat process by again a right clicking on **NetworkAB** and select **Add VL**
- Enter **201** for the VL number, select **Transmit**, and Click **OK**
- Enter **256** for the MTU and press **OK**



Network context menu

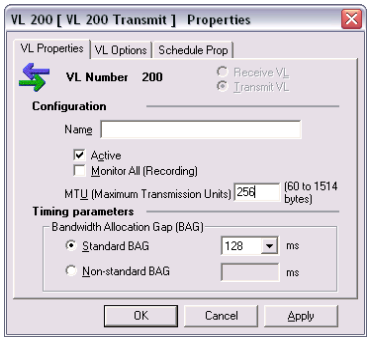


The Add VL dialog

Configure Port Data Initialization

Ports use the data initialization setting when they are created. This option will be set to initialize the data to all zeros. To set data initialization:

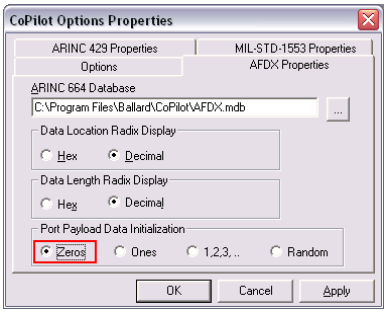
- Select **Options** from the CoPilot Project menu
- Click the **AFDX Properties** tab
- Select **Zeros** for the Port Payload Data Initialization
- Click **OK** to close the dialog.



VL property page dialog

Create Sampling Ports

- Right click on VL 201 and select **Add Port** to open the property page
- Enter **50100** for the port number and **256** for the initial port size (image on next page)



The AFDX Properties tab of the CoPilot Options dialog

► Press the **OK** button

► Uncheck **Using Functional Datasets** on the port property page and press **OK**

Repeat the steps above to create a second port.

► Right click on ➜ **VL 201** and select **Add Port**

► Enter **50101** for the port number

► Enter **128** for the initial port size

► Press the **OK** button

► Uncheck **Using Functional Datasets** on the port property page and press **OK**

Create a Queuing Port

Following the steps below adds a queuing port to VL 201. After adding the queuing port, VL 201 will contain two sampling ports and a queuing port.

► Right click on ➜ **VL 201** and select **Add Port**

► Enter **50102** for the port number

► Enter **128** for initial port size and select **Queuing**

► Press the **OK** button

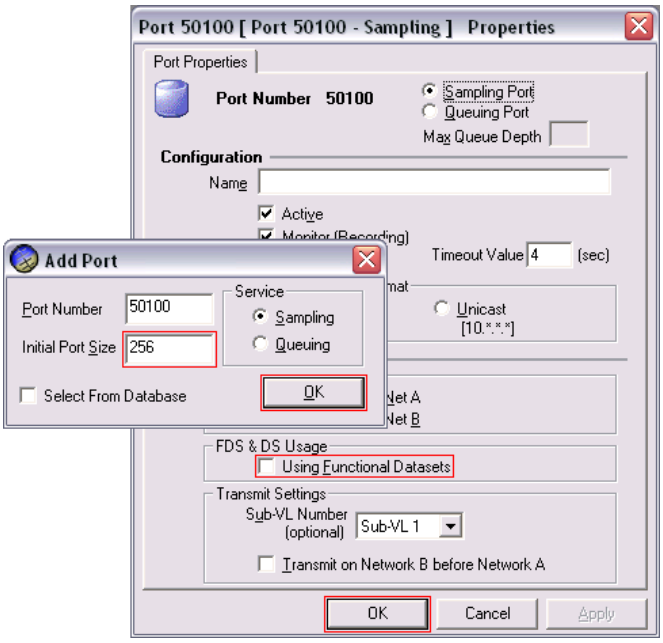
► Uncheck **Using Functional Datasets** on the port property page and press **OK**

Copy Port to Another VL

Drag and drop is a quick method for copying objects in the hardware tree. Later in Lesson 6, the database is used to load various objects in the hardware.

► Drag and drop **Port 50100** on **VL 201** to **VL 200**

Note: The object and all child objects are copied when using the drag and drop operation.



Adding a port with the Add Port dialog then setting Port properties

Error! Objects cannot be created from editing field codes.

Drag-drop to copy port

Related Topics

- Manual creation of receive VLs is done in the same way the transmit VLs were created in this lesson
- Functional Data Set (FDS) usage was disabled for ports in this lesson, but is described in detail in Lesson 7
- Transmit queuing ports only transmit data when the port (or a field on the port) is updated
- Hardware tree definitions are simplified with the use of the database as described in Lesson 6

Summary

In this lesson you learned…

- how to define a transmit VL
- how to configure port data initialization in the CoPilot Options dialog
- how to create sampling ports
- how to create queuing ports
- how to copy a port definition to another VL using drag-drop

In Lesson 3, you will schedule VL 200 for transmission.

Lesson 3: Create a Transmit Schedule

Transmit schedules control the order of transmitted frames and dictate their timing on the AFDX network. VLs are scheduled at intervals so the transmission rate of each VL does not exceed its specified Bandwidth Allocation Gap (BAG). BAG values are calculated and specified by the system designers. For less time-critical communication, sub-virtual links (sub-VLs) are used to share the allocated bandwidth between ports on the same VL. Multiple ports defined on a VL are scheduled using round-robin processing by default. CoPilot's advanced sub-VL scheduling uses sub-VL factors to set relative transmission rates for each sub-VL.

Introduced in This Lesson

VL Schedule Properties, Round-Robin Scheduling, Port Properties, Sub-VL Rate Scheduling, Sub-VL factors

Objective

Build a round-robin transmit schedule for the VLs and Ports defined in Lesson 2. Then configure a Sub-VL schedule.

Configure Round-Robin Scheduling

Note: Before you begin this lesson, open the AFDX Lesson 3 project (File | New | New Project | AFDX Lessons).

CoPilot automatically configures transmit VLs and ports on a VL to use round-robin processing.

To set the transmit properties for VL 200:

- ► Right click ➜ **VL 200** and select **Properties**
- ► Set the Standard BAG value to **64** ms
- ► Press **OK** to close the properties dialog

Non-standard BAG values are not limited to powers of 2. To change the VL 201 BAG to 75 ms:

- ► Right Click ➜ **VL 201** and select **Properties**
- ► Select the **Non-standard BAG** option and enter **75**
- ► Press **OK** to close the properties dialog

Run Round-Robin Schedule

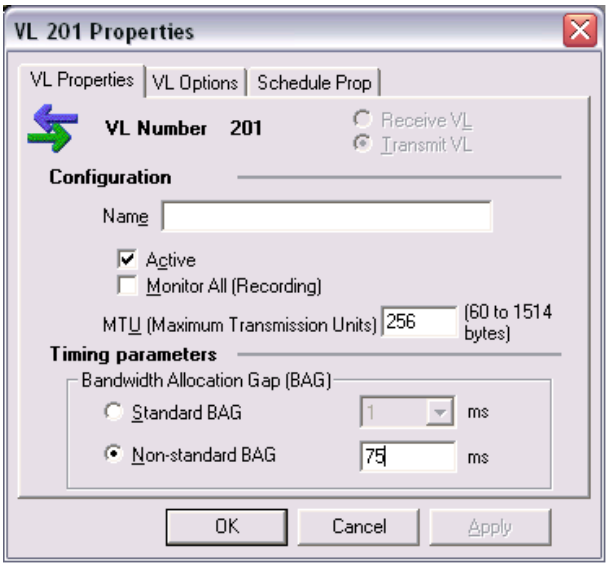
Running the schedule transmits a port from each transmit VL at the BAG specified. The ports on VLs are processed in a round-robin manner if multiple ports are defined. If queuing ports do not have data to send, the queuing port is skipped and the next port in the list is sent instead.

- ► Click the **Run** button to begin transmission

Network statistics and VL statistics can be viewed with a Network View or a VL View as shown in Lesson 1.

Note: Because real hardware is not being used, the simulated demo card transmission rates may not match the configured rates.

- ► Click the **Stop** button to stop transmitting



VL Properties to set the BAG

Error! Objects cannot be created from editing field codes.

VL View showing VL 201 statistics while running a schedule

Assign Ports to a Sub-VL

Sub-VL Rate Scheduling uses one or more Sub-VLs for scheduling. The default Sub-VL value for every port is one.

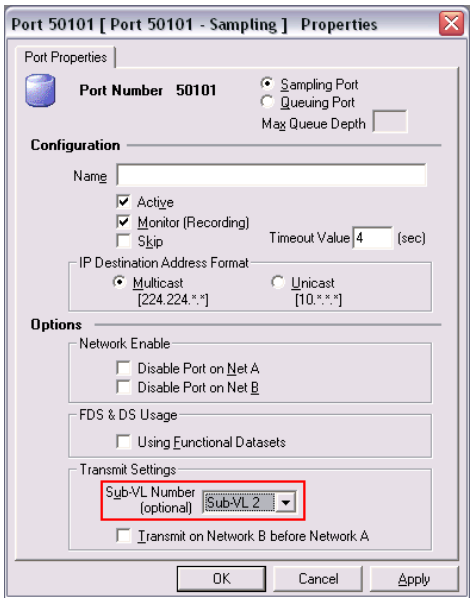
Note: Sub-VLs, with multiple ports, defined use round-robin processing for its ports.

To assign Port 50101 to Sub-VL2:

- ► Right click on **Port 50101** and select **Properties**
- ► Select **Sub-VL 2** from the Sub-VL Number list then click **OK** to close the properties dialog

Next, assign Port 50102 to Sub-VL3:

- ► Right click on **Port 50102** and select **Properties**
- ► Select **Sub-VL 3** from the Sub-VL Number list then click **OK** to close the properties dialog



Port Properties to set Sub-VL number

Sub-VL Rate Scheduling

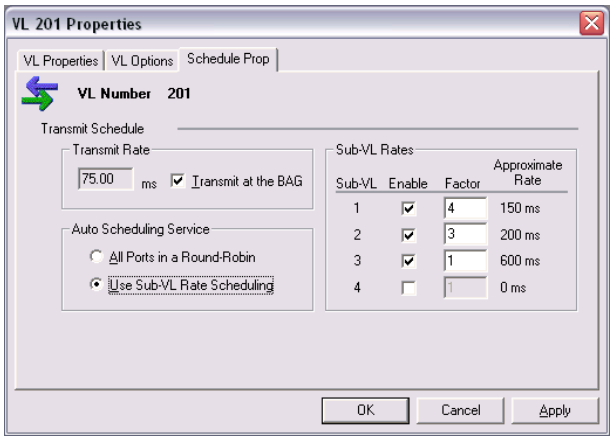
Sub-VL rate scheduling is configured on a per-VL basis to control the relative transmission rates of Sub-VLs.

- ► Right click on **VL 201** and select **Properties**
- ► Click the **Schedule Prop** tab
- ► Select **Use Sub-VL Rate Scheduling**

This schedule produces the same results as the default schedule provided all sub-VL factors are equal. Increasing a Sub-VL factor increases the relative transmission rate for that Sub-VL. To modify the Sub-VL factors

- ► Set Sub-VL 1 factor to **4,** Sub-VL 2 factor to **3**, and Sub-VL 3 factor to 1
- ► **Uncheck** the Sub-VL 4 check box and press **OK**

The project may be run at this time if desired. Note: Sub-VL rates do not affect the schedule of the demo cards.



VL Properties configuring Sub-VL schedule factors

Related Topics

- At least one port must be defined for each VL in order to schedule the VL
- At each VL transmit interval, the round robin schedule transmit the next ports on the VL
- If a queuing port is empty, the port will be skipped and the next item for that Sub-VL is sent

Summary

In this lesson you learned…

- how to configure a VL to transmit at the BAG
- how to configure a VL to transmit at an interval other than the BAG
- how to run with an automatically built schedule
- how to run a sub-VL schedule with sub-VL factors

In the next lesson, you will learn how to monitor and record the AFDX databus traffic and view it through the Monitor View window.

Lesson 4: Monitor and Record Databus Traffic

To this point, you have configured the card for transmit and receive and built transmit schedules. With CoPilot's powerful Sequential Monitor, you can record a sequential history of bus activity and view that data in several formats through the Monitor View window. Capture filtering is used to limit the collected records. Display filters limit the Monitor View to show only the items of interest from the collected records without affecting record collection.

Introduced in This Lesson

Sequential Monitor Properties dialog, Monitor View, Monitor recording controls, Display Filter dialog

Objective

Monitor the databus, view the record in the Monitor View window, pause and resume the recording. Finally, modify the display filter to filter out traffic from the Network A side of the redundant network.

Configure the Sequential Monitor

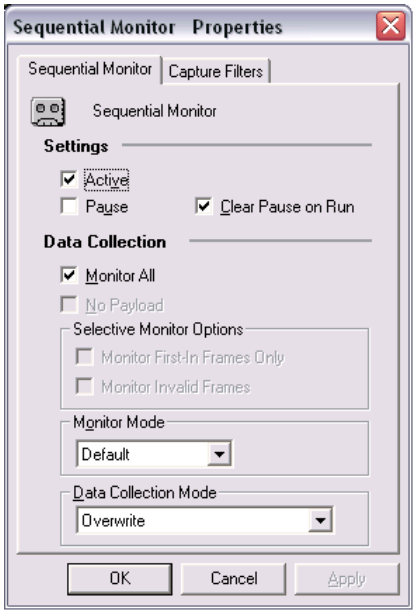
Note: Before you begin this lesson, open the AFDX Lesson 4 project (File | New | New Project | AFDX Lessons).

- ► Right click on the **Sequential Monitor** icon in the Hardware Explorer and choose **Properties** from the context menu to open the Sequential Monitor Properties window
- ► Click the check box for **Monitor All** in the Sequential Monitor tab (default tab)
- ► Click **OK** to close the properties page

The Sequential Monitor is now ready to record all activity on the databus.

- ► Click the **Run** button in the CoPilot tool bar

The record count is displayed on the same line as the Sequential Monitor icon: Sequential Monitor (248 items)



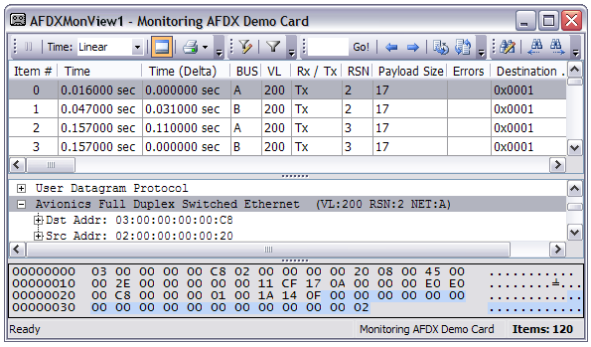
Sequential Monitor Properties dialog

Open the Monitor View Window

- ► Right click on the Sequential Monitor and choose **Open Monitor View**

The Monitor View window will open and you will see the record of current bus activity.

The top pane of the Monitor View displays the summary of the records with the timetag, bus, VL, port, payload size, etc. The middle pane contains detailed information with frame dissection capabilities for the single record selected in the top pane.



Monitor View window during simulation

Control Monitor Recording

Bus Monitor recording can be controlled through the Monitor View Pause button.

► Click the **Pause** button (upper left-hand corner of the window) to suspend recording

Notice the Pause button is depressed and the Sequential Monitor message counter stops.

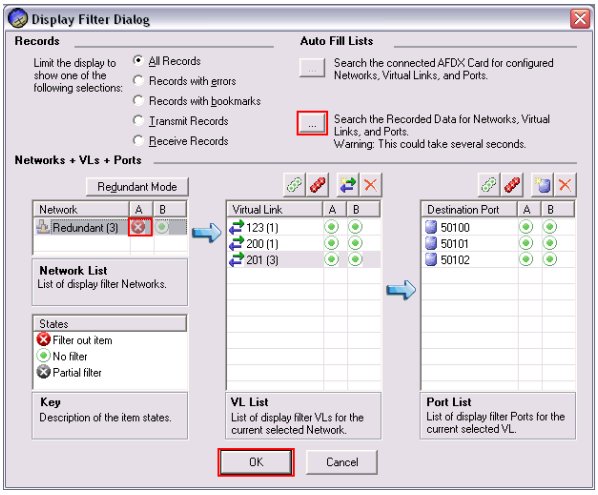
► Click the **Pause** button again to resume

► Click the **Stop** button to end the simulation

Configure the Monitor Display

The recorded items shown in the Monitor View display can be limited by display filter criteria. Modifying the display filter does not alter the recording.

► Click the **Edit** button in the Display Filter frame to open the monitor Display Filter dialog (see figure at right)



Sequential Monitor View Display Filter dialog

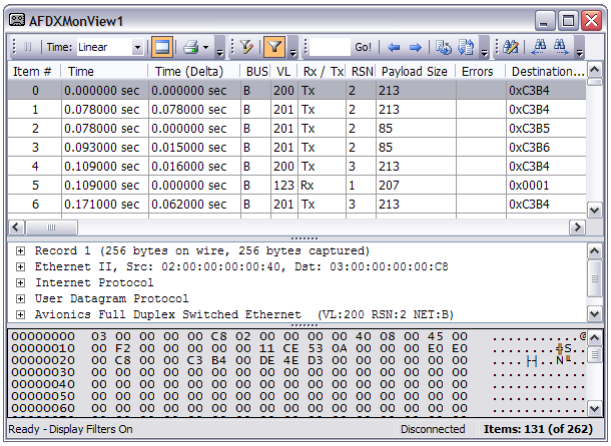
The Display Filter dialog allows numerous filter options. To limit the display to only Network B records by filtering out all Network A records:

► Click the **Search the Recorded Data** button to search through the recorded data (used to populate the various lists with active VLs and ports)

► Click the button for **Network A** on the redundant network so Network A is filtered out,

► Click **OK** to close the dialog and view the results

Only Network B items are now displayed in the Sequential Monitor View. Notice that the record numbers in the left column are sequential. The items are renumbered to not include the filtered out records.



Monitor View showing only traffic on bus (Network) B

► Click the button for **Network B** on the redundant network so Network B is filtered out,

► Click **OK** to close the dialog and view the results

Only Network B items are now displayed in the Sequential Monitor View. Notice that the record numbers in the left column are sequential. The items are renumbered to not include the filtered out records.

► Right click on the Sequential Monitor and deselect **Enable** (it is not needed for the next few lessons)

Note: **Save** and **Minimize** the Monitor View if you are maintaining a cumulative project.

Related Topics

- When an item is selected in the upper pane, the middle and bottom panes are updated to display the detailed information for that record
- The Monitor View tool bar also contains buttons for other features such as Search, Go To, and Step, etc.

Summary

In this lesson you learned…

- how to configure and record using the Sequential Monitor
- how to open a Monitor View to display recorded history
- how to pause and resume Monitor recording
- how to configure the Monitor View display

You have now covered the basics of CoPilot AFDX simulation. In the next section, you will discover how to create, interpret, modify, generate, and view AFDX data.

THIS PAGE INTENTIONALLY BLANK

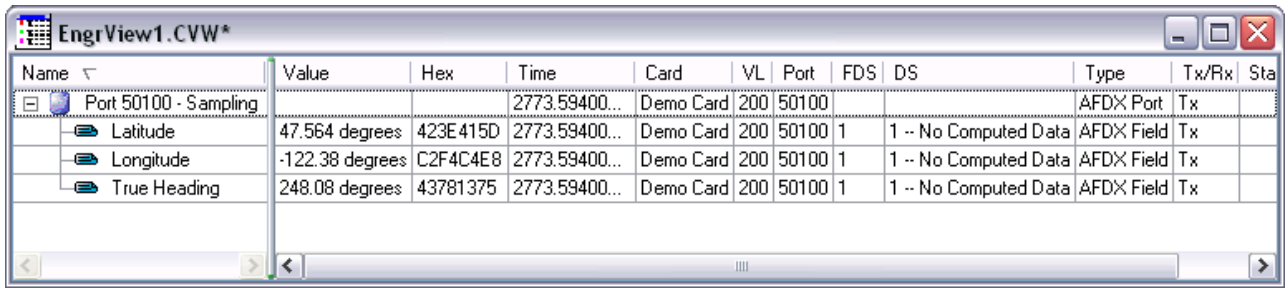
Section B: CoPilot AFDX Data

Review

In the preceding section, you learned how to auto-detect, receive frames, create VLs, create ports, define transmit schedule, record databus traffic, and view the monitor record. The lessons in Section B will guide you through the process of creating, encoding, interpreting, modifying, and displaying AFDX data in the CoPilot environment. Before you begin the six lessons in this section, read this introduction to learn about engineering units in CoPilot, how to work with view windows, and important information about the various files in a CoPilot project.

Engineering Units

Both AFDX and ARINC 664 leave the frame definition and data interpretation to the system designer. Data in each message may potentially be completely redefined for different applications. With CoPilot, the meaning of the data can be configured in projects and even stored in a user database. There is no need for the user to engage in mathematical calculations and conversions to create or interpret the raw binary values passed on the AFDX databus. CoPilot converts the ones and zeros that come across the bus into recognizable "engineering units" based on field and interpreter definitions the user chooses.

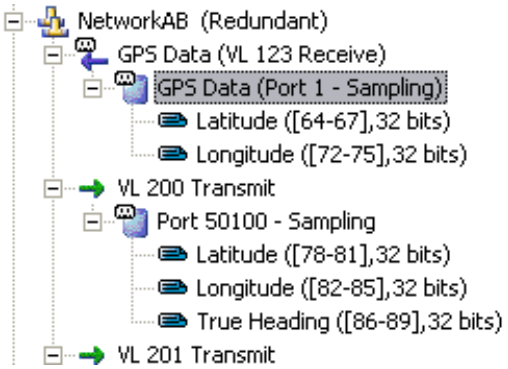


Name	Value	Hex	Time	Card	VL	Port	FDS	DS	Type	Tx/Rx	Sta
Port 50100 - Sampling			2773.59400...	Demo Card	200	50100			AFDX Port	Tx	
Latitude	47.564 degrees	423E415D	2773.59400...	Demo Card	200	50100	1	1 -- No Computed Data	AFDX Field	Tx	
Longitude	-122.38 degrees	C2F4C4E8	2773.59400...	Demo Card	200	50100	1	1 -- No Computed Data	AFDX Field	Tx	
True Heading	248.08 degrees	43781375	2773.59400...	Demo Card	200	50100	1	1 -- No Computed Data	AFDX Field	Tx	

In this way, data is given meaning within the context of the CoPilot simulation. Then, this meaning can be preserved and refined by saving to, loading from, and modifying the user-defined AFDX database.

Data Fields

Data fields are used within CoPilot to associate meaning with a bit or sequence of bits in the payload of an AFDX frame. Each field can be named, assigned units of measurement, and associated with an interpreter so that its bits may be accurately deciphered. Additionally, FDS (Functional Datasets) and DS (Datasets) structures can be applied to ports to logically group portions of the AFDX payload. The figure below illustrates the definitions of data fields (Latitude, Longitude, and True Heading).



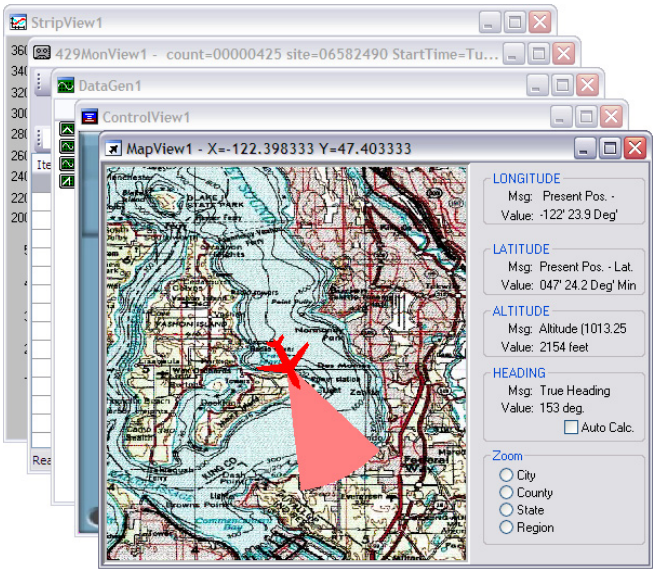
Data Interpreters

The data interpreters built into CoPilot AFDX allow you to customize how data is encoded and translated. For example, you could encode data as BNR, BCD, Boolean 32, Custom Script, Float 32, Float 64, Integer 32, Integer 64, Opaque–Fixed Length, Opaque–Variable Length, or String values.

View Window Components

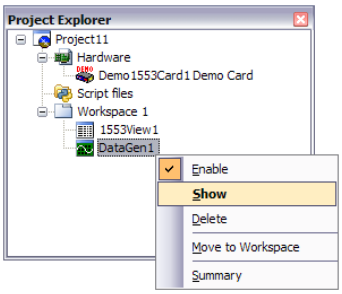
CoPilot Windows hosted in the Workspace display area are added using the View toolbars, the "Select New Hardware and Views" dialog, and through objects in the Hardware Explorer. The Message Editor, BC Schedule, and Message/Subaddress Views introduced in Section A are examples of object-generated windows. They are not saved as part of a CoPilot because their information is saved as part of objects in the Hardware Explorer.

Starting with Lesson 8, you will learn about a variety of windows (also called "components" and "views"), opened through the "Select New Hardware and Views" dialog. Independent views can be saved as part of a CoPilot project, closed, reopened, and even shared between projects. They are not tied to any specific object in the Hardware Explorer although they can interact with many objects. The components discussed in the following lessons are: Data Generator (Lesson 8), Engineering View (Lesson 9), and the Professional components in Section C (Strip View, Quick View, Control View, Map View, Python Editor View, and Script View). Once activated, they are listed in the Project Explorer.



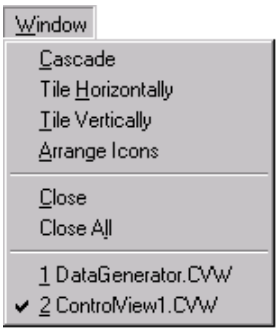
Working with View Windows

View windows can be managed through commands in the Project Explorer and the Window menu. Project Explorer menu commands (see figure below) allow the user to enable/disable, show (in the Workspace display area), delete, and record summary information for view windows. By clicking it's **Minimize** button, a view is removed from the display area but its icon is still available in the Project Explorer. A hidden/minimized window can be restored to the workspace display area by double clicking its icon in the Project Explorer or right clicking its icon to access the context menu and choosing **Show**.



Note that hidden views continue to run and affect other parts of the project (for example, the Data Generator continues to source data to fields in the Hardware Explorer). Unnecessary views may consume processing power. To disable a view without deleting it from the Project Explorer, right click on its icon and clear the checkmark by Enable (see figure on previous page).

View windows can also be selected or arranged through the Window menu (see figure below).



Hiding View Windows

The **Minimize** button in the title bar of a view window is often used in CoPilot to remove the window from the screen but preserve it in the project. Throughout this document, we refer to this as "hiding" the window.

Hidden windows continue to operate or receive updates when the CoPilot project is running, without cluttering the screen. They can be restored, placed into an active or inactive state, or deleted from the project through context menu options in the Project Explorer.

- **Hide**—To hide a window, click the **Minimize** button in the window title bar. The window will disappear from the display area, but its icon will remain in the Project Explorer and it will continue to run invisibly. To show it, double click its icon in the Project Explorer or right click its icon to access the context menu and choose **Show**.
- **Disable**—To disable a view (visible or hidden), right click on the view icon in the Project Explorer and clear the **Enable** checkmark. The view will cease to run, but will remain in the Project Explorer (and in the display area, if it is visible). To reenable it, select the Enable option.
- **Close—**To completely close a view, click the **Close** button in the window title bar. This can also be accomplished through the Delete command, accessed by right clicking on the view icon in the Project Explorer and selecting **Delete (**or press the <Del> key). The view will be removed from the Project Explorer and if needed a prompt will appear to optionally save the view to file. To reopen a saved file, choose File | Open | Open Hardware or View Files… and browse to its location.

Saving View Windows

When a new view is created, CoPilot assigns a default name, for example, "DataGen1" (see figure on previous page). When the view window is saved to disk, the .CVW extension is added. When developing a new project, it is often helpful to save view windows as you go along.

► To save a view window for the first time, choose **File | Save <*component name*> As** or click the **Save** button. You will be prompted for a file name. Although assigning descriptive names is recommended, throughout these lessons we have accepted the default file name supplied by CoPilot.

► You can also save all views and components to the project folder of the current project using the **Save Project** button or **File | Save Project**. You will be prompted to confirm the addition of new components or the overwriting of saved components.

Lesson 5: Define Data Fields and Interpreters

CoPilot's interpreters allow you to encode data in many formats, such as: BNR, BCD, Boolean 32, Custom Script, Float 32, Float 64, Integer 32, Integer 64, Opaque–Fixed Length, Opaque–Variable Length, and String.

Introduced in This Lesson

Field Configuration dialog (for adding fields), Field Selector dialog (Create New tab), Interpreter Properties dialog, Engineering Units Editor

Objective

Define a 32-bit integer field interpreter for latitude on a transmit port and copy that same field to a receive port. Edit the data value of the transmit field and run the schedule. View current field values using tooltips in the hardware tree.

Insert New Data Fields

Note: Before you begin this lesson, open the AFDX Lesson 5 project (File | New | New Project | AFDX Lessons).

Fields are used for configuring the engineering unit translations from data bits in the AFDX frame.

- Right click **Port 50100** on VL 200 and choose **Add Field**
- Set the Name to **Latitude** and the Start Byte to **78**
- Select the **Float 32-bit** Interpreter and click **OK**

The property page for the Latitude field will then appear.

- Type **degrees** for the Units
- Click Float 32-bit Interpreter tab and set Range to **-90** and **90**
- Click **Round Display**, set Display Decimals to **5** and press **OK**

A True Heading field will now be added by repeating the steps used to add the Latitude field above.

- Again, right click **Port 50100** on VL 200 and choose **Add Field**
- Set Name to **True Heading** and Start Byte to **86**
- Select the **Float 32-bit** Interpreter and click **OK**

The property page for True Heading will then appear.

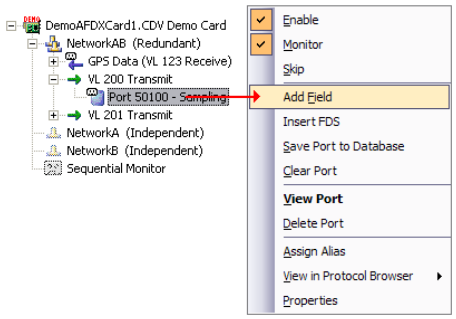
- Set Units to **degrees** and Click the Float 32-bit Interpreter tab
- Set the Range to **0** and **360** and press **OK**

Duplicate the Configured Field

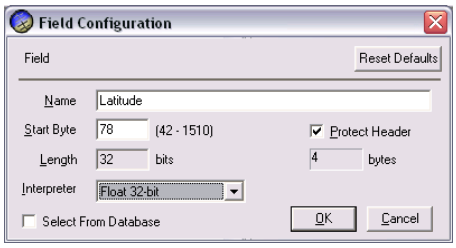
Once a data field and its interpreter are defined, this information can be quickly and easily propagated to other objects in the hardware tree.

- Drag and drop the **Latitude** field onto Port 50100 of VL201

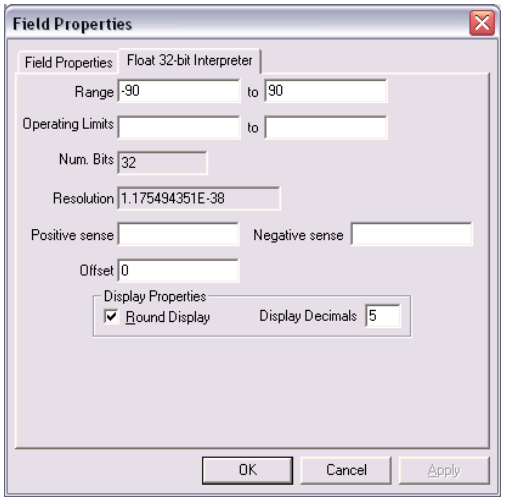
The Latitude field is then copied to the port.



Context menu for adding a field on Port 50100



Inserting a 32-bit floating point field for Latitude



Field Selector and Interpreter Properties dialogs for Latitude

Set the Data Field Value

Now that the data field is defined, data values can be easily entered and viewed in engineering units.

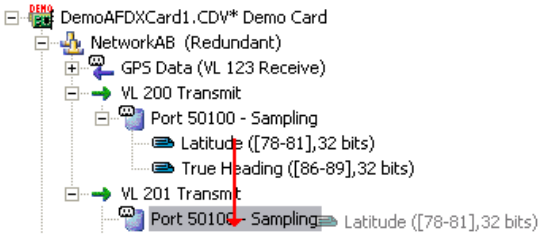
► Right click the **Latitude** field on VL 200 and choose **Edit Data** to open the editor

You can enter a value in the Engineering Units Editor with the textbox (or slidebar with some interpreters).

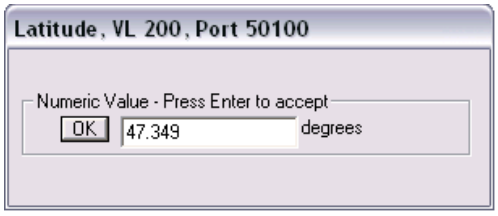
► Enter **47.349** degrees in the Numeric Value textbox and click **OK** (or press Enter) to apply the new value

► Click **OK** again to close the edit window

You can see how much easier it is to enter data here in real-world "engineering units," without having to translate into a different radix such as hexadecimal or binary.



Drag and drop to copy Latitude field to another VL



Engineering Units Editor for Latitude field

Run the Simulation

► Hover the mouse pointer over the **Latitude** field on VL 200 to view the engineering units tool tip display, which should read **47.349 degrees**

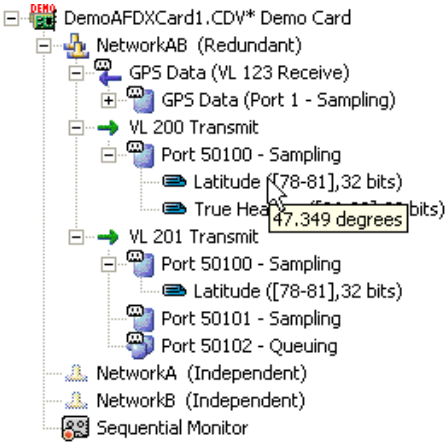
► Click the **Run** button to start the simulation

You may double click a field to edit the data rather than a right click to open the context menu then selecting Edit Data. A double click performs the object's default action (the **bold** context menu entry). To edit the Latitude field:

► Double click the **Latitude** field on VL 200 and enter the value of **47.35**

► Hover the mouse pointer again over the **Latitude** field on Port 50100 of VL 200 to view the value equal to **47.35 degrees**

► Click the **Stop** button to end the simulation



Lesson 5 Demo card tree with tool tip display for Latitude fields

Related Topics

- There are several available interpreters including: BNR, BCD, Boolean 32, Custom Script, Float 32, Float 64, Integer 32, Integer 64, Opaque–Fixed Length, Opaque–Variable Length, and String
- Data fields may be viewed through CoPilot display windows such as the Port View, Engineering View (Lesson 9) and Strip View (Lesson 10)
- The Engineering Units Editor may also be closed by clicking outside the window, pressing Enter, or pressing Esc

Summary

In this lesson you learned…

- how to create a new data field
- how to change interpreter properties
- how to copy fields to other ports
- how to modify field data using the Engineering Units Editor

Now that you have created and defined a data field, in Lesson 6 you learn how to save this field to the database and use the database to populate the hardware tree.

Lesson 6: Use the AFDX Database

Fields are used to define interpretation of payload data into human readable engineering units and strings. Fields are defined on ports or Datasets (DS) if Functional Datasets (FDS) are used. The following lesson covers FDS and DS. CoPilot allows fields and the configuration information of other objects to be stored in the CoPilot AFDX database. That information can be shared and reused—allowing data to be set and viewed in a familiar engineering units format.

Introduced in This Lesson

Save to Database dialog, Field Configuration dialog (Select from Database), Add VL from database

Objective

Learn how to save the Latitude field on Port 50100 (from Lesson 5) to the database then load the field as Longitude after modifying a few properties. Expedite the configuration process by loading a sample VL from a saved VL definition.

Save a Field and VL to Database

Note: Before you begin this lesson, open the AFDX Lesson 6 project (File | New | New Project | AFDX Lessons).

To save the Latitude field to the AFDX database:

- Right click on the **Latitude** field on Port 50100 of VL 200 and choose **Save to database** to open the Save to Database Options dialog
- Type in **Lesson 6: Latitude ([78-81],32 bits)** as the Comment and press **OK**, then **OK** when complete

To save VL 200 to the AFDX database:

- Right click **VL 200** and choose **Save VL to database** from the context menu to open the CoPilot Database Save dialog
- Type in **Lesson 6: VL 200 Transmit** as the Comment and press **OK**

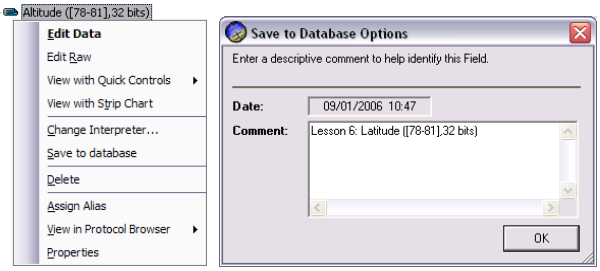
Load a Field from the Database

Loading a predefined data field from the database uses the same process as adding a field in Lesson 5, except from the Field Configuration dialog, you will select the definition from the database. The saved latitude field will be loaded as longitude.

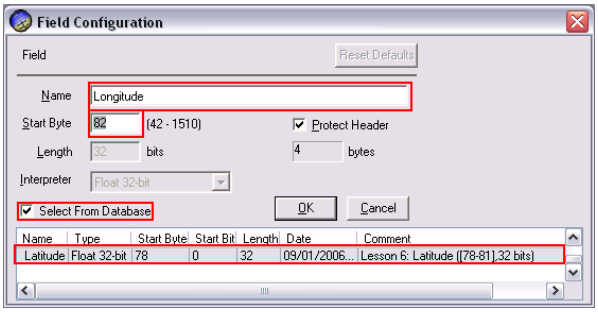
- Right click **Port 50100** of VL 200 and select **Add Field**
- Click the **Select From Database** check box
- Scroll down to select the **Latitude** field data with the comment from the previous section:
  Lesson 6:Latitude ([78-81],32 bits)

Note: Selecting an item from the database initializes items with the same parameters that were originally saved. These values may be changed when the new items are loaded.

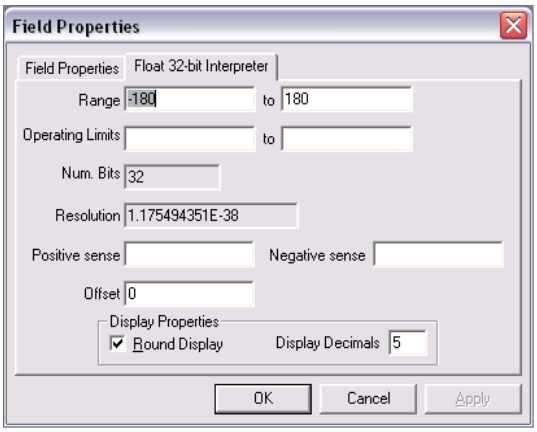
- Change the name from Latitude to **Longitude**



Field context menu and Save to database dialog



Load Latitude definition from database for adding Longitude field



Changing interpreter range of Longitude field

- ► Change the Start Byte to **82** and press **OK**
- ► Click the Float 32-bit Interpreter tab when the Field Info Properties dialog appears
- ► Set the Range from **-180** to **180**
- ► Press **OK** to finish loading the field

Load Complete VL Definition

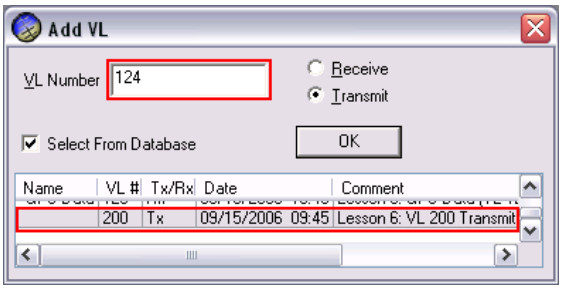
CoPilot allows loading of objects at the field, port, VL, and network levels. You could continue to load fields individually as above, but the quicker method is to save a larger collection of objects and load that. To load the VL saved earlier in this lesson:

- ► Right click on **NetworkAB** and choose **Add VL** to open the Add VL dialog
- ► Check the **Select From Database** option to display the list of available VLs to load
- ► Scroll down to select **VL 200** with the Comment **Lesson 6: VL 200 Transmit**
- ► Change the VL number to **124** and click **OK**
- ► Change the Name to **Test Load** on the VL Properties page
- ► Click **OK** to close the properties window

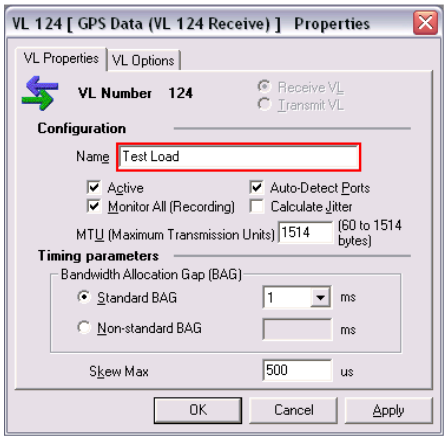
The hardware tree now has VL 124 defined with its associated ports, FDS, DS, and fields. Other items saved in the database can be used not only for configuring VLs, but also for configuring networks, ports, and fields.

Loading this VL 124 from the database was done as an example. The VL will now be deleted from the hardware tree. To delete VL 124:

- ► Right click **Test Load VL 124** and select **Delete VL**
- ► Click **OK** to confirm the deletion



Load VL definition to VL 124 from database



Test Load VL 124 configuration

Confirm the Test Load VL 124 deletion

Related Topics

- You can select the default database through the AFDX Properties tab in the Project | Options dialog
- Networks, VLs, ports, and fields are saved with their children independently of other definitions at the same level (i.e. when the Longitude field was loaded from the database, fields saved as part of VL 200, including the True Heading, were not shown as candidate fields)
- The XML database utility allows importing and exporting of the AFDX database in XML format

Summary

In this lesson you learned…

- how to save a field to the AFDX database
- how to load a field from the AFDX database
- how to load a complete VL from the database to an alternate VL number

You have now populated the hardware tree from the database and you have a variety of data fields to work with. Future lessons will group these data fields into FDS and DS, and modify the field values.

Lesson 7: FDS and DS Usage

The AFDX (and ARINC 664 Part 7) specification contains provisions for logically organizing data within the AFDX payload. Functional Datasets (FDS) contain one to four (1-4) Datasets (DS) and have an associated 4-byte Functional Status Set (FSS). A FSS is divided into four parts for each of the four possible Datasets in a FDS. Datasets contain data primitives - field definitions – for engineering unit interpretations.

Introduced in This Lesson

FDS Configuration dialog, DS Configuration dialog, Port View configuration, Port View field summary

Objective

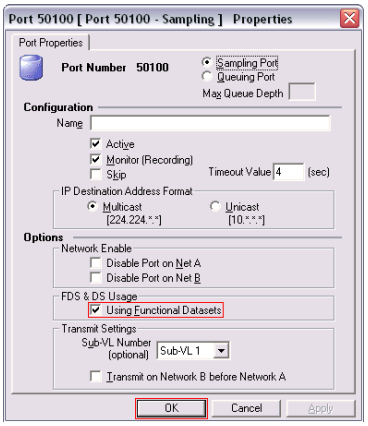
Enable FDS usage for Port 50100 on VL 200 then define a FDS with one DS. Edit the status of the DS using the Port View then hide FDS definitions.

Configure Port for FDS Use

Before you can insert functional dataset information for a port, the port must be configured for functional dataset usage. Each port individually controls its own FDS usage setting from the port's property page.

- ► Right click **Port 50100** on ➔ VL 200 and then select **Properties**
- ► Click **Using Functional Datasets** and press **OK**

FDS and DS items can now be inserted on the port.

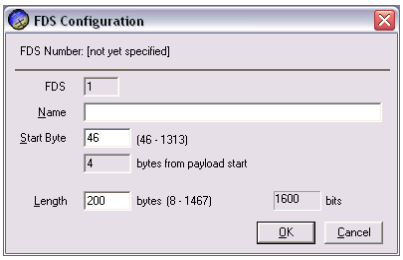


Enabling functional dataset usage on a port

Insert FDS

FDS structures are located below ports in the hardware tree. To insert a FDS:

- ► Right click **Port 50100** on ➔ VL 200 and then select **Insert FDS**
- ► Leave the Name blank (may be set if desired)
- ► Set the Start Byte to **46** so the FDS starts after the default reserved bytes
- ► Set the Length to **200**
- ► Press **OK** to insert FDS 1

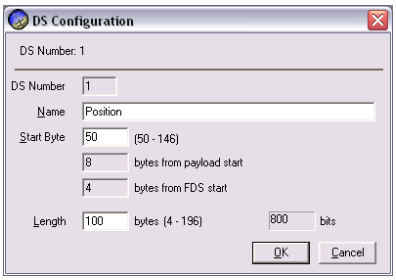


FDS configuration dialog for inserting a FDS

Insert DS

DS structures are located below FDS in the hardware tree. A FDS may contain up to four DS. Fields defined on a port within a dataset's boundaries are moved to the dataset. To insert a DS and move the defined fields:

- ► Right click the FDS previously inserted above and select **Insert DS**
- ► Set the name to **Position**
- ► Set the Start Byte to **50** and the Length to **100**
- ► Press **OK** to insert DS 1



DS configuration dialog for inserting a DS

Functional Status Values

The functional status of a DS is contained as part of the functional status set. The status is displayed and edited using the Port View. Additionally, the functional status for a DS can be seen in the DS tooltip.

► Right click **Position (DS1)** on Port 50100 and select **View DS**

Viewing a DS when not running opens a Port View in the configuration mode with the DS item selected. Fields defined on the DS are shown in the display. Right clicking on items in the port view when in this mode shows available actions. The DS status as well as field engineering unit values are displayed and edited from the Fields item of the Summary mode.

► Change the mode to **Summary**

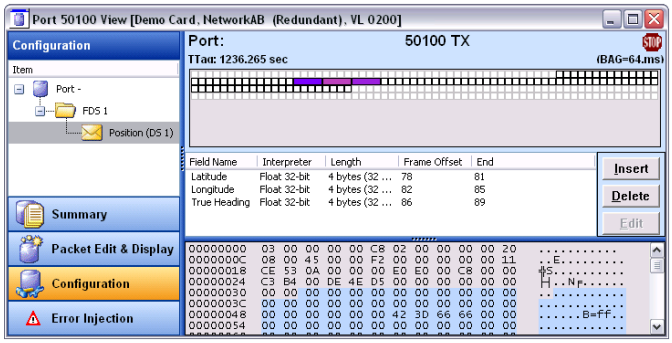
► Click the Fields item in the upper left window to display the field list

The main window displays the datasets with their defined fields. The Status column contains the DS status.

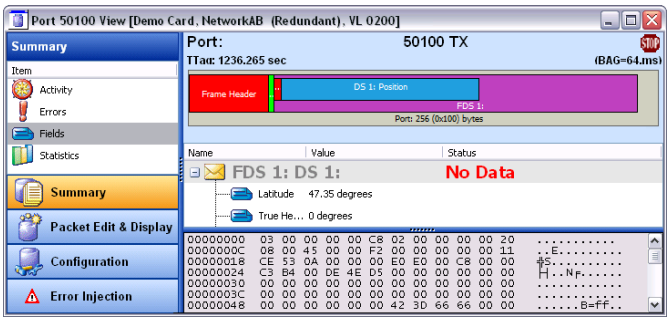
► Right click the desired DS item to show the context menu, then select **DS Status**

► Select **No Computed Data** from the sub-menu

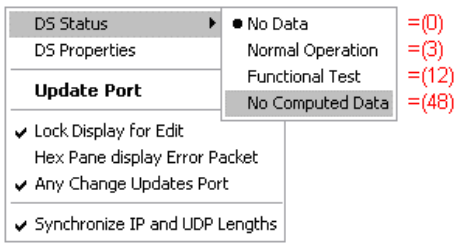
► Press **OK** to enable editing and make the change



Port View configuration mode displaying Position DS 1 fields



Port View field summary with FDS and DS information



Editing DS status with a Port View

Hide FDS

Turning off FDS usage hides FDS and DS items from the hardware tree while the definitions still exist. Fields defined on a DS are moved to the port. To hide the FDS and DS information:

► Right click **Port 50100** on VL 200 and select **Properties**

► Uncheck **Using Functional Datasets** to disable FDS items in the hardware tree and press **OK**

Related Topics

- Port Views is not only used to display and edit data, but to configure FDS, DS, and fields for a port
- The Packet Edit & Display mode of the Port View displays the current values and allows in-place editing
- Editing transmit frames in a Port View is enabled by right-clicking the Port View to lock the display for editing
- A ports frame data may be edited using the Hex display pane of the Port View when editing is enabled

Summary

In this lesson you learned…

- how to insert FDS and DS on a port
- how to view configuration and edit the DS status using a Port View
- how to hide FDS and DS configurations from the hardware tree

In Lesson 8 you will learn how to modify AFDX data values at runtime with the Data Generator and in-place editors.

Lesson 8: Generate Dynamic Data

In the previous lessons, the data values for the messages and fields were static. In this lesson you will learn how to create a continuous stream of changing data values using the CoPilot Data Generator. This dynamic data will be used to drive several graphical displays in Section D (CoPilot AFDX Professional). You will also use the Raw Editor to edit data in other radices while comparing against the engineering units value.

Introduced in This Lesson

New Component dialog, Data Generator view, Data Generator Properties dialog, Raw Editor

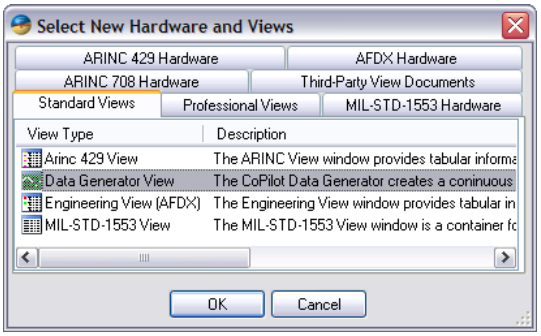
Objective

Use the Data Generator to drive dynamic data values in the Latitude, Longitude, and True Heading fields. Use the Raw Editor to change the current value of the True Heading field in the VL 200, Port 50100.

Open the Data Generator

The Data Generator can create dynamic data streams for (transmit) fields before or during simulation. The Data Generator is accessed from the New Component dialog (see the Section B introduction for more information on using view components).

- Click **New** or choose **File | New | New Hardware or Views…** to open the New Hardware and Views dialog
- Select the **Standard Views** tab, choose **Data Generator**, and click **OK**



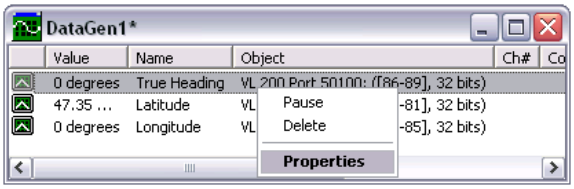
New Hardware and Views dialog with Data Generator selected

Assign Data Fields to Generator

The Data Generator can accept transmit fields.

- Drag the **True Heading** field on VL200 in the hardware tree into the Data Generator
- Open **Properties** for **Port 50100,** check **Using Functional Datasets** and press **OK** (see Lesson 7)
- Drag the **Position (DS1)** dataset on VL200 in the hardware tree into the Data Generator

Dragging objects parenting fields, such as DS1, into the Data Generator adds all fields beneath that object.

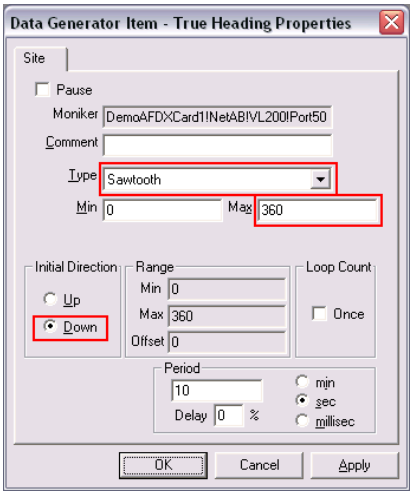


Data Generator with fields showing context menu for True Heading

Set Field Properties in Generator

- Right click on the **True Heading** row in the Data Generator and choose **Properties** to open the Data Generator Properties dialog
- Change the Data Generator type to **Sawtooth**
- Set the Min range to **0** and the Max to **360**
- Choose **Down** in the Initial Direction frame
- Click **OK** to close the Properties window

This will cause the heading to rotate full circle in a continuous loop. You will use this to create a circular hold-



Data Generator Properties for True Heading

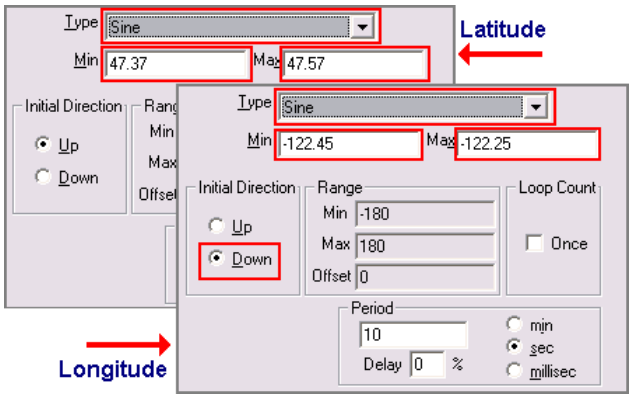
ing pattern in a moving map display in Lesson 14.

- ► Double click on **Latitude** in the Data Generator to open the Data Generator Properties window
- ► Change the Generator type to **Sine**
- ► Enter **47.37** in the Min box, **47.57** in the Max box, set initial direction to **Up** and click **OK**
- ► Double click on **Longitude** in the generator, set the type to **Sine**, the Min to **–122.45**, the Max to **122.25**, initial direction to **Down**, and click **OK**

These values will be used to position the airplane object in the moving map display in Lesson 14.

- ► Click the **Run** button to start simulation mode

Values in the Data Generator update while the bus is running (within the ranges you have set).



Latitude properties (left) and Longitude properties (right)

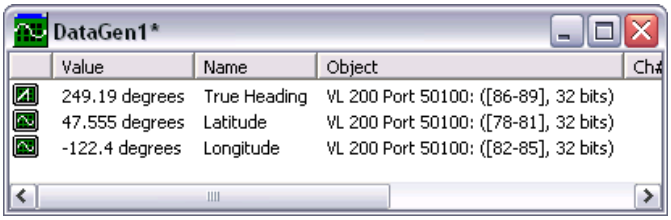
Use the Raw Editor

The Raw Editor allows you to edit data (before or during a run) in binary, octal, or hexadecimal radices, and compare those values to engineering units.

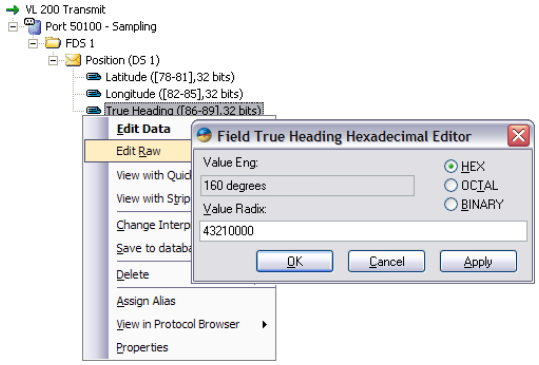
- ► Right click the **True Heading** field in the Data Generator and select **Pause**
- ► Right click the **True Heading** field in Port 50100 on VL 200 then choose **Edit Raw**
- ► Click the **Hex** button and enter the value **43210000** (hex) and click **Apply**

You will see the translation of the hexadecimal value into engineering units (161 degrees) in the Value Eng. Field of the Hexadecimal Editor.

- ► Click the buttons for **Octal** and **Binary** to view this value in each of those radices
- ► Click **OK** to close the Raw Editor
- ► Click the Stop button to end the simulation
- ► Note: If you are maintaining your own project, **Save** and **Minimize** the Data Generator.



Data Generator displaying the current data values



Raw Editor opened from the True Heading context menu

Related Topics

- The eight Generator patterns are ramping, sawtooth, sine, approach, square, data list, random, and complex ramp
- Data Generator properties may be changed and applied in real time while the simulation is running

Summary

In this lesson you learned…

- how to open the Data Generator
- how to add data fields to the Data Generator
- how to configure data generator properties for individual fields
- how to use the Raw Editor to modify the current value in various radices

In the next lesson you will learn how to view VLs, ports and fields in special display containers.

Lesson 9: View and Organize Data

The Engineering View is a window for displaying VLs, ports, and their related data fields. Items assigned to this container are dynamically updated while the project is running. Because the display columns in each Engineering View window can be custom-selected, multiple Engineering Views could each be individually configured to focus on specific types of messages or data.

Introduced in This Lesson

Engineering View, Select Columns to Display dialog

Objective

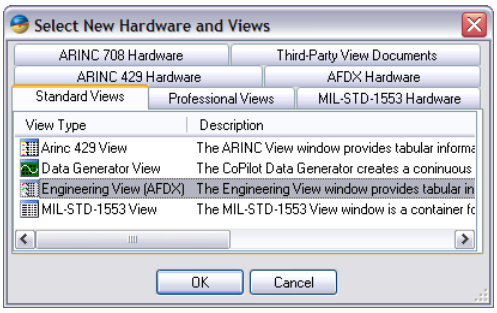
Open an Engineering View window and assign the three data fields linked to the Data Generator. Customize the display by moving, sorting, adding columns, and assigning another data field. Finally, edit a field from within the View window.

Open the Engineering View

- Click **New** or choose **File | New | New Hardware or Views…** to open the New Hardware and Views dialog
- Select the **Standard Views** tab, choose **Engineering View,** and click **OK**

The Engineering View component is now present in the display area on the right side of the CoPilot workspace.

Note: If you are working from your own project instead of opening this lesson from the New Project dialog, be sure the Data Generator is present and configured to drive the data fields (see Lesson 8).

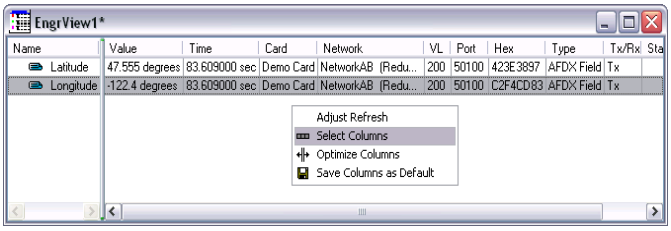


AFDX Views tab in the New Component dialog

View Field Data Dynamically

- Drag and drop **Latitude** and **Longitude** from Port 50100 on VL 200 into the Engineering View
- Click the **Run** button to start the simulation

View the changing data values in the Value column.



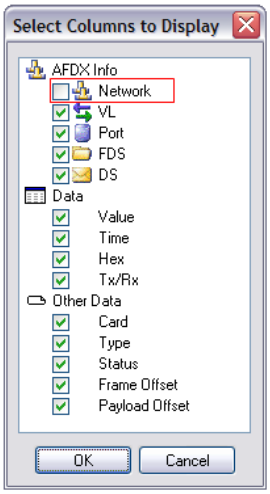
Engineering View window with three data fields and context menu

Customize the Display

In the Engineering View window, you can limit the display to information of interest. Columns widths may be adjusted or reordered for better viewing.

- Right click inside the Engineering View window (below the rows) and choose **Select Columns** from the context menu to open the Select Columns dialog
- Clear check mark for **Network**, then set the check marks for all other columns to select those columns, and click **OK**

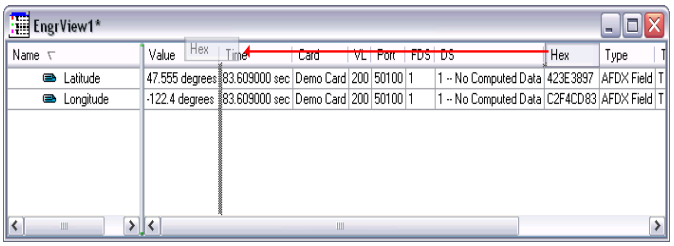
(These and any other column can be restored in the same way.) Now the Network column is hidden and the FDS and DS columns are shown.



Select Columns dialog

► Click and hold on the **Hex** column title and drag it to the left until the space between Value and Time becomes highlighted, then release

Column width can be resized by dragging the right edge of a column title or by placing the mouse along the right edge of the column title (the pointer changes to a crossbar ✢) and double clicking.



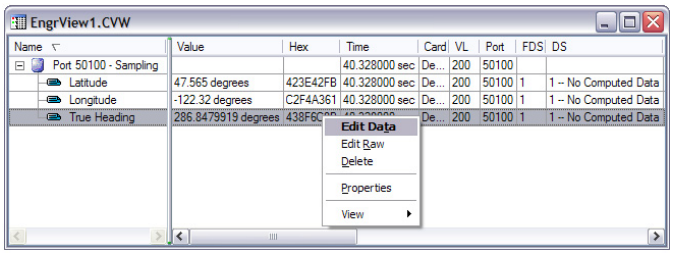
Time column being dragged between Value and Double Value

Add Items During Simulation

► Drag **Port 50100** on VL 200 into the Engineering View window to add the port and heading field

The fields are now all displayed as part of Port 50100. The True Heading field is static because it is paused in the Data Generator.

This action can be duplicated at any level in the hardware tree to add all possible items below what is dropped into the Engineering View.



Engineering View with all Port 50100 fields (and True Heading field context menu)

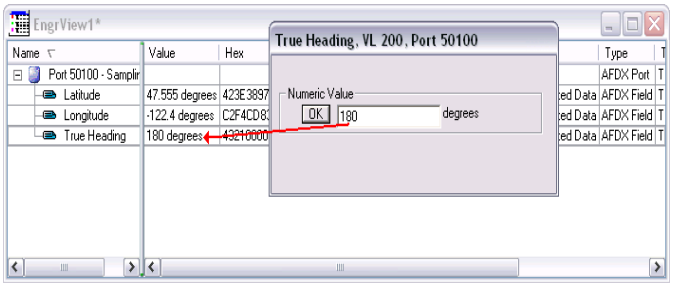
Edit Objects in Engineering View

► Right click on the **True Heading** field in Engineering View and choose **Edit Data** from the menu to open the Engineering Units Editor (see figure)

The objects in the Engineering View are linked to the actual object in the hardware tree. Changes made to an object are applied to the tree object in real time.

► Enter **180** and press **Enter** to accept, then click **OK**

The display will update in real time with the new value applied to the field in the hardware tree and the connected displays.



The True Heading field being modified within the Engineering View

► Click the **Stop** button to end the simulation

Related Topics

- Multiple Engineering View windows can be used to logically group related objects
- Right clicking on column titles allows for quick removal by hiding the displayed column
- Clicking once on the title bar sorts the rows alphabetically by the contents of that column
- Adjust Refresh from the Engineering View context menu allows changing the refresh rate (default = 2 Hz)

Summary

In this lesson you learned…

- how to open the Engineering View window
- how to add objects individually or in groups
- how to configure and order the column display
- how to edit objects directly in Engineering View

In the next lesson you will learn about the statistics calculations and displays.

Lesson 10: Statistics Calculation and Display

Statistics are useful for knowing how the system is performing. Network and VL statistics were introduced in Lesson 1. This lesson focuses on viewing port statistics and Jitter calculations.

Introduced in This Lesson

Port View Summary, Port View Activity, Port View Errors, VL jitter calculation, Jitter Wizard, independent networks

Objective

Open a Port View window to view the port's activity, errors, and statistics. Next, perform a jitter measurement, and finally use the Jitter Wizard with independent networks.

Viewing Port Statistics

Note: Before you begin this lesson, open the AFDX Lesson 10 project (File | New | New Project | AFDX Lessons).

- ► Click the **Run** button to start the simulation
- ► Right click **Port1** (VL123) and select **View Port**
- ► Click the **Summary** mode button, then select the **Statistics** item to display statistics

Port Errors and Activity

The error and activity information from the previous frame are available in the Port View. Frames that cause errors are usually due to incorrect headers.

- ► With the Port View in the Summary mode, select the **Errors** item to display errors

The activity list contains timing, redundancy, integrity, RSN and other information. To view the port's activity in a Port View:

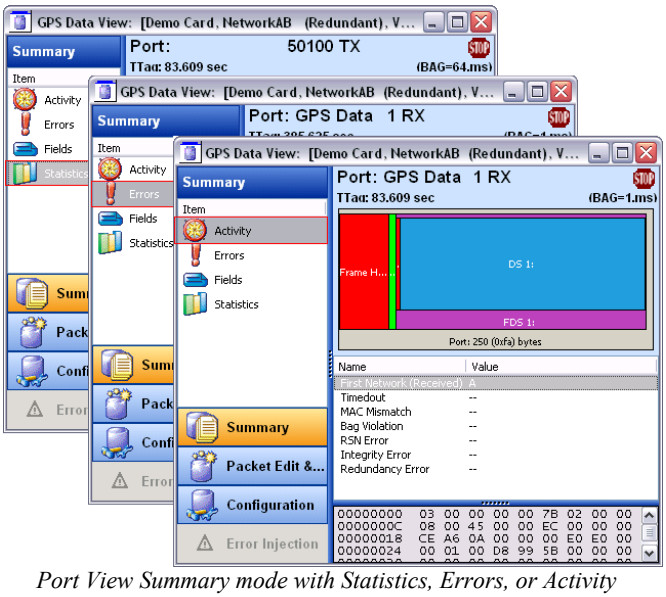
- ► Click the **Summary** mode button, then Select the **Activity** item to display activity

Jitter Calculation

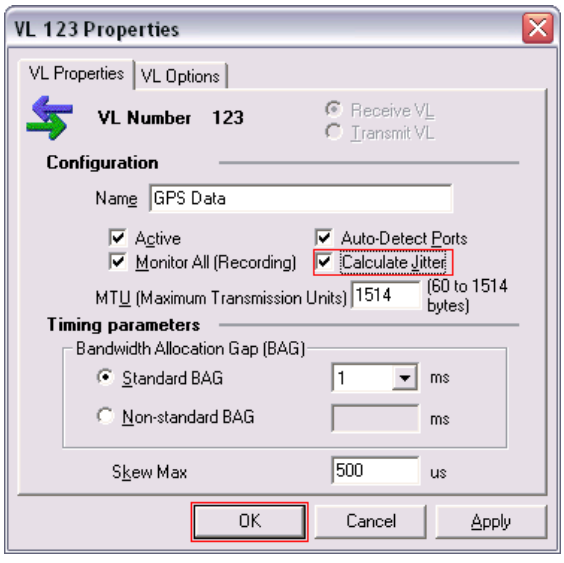
Jitter is a statistic that measures the variation between the expected reception time of a frame and the actual time the frame was received. Jitter is caused by network congestion, timing drift, route changes, and hardware implementations. AFDX hardware maintains the last jitter value and the maximum jitter value.

Jitter is calculated at the receive VL level by comparing the time between actual frames with the expected reception interval (the BAG). To enable jitter calculation on a receive VL:

- ► Click the **Stop** button to end the simulation
- ► Right click **VL123** and select **Properties**



Port View Summary mode with Statistics, Errors, or Activity



Enabling jitter calculation for a receive VL

► Click the **Calculate Jitter** option to enable jitter calculations and press **OK**.

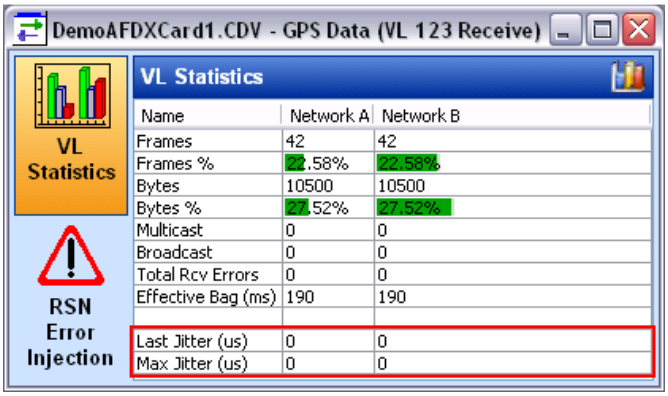
► Right click the receive **VL123** and select **View VL** to see the jitter values

► Click the **Run** button to start the simulation

The Last Jitter and Max Jitter values are displayed with the VL Statistics in the VL View.

Note: Jitter calculation is a function of the AFDX hardware and is not actually measured by the AFDX demo card. The last jitter and max jitter values are always zero with the demo card.

► Click the **Stop** button to end the simulation



Jitter statistics values in the VL View

Jitter Wizard

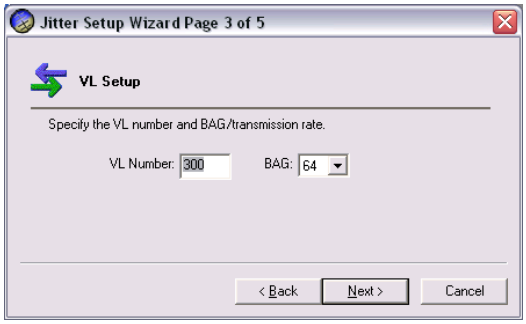
CoPilot has an automated feature for jitter testing. This feature is the Jitter Wizard. The Jitter Wizard puts the card into independent network mode where NetworkA operates independently from NetworkB. A transmit VL is then configured on one network with the corresponding receive VL on the other network. The transmitter is then set to transmit at the same rate (BAG) as the receive VL. Run the Jitter Wizard by:

► Right click the **Demo Card** and select **Jitter Wizard** then press **Next** in the Jitter Wizard

► Press the **Next** button again to proceed

► Enter **300** for the VL Number and press **Next**

► Enter **54321** for the Port Number and press **Next**, then press **Finish**



Setting the VL number in the Jitter Wizard

Setting the Port number in the Jitter Wizard

Note: Using real hardware with the Jitter Wizard, you can directly connect the two networks with a crossover cable to measure the Jitter of the AFDX hardware. Alternatively, connect the networks through switches to determine how paths and loading affect the jitter.

► Right click **NetworkAB** and select **Enable** to enable the redundant network

Related Topics

- Port View and VL View statistics may be reset by right clicking on the statistics data and selecting Reset Statistics
- Enabling an independent network or using the Jitter Wizard will disable the redundant network (NetworkAB)
- Jitter measurements assumes the source VL is transmitting at the BAG interval of the receive VL

Summary

In this lesson you learned…

- how to view port statistics, activity, and errors with a Port View
- how to configure receive VLs to calculate jitter
- how to use the Jitter Wizard with independent networks

In the next section you will learn about the optional features available with CoPilot Professional.

THIS PAGE INTENTIONALLY BLANK

Section C: CoPilot Professional

In Section B you learned how to create data fields and interpreters, assign "engineering units," save and download those data definitions from a database, modify values through data editors and the Data Generator, and view that data in the Engineering View display window. You have now completed the CoPilot AFDX Standard portion of this guide. The remaining five lessons introduce you to the advanced features available with CoPilot AFDX Professional. This includes a variety of graphical displays (discussed in Lessons 11 through 13), a scripting engine for customizing CoPilot (Lesson 14) and software playback, a powerful post-analysis tool (Lesson 15). Refer to the *CoPilot User's Manual* for additional Professional features not discussed in this guide such as ATE (Automated Test Environment) Test Manager, the Script Debugger and the Command Prompt.

CoPilot Professional

CoPilot is licensed as either CoPilot Standard or CoPilot Professional. CoPilot Professional contains all the features of CoPilot Standard plus powerful graphical displays, ATE (using Python scripting technology), Test Manager, and Visual Basic Scripting. CoPilot Professional features described in this getting started guide include…

- graphical strip chart displays (Strip View, Lesson 11)
- a library of virtual controls and aircraft instruments (Quick View and Control View, Lessons 12 and 13)
- moving map displays (Map View, Lesson 14)
- flexible script routines used to extend the capabilities of CoPilot for running advanced simulations, generating complex data, creating reports, running tests and more (Python Code Editor, Lesson 15)
- VBScript routines (Script View, Lesson 16)

Professional Mode

To run CoPilot Professional components, CoPilot requires all hardware devices (included in the project) to have Professional license keys. The AFDX Demo card is enabled with Professional capability (and does not require a physical hardware key). Unkeyed hardware devices will not be run when professional components are part of a project.

Professional Views Toolbar

The Professional Views Toolbar provides quick access for creating new display views. The Professional Views Toolbar and other toolbars, can be shown/hidden from the **View** | **Toolbars** menu item. Users can also customize and add additional toolbars as described in the CoPilot User's Manual.

Introduced in This Section

The lessons in this section demonstrate the professional features available in CoPilot Professional. Professional features include graphical display components and scripting capabilities.

- Lesson 11 Strip charts can be used to display a history of one or more streams of data.
- Lesson 12 Aircraft instruments and displays are automatically generated from fields.
- Lesson 13 Simulated controls can also be custom designed and explicitly linked to a field values.
- Lesson 14 Positional data may be used to display an aircraft position on a moving map display.
- Lesson 15 CoPilot can be customized and controlled through script routines that access and modify the properties of CoPilot objects.

Lesson 11: Create a Strip Chart Display

The Strip View window is a dynamic, two-dimensional charting control for displaying one or more streams of real-time data. The zoom function allows for quick review of the dataset and in-depth analysis of data trends. Strip charts can be launched from individual fields during a simulation.

Introduced in This Lesson

Strip View, Strip View Properties dialog

Objective

View two fields in Strip View during simulation, configure the display, and review the dataset.

Launch a Strip Chart

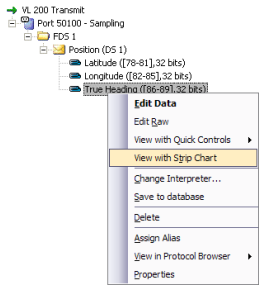
Note: Before you begin this lesson, open the AFDX Lesson 11 project (File | New | New Project | AFDX Lessons).

- ► Click the **Run** button to start the simulation
- ► Open the **DataGen1.CVW** and right click on **True Heading**, then click **Pause** to clear the Pause state
- ► Right click on the **True Heading** field on VL 200 and choose **View with Strip Chart**

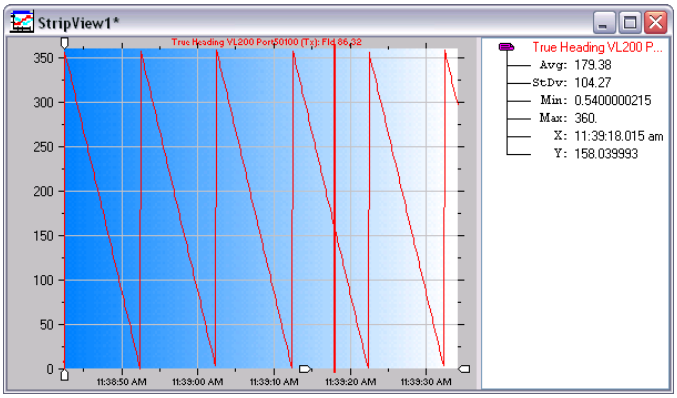
A Strip View window automatically opens and begins charting the True Heading field. Strip Views can also be opened from the New Component dialog (when the simulation is not running).

- ► Resize to match figure at right (for better viewing)

Notice the average, standard deviation, and minimum/maximum statistics displayed in the right half of the Strip View window. When you click in the chart area (mouse pointer will become a small hand icon), the X and Y fields report the time of day and data coordinates marked by the cursor (see figure).



The Strip Chart command in the field context menu



Resized Strip View window with True Heading field

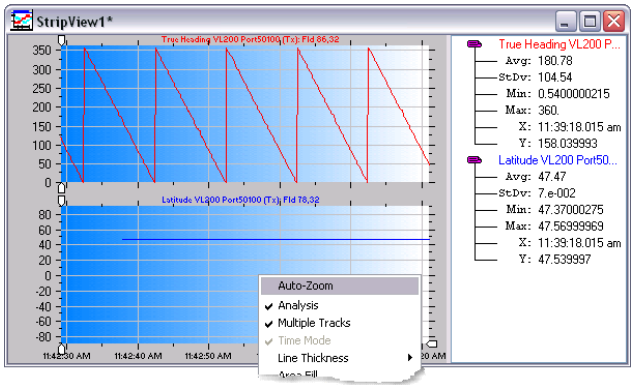
Add a Field while Running

- ► Drag and drop the **Latitude** field from VL 200 (in the hardware tree) into to the Strip View window

Now each data value is tracked in a separate chart. Notice that the y-axis for each track reflects the minimum and maximum values defined in their respective data field interpreters. You can also choose to display all fields in a single track in the Strip View context menu.

Zoom in on Data Ranges

The Latitude chart area is difficult to read because the actual data values cover less than one percent of the 180-degree range defined by the interpreter.



Strip View with Latitude field and context menu

► Right click in the Latitude chart area and choose **Auto-Zoom** (see figure on previous page)

Only the range of actual values is now seen (notice the new scale in the y-axis). You can also adjust the zoom by dragging the zoom buttons along the y-axis.

Modify Strip View Properties

In addition to the context menu commands, strip charts can be customized through the Strip View Properties dialog.

► Right click on the chart area and choose **Properties** from the context menu

► In the Properties dialog, click the **Area Fill** check box and click **Apply** to view the results

The amount of detail and smoothness of lines is affected by the *buffer size* and *sample rate* (these can be set in Edit mode). The background can be a gradient fill as shown, or a single color of your choice.

► Click **OK** to close the properties dialog

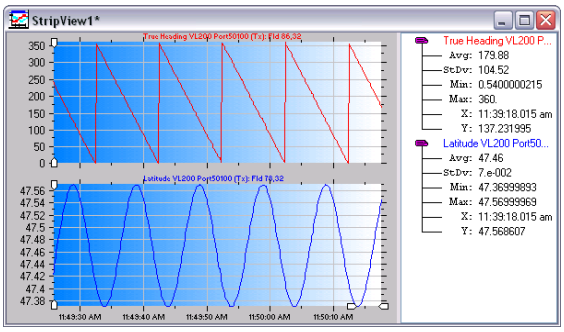
Analyze Chart after Simulation

► Click **Stop** (after running the simulation long enough for the chart to have a sizeable sample)

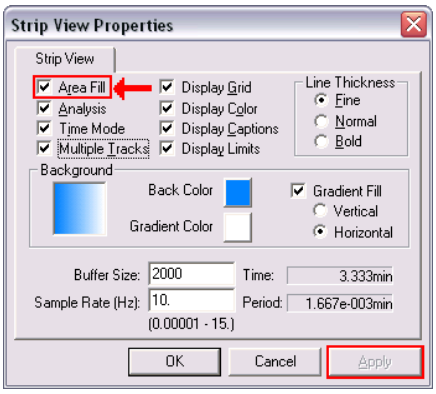
► Position your mouse near the left handle on the x-axis (bottom) until it changes to a double arrow, then drag back and forth (both handles move together) to view the history of data values

Examine the data statistics in the right-hand pane. Compare these values to the settings created for these fields in the Data Generator in Lesson 8.

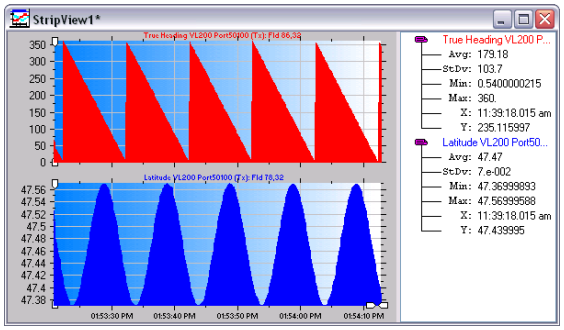
As you review the data history, you can click to place the cursor and examine the X and Y statistics to determine the exact data value and time of day at the cursor point.



Strip View after Latitude track has been auto-zoomed



Strip View Properties window



Strip View with changes from Properties window

Related Topics

- There is also a strip chart control in the Control View window (Lessons 11 and 12 discuss Control View)

Summary

In this lesson you learned…

- how to open a Strip View window
- how to add data fields
- how to configure the display and modify Strip View properties
- how to view the data history after a run

In this lesson you were introduced to Strip View, the first of several CoPilot Professional graphical display components. The next two lessons will introduce Quick View and Control View, two windows that host graphical ActiveX controls.

Lesson 12: Generate Controls Automatically

A host of graphical controls and virtual instruments are available in CoPilot for displaying and modifying data. In this lesson, you will learn how to generate automatic controls in Quick View and Control View from data fields in the hardware tree. Quick Views are pre-configured controls that can be launched in one step from data fields in the hardware tree. Control View is a display window that can host multiple controls and instruments. You can add controls to Control View automatically (Lesson 12) or build and link custom controls manually (Lesson 13).

Introduced in This Lesson

Quick Controls submenu (for data fields), Quick View, Control View

Objective

Automatically create an engineering unit Quick Control and modify the control properties. Next, generate a slider Quick Control from a field and configure it as a source to edit data values. Then open a Control View window and add a slider control. Modify the field value and observe the results in the engineering unit Quick View.

Create Quick View Sink Control

Note: Before you begin this lesson, open the AFDX Lesson 12 project (File | New | New Project | AFDX Lessons).

The easiest way to quickly view or modify a data field graphically is through CoPilot Professional Quick View. First, you will create a Quick View as a sink for displaying the current value of a field.

- Right click on the **True Heading** field, point to **View with Quick Controls** to expand the submenu, and click on **Engineering Units** (see figure at right)

A Quick View engineering unit control will open in the display area. The value in the control reflects the current field value as it is modified by the data generator.

Customize Quick Controls by changing the properties.

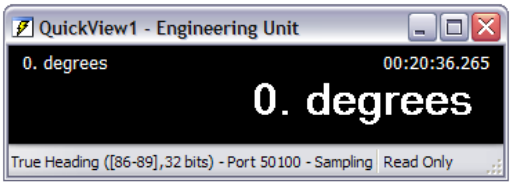
- Right click the background of the **Engineering Unit** control and select **Properties**
- Click **ForeColor** then click **Green** from the Color Palette selection
- Press **OK** to close the Engineering Unit Properties dialog window

There are a variety of additional Quick Controls available for displaying different types of data. For example, a two-state LED or Toggle control could be selected to represent a discrete value. The Slider control will display or modify the field value as determined by the source/sink mode.

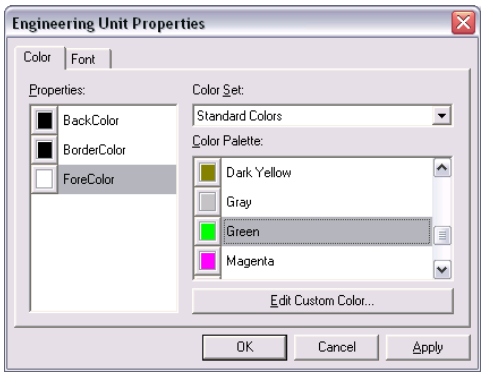
Edit Using a Quick View Control

Now that you have used a Quick View to display data, you will create a Quick View slider to source data.

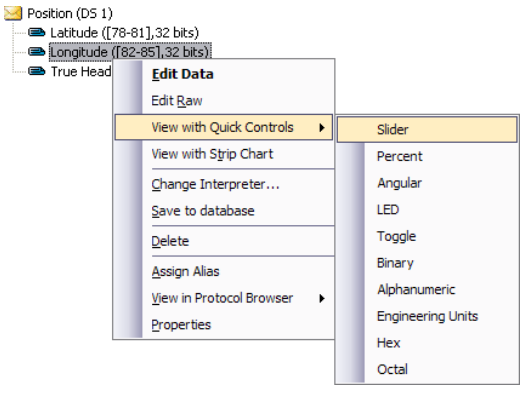
- Click the **Run** button to start the simulation
- Open the **DataGen1.CVW** and right click on **True Heading**, then click **Pause** to set the Pause state



Quick View Engineering Units control for True Heading



Quick View Engineering Units control properties

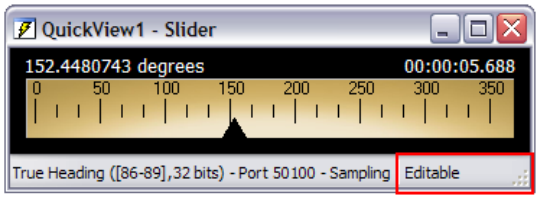


The Quick View controls menu for True Heading field

► Right click on the **True Heading** field, point to **Quick Controls** to expand the submenu, and click on **Slider** (see figure)

A Quick View slider control will open in the display area. The slider defaults as editable for transmit data.

► Click the Quick View slider needle value and drag it left and right (the value is set when released)



Quick View slider for True Heading configured as a data source

Generate a Control View Control

Next, let's produce similar results in a Control View.

► Click the **Stop** button to halt the simulation

► Click the **New** button to open the Hardware and Views dialog

► Select the **Professional Views** tab, choose **Control View**, and click **OK**

A blank Control View will open in the display area.

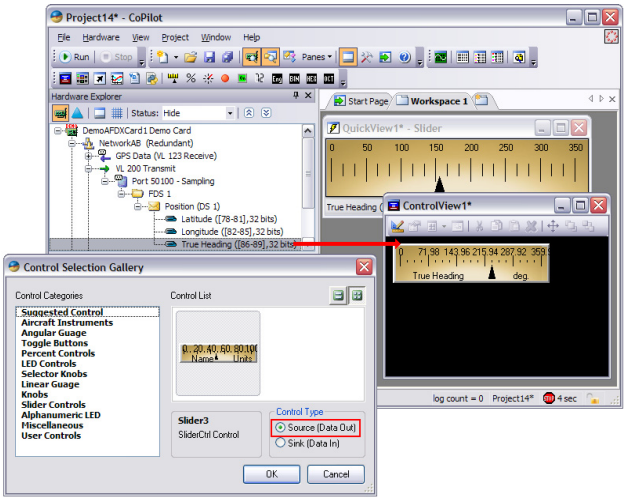
► Click the **Run** button to start the simulation

With the simulation running, now add a slider control to the Control View.

► Drag and drop the **True Heading** field from **Port 50100** on VL 200 into the Control View

The Control Selection Gallery will open

► Set the Control Type to **Source** and accept the suggested control by clicking the **OK** button



Slider control generated from the True Heading field

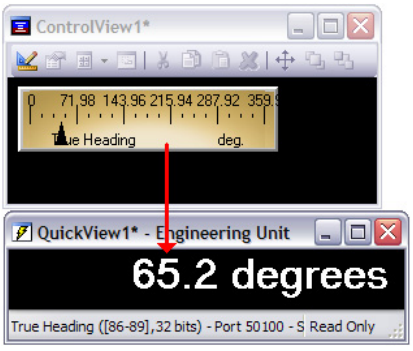
The slider control appears, displaying the current value of the True Heading field. (The upper-left corner of the control is positioned at the where you release the mouse.)

► Click and drag the needle in the slider control up and down to change the True Heading value

A field value change updates the Quick View (sink).

► Click the **Stop** button to halt the simulation

► If you are maintaining a cumulative project, save the Control View window (for the next lesson).



Slider control in Control View used to edit True Heading field

Related Topics

- Control View can also host third-party ActiveX controls
- Dragging a transmit field into a Control View with the <SHIFT> key down creates the control as a sink for display

Summary

In this lesson you learned…

- how to automatically generate a Quick View control as a sink for display
- how to automatically generate a Quick View control as a source for editing
- how to open Control View and add a control automatically

In Lesson 12 you will learn how to create a new control from the library of aircraft instruments included with Control View and link it manually to a data source.

Lesson 13: Simulate an Aircraft Instrument

A library of virtual controls and gauges can be accessed through the Control View window. In this lesson you will select, modify, and manually link a control to an AFDX field.

Introduced in This Lesson

Control Palette, Control Properties dialog, Control Property Browser, Select Source and Sink dialog

Objective

Create and configure a Heading Indicator control and link it to the True Heading field in the hardware tree.

Create a Control Manually

Note: Before you begin this lesson, open the AFDX Lesson 13 project (File | New | New Project | AFDX Lessons).

► Click on the **Design Mode** button in the Control View toolbar

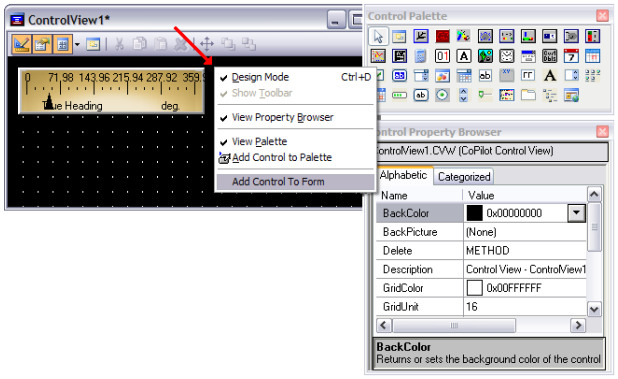
The Control Palette and Control Property Browser appear.

► Right click on the View near the top right corner of the Altimeter control and select **Add Control To Form** from the context menu

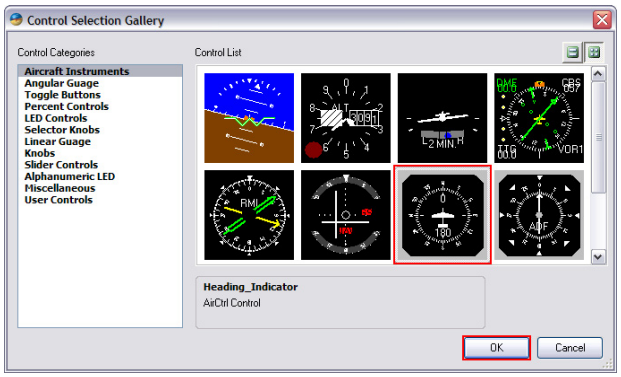
(The upper-left corner of the new control will be positioned at the point where you right clicked the view.) The Control Selection Gallery dialog appears, displaying categories of preconfigured controls.

► Select the **Heading_Indicator** from the **Aircraft Instruments** Category (see figure)

► Click **OK** to close the Control Selection Gallery and add the new control to the view



Open the Control Selection Gallery Window



Control Selection Gallery With Heading_Indicator Selected

Configure Controls in Control View

A convenient way to configure multiple controls (and even the Control View window itself) is to use the Control Property Browser. The Browser contains the same properties as the control property pages, plus additional properties.

► Click on the slider (from Lesson 12) to select it

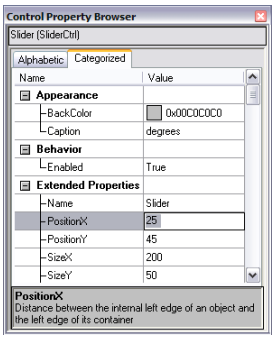
When this control is selected (green and blue handles appear), its properties are loaded into the Control Property Browser and its name (SliderCtrl1) appears at the top of the browser.

► In the Control Property Browser, click on the **Categorized** tab and locate the **Extended Properties** group

► Enter **Slider** in the name field, **25** in PositionX and **45** in PositionY, **200** in SizeX, and **50** in SizeY (see figure)

► Select the Heading Indicator control (AirCtrl2), rename it to **HeadingIndicator** (the scriptable name)

► Enter **230** for PositionX, **45** for PositionY, and **150** for SizeX and SizeY



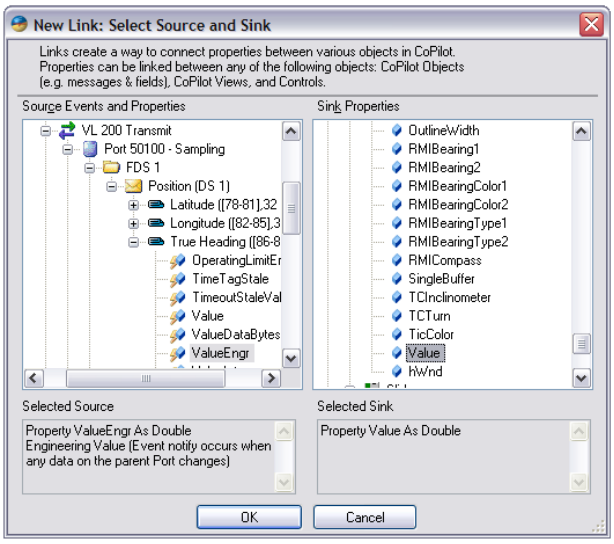
The Slider control in the Control Properties Browser

Now these two controls are positioned and the Heading Indicator is ready to be linked to a data source.

Link Control to Data Source

- ► Click **Project | New Link…** from the CoPilot menu bar to open the Select Source and Sink dialog
- ► In the *Sources* pane, click the plus ⊟ symbol to expand the **DemoAFDXCard1** tree, then expand **NetworkAB**, then **VL 200**, then **Port 50100**, then **FDS 1**, then **Position (DS 1)**, then **True Heading**
- ► Under True Heading, choose **ValueEngr**
- ► In the *Sinks* pane on the right, click the plus ⊞ symbol to expand the **ControlView1** tree, then expand **HeadgIndicator**
- ► Scroll down and select the **Value** property and click **OK** to close the Source and Sink dialog

The ValueEngr property is most often used for data fields and the Value property works for many of the controls. A reference to each control and their properties is contained in the *CoPilot User's Manual* and help documentation.



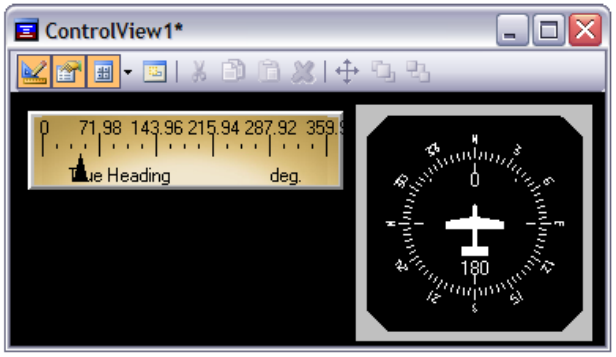
Select Source and Sink dialog linking the field and control

Run the Simulation

- ► Click the **Run** button to start the simulation

Now you can observe heading data through a custom control. There are dozens of other controls and all of them can be customized through control properties. By combining them, you could create a virtual instrument panel with custom links to AFDX fields in the Hardware Explorer.

- ► Open the **DataGen1.CVW** and right click on **True Heading**, then click **Pause** to clear the Pause state
- ► Click the **Stop** button to end the simulation



Linked and operating Heading Indicator control

Related Topics

- The Edit Links dialog (Project | Links Status…) allows you to view, modify, and delete links
- Control View can host third-party ActiveX controls, and you can add buttons for third-party controls to the Palette

Summary

In this lesson you learned…

- how to create a control from the library of aircraft controls
- how to configure controls
- how to link data to the control

In this lesson you have learned how single fields are linked to a control. In Lesson 14 you will learn about a powerful moving map window that displays aircraft position based on links to several data fields.

Lesson 14: Create a Moving Map Display

Map View is a container for displaying aircraft position on a fixed or moving map image. The Map View window links positional data to a real-time moving map display. Normally you would obtain your positional data from receive data fields on an active AFDX databus. This example uses data streams from the Data Generator (established in Lesson 8) to demonstrate how Map View displays incoming data.

Introduced in This Lesson

Map View, Map View Properties, TerraServer Properties

Objective

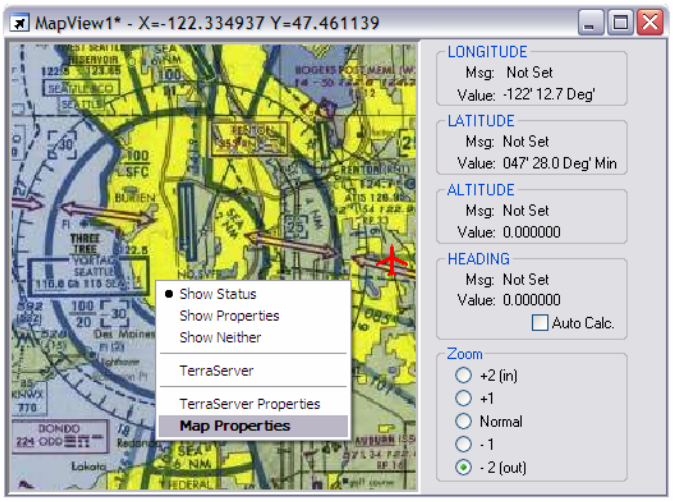
Create a moving map display of a holding pattern by configuring a Map View window, linking positional data fields, and running the simulation. Then, change settings and maps while the simulation is running.

Open a Map View Window

Note: Before you begin this lesson, open the AFDX Lesson 14 project (File | New | New Project | AFDX Lessons).

- ► Click the **New** button to open the New Hardware and Views dialog
- ► Select the **Professional Views** tab, choose **Map View** and click **OK**

The Map View window is divided into the map display (Map pane) and the data links (Status pane). Map View defaults to a Seattle, Washington sectional map with a center point of 47.47° latitude and –122.35° longitude over SeaTac airport. The background bitmap may be changed at any time, but the data values created through the Data Generator relate to this map.



Default Map View showing Map pane, Status pane, and context menu

Link Data and View Moving Map

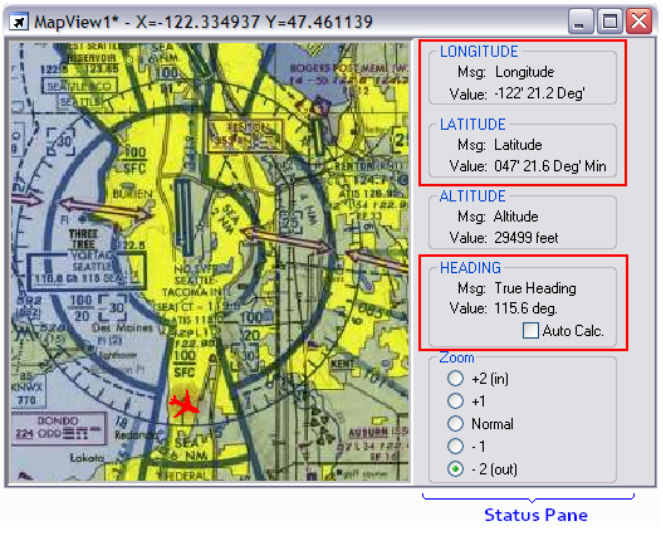
Note: Be sure the Data Generator is present and configured to drive three of the data fields (see Lesson 8). If extra views are open from previous lessons, you can hide them to the Project Explorer with the Minimize button.

- ► Drag and drop the **True Heading** field on VL 200 on to the Heading sink in the Status pane

Notice how the plane object reacts to each new link.

- ► Drag and drop the **Latitude**, and **Longitude** fields from VL 200 in the hardware tree onto their labeled sinks in the Status pane
- ► Resize the Map View by dragging the outside edges and the pane divider to resemble the figure at right
- ► Click the **Run** button to start the simulation

Observe how the plane is flying in a circular holding pattern around the airport.

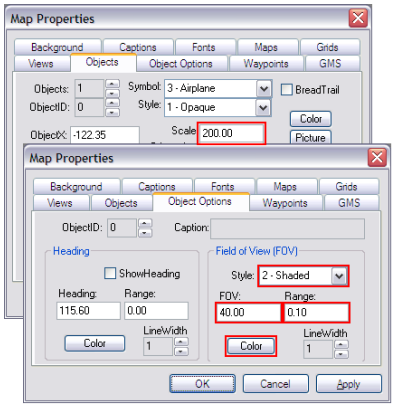


Holding pattern moving map display in the Map View window

Configure Map View

The Map View Properties dialog contains many options to customize the display. For starters, let's make the airplane object easier to observe.

- ► Right click in the map display area and choose **Map Properties** from the context menu
- ► Click on the **Objects** tab and change the scale of the default object (an airplane) to **200.00**
- ► Click on the **Object Options** tab
- ► In the *Field of View* frame, enter **40.00** in FOV, **0.10** in Range, click the **Color** button, choose **light red**, click **OK** to close the color dialog, choose a style of **2 – Shaded**, and click **OK**



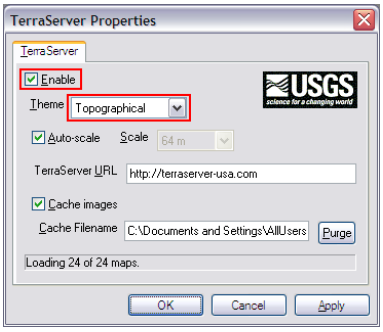
Objects and Object Options tabs with settings described in text

Display a TerraServer Map

Map View can link to Microsoft's TerraServer® web-site and download topographical maps or black-and-white satellite photomaps for the defined coordinates.

Note: You must be connected to the Internet (with a reasonably fast connection) to use this feature.

- ► Right click in the map pane and choose **Terra-Server Properties** from the context menu
- ► Click the **Enable** checkbox and choose **Topo-graphical** from the Theme listbox (see figure) and click **OK** to the close TerraServer Properties dialog



TerraServer Properties windows with specified settings

After the maps are loaded (the CoPilot status bar will report on loading maps), they will appear (see figure).

- ► Right click in the map pane and choose **Map Properties** from the context menu
- ► Click on the **Maps** tab, change the mapping mode to **Floating**, and click **OK**

Observe how the map display tracks the path of the airplane when in Floating mode.

- ► Click the **Stop** button to end the simulation

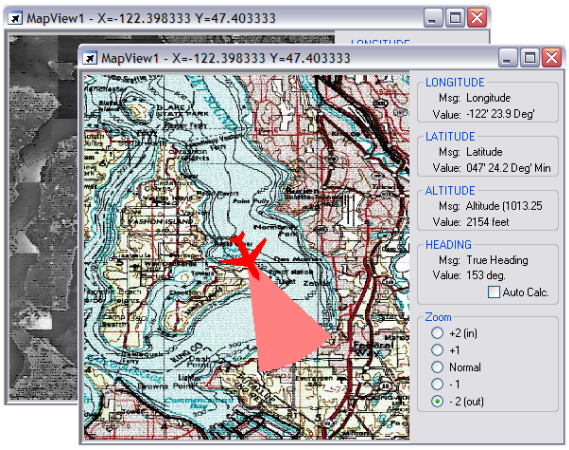


Photo (behind left) and topographical TerraServer maps

Related Topics

- The zoom buttons in the status pane adjusts the display to four levels of magnification

Summary

In this lesson you learned…

- how to open a Map View window and configure its properties
- how to link data fields to Map View properties
- how to use TerraServer maps

In the next lesson you will learn how to customize CoPilot with scripting using the Script View control.

Lesson 15: Customizing through ATE Scripting

ATE (Automated Test Environment) is a framework for creating and running tests in the CoPilot software. Python scripts are used to perform the innumerable tasks possible using the framework in conjunction with the CoPilot Object Model, which allows logical access to much of the functionality in CoPilot via ATE. This lesson shows just one possibility of how to use the tools available in CoPilot ATE to accomplish a simple automated task.

Introduced in This Lesson

ATE (Automated Test Environment), Python Code Editor, Object Browser, Alias properties, methods and events

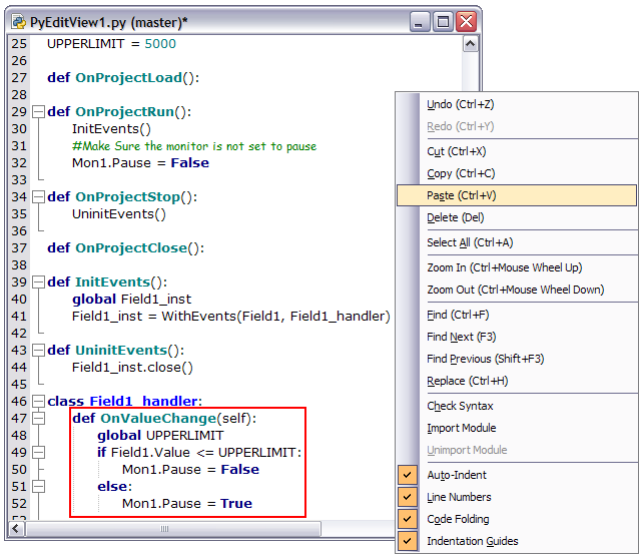
Objective

Use ATE Python scripting to control the monitor so that bus traffic is only recorded during takeoff and landing conditions.

Open View and Copy Script

- Click the **New** button, select the **Professional Views** tab, choose **Python Code Editor**, and click **OK**
- Open the example script **MonitorControlAFDX.py** by right clicking the file folder (copied to "…\My Documents\My CoPilot Projects\Samples\Scripts" during first run) and selecting "Edit with IDLE"
- Copy the entire file and replace the contents of the Python Editor in CoPilot (through the "Select All" and "Paste" context menu commands)

The Scripts folder (copied to "…\My Documents\My CoPilot Projects\Samples\Scripts" during first run) contains example scripts that you can use or modify.



The Python Editor with Monitor Control script and context menu

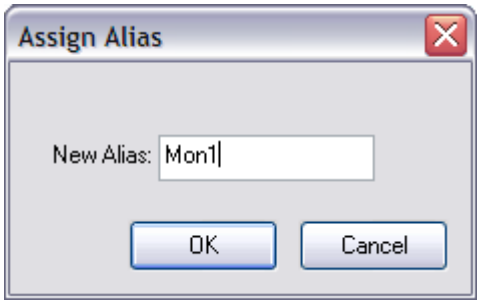
Translate Objectives into Code

The Monitor Control script records bus traffic during takeoff and landing only. Examine the code in the Python Editor (top figure at right). The event handler subroutine (OnValueChange) is triggered each time the Altitude value (Field1) is changed. When the value of Field1 is above 5,000 feet (UPPERLIMIT), the Sequential Monitor (Mon1) is paused. When it is below 5,000 feet, monitor recording is resumed.

Access Objects via Script

In Lesson 6 you used the Pause button in the Monitor View window to control monitor recording. Although it may not have been obvious, "Pause" is a property of the monitor object. In Lesson 13, you linked the "ValueEngr" property of the True Heading field to a control. Now we will use Python scripting to access those same properties.

- If the Object Browser is not active, show it by selecting **View | Other Windows | Object Browser**
- Drag the **Sequential Monitor** object into the Object Browser and rename the alias **Mon1**

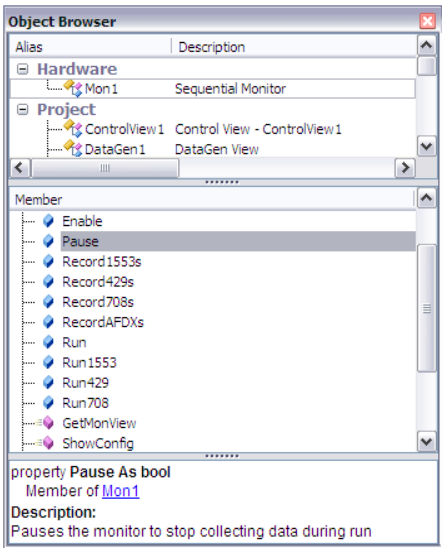


The Sequential Monitor object being assigned an alias

To make code easier to reuse, example scripts refer to aliases by generic names (such as Mon1) and aliases in the Object Browser are renamed to match. Look at the center pane of the Object Browser. The Pause property is used in the script. (Methods are represented by a magenta icon and events by yellow icons.)

- ► Right click **Port 50100** on VL 200 and choose **Add Field**
- ► Set the Name to **Altitude**, and the Start Byte to **90** then click **OK**, set Units to **feet**, then click **OK** again
- ► Drag the **Altitude** field from VL 200 into the Object Browser and rename it **Field1**

If you browse to the OnValueChange method for Field1, you will notice that the yellow lightning bolt icon indicates that the method is called in response to an event.



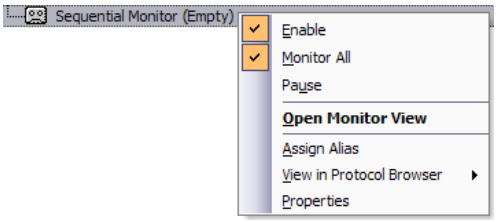
The Pause property of the Mon1 object

Prepare the Data

- ► Double click on the **Data Generator** icon in the Project Explorer to view it in the display pane
- ► Drag the **Altitude** field from VL 200 into the Data Generator
- ► As done with the Data Generator in Lesson 8, set the type to **Sine**, the Min to **0**, and the Max to **20000**, then click **OK**

The Bus Monitor has been disabled since Lesson 5.

- ► Right click on the Sequential Monitor object in the hardware tree and select **Enable** to reenable monitoring (match figure at right)



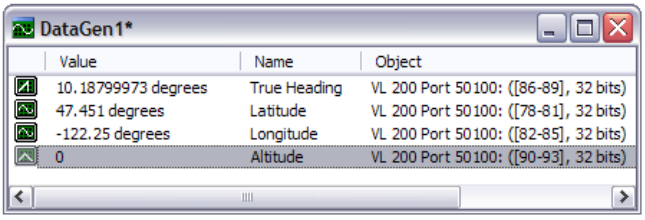
Correctly configured monitors

Execute the Script

- ► Click the **Run** button to start the simulation and trigger the OnProjectRun function

Watch the Sequential Monitor icon in the Hardware Explorer. The monitor pauses and the record count freezes when Altitude goes above 5,000 feet, then resumes recording when Altitude falls below 5,000 feet.

- ► Click the **Stop** button to end the simulation and trigger the OnProjectStop subroutine.



The Altitude field in the Data Generator just past 5,000 feet

The Sequential Monitor paused by the script

Related Topics

- Example scripts (both Python and VB) and detailed scripting instructions are included with CoPilot documentation

Summary

In this lesson you learned…

- how to open a Python Editor window and copy a script
- how to add and rename aliases
- how to run a script and see the results

Now that you have completed these introductory lessons, you are ready to use CoPilot! To learn more about the topics introduced in this guide, please refer to the *CoPilot User's Manual*.

Lesson 16: Legacy CoPilot Scripting

This lesson included for legacy purposes and has been replaced by lesson 15.

Scripting allows users to customize and extend the functionality of the CoPilot. Scripts could be used to automate configuration tasks, respond to bus events, start and stop monitor recording, create a sequence of unique data responses based on the value of incoming messages, or perform other tasks. Scripts could also be used to transfer information between CoPilot objects and between other applications (such as Microsoft Excel or LabView).

Introduced in This Lesson

Script View, Script Objects, Script Object's properties and methods, Select Import File dialog

Objective

Use Script View to control the monitor so that bus traffic is only recorded during takeoff and landing conditions.

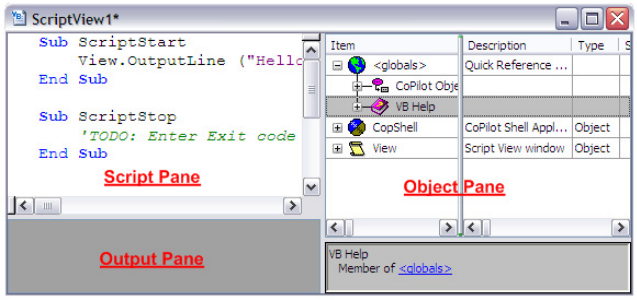
Open View and Import Script

- Click the **New** button, select the **Professional Views** tab, choose **Script View**, and click **OK**

Script View is divided into three panes (see figure).

- Right click in the Script Pane to access the context menu, point to **Import Script** to expand the sub-menu, then click **Overwrite** (see figure)
- In the Select Import File dialog, click the **Browse** button, navigate to the Scripts folder, select the **MonitorControlAFDX.txt** script, and click **OK**

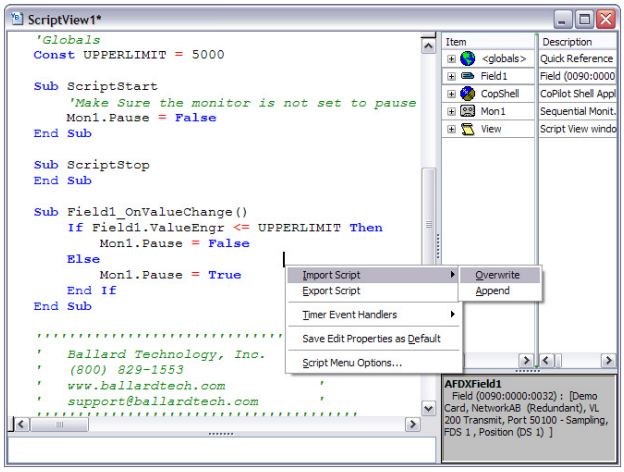
The Scripts folder (copied to "…\My Documents\My CoPilot Projects\Samples\Scripts" during first run) contains many example scripts that you can use or modify.



An example Script View window with each pane labeled

Translate Objectives into Code

The Monitor Control script records bus traffic during takeoff and landing only. Examine the code in the Script Pane (middle figure at right). The event handler subroutine (Field1_OnValueChanged) is triggered each time the Altitude value (Field1) is changed. When the value of Field1 is above 5000 feet (UPPERLIMIT), the Sequential Monitor (Mon1) is paused. When it is below 5000 feet, monitor recording is resumed.

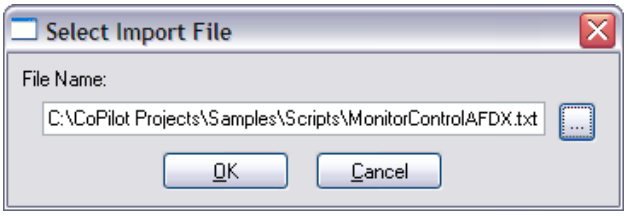


The Script pane with Monitor Control script and context menu

Access Objects via Script

In Lesson 6 you used the Pause button in the Monitor View window to control monitor recording. Although it may not have been obvious, "Pause" is a property of the monitor object. In Lesson 14, you linked the "ValueEngr" property of the True Heading field to a control. Now we will use scripting to access those same properties.

- Drag the **Sequential Monitor** object into the Object Pane of the Script View window
- Double click on the default name (AFDXMon1) and rename it **Mon1** (as identified in the script)



The import dialog with the default path to MonitorControl.txt

To make code easier to reuse, example scripts refer to objects by generic names (such as Mon1) and objects in the Object pane are renamed to match. Click the plus icon for Mon1 in the Object pane to see its properties. The Pause property is used in the script. (Methods are represented by a icon and events by blue icons.)

- Right click **Port 50100** on VL 200 and choose **Add Field**
- Set the Name to **Altitude**, and the Start Byte to **90** then click **OK**, set Units to **feet**, then click **OK** again
- Drag the **Altitude** field from VL 200 into the Object Browser and rename it **Field1**

If you browse to the OnValueChange method for Field1, you will notice that the icon is the blue lightning bolt cating that the method is called in response to an event.

Prepare the Data

- Double click on the **Data Generator** icon in the Project Explorer to view it in the display pane
- Drag **Altitude** field on VL 200 into Data Generator
- As done with the Data Generator in Lesson 8, set the type to **Sine**, the Min to **0**, and Max to **20000**, then click **OK**

The Sequential Monitor was disabled in Lesson 5, but needs to be activated for this lesson.

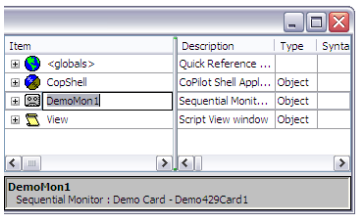
- Right click on the Sequential Monitor object in the hardware tree and select **Enable** to reenable monitoring (match figure at right)

Execute the Script

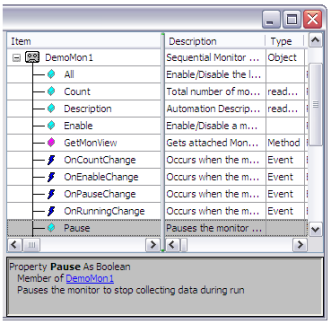
- Click the **Run** button to start the simulation and trigger the Script Start subroutine

Watch the Sequential Monitor icon in the Hardware Explorer. The monitor pauses and the record count freezes when Altitude goes above 5,000 feet, then resumes recording when Altitude falls below 5,000 feet.

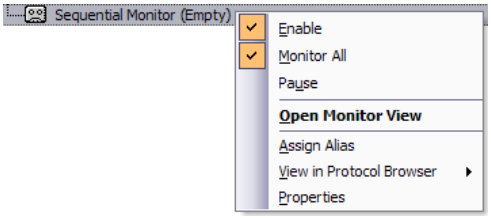
- Click the **Stop** button to end the simulation and trigger the Script Stop subroutine.



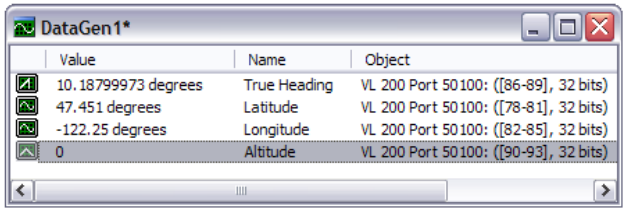
The Sequential Monitor object being renamed



The Pause property of the Mon1 object



Correctly configured monitors



The Altitude field in the Data Generator just past 5,000 feet

The Sequential Monitor paused by the script

Related Topics

- Example scripts and detailed scripting instructions are included with CoPilot documentation

Summary

In this lesson you learned…

- how to open a Script View window and import a script
- how to add and rename script objects
- how to run a script and see the results

This lesson included for legacy purposes and has been replaced by lesson 15.

THIS PAGE INTENTIONALLY BLANK

Ballard TECHNOLOGY

www.ballardtech.com

Your Source for Avionics Databus Tools

Protocols Supported

- MIL-STD-1553
- ARINC 429/575
- ARINC 629
- ARINC 664/AFDX
- ARINC 708/453
- ARINC 717/573
- Space Shuttle (MIA and MDM)
- Custom and non-standard protocols

Platforms

- PCI
- cPCI
- PMC
- PCMCIA
- USB
- Ethernet
- PC/104 and PC/104 plus
- ISA
- VMEbus
- Industry Pack
- Handheld

Software

- CoPilot for 1553, 429, AFDX/664, and 708 boards
- BTIDriver™ universal API
- ARINC 708 Utility GUI
- ARINC 717 Monitor GUI
- Drivers for DOS® and Windows® (all versions)

Increase Your Productivity with CoPilot AFDX

Just the Beginning

In fifteen brief lessons, you learned how easy it is to simulate and test AFDX networks using CoPilot AFDX, and you have only scratched the surface! If you look back at the windows, dialogs, and menus shown in these lessons, you will discover numerous options that were not demonstrated. For further information, refer to the *CoPilot User's Manual*. In addition, you can count on additional features being continuously added to CoPilot to meet the emerging requirements of a growing number of users.

Take a CoPilot Test Drive

The best way to discover how CoPilot AFDX can increase your own productivity is to try it. Ballard Technology would be happy to send you an evaluation copy of CoPilot. With it you can duplicate the examples illustrated in these lessons and examine all the features available through CoPilot menus, tool bars, and windows using the software-generated AFDX Demo card. If you already have an LC-AFDX-1, LP-AFDX-1, or LM-AFDX-1 board, Ballard can install a temporary key that will allow you to evaluate the full capability of our hardware and software.

Expand Use to Other Databuses

Today's complex avionics systems often utilize multiple protocols, drawing on the strengths of each one. The examples in this document illustrate the use of CoPilot for MIL-STD-1553 systems, but CoPilot supports ARINC 429, ARINC 664/AFDX and ARINC 708 systems as well. Consequently, with the appropriate Ballard Technology avionics databus cards and CoPilot software, users can monitor, simulate, analyze, compare, and integrate activity on several databuses at the same time. In CoPilot, users have the advantage of a common environment with features and tools suited to the unique characteristics of each avionics protocol.

Contact Ballard

Our experienced engineering staff is available to discuss your requirements for avionics databus tools and interfaces. For more information about our products or support in the use of this product, call Customer Service. Our hours are 8:00 AM to 5:00 PM Pacific Time, though support and sales engineers are often available outside those hours. We invite your questions and comments on any of our products.

Ballard TECHNOLOGY

Phone: 1-800-829-1553 425-339-0281 Fax: 425-339-0915
E-mail: *support@ballardtech.com*
Web: *www.ballardtech.com*