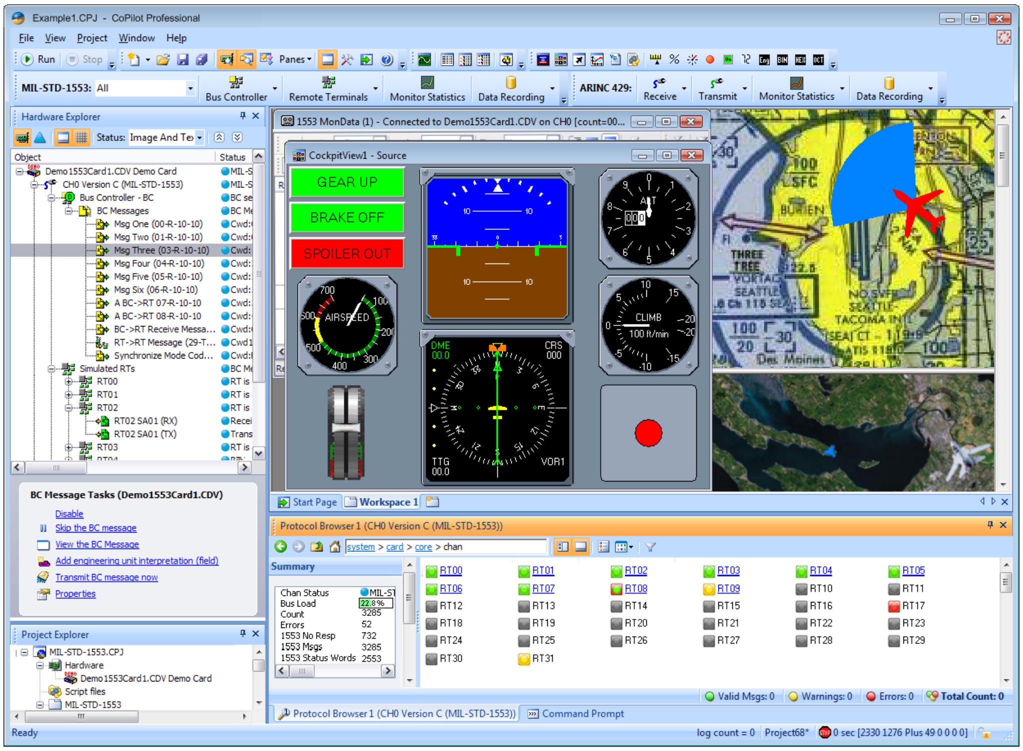
Getting Started Guide to

CoPilot 1553

Version 5
Rev B



Ballard
TECHNOLOGY

Ballard
TECHNOLOGY

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

MA136-20090727

Contents

THIS PAGE INTENTIONALLY BLANK

Introduction

CoPilot 1553

The term “CoPilot 1553” is used throughout this manual to identify the MIL-STD-1553 specific features available with CoPilot. CoPilot 1553 is part of the multi-protocol CoPilot software 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 with MIL-STD-1553, you can set up and run Bus Controller schedules, simulate RTs, or engage the Bus Monitor with a few clicks of the mouse. You can also observe and change data in engineering units and other radices while the bus is running. Although CoPilot is capable of simultaneously supporting multiple protocols and multiple hardware interfaces, this getting started guide focuses on the operation of a single MIL-STD-1553 demonstration device (a.k.a. 1553 Demo Card).

Ballard Hardware

CoPilot 1553 operates in conjunction with Ballard’s OmniBus (PCI, cPCI, USB, Ethernet, PMC, VME), Avionics BusBox (Ethernet), CM1553-3 (PCMCIA), LP1553-3 (PCI), LC1553-3 (cPCI), and BUSBox (USB) family of products. These boards can be ordered with varying levels of capabilities, summarized in the table below.

	Level A	Level B4	Level B32	Level C	Level D	Single Function	Multi Function
Bus Controller	✓	✓	✓	✓	✓	✓++	✓
Simultaneous Terminals	1	4	32	32	32	32++	32
Monitor	✓	✓	✓	✓	✓	✓++	✓
Filtering for Terminal Address	✓	✓	✓	✓	✓	✓++	✓
Filtering for Subaddress		✓	✓	✓	✓	✓++	✓
Concurrent Terminal Monitoring				✓	✓		✓
Protocol Error Injection				✓	✓		✓
Variable Transmit Amplitude					✓		
Zero Crossing Distortion					✓		

++ Single level functionality operates a product as either a BC, or RTs, or a Monitor

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 1553* is a tutorial designed to help you quickly learn the basics of CoPilot for MIL-STD-1553 and introduce you to the optional features available with CoPilot Professional. This guide is composed of sixteen brief lessons in three sections. Sections A and B of this guide describe features available in CoPilot Standard. Section C of this document 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.

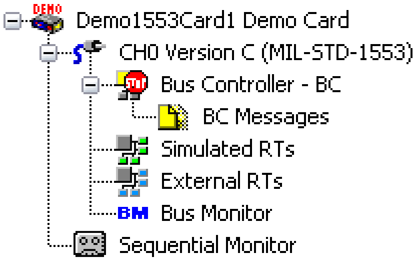
Note: This guide assumes the reader is familiar with MIL-STD-1553, the avionics databus protocol used in military applications. No effort has been made to define terms that are part of the MIL-STD-1553 specification.

1553 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 ones, 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.

1553 Demo Card

The lessons in this guide use the built-in 1553 demonstration card (see figure below). Using this demonstration card, you can duplicate on your screen what is described in the following lessons. This allows you to “try out” the features and capabilities of CoPilot without the need for a keyed Ballard 1553 card or access to an active MIL-STD-1553 databus. The 1553 Demo card emulates a 1553 BUSBox Level C card (see table above for a summary of Level C capabilities). Demo card features include multi-terminal simulation, concurrent monitoring, and assumes a CoPilot 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** link and follow the directions. If the menu screen does not automatically launch, run the **INSTALL.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.

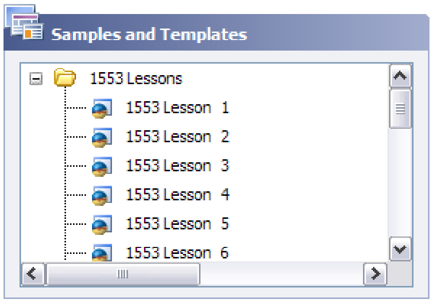
Note that files and projects saved in CoPilot version 5 will not be able to be opened by previous version of CoPilot. Be sure to save a copy of the files you still intend to open with the previous version.

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 **1553 Lessons** from the Samples and Templates on the Start Page (see figure below)
- ► Choose **1553 Lesson 2**

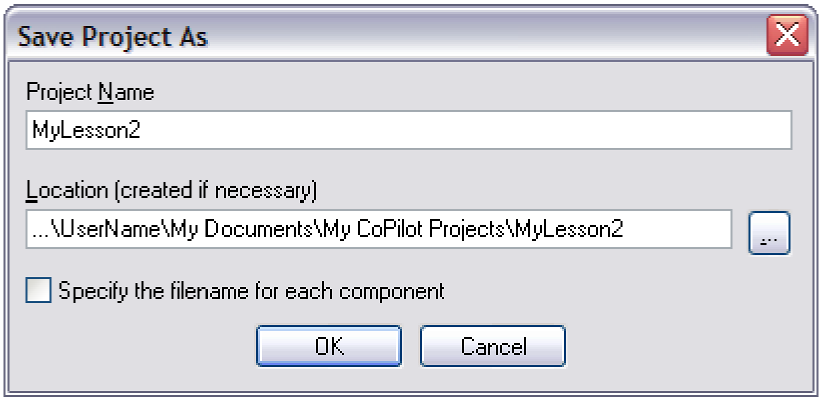


Create User Projects

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., MyLesson2) 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)



Save Project As

Project Name: MyLesson2

Location (created if necessary): ...\UserName\My Documents\My CoPilot Projects\MyLesson2

□ Specify the filename for each component

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.

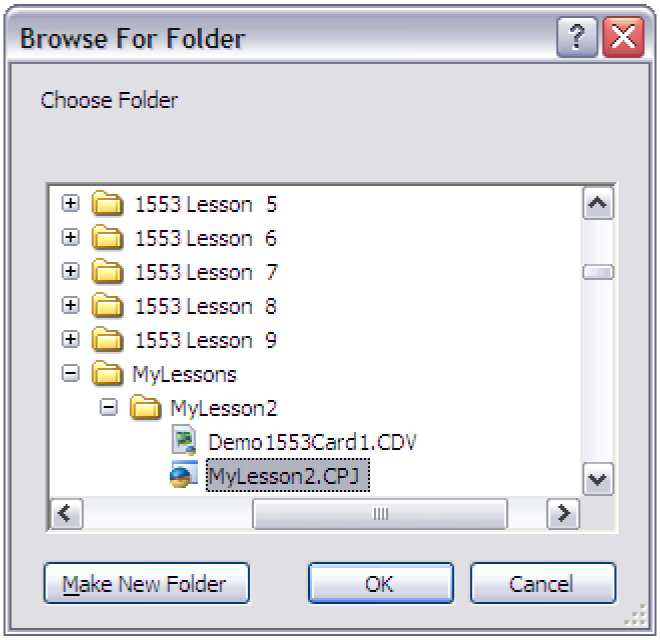
Begin the Lessons

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 CoPilot User's Manual (on the CoPilot CD)
- Samples and Templates available from the Start Page
- The 1553 example projects and scripts (on the CoPilot CD)

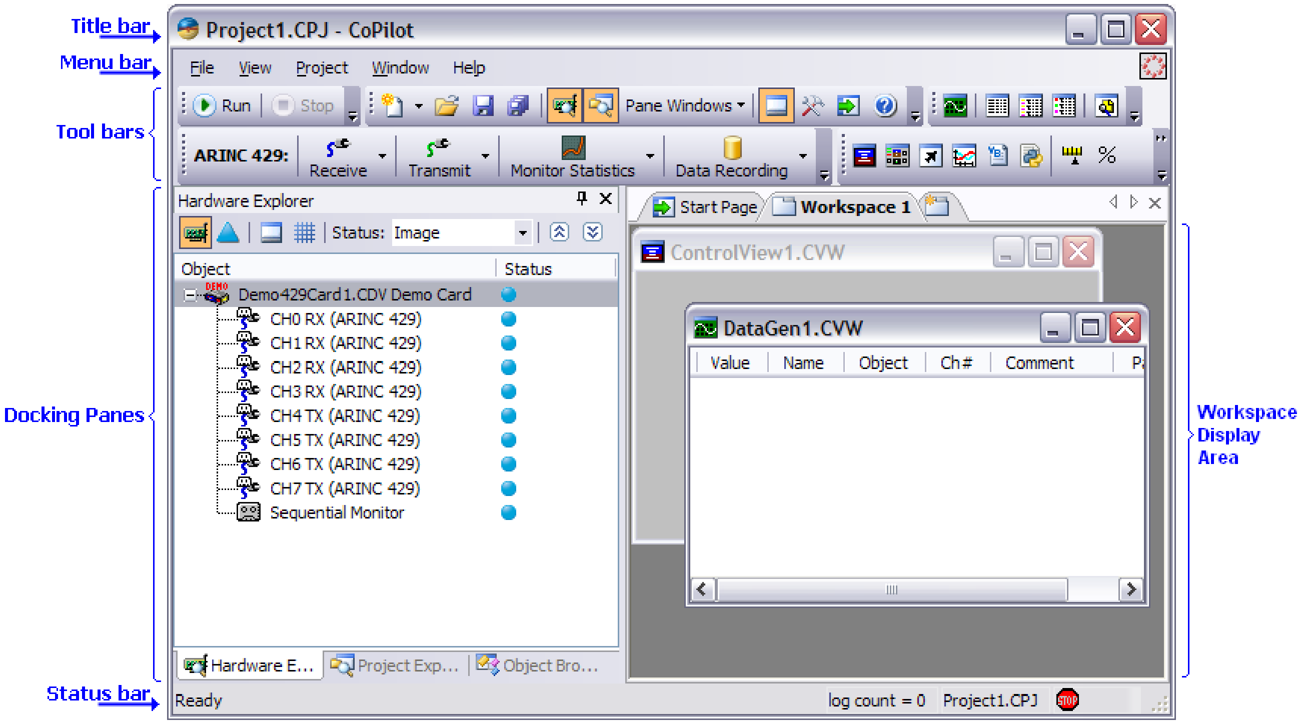
Section A: MIL-STD-1553 Basics

The six lessons in Section A present the fundamental MIL-STD-1553 simulation capabilities of CoPilot 1553. The topics covered are Quick Start Analyzer, BC message, BC schedule, RT and SA detection, shadowing, simulation, and the Bus Monitor. All of the functionality in this section is available with CoPilot Standard. 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.

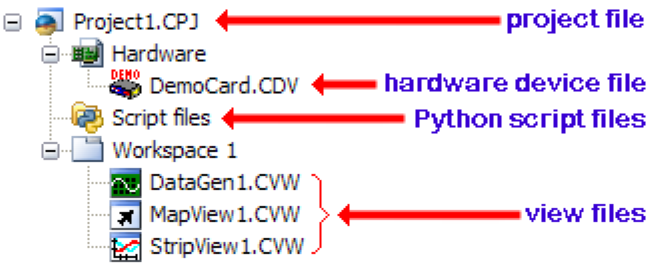
The CoPilot Project

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 1553 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.

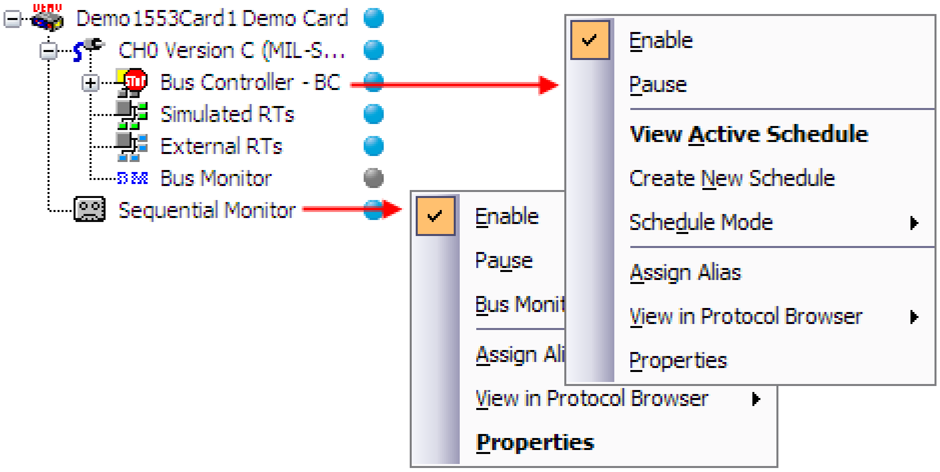
- **Edit Mode**—This mode is the default state for CoPilot during which most data initialization, configurations, and other settings are specified. No interaction with the databus or Ballard board takes place in Edit mode.
- **Simulation Mode**—When Simulation mode is initiated through the Run button, 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).



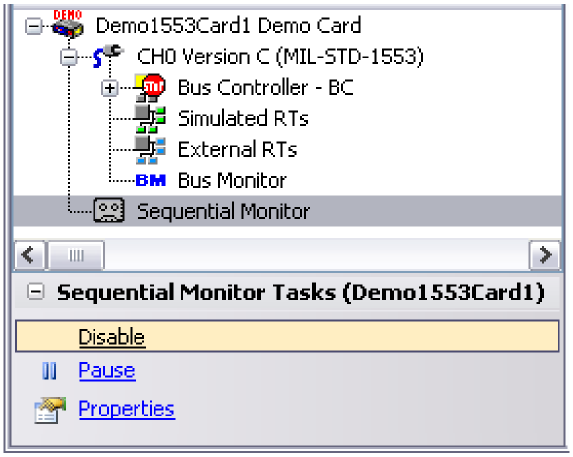
Context Menus are shown from a Right Mouse Click

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

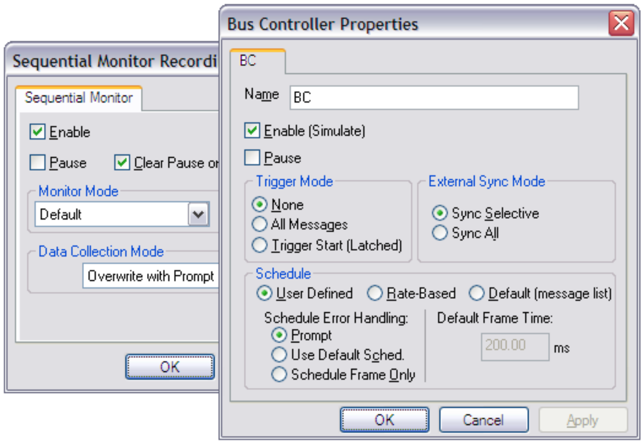
Selecting an item in the Hardware Explorer shows a list of available “quick tasks” for that item. These tasks lists contain the actions most commonly performed on that item. However, if more than one item is selected, then the task list will reflect those common tasks that can be performed by all the 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.



Sequential Monitor Recordi

Sequential Monitor

☑ Enable

□ Pause ☑ Clear Pause or

Monitor Mode: Default

Data Collection Mode: Overwrite with Prompt

OK

Bus Controller Properties

BC

Name: BC

☑ Enable (Simulate)

□ Pause

Trigger Mode: ● None ○ All Messages ○ Trigger Start (Latched)

External Sync Mode: ● Sync Selective ○ Sync All

Schedule: ● User Defined ○ Rate-Based ○ Default (message list)

Schedule Error Handling: ● Prompt ○ Use Default Sched. ○ Schedule Frame Only

Default Frame Time: 200.00 ms

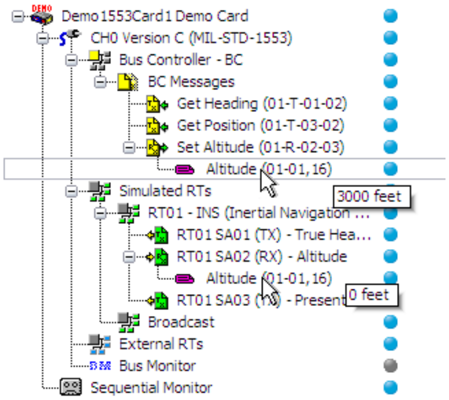
OK Cancel Apply

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). Messages, subaddresses, 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 Explorer pane use tool tips to display additional information. In addition, many properties pages, buttons, and other objects and windows throughout CoPilot use tool tip displays. To display the tool tip for an item, hover the mouse over the item without moving. A popup will appear with status, configuration, or data information. (see figure below).



```
Demo1553Card1 Demo Card
  CH0 Version C (MIL-STD-1553)
    Bus Controller - BC
      BC Messages
        Get Heading (01-T-01-02)
        Get Position (01-T-03-02)
        Set Altitude (01-R-02-03)
          Altitude (01-01, 16)        3000 feet
    Simulated RTs
      RT01 - INS (Inertial Navigation ...
        RT01 SA01 (TX) - True Hea...
        RT01 SA02 (RX) - Altitude
          Altitude (01-01, 16)
        RT01 SA03 (TX) - Present      0 feet
      Broadcast
    External RTs
    Bus Monitor
  Sequential Monitor
```

Introduced in This Section

The six lessons in this section (MIL-STD-1553 Basics) demonstrate how CoPilot 1553 facilitates communication with one or more MIL-STD-1553 databuses. The following section will demonstrate additional features of CoPilot that simplify the configuration and analysis of information sent and received over the databus.

- Lesson 1 The MIL-STD-1553 Quick Start Analyzer is used to discover and analyze databus activity.
- Lesson 2 A 1553 BC message is created and asynchronously sent on the databus.
- Lesson 3 Default and custom user schedules are used to autonomously send multiple BC messages.
- Lesson 4 Detects RT (remote terminal) and SA (subaddress) activity by shadowing an external RT.
- Lesson 5 Configures RTs and SAs for simulation and modifies data at runtime.
- Lesson 6 Controls the logging of databus activity and views the records in the Sequential Monitor View.

THIS PAGE INTENTIONALLY BLANK

Lesson 1: 1553 Quick Start Analyzer

The fastest way to determine the current activity on your MIL-STD-1553 buses is by launching the MIL-STD-1553 Quick Start Analyzer from the Start Page in CoPilot. This feature configures the hardware to capture and record all databus activity to file, view the sequential monitor record file as it is being captured, and monitor the databus activity using the Protocol Browser. The Analyzer Setup allows for the configuration of multiple 1553 channels for hardware installed in the system.

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 1553 lesson projects and maintaining a personal cumulative project.

Introduced in This Lesson

Start Page, MIL-STD-1553 Quick Start Analyzer, Protocol Browser

Objective

Setup MIL-STD-1553 channels to quickly view the databus activity and analyze RT statistics, errors and warnings.

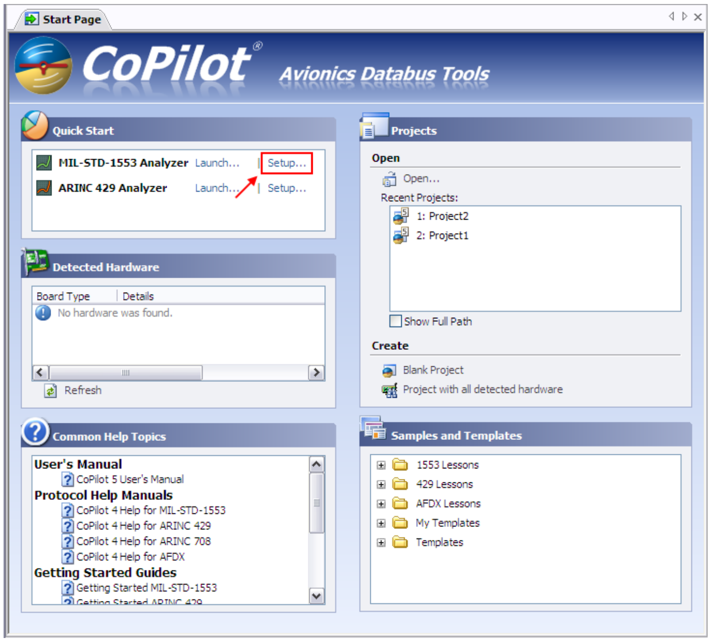
Show the Start Page

This first lesson is the only lesson that does not begin with opening a template lesson. To begin this lesson, be sure that the CoPilot Start Page is shown:

- ▶ Select **View | Start Page**

The Start Page should then be shown in CoPilot as shown in the figure at right.

Note that the Start Page can be shown at any time; even after it is closed or while CoPilot is running.



The CoPilot Start Page

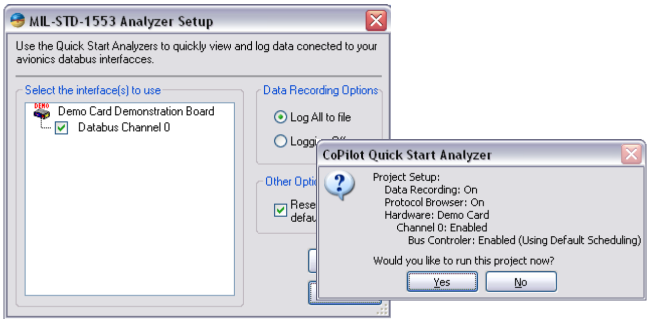
Setup the 1553 Analyzer

First, the MIL-STD-1553 Quick Start Analyzer needs to be configured.

- ▶ Select **Setup…** next to the MIL-STD-1553 Analyzer item from the Quick Start panel of the Start Page

The figure on the lower right shows the MIL-STD-1553 Analyzer Setup dialog. To configure the MIL-STD-1553 Analyzer setup:

- ▶ Verify the **Databus Channel 0** is checked
- ▶ Click **Log All to file** to specify that all databus traffic will be recorded to file
- ▶ Verify **Reset layout to default** on start is checked to ensure optimal window positioning
- ▶ Click the **OK** button to accept the configuration



MIL-STD-1553 Analyzer Setup

Use the Quick Start Analyzers to quickly view and log data conected to your avionics databus interfacces.

Select the interface(s) to use

Demo Card Demonstration Board
☑ Databus Channel 0

Data Recording Options

● Log All to file
○ Loggi

Other Optio

☑ Rese defau

CoPilot Quick Start Analyzer

Project Setup:
Data Recording: On
Protocol Browser: On
Hardware: Demo Card
Channel 0: Enabled
Bus Controler: Enabled (Using Default Scheduling)

Would you like to run this project now?

Yes No

Configuring the 1553 Analyzer

Run the Quick Start Analyzer

After the 1553 Quick Start Analyzer is configured, the setup is displayed. This information lists the configuration and prompts before the analyzer is started.

- ► Click the **Yes** button to begin the analysis

Notice that because this is demonstration hardware, three BC messages are automatically created to show meaningful bus activity.

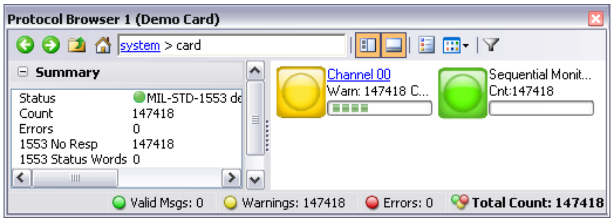
Analyze the Activity on the Bus

The Quick Start Analyzer has configured the hardware and the Protocol Browser should be displayed (see figure to right). Protocol Browser panes allow you to drill in and out of the various levels of the hardware. The same protocol browser can be used to browse between multiple channels or protocols (such as ARINC 429 and MIL-STD-1553) if more than one exists in a project.

- ► Notice the state of Channel 00 is shown with a yellow icon to show both the current state and the historical state have warnings
- ► Click the **Display Legend** button in the toolbar of Protocol Browser 1 to show the Protocol Browser legend that defines the icon states (see figure middle right)
- ► Click the **Close** button to close the Legend

Additional information is available with the summary information (on the parent sidebar) of the Protocol Browser and with item tooltips.

- ► Click the **Channel 00** item in the Protocol Browser to drill into the channel
- ► Change the view mode by clicking the **View** button on the toolbar, then clicking the Small List Mode menu item to view all RTs
- ► To complete this lesson, press the **Stop** button



Protocol Browser 1 (Demo Card)

system > card

Summary

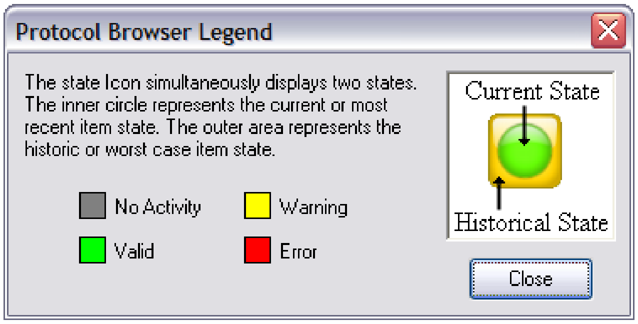
Status	MIL-STD-1553 de
Count	147418
Errors	0
1553 No Resp	147418
1553 Status Words	0

Channel 00
Warn: 147418 C...

Sequential Monit...
Cnt:147418

Valid Msgs: 0 Warnings: 147418 Errors: 0 **Total Count: 147418**

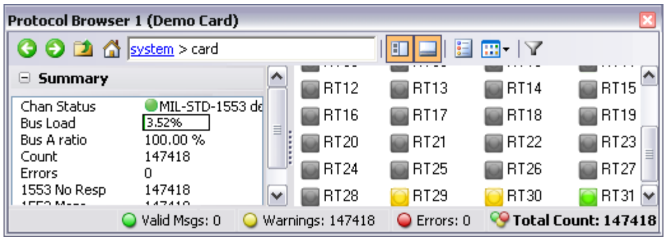
Protocol Browser pane for the 1553 Demo Card



Protocol Browser Legend

The state Icon simultaneously displays two states. The inner circle represents the current or most recent item state. The outer area represents the historic or worst case item state.

Protocol Browser legend



Protocol Browser 1 (Demo Card)

system > card

Summary

Chan Status	MIL-STD-1553 de
Bus Load	3.52%
Bus A ratio	100.00 %
Count	147418
Errors	0
1553 No Resp	147418

Valid Msgs: 0 Warnings: 147418 Errors: 0 **Total Count: 147418**

Protocol Browser showing RTs with activity

Related Topics

- It is possible to show/hide the Parent Information Sidebar with the button
- The status bar of Protocol Browser panes can be shown/hidden by clicking the button
- Click hyperlinks in the Protocol Browser to drill into items or drag and drop items from the Hardware Explorer
- Protocol Browser panes can be undocked (floated), pinned, moved, or even shown on another monitor
- The **filter** button limits the view to display to only RTs with activity

Summary

In this lesson you learned…

- how to display the Start Page
- how to setup a MIL-STD-1553 Quick Start Analyzer
- how to navigate the displayed items in the Protocol Browser
- how to analyze databus activity with the Protocol Browser

Building on these ideas, you will learn in Lesson 2 how to create and send 1553 BC Messages.

Lesson 2: Create and Send a 1553 Message

The Bus Controller (BC) initiates activity on a MIL-STD-1553 bus. You can use CoPilot's Bus Controller mode to test real RTs on a 1553 databus. Within CoPilot 1553, all BC messages are composed using the Message Editor, and all messages are stored and organized within the Message List.

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 1553 lesson projects and maintaining a personal cumulative project.

Introduced in This Lesson

Message Editor, Data Editor, Message List

Objective

Define and transmit a single message named Set Altitude instructing RT01, SA02 to receive three data words.

Open the Lesson 2 Project

Be sure that the 1553 Lesson 2 project is open:

- ▶ Select **File | New | New Project**
- ▶ Click the **1553 Lessons** tab
- ▶ Select **1553 Lesson 2**

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 on page 1 for complete instructions.

Create a BC Message

First, create a BC message using the Message Editor.

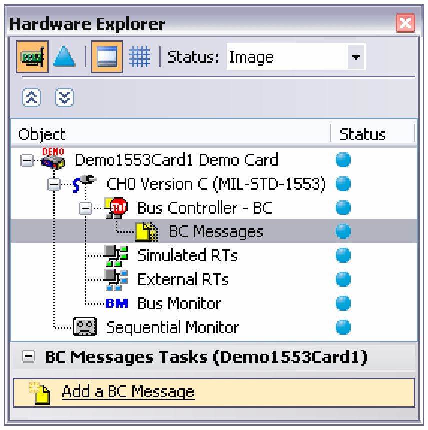
- ▶ Select the BC Messages icon choose **Insert a BC Message** from the task list to open the Message Editor (see figure above right)
- ▶ Type **Set Altitude** in the Message Name field

The first step in implementing "engineering units" (see Section B) is giving the message a meaningful name.

- ▶ Click on the **BC→RT** option button to define an RT receive message
- ▶ Type **1** in the RT cell, **2** in the SA cell and **3** in the WC cell to address the message to RT01, receive subaddress 2, with 3 data words
- ▶ Click the **Apply** button to apply the current choices (Do not close the window yet)

When you press Apply, the message is added to the Message List in the Hardware Explorer tree (see figure to right). The message in the tree is still linked to the Message Editor window, and any changes you make in the editing window are applied with the Apply button.

Notice that this message in the Hardware Explorer is identified as an RT receive message by its icon:


Hardware Explorer

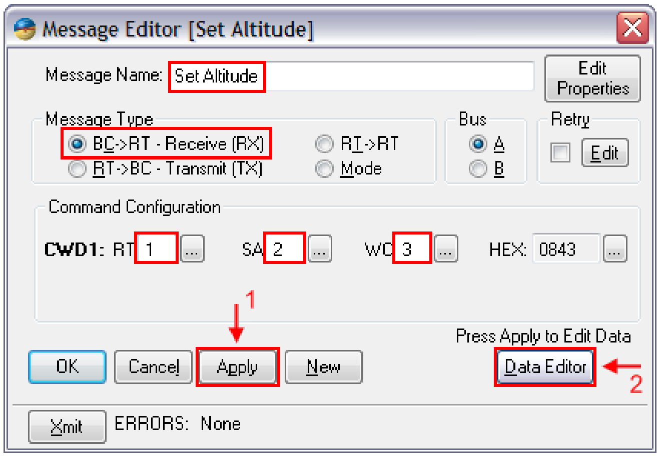
Status: Image

Object	Status
Demo1553Card1 Demo Card	●
CH0 Version C (MIL-STD-1553)	●
Bus Controller - BC	●
BC Messages	●
Simulated RTs	●
External RTs	●
Bus Monitor	●
Sequential Monitor	●

BC Messages Tasks (Demo1553Card1)

Add a BC Message

Hardware Explorer task list for BC Messages


Message Editor [Set Altitude]

Message Name: Set Altitude

Message Type

- ● BC->RT - Receive (RX)
- ○ RT->BC - Transmit (TX)
- ○ RT->RT
- ○ Mode

Bus

- ● A
- ○ B

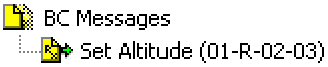
Retry □

Command Configuration

CWD1: RT 1 SA 2 WC 3 HEX: 0843

ERRORS: None

The Set Altitude message defined in the Message Editor


BC Messages

Set Altitude (01-R-02-03)

The Set Altitude message in the Message List

Edit the Data

- ▶ Click the **Data Editor** button (in the Message Editor) to open the Data Initialization window

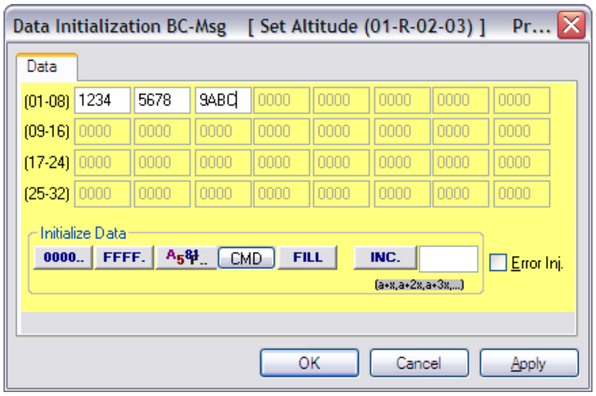
Notice that only three data cells are available. This is because the word count for this message is "3."

- ▶ Click in the first data cell and type **1234**, type **5678** in the next, and **9ABC** in the third (you can use the Tab key to advance to the next data cell)

Notice that data is entered in hexadecimal radix. Later we'll view and modify data in other radices.

Icons and colors are used throughout CoPilot to identify objects. The Data Initialization window has a yellow background to indicate it is associated with a BC message.

- ▶ Click the **OK** button to apply the new data, close the Data Editor, and return to the Message Editor
- ▶ Click **OK** to close the Message Editor



(01-08)	1234	5678	9ABC	0000	0000	0000	0000	0000
(09-16)	0000	0000	0000	0000	0000	0000	0000	0000
(17-24)	0000	0000	0000	0000	0000	0000	0000	0000
(25-32)	0000	0000	0000	0000	0000	0000	0000	0000

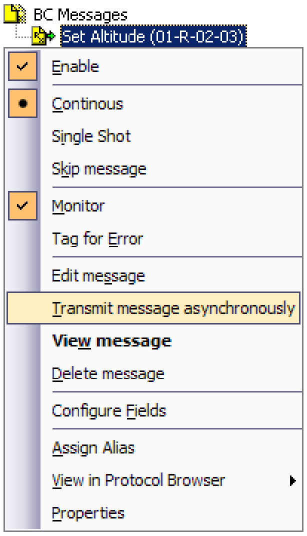
Data Initialization for Set Altitude BC Message

Transmit a Single Message

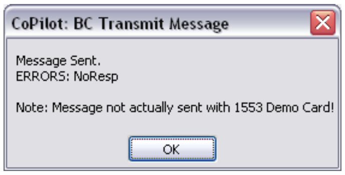
Although you have now created a message, it is not scheduled and will not be automatically transmitted. However, you can transmit this single message once on the databus.

- ▶ Right click the **Set Altitude** message in the BC Message List to show the context menu and choose **Transmit message asynchronously** from the context menu (see figure at right)

A warning message will appear since the demo card is a virtual card and not attached to a databus (click **OK** to clear the warning dialog). If you had an actual Ballard board, you would get a confirmation message box with a transmission report (see second figure at right).



Set Altitude context menu



Message Sent.
ERRORS: NoResp

Note: Message not actually sent with 1553 Demo Card!

Actual transmit confirmation box

Related Topics

- It is also possible to transmit a single message from the Message Editor using the Transmit Xmit button
- The Data Editor dialog can quickly initialize data to zeros 0000.., random numbers A5…, etc (see figure above).
- Messages may be tagged for error injection (only available for Ballard 1553 boards Level C or above)

Summary

In this lesson you learned…

- how to access quick tasks on objects in the Hardware Explorer
- how to create a 1553 message using the Message Editor
- how to enter data using the Data Editor
- how to access context menus for Hardware Explorer objects
- how to transmit a single message on the (simulated) 1553 bus

Building on these ideas, in the next lesson you will learn how to create a Bus Controller schedule.

Lesson 3: Run Default and Custom BC Schedules

The Bus Controller (BC) schedule directs the flow of information on the 1553 databus. The BC schedule window allows the user to control the selection and frequency of transmitted messages and define schedule elements such as frames, subframes, gaps, etc. An automatic default schedule can also be selected.

Introduced in This Lesson

Default and custom user BC Schedules, BC icons, BC Schedule window

Objective

First, run a default BC schedule. Then, create and transmit a custom user schedule consisting of one RT transmit message (Get Position) and one RT receive message (Set Altitude).

Run a Default Schedule

Note: Before you begin this lesson, open the 1553 Lesson 3 project (File | New | New Project | 1553 Lessons) or use the project you saved from Lesson 2 (so that the Set Altitude message will be present).

The stop sign on the Bus Controller icon indicates that no user BC Schedules have been defined.

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

The dialog (shown at right) will appear, giving you the option to temporarily run a default BC schedule.

► Click **Yes** to run a default schedule

The default schedule transmits all messages in the BC Message list (in this case, a single message) within a single frame. Notice the green "D" icon on the BC indicating that a default schedule is running.

Note: The two RT icons that appear (see figure at right) will be discussed in the next lesson.

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

Create a Custom BC Schedule

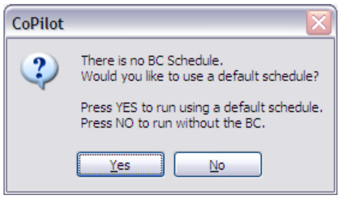
It is almost as easy to define a schedule explicitly as it is to run a default. First, open the schedule window.

► Double click the BC icon or right click and choose **View Active Schedule** from the context menu to open a BC Schedule edit window

► Click the Add Subframe button in the Frame toolbar and click **OK** on the Subframe dialog to create and insert a new subframe into the schedule

► Drag the **Set Altitude** message from the message list and drop it on the BC Schedule window

(The "drag and drop" procedure is used throughout CoPilot to link objects, copy configurations, etc.)

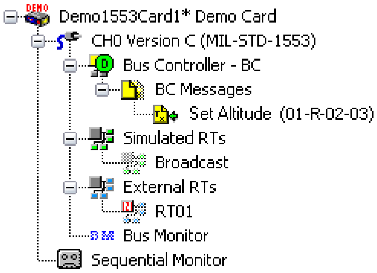


CoPilot

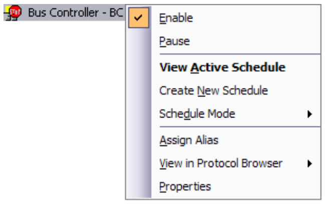
There is no BC Schedule.
Would you like to use a default schedule?

Press YES to run using a default schedule.
Press NO to run without the BC.

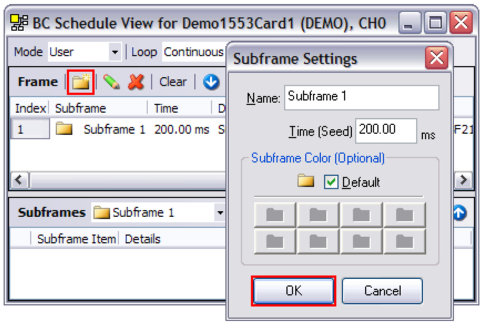
Default schedule dialog



The 1553 Demo Card running a default schedule



Bus Controller context menu



BC Schedule View for Demo1553Card1 (DEMO), CH0

Mode: User | Loop: Continuous

Index	Subframe	Time	D
1	Subframe 1	200.00 ms	S

Subframes: Subframe 1

Subframe Item | Details

Subframe Settings

Name: Subframe 1

Time (Seed): 200.00 ms

Subframe Color (Optional)

☑ Default

User BC Schedule with one Subframe

Now, with just a few clicks of the mouse, we have duplicated the default schedule! Buttons within the BC Schedule window make it easy to add messages, gaps, conditional branching, and multiple subframes with differing subframe times.

Define & Add a New Message

It is possible to access the Message Editor from within the BC Schedule window to create a new message.

- ► Click on the **Add Message** button and select **Create New** to open the Message Editor (introduced in Lesson 2)
- ► Type **Get Position** in the Message Name box
- ► Click on the **RT→BC (TX)** option button to define an RT transmit message
- ► Type **1** in the RT cell, **3** in the SA cell and **2** in the WC cell to address this message to RT01, transmit SA03, with 2 data words
- ► Click **OK** to add this message to the BC Schedule

Get Position is added to the same frame as Set Altitude (and it is added to the Message List in the Hardware Explorer). Based on the frame time, each of the two messages will be transmitted once every 200 milliseconds.

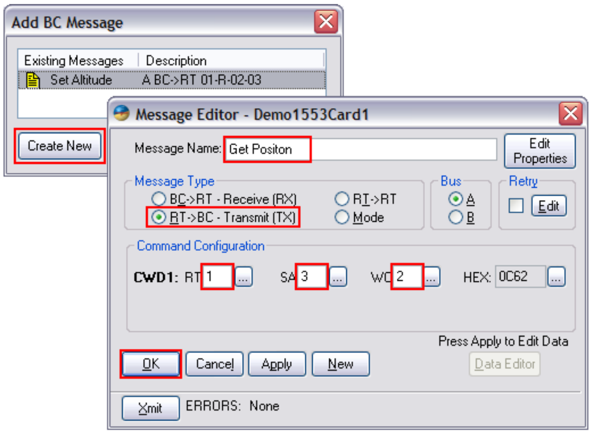
- ► Click the **Close** button in the upper right-hand corner to close the BC Schedule window

Transmit the Custom Schedule

- ► Click the **Run** button in the CoPilot toolbar to start the simulation

The BC schedule is now transmitting. In later lessons we will view messages in individual windows as they are transmitted and collectively in the Bus Monitor window.

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


Add BC Message

Existing Messages	Description
Set Altitude	A BC->RT 01-R-02-03

Message Editor - Demo1553Card1

Message Name: Get Positon

Message Type: ○ BC->RT - Receive (RX) ● RT->BC - Transmit (TX) ○ RT->RT ○ Mode

Bus: ● A ○ B

Retry: □

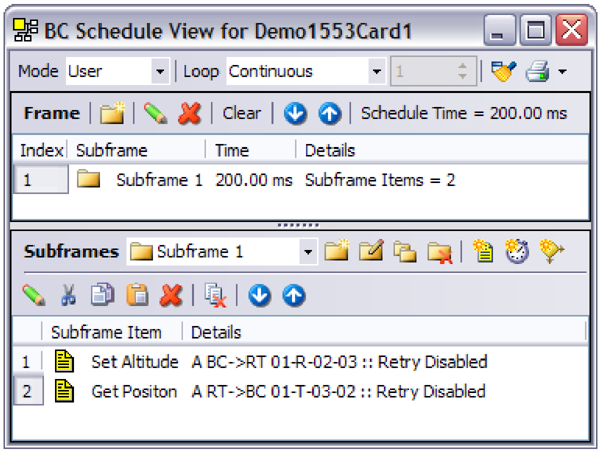
Command Configuration

CWD1: RT 1 SA 3 WC 2 HEX: 0C62

Press Apply to Edit Data

ERRORS: None

Get Position message


BC Schedule View for Demo1553Card1

Mode User | Loop Continuous | 1

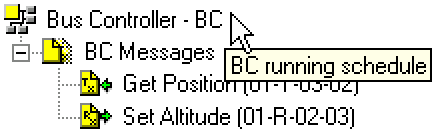
Frame | Clear | Schedule Time = 200.00 ms

Index	Subframe	Time	Details
1	Subframe 1	200.00 ms	Subframe Items = 2

Subframes: Subframe 1

	Subframe Item	Details
1	Set Altitude	A BC->RT 01-R-02-03 :: Retry Disabled
2	Get Positon	A RT->BC 01-T-03-02 :: Retry Disabled

BC Schedule with two messages


Bus Controller - BC
BC Messages
Get Position (01-T-03-02)
Set Altitude (01-R-02-03)

Bus Controller tool tip during run time

Related Topics

- The BC context menu and properties dialog contain many configuration options not covered in this lesson
- You can configure the BC to run a default schedule even if multiple custom schedules are defined
- A rate-based schedule automatically creates a schedule using the message rate from each BC message

Summary

In this lesson you learned…

- how to run a default BC Schedule
- how to access and create a custom BC Schedule
- how to assign existing messages to the schedule
- how to create and add new messages from within the schedule window

In the next lesson, you will automatically detect an RT and subaddresses, shadow that RT, and examine its data.

Lesson 4: Detect External Shadow RTs and SAs

CoPilot 1553 incorporates many automatic features that make configuration tasks easy and provide quick results to increase your productivity. In this lesson, we will see how CoPilot 1553 automatically detects RTs and subaddresses on the MIL-STD-1553 databus. Then, we'll monitor the responses of an external RT by configuring an external shadow RT.

Introduced in This Lesson

RT icons in the Hardware Explorer; TSA tab in the RT Configuration Properties dialog; SA auto-detection

Objective

Automatically detect whether or not an RT is responding to the Bus Controller. Examine the Terminal Simulation Assignment (TSA) window and shadow the detected RT. Run the simulation again to auto-detect two subaddresses.

Automatically Detect an RT

Note: Before you begin this lesson, open the 1553 Lesson 4 project (File | New | New Project | 1553 Lessons) or use the project you saved from Lesson 3.

CoPilot can automatically detect RTs on the MIL-STD-1553 databus.

▶ Click the **Run** button to transmit the custom user BC schedule (from Lesson 3)

Because there are no RTs on our demo bus, there is no response from RT01. A dimmed RT icon appears in the Hardware Explorer because CoPilot detected a BC message addressed to this terminal. The red "N" symbol indicates a "no response" condition for this RT. (A dimmed RT31 labeled "Broadcast," reserved by default for broadcast messages, appears as well.)

▶ Click the **Stop** button to stop transmitting

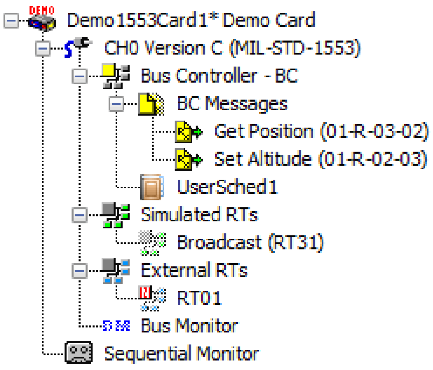
Examine Simulation Options

▶ Right click on the Simulated RTs and choose **Configure RTs** from the context menu

The RT Configuration Properties dialog with the Terminal Simulation Assignment (TSA) window will open. The TSA tab allows you to select which terminals (Bus Controller, RTs, and Bus Monitor) will be activated (simulated) through the Ballard board (see figure).

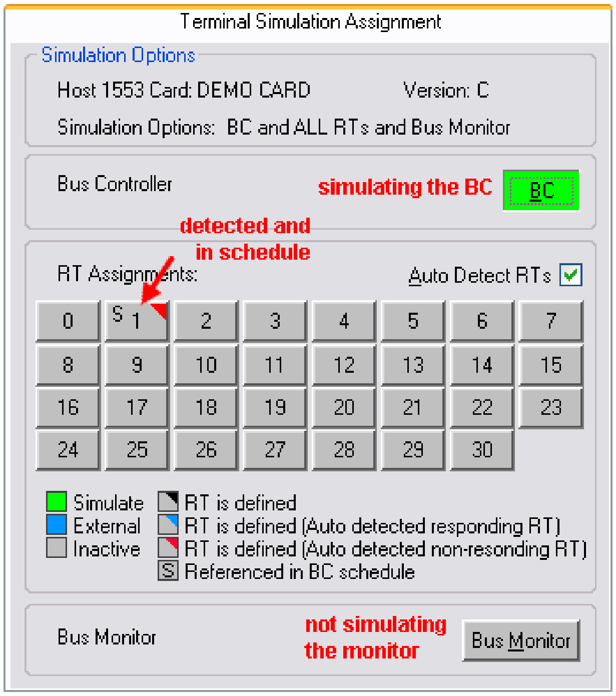
Notice, the RT01 key is marked by an "S" and a red triangle. The "S" indicates that this RT is addressed in the BC schedule and the red triangle indicates that the auto-detected RT did not respond to the BC message.

Note: The first three lessons could be run on a single-terminal 1553 board (Level A). The multiple-terminal simulation illustrated next would require a Level B4 or higher Ballard board.



```
Demo1553Card1* Demo Card
  CH0 Version C (MIL-STD-1553)
    Bus Controller - BC
      BC Messages
        Get Position (01-R-03-02)
        Set Altitude (01-R-02-03)
      UserSched1
    Simulated RTs
      Broadcast (RT31)
    External RTs
      RT01
    Bus Monitor
  Sequential Monitor
```

Hardware Explorer tree with detected external non-responding RT



Terminal Simulation Assignment

Simulation Options

Host 1553 Card: DEMO CARD Version: C

Simulation Options: BC and ALL RTs and Bus Monitor

Bus Controller

simulating the BC

detected and in schedule

RT Assignments: Auto Detect RTs ☑

0	S 1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	

🟩 Simulate
🟦 External
□ Inactive
◩ RT is defined
◩ RT is defined (Auto detected responding RT)
◩ RT is defined (Auto detected non-resonding RT)
S Referenced in BC schedule

Bus Monitor

not simulating the monitor

The state of the Hardware Explorer is represented in the TSA window

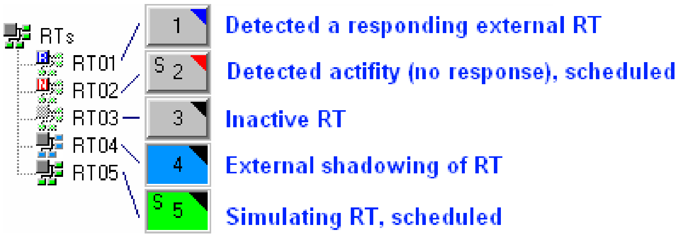
External Monitoring an RT

CoPilot 1553 allows you to "shadow monitor" an external RT on the bus; that is, to passively monitor and view the activity of an RT without interacting with the bus. Although we are not connected to a real terminal, the 1553 demo card can be used to illustrate the concept by "shadow monitoring" the messages to RT01.

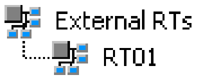
- ► Click the **RT01 button** S 1 twice until it is depressed with a blue background (S 1).
- ► Click **OK** to close the TSA window

A green color indicates that the RT will be simulated through the Ballard 1553 board and a blue color indicates the external RT will shadow monitored. Notice that changing between simulated and external moves the RT in the Hardware Explorer to either Simulated RTs or External RTs (see figure at right).

Shadowing allows you to view the changing data values through CoPilot views and displays (such as MIL-STD-1553 Message Views) as they are being transmitted and received by the external RT. With CoPilot Professional, you can view data from shadowed RTs through strip charts, controls, or moving map displays.



Sample Hardware Explorer icons and buttons for RTs in TSA window


External RTs
RT01

RT01 in shadow mode

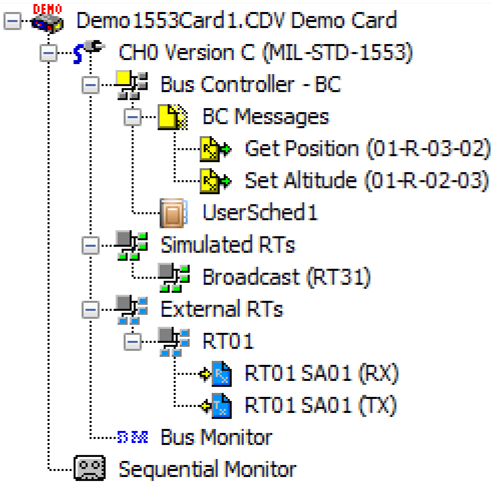
Automatically Detect SAs

CoPilot 1553 automatically detects subaddress activity associated with simulated or external RTs.

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

Note that the two detected subaddresses appear in the Hardware Explorer (see figure at right). Data for external SAs can be viewed, but not changed. This external SA data may, however, be linked to and viewed through CoPilot display windows.

- ► Click the **Stop** button to return to Edit mode


Demo1553Card1.CDV Demo Card
CH0 Version C (MIL-STD-1553)
Bus Controller - BC
BC Messages
Get Position (01-R-03-02)
Set Altitude (01-R-02-03)
UserSched1
Simulated RTs
Broadcast (RT31)
External RTs
RT01
RT01 SA01 (RX)
RT01 SA01 (TX)
Bus Monitor
Sequential Monitor

Hardware Explorer with two auto-detected external subaddresses

Related Topics

- RT31 is designed to receive broadcast messages but may be reassigned as a standard terminal if needed
- You can manually disable the auto-detection of RTs and SAs (through their context menus)

Summary

In this lesson you learned…

- how to automatically detect RTs addressed by the Bus Controller
- how to use the Terminal Simulation Assignment window to shadow monitor external RT activity
- how to detect and display subaddresses through the auto-detect SAs feature

In the next lesson you will learn how to simulate an RT, manually add subaddresses, and modify subaddress data respond to a BC message.

Lesson 5: Configure and Simulate RTs and SAs

Remote terminals respond to commands from the Bus Controller. These BC commands may direct the RT to receive data, to transmit data to the BC or another RT, to receive bus-wide broadcast commands, or to respond to mode codes. In most cases, 1553 protocol requires the RT to respond to BC commands with a status word. CoPilot allows the user to simulate all aspects of RT behavior as defined by MIL-STD-1553.

Introduced in This Lesson

RT Properties dialog, SA Properties dialog, Message View window, RT Subaddress View window

Objective

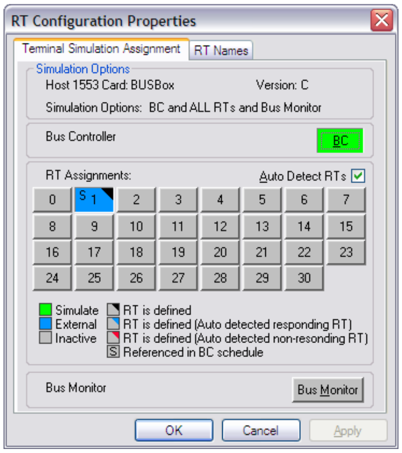
Simulate an INS (Inertial Reference System) terminal at RT address 1. Configure transmit subaddress 1 of RT01, define two data words in subaddress 3, and modify those values during run time.

Simulate the Detected RT

In Lesson 4 we illustrated how an external RT could be shadow monitored. In this lesson, we will simulate the behavior of a real RT (RT01).

- ► Right click on Simulated RTs in the Hardware Explorer and choose **Configure RTs** to open the RT Configuration dialog
- ► In the Terminal Simulation tab, click twice on the **RT01** S 1 button to select the simulate option (light green) and click **OK** to close the dialog

The light green button color S 1 indicates that this terminal is selected for simulation. The black triangle indicates that this RT was configured by the user (in this case, configured for simulation).



RT Configuration Properties

Host 1553 Card: BUSBox   Version: C

Simulation Options: BC and ALL RTs and Bus Monitor

Auto Detect RTs ☑

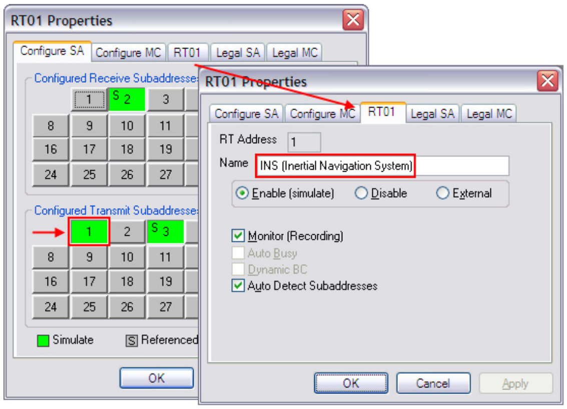
RT01 selected for explicit simulation in the TSA window

Configure the RT

- ► Right click RT01 and choose **Properties** to open the Properties dialog

Subaddresses are activated automatically when the Auto-detect SAs option is selected. Since the option is active by default, receive SA02 and transmit SA03 were detected when CoPilot was run in Lesson 4.

- ► Click the transmit **SA01** button (see figure)
- ► Click the **RT01** tab and enter **INS (Inertial Navigation System)** in the Name field
- ► Click **OK** to close the properties dialog



RT01 Properties

RT Address: 1

Name: INS (Inertial Navigation System)

● Enable (simulate) ○ Disable ○ External

☑ Monitor (Recording)

□ Auto Busy

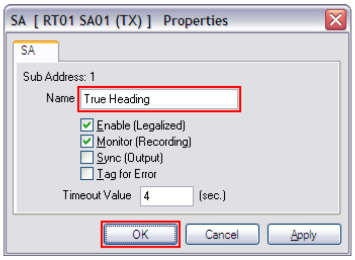
□ Dynamic BC

☑ Auto Detect Subaddresses

RT01 Properties window: Configure SA tab andRT01 tab

Name the Subaddresses

- ► Right click transmit SA01 and choose **Properties** to open the SA Properties dialog
- ► Type **True Heading** in the Name box and click **OK** to close the window



SA [RT01 SA01 (TX)] Properties

Sub Address: 1

Name: True Heading

☑ Enable (Legalized)

☑ Monitor (Recording)

□ Sync (Output)

□ Tag for Error

Timeout Value 4 (sec.)

SA Properties dialog for SA01

- ▶ Repeat this process to name **SA02** (type **Altitude**) and **SA03** (type **Present Position**)

Create a BC Message from a SA

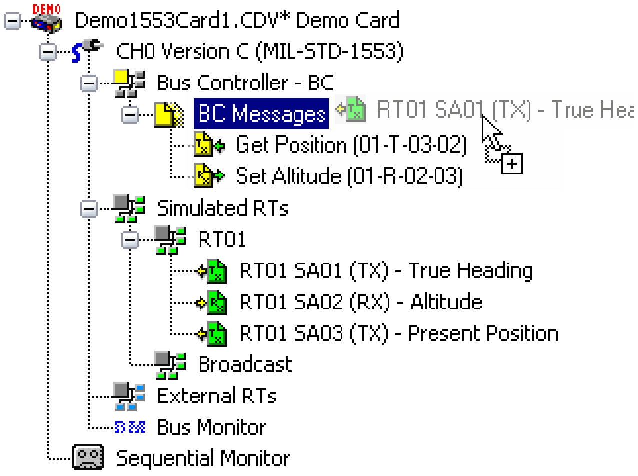
Quickly create a BC message by dragging a SA.

- ▶ Drag and drop the **True Heading** SA from RT01 onto the BC Messages branch (Message List)

The Message Editor (see Lesson 2) will automatically open so you can complete the necessary information.

- ▶ Enter **1** for the Subaddress and **2** in the Word Count field, type **Get Heading** in the Message Name field, and click **OK**

Note: Do not add the Get Heading message to the BC schedule at this time.


Demo1553Card1.CDV* Demo Card
CH0 Version C (MIL-STD-1553)
Bus Controller - BC
BC Messages
RT01 SA01 (TX) - True Hea
Get Position (01-T-03-02)
Set Altitude (01-R-02-03)
Simulated RTs
RT01
RT01 SA01 (TX) - True Heading
RT01 SA02 (RX) - Altitude
RT01 SA03 (TX) - Present Position
Broadcast
External RTs
Bus Monitor
Sequential Monitor

Dragging True Heading onto the Message List

Define and View SA Data

The Get Position message asks for two data words.

- ▶ Double click on the **Present Position** subaddress to open the Subaddress View window
- ▶ Enter **4321** in the first data cell and **DDDD** in the second (do not close Subaddress View window)
- ▶ Double click on **Get Position** to open view

The data words for Get Position are zeros because in previous runs, there was no *simulated* RT to respond.

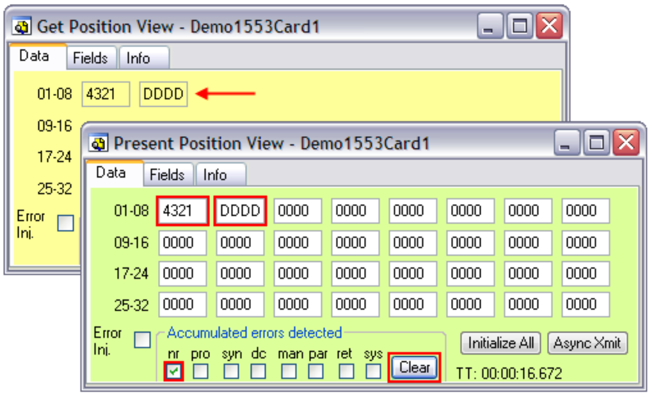
- ▶ Click the **Run** button in the CoPilot toolbar

Present Position (SA03) responds to the BC Schedule by transmitting two data words, and you see the data in the Message View window update in real time.

- ▶ Click the **Clear** Clear button in the Accumulated Errors frame of **Get Position** and **Present Position** to clear the no response (nr) flag

This flag remains clear because the RT is responding.

- ▶ Click the **Stop** button to end the simulation
- ▶ **Close** both view windows


Get Position View - Demo1553Card1
Data
Fields
Info
Present Position View - Demo1553Card1
Data
Fields
Info

Get Position View:

01-08	4321	DDDD
09-16		
17-24		
25-32		

Error Ini. ☐

Present Position View:

01-08	4321	DDDD	0000	0000	0000	0000	0000	0000
09-16	0000	0000	0000	0000	0000	0000	0000	0000
17-24	0000	0000	0000	0000	0000	0000	0000	0000
25-32	0000	0000	0000	0000	0000	0000	0000	0000

Error Ini. ☐ Accumulated errors detected: nr ☑ pro ☐ syn ☐ dc ☐ man ☐ par ☐ ret ☐ sys ☐ Clear | Initialize All | Async Xmit | TT: 00:00:16.672

Get Position Message View and Present Position SA View

Related Topics

- Other tabs in the RT Properties page configure mode code options and status bits

Summary

In this lesson you learned…

- how to simulate and configure an RT
- how to simulate and configure a subaddress
- how to create a BC message from a subaddress
- how to view a subaddress and modify its data

In the next lesson you will monitor these messages and view them through the Monitor View window.

Lesson 6: Monitor and Record the Databus

To this point, you have viewed the current value of selected messages using Message View and Subaddress View. With CoPilot's powerful Bus Monitor, you can capture a sequential history of bus activity and view that data in several formats through the Monitor View window.

Bus Monitor and Sequential Monitor

MIL-STD-1553 Bus Monitor properties are defined using the Bus Monitor BM icon in the Hardware Explorer. Multi-channel and/or multi-protocol boards (such as BUSBox or OmniBus) have both a Bus Monitor for configuration and a Sequential Monitor that manages the data collection. Since the 1553 Demo Card is an emulation of a BUSBox card, both monitor icons will appear in the Hardware Explorer tree. If you are working with a single-channel Ballard 1553 board, all references to Sequential Monitor in the instructions below should be translated to Bus Monitor.

Introduced in This Lesson

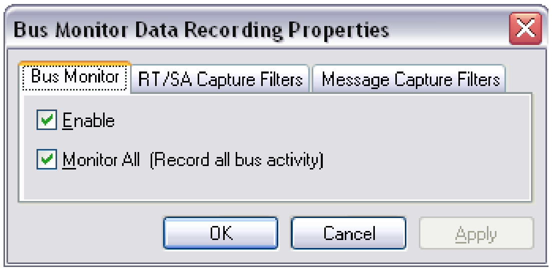
Monitor View, recording controls, Monitor Data (MonData) files, Display Filter dialog, Bus Monitor Properties dialog

Objective

Monitor the databus, view the record in the Monitor View window, pause and resume the recording, then filter the Get Position message out of the display. Finally, change the Monitor View display from Full to Form mode.

Configure the Bus Monitor

- ▶ Right click on the dimmed Bus Monitor BM icon in the Hardware Explorer (disabled by default) and choose **Properties** from the context menu to open the Bus Monitor Properties window
- ▶ Check the boxes for **Enable** and **Monitor All** in the Bus Monitor tab (default tab)
- ▶ Click **OK** to close the properties page



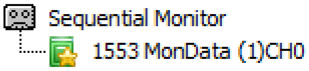
☑ Enable

☑ Monitor All (Record all bus activity)

Bus Monitor tab in the Bus Monitor Properties dialog

The Bus Monitor icon will change to indicate that it is enabled BM. It is now configured to capture all bus traffic.

- ▶ Click the **Run** button in the CoPilot toolbar



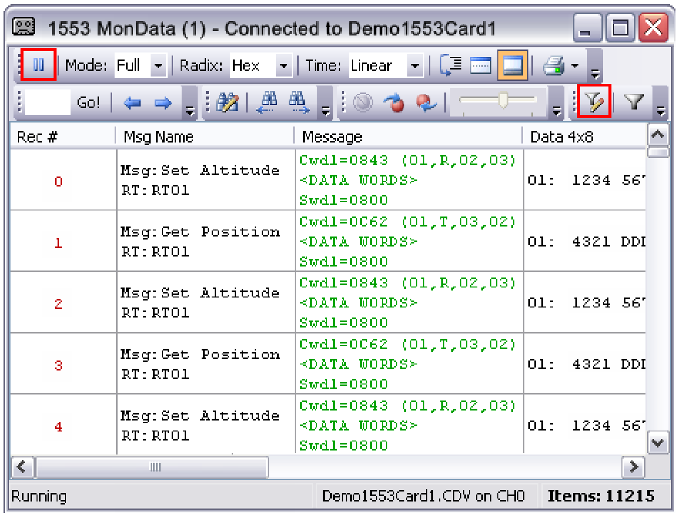
Sequential Monitor with one Monitor Data File

A new Monitor Data Icon is added under the Sequential Monitor holding the newly captured data. The status column text displays the record count.

Open the Monitor View Window

- ▶ Right click on the Monitor Data and choose **View Monitor Data**

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



Mode: Full | Radix: Hex | Time: Linear

Rec #	Msg Name	Message	Data 4x8

0	Msg:Set Altitude
	RT:RT01
		Cwd1=0843 (01,R,02,03)
		<DATA WORDS>
		Swd1=0800
			01: 1234 56

1	Msg:Get Position
	RT:RT01
		Cwd1=0C62 (01,T,03,02)
		<DATA WORDS>
		Swd1=0800
			01: 4321 DDI

2	Msg:Set Altitude
	RT:RT01
		Cwd1=0843 (01,R,02,03)
		<DATA WORDS>
		Swd1=0800
			01: 1234 56

3	Msg:Get Position
	RT:RT01
		Cwd1=0C62 (01,T,03,02)
		<DATA WORDS>
		Swd1=0800
			01: 4321 DDI

4	Msg:Set Altitude
	RT:RT01
		Cwd1=0843 (01,R,02,03)
		<DATA WORDS>
		Swd1=0800
			01: 1234 56

Running Demo1553Card1.CDV on CH0 **Items: 11215**

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 Monitor View display can be limited by display filter criteria (monitor record is unaffected).

- ▶ Click the **Edit Filter** button in the Display Filter frame to open the Monitor Display Filter Properties dialog (see figure at right)

As you can see in the RT/SA Filters tab (default tab), you can filter out specific RTs and SAs.

- ▶ Click on the **Message Filters** tab and click on **Get Position** to clear its checkmark

Note: The Message Filters tab lists all messages defined in the Message List even if they are not used in the BC schedule (Get Heading is not in the schedule).

- ▶ Click **OK** to close the dialog and view the results

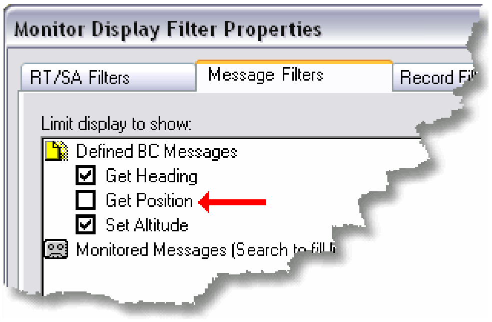
Notice that the record numbers in the left column are not sequential (e.g., 0, 2, 4, etc.) The filtered out messages are not displayed.

The Full display (default view mode) is a color-coded multi-line display. You can see several records at once.

- ▶ Click the display Mode droplist to change to the **Form** display mode

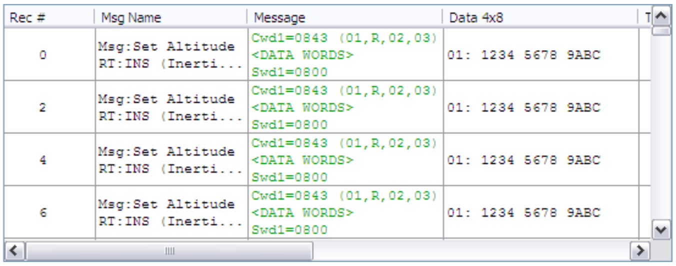
In Form view display mode, messages are displayed in a one-per-page format so that changes in command word, error conditions, or data are easily detected.

- ▶ Right click on the Bus Monitor BM and deselect **Enable** (it is not needed for the next few lessons)



Limit display to show:
Defined BC Messages
☑ Get Heading
☐ Get Position
☑ Set Altitude
Monitored Messages (Search to fill i

Display Filter dialog with Get Position selected for filtering



Rec #	Msg Name	Message	Data 4x8

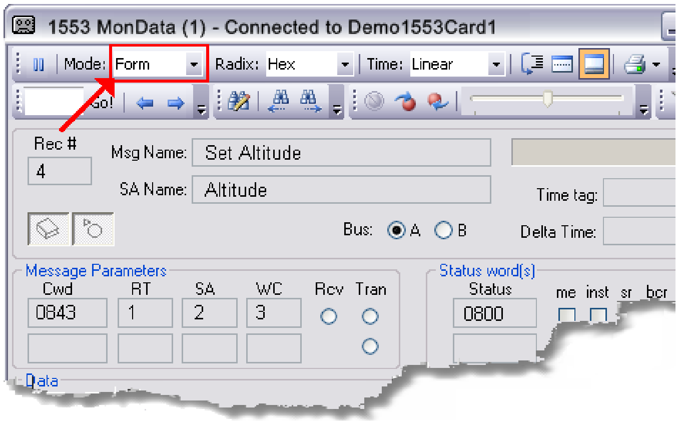
0	Msg:Set Altitude
	RT:INS (Inerti...
		Cwd1=0843 (01,R,02,03)
		<DATA WORDS>
		Swd1=0800
			01: 1234 5678 9ABC

2	Msg:Set Altitude
	RT:INS (Inerti...
		Cwd1=0843 (01,R,02,03)
		<DATA WORDS>
		Swd1=0800
			01: 1234 5678 9ABC

4	Msg:Set Altitude
	RT:INS (Inerti...
		Cwd1=0843 (01,R,02,03)
		<DATA WORDS>
		Swd1=0800
			01: 1234 5678 9ABC

6	Msg:Set Altitude
	RT:INS (Inerti...
		Cwd1=0843 (01,R,02,03)
		<DATA WORDS>
		Swd1=0800
			01: 1234 5678 9ABC

Monitor View with Get Position filtered out



Mode: Form Radix: Hex Time: Linear

Rec # 4

Msg Name: Set Altitude

SA Name: Altitude

Time tag:

Bus: ● A ○ B

Delta Time:

Message Parameters

Cwd	RT	SA	WC	Rcv	Tran
0843	1	2	3	○	○
					○

Status word(s)

Status	me	inst	sr	bcr
0800	□	□		

Form button—Monitor View in Form display mode

Related Topics

- View1/View2 display modes (same button as Full and Form) can be uniquely named and configured by the user
- The Monitor View toolbar also contains buttons for other features such as Search, Go To, and Step, etc.

Summary

In this lesson you learned…

- how to configure and simulate a Bus Monitor
- how to pause and resume Monitor recording
- how to view logged data from a monitor data file
- how to configure the Monitor View display

You have now covered the basics of CoPilot 1553 simulation. In the next section, you will discover how to create, interpret, modify, generate, and view data using fields, editors, and display windows.

THIS PAGE INTENTIONALLY BLANK

Section B: Additional Features

Review

In the preceding section, you learned how to use the 1553 Quick Start Analyzer, create BC messages, define a BC schedule, simulate an RT, configure subaddresses, monitor bus activity, and view the monitor record. The lessons in Section B will guide you through the process of creating, encoding, interpreting, modifying, displaying, and playing back MIL-STD-1553 data in the CoPilot environment.

Introduced in This Section

All of the features described in this section and in Section A are available in the standard version of CoPilot for MIL-STD-1553. (Section C discusses CoPilot 1553 Professional features). The "additional features" in this section are differentiated from the basic concepts introduced in the first section, however, they are still just as easy to use. The first two lessons in this section deal with engineering unit interpretations on data fields. These lessons are followed by lessons that automatically generate changing data and show how data can be displayed for analysis. This section concludes with a lesson on software playback.

- Lesson 7 Data fields and field interpreters are created for engineering unit conversion of data and editing.
- Lesson 8 The CoPilot MIL-STD-1553 database is used for storing and loading configuration information.
- Lesson 9 The Raw Editor and Data Generator allow data field values to be edited and dynamically modified.
- Lesson 10 Information on the MIL-STD-1553 bus is displayed using 1553 display windows.
- Lesson 11 Captured data may be replayed through CoPilot displays (like a moving map view) using the software playback feature.

Looking Ahead

Before you begin the lessons in this section, read on to learn about engineering units in CoPilot, how to work with view windows, and important information about the various files in a CoPilot project. CoPilot Standard also provides the ability to playback on the databus. Although hardware playback is not possible with a demonstration card, the operation of hardware playback with real hardware functions similarly to the software playback feature described in this guide. For more information on retransmitting recorded data with hardware playback, please refer to the CoPilot User's Manual or the online help.

Engineering Units

MIL-STD-1553 does not specify the meaning or field position of data in transferred messages. Therefore, data in each message may be completely redefined by the user for each application. With CoPilot, 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 MIL-STD-1553 databus. CoPilot converts the raw data that come across the bus into recognizable "engineering units" based on field and interpreter definitions supplied by the user.

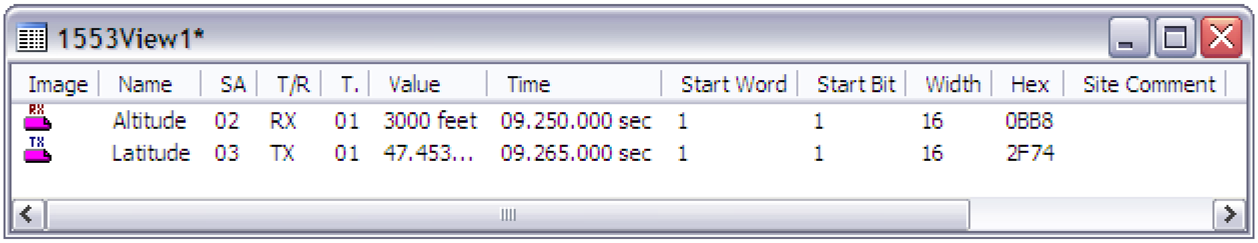
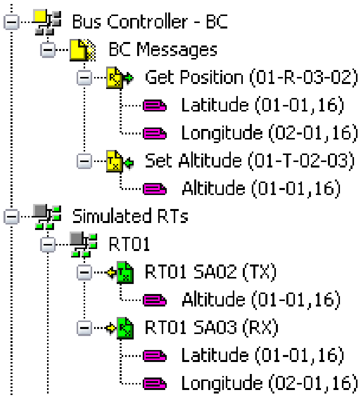


Image	Name	SA	T/R	T.	Value	Time	Start Word	Start Bit	Width	Hex	Site Comment
RX	Altitude	02	RX	01	3000 feet	09.250.000 sec	1	1	16	0BB8	
TX	Latitude	03	TX	01	47.453...	09.265.000 sec	1	1	16	2F74	

In this way, data is given meaning within the context of the CoPilot simulation. This meaning can be preserved and refined by saving to, loading from, or modifying the user-defined 1553 database.

Data Fields

Data fields are used within CoPilot to associate meaning with a bit or sequence of bits in a MIL-STD-1553 message or RT subaddress. Each field can be named, assigned units of measurement, and associated with an interpreter so that its bits may be accurately deciphered. The figure below illustrates data fields (Altitude, Latitude, and Longitude) in BC messages and RT subaddresses.



- Bus Controller - BC
  - BC Messages
    - Get Position (01-R-03-02)
      - Latitude (01-01,16)
      - Longitude (02-01,16)
    - Set Altitude (01-T-02-03)
      - Altitude (01-01,16)
- Simulated RTs
  - RT01
    - RT01 SA02 (TX)
      - Altitude (01-01,16)
    - RT01 SA03 (RX)
      - Latitude (01-01,16)
      - Longitude (02-01,16)

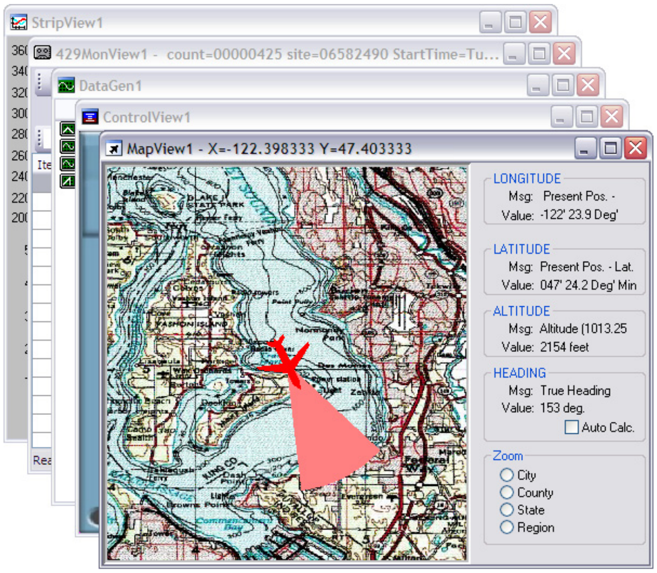
Data Interpreters

The data interpreters built into CoPilot 1553 allow you to customize how data is encoded and translated. For example, you could encode data in 2's complement or BCD (Binary Coded Decimal) format, encode alphanumeric characters in ASCII format, or indicate conditions or codes using discretes.

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 9), Engineering View (Lesson 10), 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.



StripView1
429MonView1 - count=00000425 site=06582490 StartTime=Tu...
DataGen1
ControlView1
MapView1 - X=-122.398333 Y=47.403333

LONGITUDE
Msg: Present Pos. -
Value: -122' 23.9 Deg'

LATITUDE
Msg: Present Pos. - Lat.
Value: 047' 24.2 Deg' Min

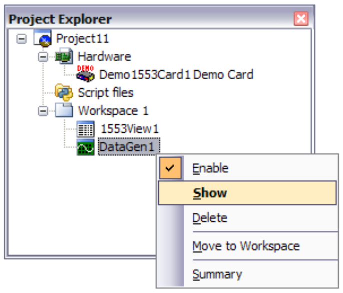
ALTITUDE
Msg: Altitude (1013.25
Value: 2154 feet

HEADING
Msg: True Heading
Value: 153 deg.
□ Auto Calc.

Zoom
○ City
○ County
○ State
○ Region

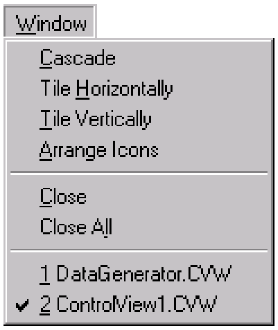
Working with View Windows

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 its **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 above).

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 <Delete> 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

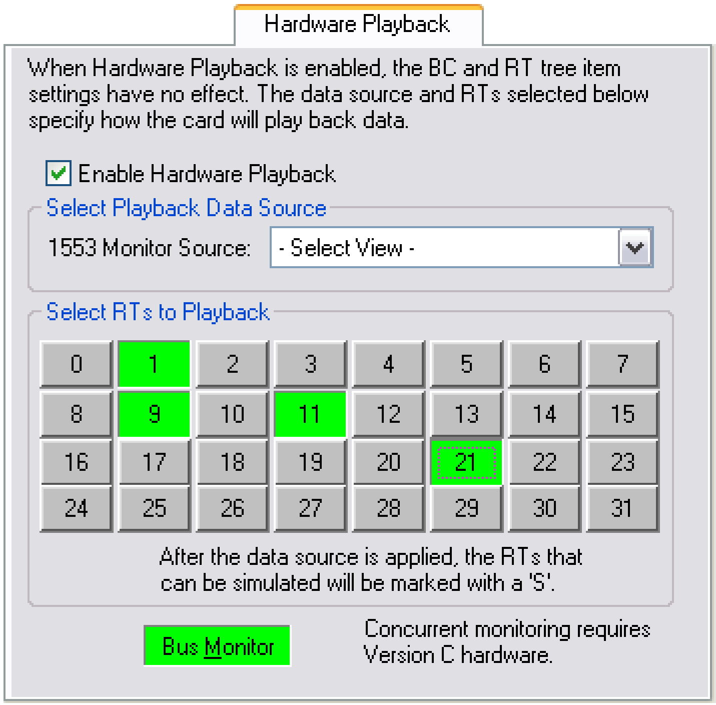
When a new view is created, CoPilot assigns a default name, for example, “DataGen1” (see figure on previous page). When the view window is saved to disk, the .CVW extension is added. When developing a new project, it is often helpful to save view windows as you go along.

► 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.

Playback

The CoPilot software supports both software and hardware playback. Software playback is the replaying of recorded sequential monitor data through the displays and hardware objects to analyze the databus values and activity as if it were being transmitted again with the ability to start/stop/pause and control the playback rate. Although the 1553 demonstration card does not support hardware playback, hardware playback is used with real hardware to replay previously recorded information on the bus to communicate with and test external equipment. See the user’s manual for additional information about the operation and advanced features of hardware playback.



Hardware Playback

When Hardware Playback is enabled, the BC and RT tree item settings have no effect. The data source and RTs selected below specify how the card will play back data.

☑ Enable Hardware Playback

Select Playback Data Source

1553 Monitor Source: - Select View -

Select RTs to Playback

0	[✓ 1]	2	3	4	5	6	7
8	[✓ 9]	10	[✓ 11]	12	13	14	15
16	17	18	19	20	[✓ 21]	22	23
24	25	26	27	28	29	30	31

After the data source is applied, the RTs that can be simulated will be marked with a 'S'.

Bus Monitor

Concurrent monitoring requires Version C hardware.

Hardware playback configuration (not available with the 1553 demo card)

THIS PAGE INTENTIONALLY BLANK

Lesson 7: Define Data Fields and Interpreters

CoPilot allows you to insert a data field of varying lengths in any data word, at any starting bit, and even spanning two or more data words. CoPilot's interpreters allow you to encode data in many formats, such as 2's complement or BCD (Binary Coded Decimal) format, alpha-numeric characters in ASCII format, or discretes indicating conditions or states.

Introduced in This Lesson

Configure Fields dialog, Field Selector dialog (Create New tab), Interpreter Properties dialog, Engineering Units Editor

Objective

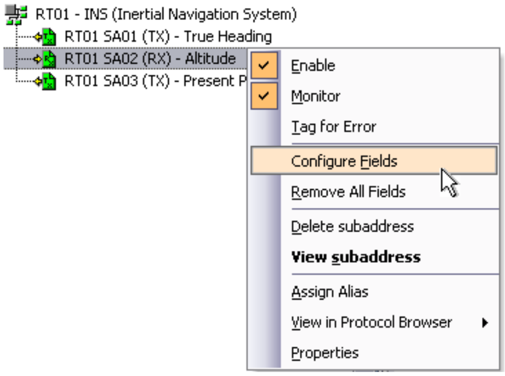
Define a one-word data field and interpreter for inertial altitude to an RT subaddress and assign that same field to a BC message. Set the value of these fields in the BC message and run the schedule.

Insert a New Data Field

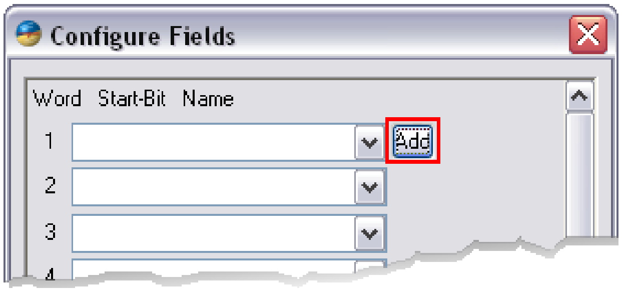
- ▶ Right click on SA02 - Altitude and choose **Configure Fields** from the context menu to open the Configure Fields dialog
- ▶ Select data word **1** by clicking its listbox (highlight will appear) and click the **Add** Add button to open the Field Selector dialog (see figure at right)

Leave the default starting bit, field width, and interpreter type (see figure below right) as they are.

- ▶ Replace "Field1" with **Altitude** in the Name box
- ▶ Enter **feet** in the Units box and click **OK** to open the Interpreter Properties dialog



Context menu for SA02



Configure Fields dialog with Word 1 selected

Define the Data Interpreter

Change the BNR (binary) interpreter default from a signed value to an unsigned value.

- ▶ Click the **BNR** tab (Interpreter Properties dialog) and select the **Unsigned Values** option button

CoPilot automatically sets the low range to 0 (zero).

- ▶ Enter **65535** in the second range box (notice that the Resolution field updates as you type)
- ▶ Click **OK** to close the Interpreter Properties page and **OK** to close the Configure Fields dialog

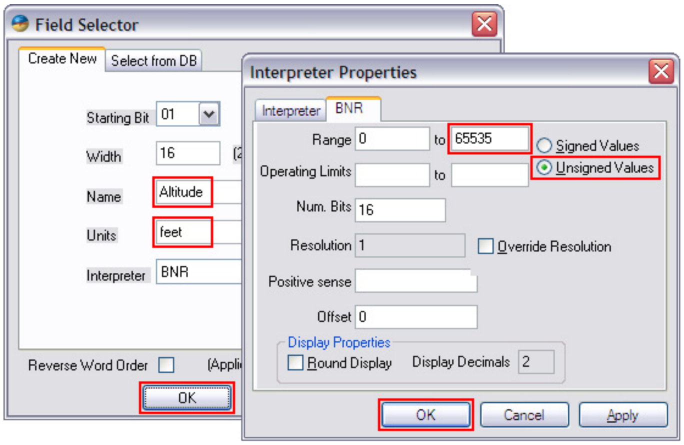
The Altitude field is now visible in the Hardware Explorer.

Copy 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 Explorer.

- ▶ Drag and drop the **Altitude** field of the SA on to the **Set Altitude** BC message

The Altitude field and all its properties are copied into the message (see figure on next page).



Starting Bit 01
Width 16
Name Altitude
Units feet
Interpreter BNR
Reverse Word Order □ (Appli

Range 0 to 65535
○ Signed Values
● Unsigned Values
Operating Limits to
Num. Bits 16
Resolution 1 □ Override Resolution
Positive sense
Offset 0
□ Round Display Display Decimals 2

Field Selector and Interpreter Properties dialogs for Altitude

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 on the Altitude field in the Set Altitude BC message (not SA02) and choose **Edit Data** to open the Engineering Units Editor

You can enter a new value in the Engineering Units Editor through the textbox or using the slidebar.

- ▶ Enter **3000** feet in the Numeric Value textbox and press the **Enter** key to apply the new value

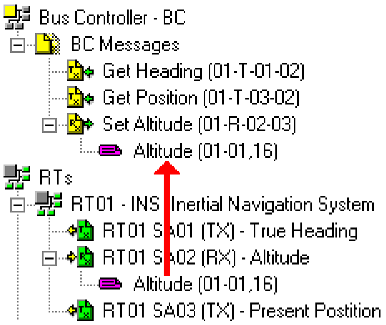
Note: By setting a new value in this message by means of a field, you have *overwritten* the previous data word value (1234 hex / 4660 decimal) from Lesson 1.

- ▶ Click outside of the Engineering Units Editor window or press **ESC** to close the editor

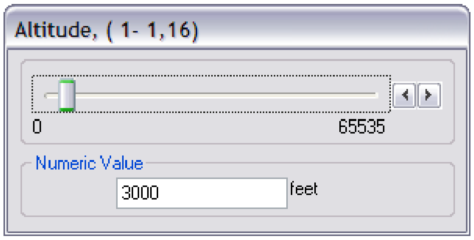
You can see how much easier it is to enter data here in real-world "engineering units," without having to translate into a different radix such as hex or binary.

Run the Simulation

- ▶ Hover the mouse pointer over the Altitude field in SA02 to view the engineering units tool tip display, which should read "4660 feet" (1234 hex)
- ▶ Click the **Run** button to start the simulation
- ▶ Note: If a monitor warning dialog appears, select **Don't ask me again** and click **Continue** to clear it
- ▶ Hover the mouse pointer again over the Altitude field in SA02 to view the value after is was updated by the BC message (should be "3000 feet")
- ▶ Click the Stop button to end the simulation

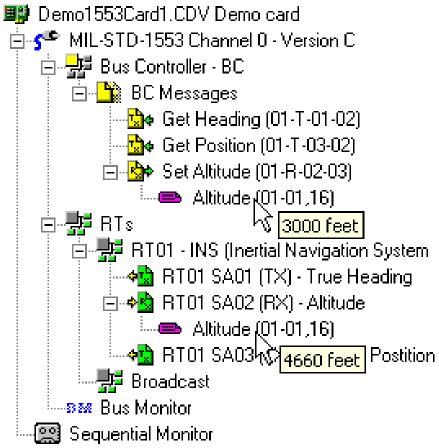

Bus Controller - BC
BC Messages
Get Heading (01-T-01-02)
Get Position (01-T-03-02)
Set Altitude (01-R-02-03)
Altitude (01-01,16)
RTs
RT01 - INS Inertial Navigation System
RT01 SA01 (TX) - True Heading
RT01 SA02 (RX) - Altitude
Altitude (01-01,16)
RT01 SA03 (TX) - Present Postition

Altitude field propagated to the Set Altitude message



Numeric Value: 3000 feet

Engineering Units Editor for Altitude field in Set Altitude BC Message


Demo1553Card1.CDV Demo card
MIL-STD-1553 Channel 0 - Version C
Bus Controller - BC
BC Messages
Get Heading (01-T-01-02)
Get Position (01-T-03-02)
Set Altitude (01-R-02-03)
Altitude (01-01,16)
3000 feet
RTs
RT01 - INS (Inertial Navigation System
RT01 SA01 (TX) - True Heading
RT01 SA02 (RX) - Altitude
Altitude (01-01,16)
RT01 SA03
4660 feet
Postition
Broadcast
Bus Monitor
Sequential Monitor

Demo card with tool tip displays for Altitude fields

Related Topics

- There are seven interpreters: BNR, BCD, Decimal, Discrete, IEEE 32-bit, Simple Scalar, and Custom Script
- Data fields may be viewed with CoPilot displays such as an Engineering View (Lesson 10) and Strip View (Lesson 12)

Summary

In this lesson you learned…

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

Now that you have created and defined a data field, in Lesson 8 you learn how to save this field to the database and use the database to populate the Hardware Explorer.

Lesson 8: Use the 1553 Database

CoPilot allows users to define fields for BC messages and RT subaddresses and store them in the CoPilot 1553 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

Field Selector dialog (Select from DB tab) for loading database definitions, Choose RT Assignment dialog

Objective

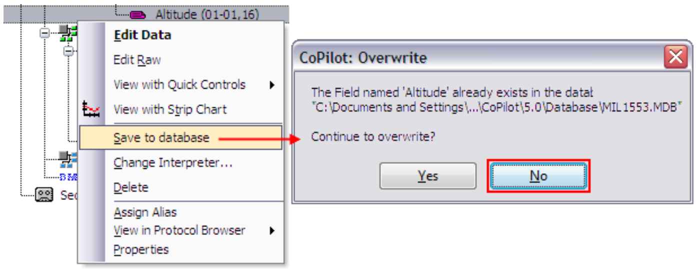
Learn how to save the Altitude field (from Lesson 7) to the database, and then load the Latitude field from the database. Expedite the configuration process by loading the complete RT definition for RT01 from the database.

Save a Field to the Database

Although we do not wish to overwrite the definition of Altitude currently in the database, we will examine the process of saving a data field to the 1553 database.

- ▶ Right click on the Altitude field in BC message Set Altitude and choose **Save to database** from the context menu to open the CoPilot Database Save dialog (**do not click Yes**)
- ▶ Click **No** to avoid overwriting this field in the 1553 database

Note: Use caution when saving to the database. Once overwritten, a previously stored definition is lost.



The Field named 'Altitude' already exists in the datal "C:\Documents and Settings\...\CoPilot\5.0\Database\MIL1553.MDB"

Continue to overwrite?

Saving the Altitude field to the database from the context menu

Load Fields from the Database

Loading a predefined data field from the database uses the same process as in Lesson 7, except that in the Field Selector, you will choose the database tab.

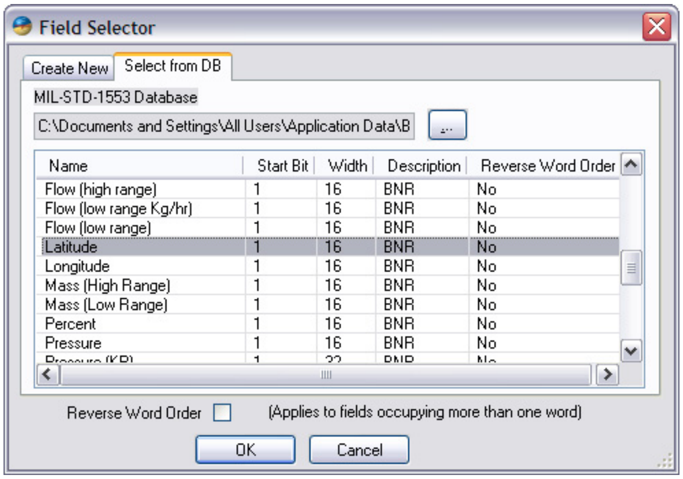
- ▶ Right click on SA03 - Present Position and choose **Configure Fields** from the context menu to open the Configure Fields dialog
- ▶ Select data word **1** by clicking its listbox and click **Add** Add to open the Field Selector dialog
- ▶ In the Field Selector dialog, choose the **Select from DB** tab to access the 1553 database

Notice that the fields are sorted by name, start bit, field width, and interpreter type.

- ▶ Scroll down, select the **Latitude** field, and click **OK** to open the Interpreter Properties dialog

If you wished to modify the properties of this field before inserting it, you would do so here through the Interpreter Properties dialog (see Lesson 7).

- ▶ Examine the properties in the two tabs if you wish, then click **OK** to close the Interpreter Properties dialog (do not close the Configure Fields dialog)



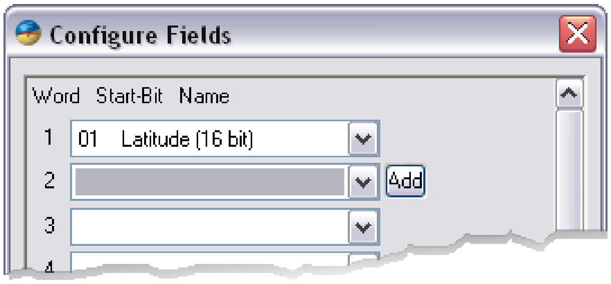
MIL-STD-1553 Database

C:\Documents and Settings\All Users\Application Data\B

Name	Start Bit	Width	Description	Reverse Word Order
Flow (high range)	1	16	BNR	No
Flow (low range Kg/hr)	1	16	BNR	No
Flow (low range)	1	16	BNR	No
Latitude	1	16	BNR	No
Longitude	1	16	BNR	No
Mass (High Range)	1	16	BNR	No
Mass (Low Range)	1	16	BNR	No
Percent	1	16	BNR	No
Pressure	1	16	BNR	No
Pressure (KP)	1	32	BNR	No

Reverse Word Order □ (Applies to fields occupying more than one word)

Select from Database tab with Latitude field highlighted



Word	Start-Bit	Name
1	01	Latitude (16 bit)
2		
3		
4		

Configure Fields dialog with Latitude field (and Word 2 selected)

The Latitude Field is listed in Word 1 of SA 3 in the Configure Fields dialog (see figure on previous page). Notice that the Edit button is now available for that word.

- ▶ Select data word **2** by clicking its listbox (highlight will appear) and click the **Add** button to open the Field Selector dialog
- ▶ Repeat the same steps to add the **Longitude** field, then click **OK** to close the Configure Fields dialog

The Latitude and Longitude fields now appear in words 1 and 2 of SA03 - Present Position.

Load Complete RT Definition

You could continue to load fields one at a time as above, but there is an even quicker way to finish configuring RT01. Complete RT definitions can be saved to and loaded from the 1553 database.

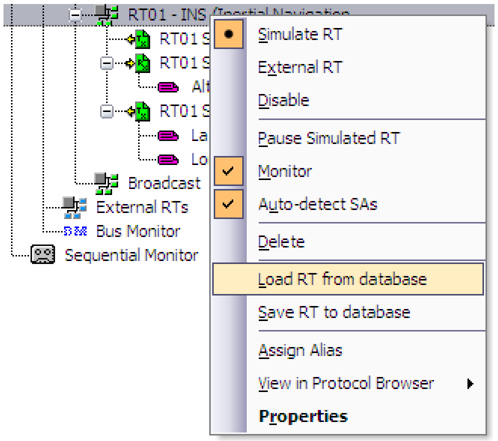
- ▶ Right click on RT01 and choose **Load RT from database** to open the Choose RT Assignment window

A dialog will appear warning you that this action will overwrite existing SAs and fields.

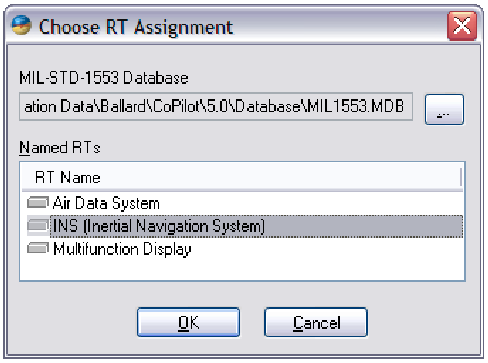
- ▶ Click **YES** to clear the RT and close the dialog
- ▶ Select **INS (Inertial Navigation System)** and click **OK** to load the RT

The RTs branch in the Hardware Explorer should now look like the figure to the right. Although your previous RT01 has been overwritten, you can see that the new RT01 contains the SAs and fields you had before, plus newly added items.

In addition to loading an RT from the database, an RT's configuration can be saved to the database then loaded from the database at a later time.



Context menu for RT01

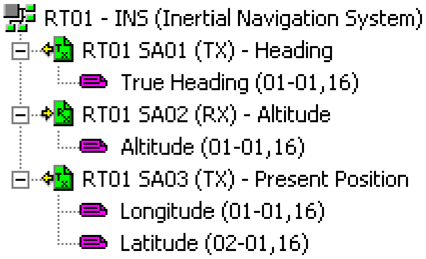


MIL-STD-1553 Database: ation Data\Ballard\CoPilot\5.0\Database\MIL1553.MDB

Named RTs

RT Name
Air Data System
INS (Inertial Navigation System)
Multifunction Display

Choose RT Assignment dialog



RT01 - INS (Inertial Navigation System)
RT01 SA01 (TX) - Heading
True Heading (01-01,16)
RT01 SA02 (RX) - Altitude
Altitude (01-01,16)
RT01 SA03 (TX) - Present Position
Longitude (01-01,16)
Latitude (02-01,16)

Complete INS terminal (RT01) as loaded from the database

Related Topics

- When loading from the database, the **Browse** ... button can be used to change databases
- You can select the default database through the MIL-STD-1553 Properties tab in the Project | Options dialog

Summary

In this lesson you learned…

- how to save a field to the 1553 database
- how to load a field from the database
- how to load a complete RT from the database

Now that you have populated your tree from the database and you have a variety of data fields to work with, you will learn in Lesson 9 about writing and generating data in CoPilot 1553.

Lesson 9: Generate Dynamic Data

In the previous lessons, the data values for message 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 C (CoPilot Professional). You will also use the Raw Editor to edit data in various radices while comparing against the engineering units value.

Introduced in This Lesson

New Hardware and Views dialog, Data Generator view, Data Generator Properties dialog, Raw Editor

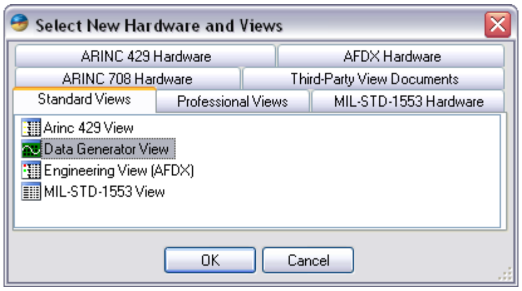
Objective

Use the Data Generator to drive dynamic data values in the Latitude, Longitude, and True Heading fields. Use the Raw Editor to change the value of the Altitude field in the BC Set Altitude message.

Open the Data Generator

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

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



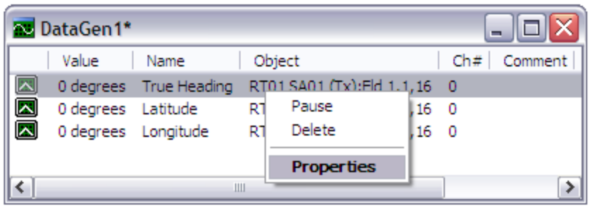
New Hardware and Views dialog with Data Generator selected

Assign Data Fields to Generator

The Data Generator can accept transmit data fields from messages and subaddresses.

- Click on the **True Heading** field in the Hardware Explorer and drag it to the Data Generator
- Drag and drop the **Present Position** SA03 icon in the Hardware Explorer into the Data Generator

Dragging a parent object into the Data Generator automatically adds all fields that branch beneath that object.


DataGen1*

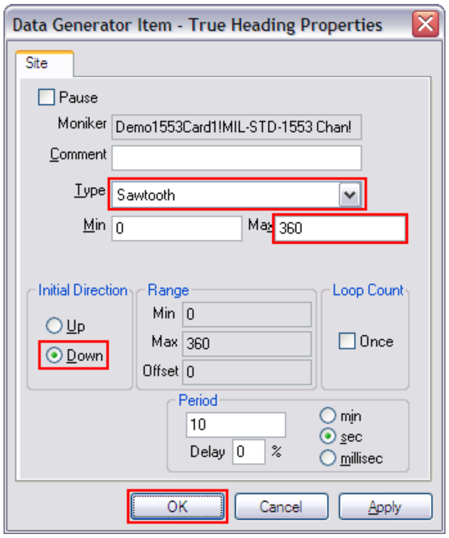
Value	Name	Object	Ch#	Comment
0 degrees	True Heading	RT01 SA01 (Tx):Fld 1.1,16	0	
0 degrees	Latitude	RT ,16	0	
0 degrees	Longitude	RT ,16	0	

Data Generator 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 holding pattern in a moving map display in Lesson15.


Data Generator Item - True Heading Properties

Site

☐ Pause

Moniker: Demo1553Card1!MIL-STD-1553 Chan!

Comment:

Type: Sawtooth

Min: 0 Max: 360

Initial Direction: ○ Up ● Down

Range: Min 0, Max 360, Offset 0

Loop Count: ☐ Once

Period: 10 Delay 0 % ○ min ● sec ○ millisec

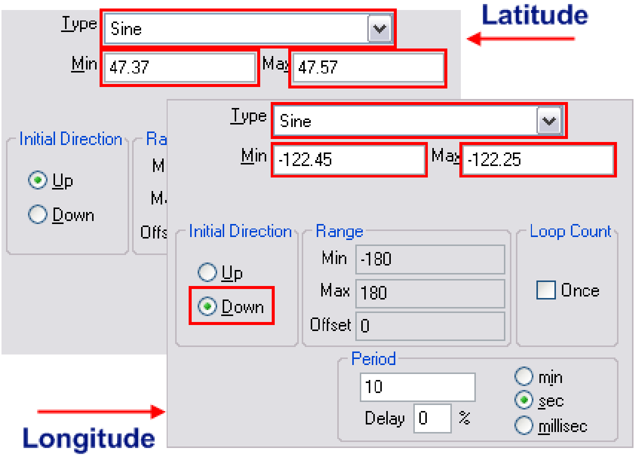
Data Generator Properties for True Heading

- ▶ 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**, the initial direction to **Down**, and click **OK**

These values will be used to position the airplane object in the moving map display in Lesson 15.

- ▶ 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: Type: Sine; Min: 47.37; Max: 47.57; Initial Direction: ● Up ○ Down

Longitude: Type: Sine; Min: -122.45; Max: -122.25; Initial Direction: ○ Up ● Down; Range: Min -180, Max 180, Offset 0; Loop Count: □ Once; Period: 10, Delay 0 %, ○ min ● sec ○ millisec

Latitude properties (left) and Longitude properties (right)

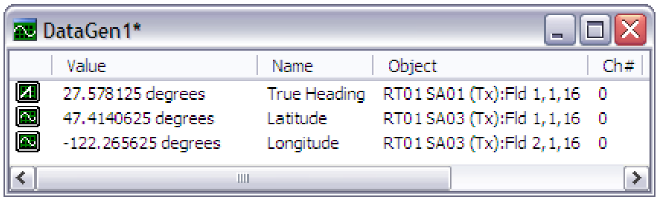
Use the Raw Editor

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

- ▶ Right click on the **Altitude** field in the *Set Altitude BC message* and choose **Edit Raw** from the context menu to open the Raw Editor
- ▶ Click the **Hex** button (value is 0BB8 / 3000 feet)
- ▶ Enter the value **0FA0** (hex) and click **Apply**

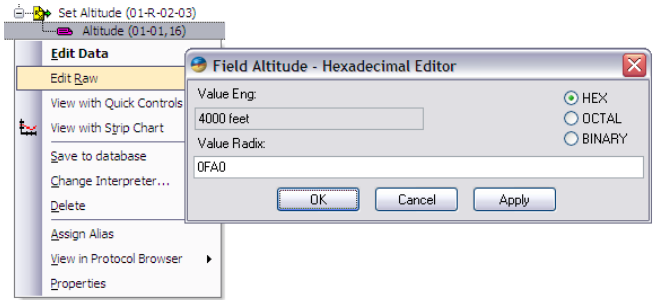
You will see the translation of 0FA0 hexadecimal in engineering units (4000 feet) in the Value Eng. field.

- ▶ 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, **Minimize** the Data Generator then save the project.



Value	Name	Object	Ch#
27.578125 degrees	True Heading	RT01 SA01 (Tx):Fld 1,1,16	0
47.4140625 degrees	Latitude	RT01 SA03 (Tx):Fld 1,1,16	0
-122.265625 degrees	Longitude	RT01 SA03 (Tx):Fld 2,1,16	0

Data Generator with changing data values



Value Eng: 4000 feet
Value Radix: 0FA0
● HEX ○ OCTAL ○ BINARY

Raw Editor opened from the Altitude 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 generator properties for individual fields
- how to use the Raw Editor

In the next lesson you will learn how to view messages, subaddresses and fields in a special display container.

Lesson 10: View and Organize Data

The Engineering View is a window for displaying BC messages, RT subaddresses, fields, and objects from other protocols such as ARINC 429 and AFDX. Items assigned to this view 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

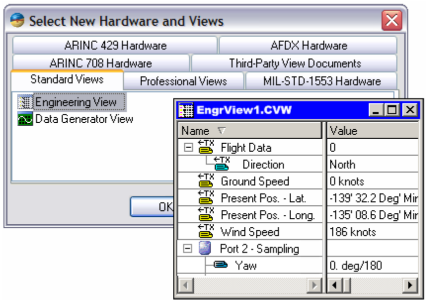
Open a Engineering View window and assign the three data fields linked to the Data Generator. Customize the display by moving, sorting, adding columns, and assigning another data field. Finally, edit a field from within an Engineering View window.

Open an Engineering View

- ▶ Click the **New** button to open the New Hardware and Views dialog
- ▶ Select the **Standard Views** tab, choose **Engineering View**, and click **OK**

The Engineering View 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 9).



EngrView1.CVW

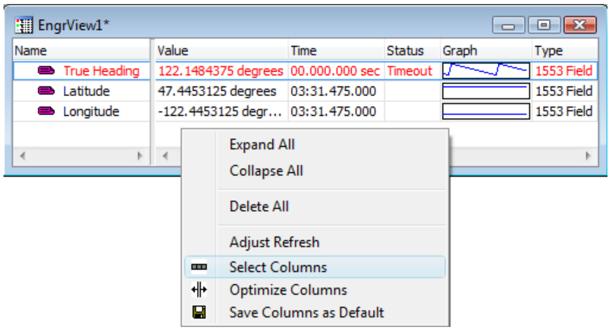
Name	Value
Flight Data	0
Direction	North
Ground Speed	0 knots
Present Pos. - Lat.	-139' 32.2 Deg' Mir
Present Pos. - Long.	-135' 08.6 Deg' Mir
Wind Speed	186 knots
Port 2 - Sampling	
Yaw	0. deg/180

Standard Views tab in the New Hardware and Views dialog

View Field Data Dynamically

- ▶ Drag and drop **True Heading**, **Latitude**, & **Longitude** fields from RT1 into EngrView1
- ▶ Click the **Run** button to start the simulation

View the changing data values in the Value column.


EngrView1*

Name	Value	Time	Status	Graph	Type
True Heading	122.1484375 degrees	00.000.000 sec	Timeout		1553 Field
Latitude	47.4453125 degrees	03:31.475.000			1553 Field
Longitude	-122.4453125 degr…	03:31.475.000			1553 Field

Engineering View window with three data fields and context menu

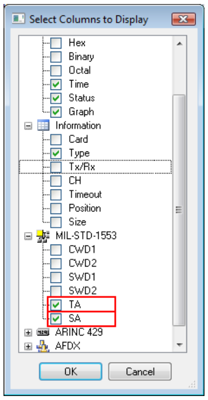
Customize the Display

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

- ▶ Right click inside the Engineering View (below the items) and choose **Select Columns** from the context menu to open the Select Columns dialog
- ▶ In the MIL-STD-1553 section, set the check marks for SA, and TA and click **OK**

(These and any other columns can be shown/hidden in the same way.) Now the SA column will be moved.

- ▶ Move the scroll bar until the **SA** column is visible
- ▶ Click and hold on the **SA** column title and drag it to the left until the space between Value and Time

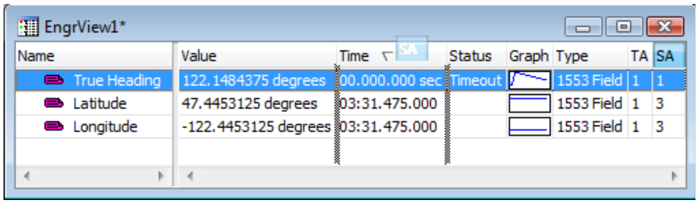

Select Columns to Display

☐ Hex
☐ Binary
☐ Octal
☑ Time
☑ Status
☑ Graph
Information
☐ Card
☑ Type
☐ Tx/Rx
☐ CH
☐ Timeout
☐ Position
☐ Size
MIL-STD-1553
☐ CWD1
☐ CWD2
☐ SWD1
☐ SWD2
☑ TA
☑ SA
ARINC 429
AFDX

Select Columns dialog

becomes highlighted, then release

The True Heading field is timed out because that subaddress is not referenced in the BC Schedule (see figure top right). Column width can be resized by dragging the right edge of a column title, by placing the mouse along the right edge of the column title (the pointer changes to a crossbar ✛), and double clicking (to auto-size the column).



EngrView1*

Name	Value	Time	Status	Graph	Type	TA	SA
True Heading	122.1484375 degrees	00.000.000 sec	Timeout		1553 Field	1	1
Latitude	47.4453125 degrees	03:31.475.000			1553 Field	1	3
Longitude	-122.4453125 degrees	03:31.475.000			1553 Field	1	3

SA column being dragged between Value and Time

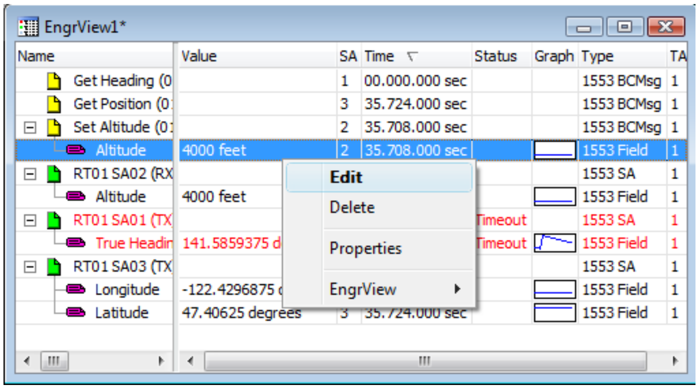
Add Items During Simulation

- ► Drag the **Altitude** field from the BC message Set Altitude in the Hardware Explorer into the Engineering View window

This field is displayed as a BC message receive field. Its data is static because it is not in the Data Generator.

- ► Drag and drop the **1553 Demo Card** object into the Engineering View window

This action automatically adds (or updates) all objects that branch beneath it to the view window.



EngrView1*

Name	Value	SA	Time	Status	Graph	Type	TA
Get Heading (0		1	00.000.000 sec			1553 BCMsg	1
Get Position (0		3	35.724.000 sec			1553 BCMsg	1
Set Altitude (0		2	35.708.000 sec			1553 BCMsg	1
Altitude	4000 feet	2	35.708.000 sec			1553 Field	1
RT01 SA02 (RX						1553 SA	1
Altitude	4000 feet					1553 Field	1
RT01 SA01 (TX				Timeout		1553 SA	1
True Headin	141.5859375 d			Timeout		1553 Field	1
RT01 SA03 (TX						1553 SA	1
Longitude	-122.4296875 d					1553 Field	1
Latitude	47.40625 degrees	3	35.724.000 sec			1553 Field	1

Engineering View with entire tree (with field context menu)

Edit Objects in Engineering View

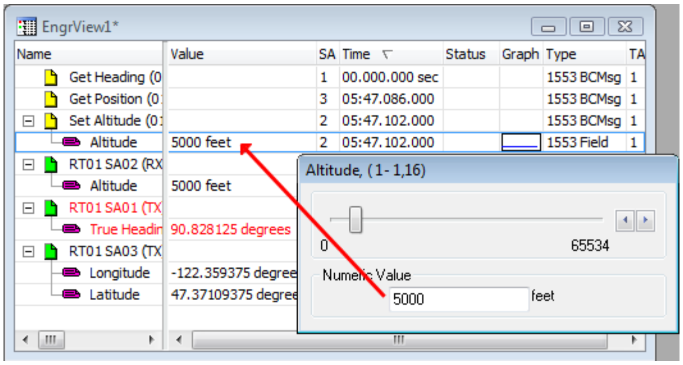
- ► Right click on the BC Message **Altitude** field in Engineering View and choose **Edit Data** from the menu to open the Engineering Units Editor (see figure)

The objects displayed in Engineering Views are linked to the actual object in the Hardware Explorer. Changes made from the Engineering View window are applied in the hardware objects in real time.

- ► Enter **5000** feet and press **Enter** to accept

The display will update in real time with the new value applied first in the BC message, then in the SA Altitude field row as it receives the new data.

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



EngrView1*

Name	Value	SA	Time	Status	Graph	Type	TA
Get Heading (0		1	00.000.000 sec			1553 BCMsg	1
Get Position (0		3	05:47.086.000			1553 BCMsg	1
Set Altitude (0		2	05:47.102.000			1553 BCMsg	1
Altitude	5000 feet	2	05:47.102.000			1553 Field	1
RT01 SA02 (RX							
Altitude	5000 feet						
RT01 SA01 (TX							
True Headin	90.828125 degrees						
RT01 SA03 (TX							
Longitude	-122.359375 degree						
Latitude	47.37109375 degree						

Altitude, (1- 1,16)

0 65534

Numeric Value 5000 feet

The Altitude fields data modified from the Engineering View window

Related Topics

- Engineering View windows can be used to logically group related objects
- Right clicking on column titles allows for quick removal of a displayed column
- Clicking once on the title bar sorts the rows alphabetically by the contents of that column

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 from the Engineering View

In the next lesson you will learn how to analyze recorded data using software playback.

Lesson 11: Analyze Data with Software Playback

The speed of activity on a MIL-STD-1553 databus makes it difficult for users to identify or respond to anomalies as they occur. One solution is to replay monitored data at a user-controlled pace in CoPilot's software playback mode. In Software Playback mode, interaction with the Ballard card and the MIL-STD-1553 databus is suspended and previously recorded monitor data is replayed through CoPilot as if it were coming from the databus (see first figure below).

Introduced in This Lesson

Software Playback mode, Monitor View playback controls

Objective

Replay a monitor record through Open Views and use the Monitor View playback toolbar to control software playback.

Create Playback Data Source

- ▶ Start this lesson by opening the 1553 Lesson 11 project (File | New | New Project | 1553 Lessons) to obtain the necessary configurations and settings

Your first step is to create the monitor file that will drive software playback.

- ▶ Right click Bus Monitor and select **Enable**
- ▶ Press the **Run** button to start the simulation
- ▶ After the monitor has recorded about 200 records to create , press the **Stop** button
- ▶ Right click Bus Monitor BM then select **Enable** again to disable it

Configure CoPilot for Playback

During runtime, playback is controlled from the Monitor View using the Playback toolbar and other controls.

- ▶ Double click on the **Monitor Data** to display the monitor records in a Monitor View window
- ▶ Right click on the **Demo1553Card1** icon and choose **Software Playback** (see figure)

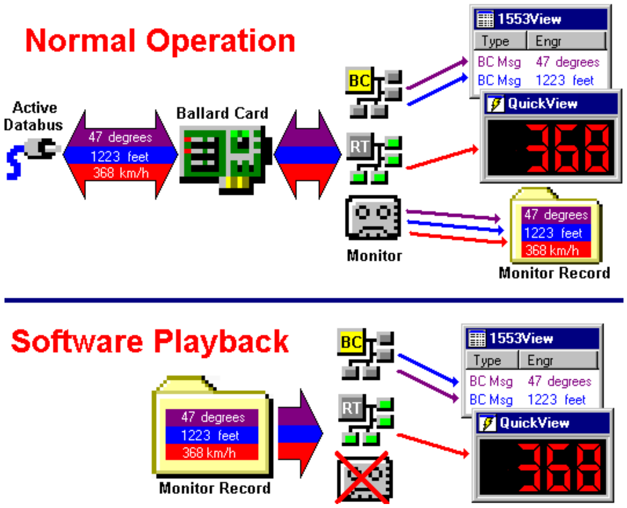
Notice the playback icon that appears on the card.

Run and Observe Playback

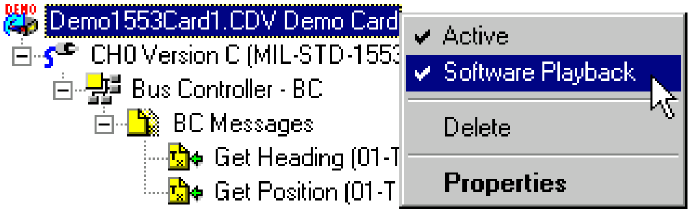
- ▶ Press the **Run** button to start the simulation and software playback

While software playback is running, data is replayed from recorded data. Observe how the highlight in Monitor View tracks with the data record currently being played back. When software playback is running, the recorded data can be reviewed through all the CoPilot view windows, including: strip charts, controls, data editors, etc.

Note: The Data Generator is automatically paused during software playback to prevent the data in view windows from being driven by two sources.



An illustration of how software playback works



Demo card in software playback mode

Control Playback during Run

Let's learn to use some of the playback controls.

- ▶ Move the **speed** control to various positions and observe the effect in Monitor View
- ▶ Put the **speed** control back to near the center position
- ▶ Click the **Pause** button to suspend playback
- ▶ Enter **100** in the Go To Message textbox and click the **Go** Go! button to jump to message 100
- ▶ Right click on row **100** and choose **Breakpoint Record** from the context menu (see figure)

A breakpoint icon is placed by the record number.

- ▶ Click the **Pause** button again to resume

Wait until playback encounters the breakpoint at record 100 and enters a paused state.

- ▶ Go to and place a breakpoint at record **150**
- ▶ Click the **Pause** button again to resume

Playback will run from record 100 to 150 and pause. While paused, you advance playback in single-record steps, forwards and backwards.

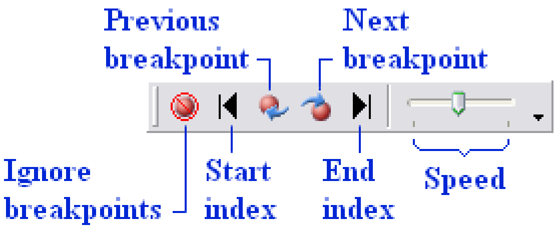
- ▶ Experiment with the **Step** button, then click the **Pause** button to resume (re-pauses at 100)

Wait for playback to pause at record 100 again.

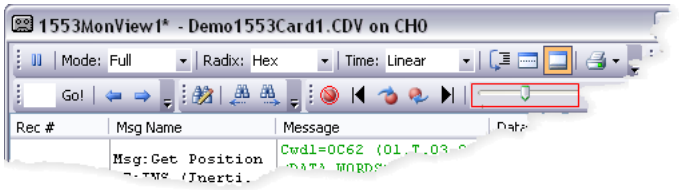
- ▶ Click the **Go To Next Breakpoint** button

This action advances the playback resume point from record 100 to 150. When playback is resumed, all records between rows 100 and 150 are skipped. You can also use the Set as Next Message menu command.

- ▶ Click the **Pause** button to resume playback
- ▶ Click the **Stop** button to end the simulation
- ▶ Finally, right click on the **Demo1553Card1** icon and clear **Software Playback**



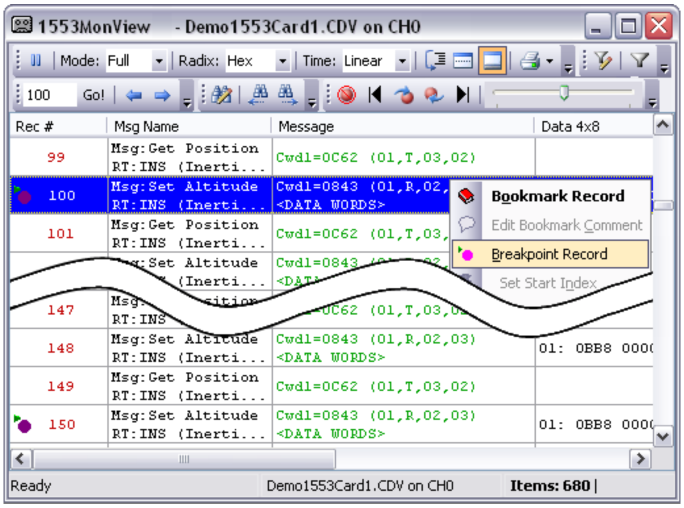
Playback toolbar in the Monitor View window


1553MonView1* - Demo1553Card1.CDV on CH0

Mode: Full | Radix: Hex | Time: Linear

Rec #	Msg Name	Message	Data
	Msg:Get Position RT:INS (Inerti.	Cwd1=0C62 (01,T,03,0 <DATA WORDS>	

Monitor View with playback speed decreased


1553MonView - Demo1553Card1.CDV on CH0

Mode: Full | Radix: Hex | Time: Linear

100 Go!

Rec #	Msg Name	Message	Data 4x8
99	Msg:Get Position RT:INS (Inerti...	Cwd1=0C62 (01,T,03,02)	
100	Msg:Set Altitude RT:INS (Inerti...	Cwd1=0843 (01,R,02, <DATA WORDS>	
101	Msg:Get Position RT:INS (Inerti...	Cwd1=0C62 (01,T,03,	
	Msg:Set Altitude (Inerti...	Cwd1=0843 <DATA	
147	Msg RT:INS	0C62 (01,T,03,0	
148	Msg:Set Altitude RT:INS (Inerti...	Cwd1=0843 (01,R,02,03) <DATA WORDS>	01: 0BB8 000
149	Msg:Get Position RT:INS (Inerti...	Cwd1=0C62 (01,T,03,02)	
150	Msg:Set Altitude RT:INS (Inerti...	Cwd1=0843 (01,R,02,03) <DATA WORDS>	01: 0BB8 000

Ready | Demo1553Card1.CDV on CH0 | Items: 680

Paused Monitor View with records 100 & 150 marked with breakpoints

Related Topics

- You can replay a subset of the monitor record by setting the start and end index
- Any Monitor View file in the project may be selected as the source for software playback

Summary

In this lesson you learned…

- how to create a playback source
- how to engage playback mode and replay a monitor record
- how to control playback using the controls in Monitor View

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

THIS PAGE INTENTIONALLY BLANK

Section C: CoPilot Professional

You have now completed the CoPilot Standard portion of this guide. In the first two sections you learned how to collect and display labels and fields in the Sequential Monitor and Engineering View windows. The information was defined and displayed using engineering units. The MIL-STD-1553 database was used for defining RTs, SA, and fields. In this section, the MIL-STD-1553 database is also used to store and retrieve graphical virtual instrument definitions for displaying engineering units. The additional CoPilot Professional capabilities described in this *Getting Started Guide* include scripting, strip charts, and moving map displays. Refer to the *CoPilot User's Manual* for additional Professional feature details not discussed in this guide such as running tests with 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…

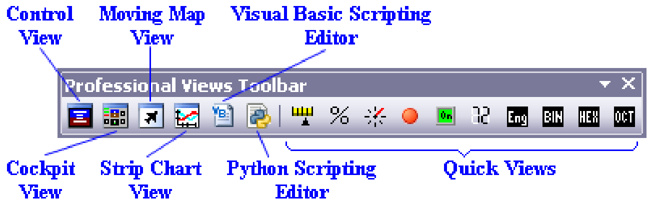
- graphical strip chart displays (Strip View, Lesson 12)
- a Control Selection Gallery of virtual controls and aircraft instruments (Quick View and Control View, Lessons 13 and 14)
- moving map displays (Map View, Lesson 15)
- flexible script routines used to extend the capabilities of CoPilot for running advanced simulations, generating complex data, creating reports, running tests and more (Python Code Editor, Lesson 16)
- VBScript routines (Script View, Lesson 17)

Professional Mode

To run CoPilot Professional components, CoPilot requires all hardware devices (included in the project) to have Professional license keys. The 1553 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 12 Strip charts can be used to display a history of one or more streams of data.
- Lesson 13 Simulated aircraft instruments can be automatically generated from fields, customized, and then permanently associated with a specific field using the 1553 database.
- Lesson 14 Simulated controls can also be custom designed and explicitly linked to a field values.
- Lesson 15 Positional data may be used to display an aircraft position on a moving map display.
- Lesson 16 CoPilot can be customized and controlled through script routines that access and modify the properties of CoPilot objects.

Lesson 12: Create a Strip Chart Display

The Strip View window is a dynamic, two-dimensional charting control for plotting one or more streams of real-time data. The Strip View displays the engineering value history of the selected parameters. The zoom function allows for quick review of the data set and in-depth analysis of data trends. Strip charts can be launched from individual fields during a simulation and additional parameters can be added from the Hardware Explorer through a drag-and-drop operation.

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 data set.

View Data with a Strip Chart

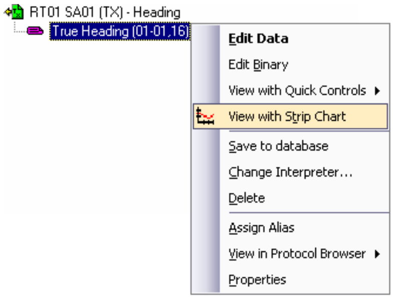
Note: If you are working from your own project instead of opening this lesson from the Samples and Lessons panel, be sure the Data Generator is present and configured to drive the data fields (see Lesson 9).

- ► Click the **Run** button to start the simulation
- ► Right click on the **True Heading** field and choose **View with Strip Chart** from the menu (see figure)

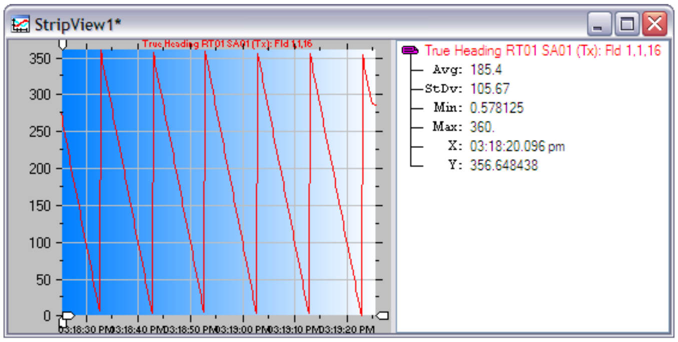
A Strip View window automatically opens and begins charting the True Heading field. Strip Views can also be opened from the New Hardware and Views dialog (when the simulation is not running).

- ► 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 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



```
True Heading RT01 SA01 (Tx): Fld 1,1,16
  Avg: 185.4
StDv: 105.67
  Min: 0.578125
  Max: 360.
    X: 03:18:20.096 pm
    Y: 356.648438
```

Resized Strip View window with True Heading field

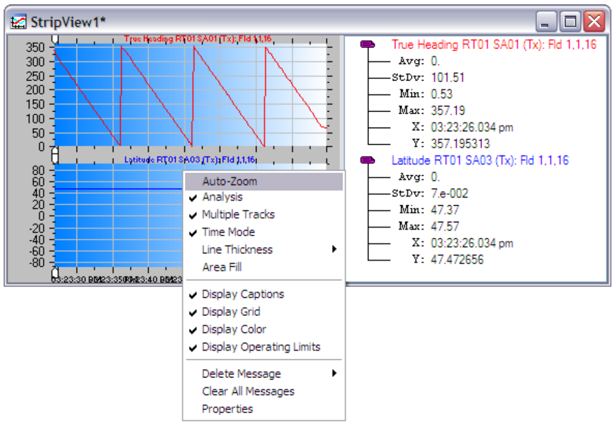
Add a Field while Running

- ► Drag and drop the **Latitude** field from the Hardware Explorer into 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.

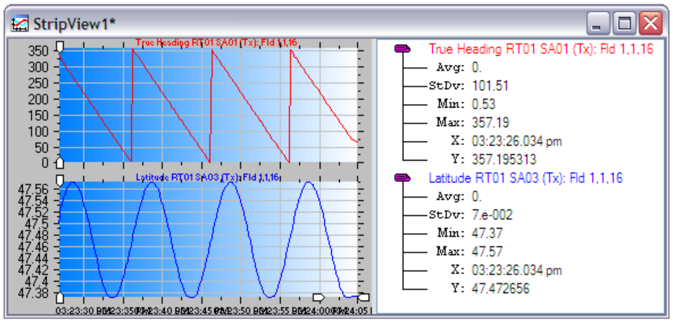


```
True Heading RT01 SA01 (Tx): Fld 1,1,16
  Avg: 0.
StDv: 101.51
  Min: 0.53
  Max: 357.19
    X: 03:23:26.034 pm
    Y: 357.195313
Latitude RT01 SA03 (Tx): Fld 1,1,16
  Avg: 0.
StDv: 7.e-002
  Min: 47.37
  Max: 47.57
    X: 03:23:26.034 pm
    Y: 47.472656
```

Strip View with Latitude field added and context menu

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

Now we see only the range of actual values (notice the new scale in the y-axis). You can also adjust the zoom by dragging the zoom buttons ▽△ along the y-axis.



StripView1*

True Heading RT01 SA01 (Tx): Fld 1,1,16
350
300
250
200
150
100
50
0
Latitude RT01 SA03 (Tx): Fld 1,1,16
47.56
47.54
47.52
47.5
47.48
47.46
47.44
47.42
47.4
47.38
03:23:30 PM 03:23:35 PM 03:23:40 PM 03:23:45 PM 03:23:50 PM 03:23:55 PM 03:24:00 PM 03:24:05

True Heading RT01 SA01 (Tx): Fld 1,1,16
Avg: 0.
StDv: 101.51
Min: 0.53
Max: 357.19
X: 03:23:26.034 pm
Y: 357.195313

Latitude RT01 SA03 (Tx): Fld 1,1,16
Avg: 0.
StDv: 7.e-002
Min: 47.37
Max: 47.57
X: 03:23:26.034 pm
Y: 47.472656

Strip View after Latitude track has been auto-zoomed

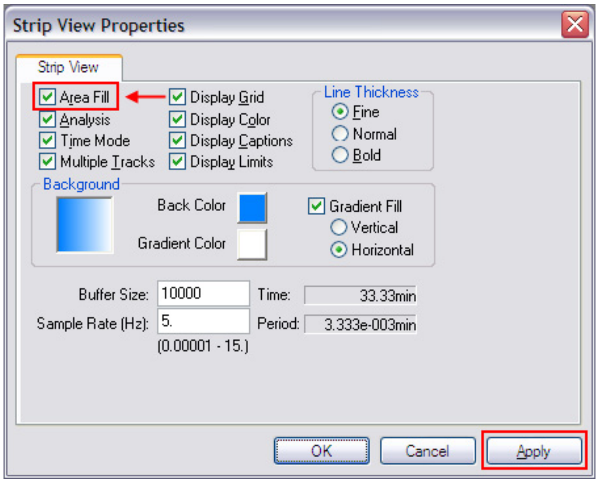
Modify Strip View Properties

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



Strip View Properties

Strip View

☑ Area Fill
☑ Analysis
☑ Time Mode
☑ Multiple Tracks
☑ Display Grid
☑ Display Color
☑ Display Captions
☑ Display Limits

Line Thickness
● Fine
○ Normal
○ Bold

Background
Back Color
Gradient Color
☑ Gradient Fill
○ Vertical
● Horizontal

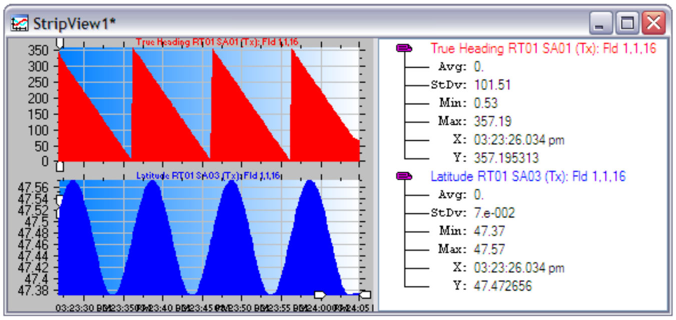
Buffer Size: 10000
Time: 33.33min
Sample Rate (Hz): 5.
Period: 3.333e-003min
(0.00001 - 15.)

Strip View Properties window

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 we 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.



StripView1*

True Heading RT01 SA01 (Tx): Fld 1,1,16
350
300
250
200
150
100
50
0
Latitude RT01 SA03 (Tx): Fld 1,1,16
47.56
47.54
47.52
47.5
47.48
47.46
47.44
47.42
47.4
47.38
03:23:30 PM 03:23:35 PM 03:23:40 PM 03:23:45 PM 03:23:50 PM 03:23:55 PM 03:24:00 PM 03:24:05

True Heading RT01 SA01 (Tx): Fld 1,1,16
Avg: 0.
StDv: 101.51
Min: 0.53
Max: 357.19
X: 03:23:26.034 pm
Y: 357.195313

Latitude RT01 SA03 (Tx): Fld 1,1,16
Avg: 0.
StDv: 7.e-002
Min: 47.37
Max: 47.57
X: 03:23:26.034 pm
Y: 47.472656

Strip View with changes from Properties window

Related Topics

- There is also a strip chart control in the Control View window (Lessons 13 and 14 discuss Control View)

Summary

In this lesson you learned…

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

In this lesson you were introduced to Strip Views, one of many CoPilot Professional graphical display components. The next two lessons introduce Quick Views and Control Views, displays that host virtual instruments and ActiveX controls.

Lesson 13: Generate Controls Automatically

A variety 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 Explorer. Quick Views are pre-configured controls that can be launched in one step from data fields in the Hardware Explorer. The Control View is a display window that can host multiple controls and instruments. You can add controls to Control View automatically through a simple drag-and-drop or build and link custom controls manually (described in the following lesson).

Introduced in This Lesson

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

Objective

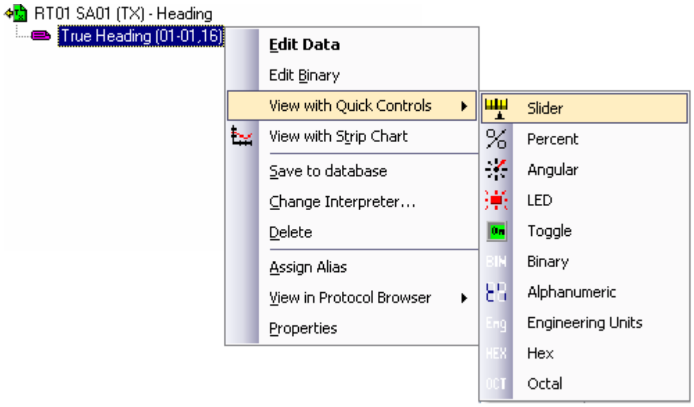
Automatically generate a slider Quick Control from a field and configure it as a data sink. Then open a Control View window and add an altimeter control from another field. Modify the field value and observe the results in the altimeter.

Generate a Quick View Control

Note: First open the 1553 Lesson 13 project (from Samples and Templates on the Start Page).

The easiest way to view and/or modify a data field graphically is with CoPilot Professional Quick Views.

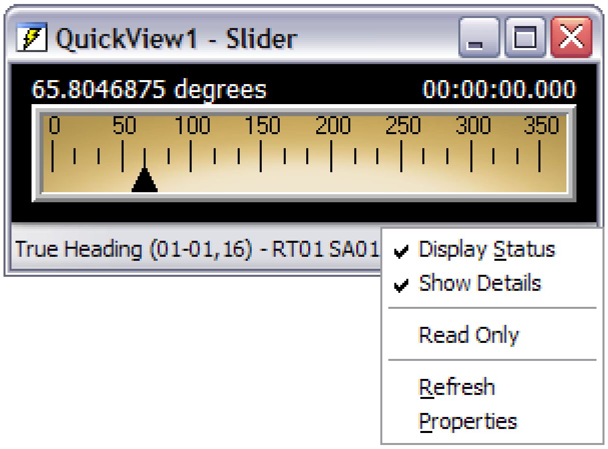
- ▶ Click the **Run** button to start the simulation
- ▶ Right click on the **True Heading** field, point to **View With Quick Controls** to expand the submenu, and click on **Slider** (see figure at right)



The Quick View controls menu for True Heading field

A Quick View slider control will open in the display area. Because True Heading is a transmit field the slider defaults to editable. Since we are already sourcing data for this field in the Data Generator, we'll configure the slider to be Read Only.

- ▶ Right click on the slider control and choose **Read Only** from the context menu (see figure)



Quick View slider for True Heading

Now the value of this field cannot be modified using the Quick View.

There are many 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 Engineering Units control will display both the value and the units of measurement (for example, "182 degrees").

Generate a Control View Control

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

- ▶ Click the **Stop** button to halt the simulation
- ▶ Click the **New** button to open the New Hardware and Views dialog
- ▶ Select the **Professional Views** tab, choose **Control View**, and click **OK**

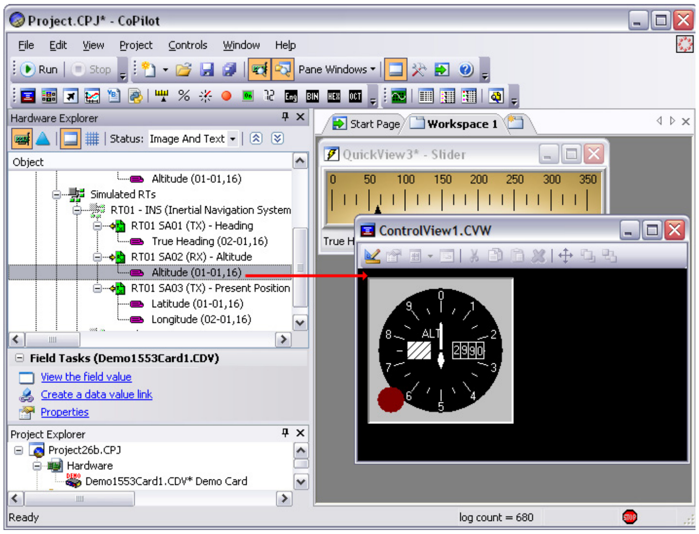
A blank Control View will open in the display area.

- ▶ Drag and drop the **Altitude** field from RT01, SA02 into the form (see figure on previous page)

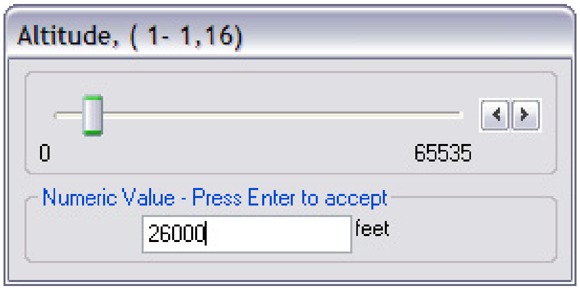
(The upper-left corner of the control will be positioned at the point where you release the mouse. We'll reposition this control in the next lesson.) The altimeter control appears, displaying the value of the Altitude field (5,000 feet).

- ▶ Click the **Run** button to start the simulation
- ▶ Double click the **Altitude** field in the Set Altitude message to open the Engineering Units Editor
- ▶ Enter a new value of **26,000** feet and hit **Enter** (do not close the editor yet)

Notice how the altimeter displays changes. There is now a 20 in the ten thousands box, and the needle is pointing to 6 (see figure).

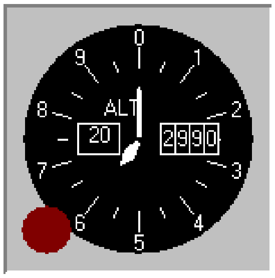


Altimeter control generated from the Altitude field



26000 feet

Engineering Units Editor for BC message Altitude field



The altimeter control displaying 26,000 feet

- ▶ In the Engineering Units Editor, change the value back to **5,000** feet and click away (anywhere) to close the editor
- ▶ Click the **Stop** button to halt the simulation
- ▶ If you are maintaining a cumulative project, save the Control View window (for the next lesson).

Control Links in the Database

An altimeter control was generated from the Altitude field because that field had been previously saved to the database with a link to the altimeter control. Fields without control links are represented by a default control. The default control varies by interpreter type.

In the next lesson, you will create a new control and link it to a data field. That link can be preserved in the 1553 database as part of a field or RT definition. Then, dragging that field into Control View will automatically reinstate the saved control.

Related Topics

- The Control View can host user defined and third-party ActiveX controls

Summary

In this lesson you learned…

- how to automatically generate a Quick View control
- how to open Control View and add a control automatically
- about saving field-control links to the 1553 database

In the next lesson 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 14: 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 a 1553 field.

Introduced in This Lesson

Control Selection Gallery, Control Palette, Control Properties dialog, Control Property Browser, New Link: Select Source and Sink dialog

Objective

Create and configure a Heading Indicator control and link it to the True Heading field in the Hardware Explorer.

Create a Control Manually

Note: Before you begin this lesson, open the 1553 Lesson 14 project (File | New | New Project | 1553 Lessons) or use the project you saved from Lesson 13 so that Control View is present.

- ▶ 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

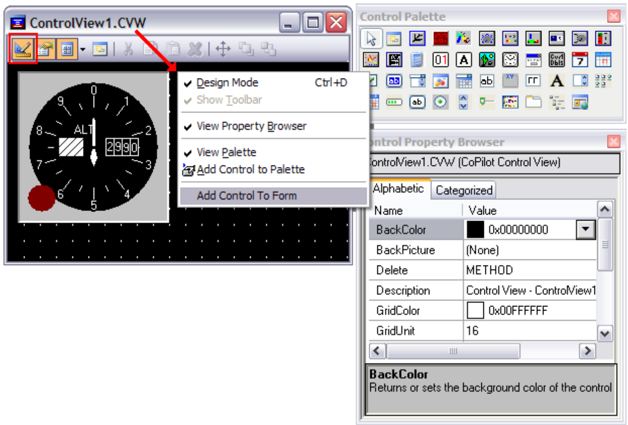
Configure Controls in Control View

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 altimeter (from Lesson 13) 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 (AirCtrl1) appears at the top of the browser.

- ▶ In the Control Property Browser, click on the **Categorized** tab and locate the **Extended Properties** group
- ▶ Enter **Altimeter** in the name field, **25** in PositionX and **45** in PositionY, and **150** in SizeX and Y (see figure)
- ▶ Select the Heading Indicator control (AirCtrl2), rename it to **HeadingIndicator** (the scriptable name)
- ▶ Enter **190** for PositionX, **45** for PositionY, and **150** for SizeX and SizeY



Control Property Browser

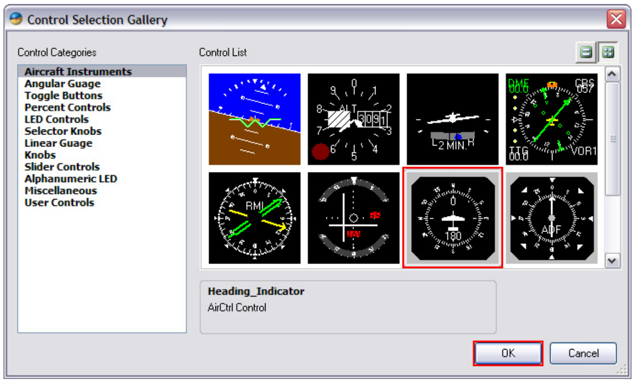
ControlView1.CVW (CoPilot Control View)

Alphabetic | Categorized

Name	Value
BackColor	0x00000000
BackPicture	(None)
Delete	METHOD
Description	Control View - ControlView1
GridColor	0x00FFFFFF
GridUnit	16

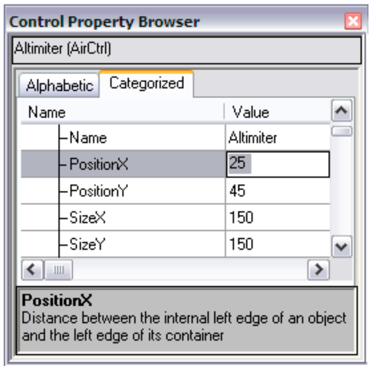
BackColor
Returns or sets the background color of the control

Open the Control Selection Gallery Window



Heading_Indicator
AirCtrl Control

Control Selection Gallery With Heading_Indicator Selected



Control Property Browser

Altimiter (AirCtrl)

Alphabetic | Categorized

Name	Value
Name	Altimiter
PositionX	25
PositionY	45
SizeX	150
SizeY	150

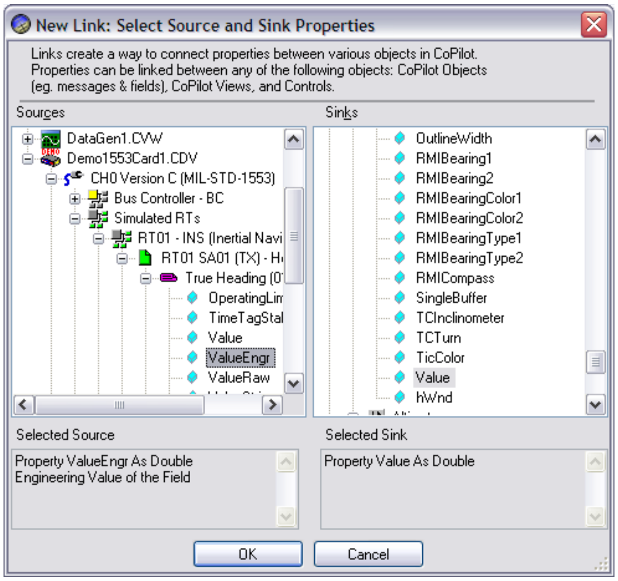
PositionX
Distance between the internal left edge of an object and the left edge of its container

The Altimeter 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 **Demo1553Card1** tree, then expand **CH0 Version C (MIL-STD-1553)**, then **Simulated RTs**, then **RT01**, then **SA01**, 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 **HeadingIndicator**
- ▶ Scroll down and select the **Value** property and click **OK** to close the Source and Sink dialog



Links create a way to connect properties between various objects in CoPilot. Properties can be linked between any of the following objects: CoPilot Objects (eg. messages & fields), CoPilot Views, and Controls.

Selected Source: Property ValueEngr As Double
Engineering Value of the Field

Selected Sink: Property Value As Double

Select Source and Sink dialog linking the field and control

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.

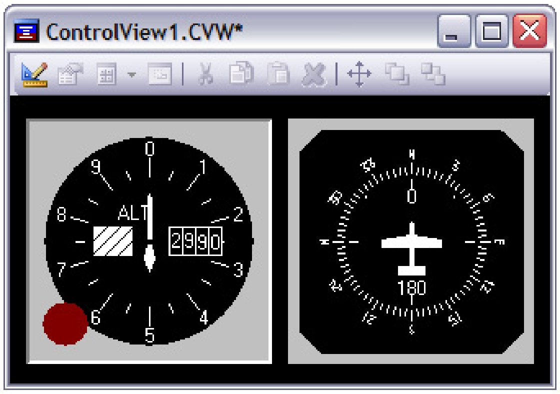
Run the Simulation

Note: Be sure the Data Generator is in the Project Explorer and configured to drive the True Heading field (Lesson 9).

- ▶ 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 1553 fields in the Hardware Explorer.

- ▶ 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 the next lesson, you will learn about a powerful Moving Map display that displays aircraft position.

Lesson 15: Create a Moving Map Display

Map View is a container for displaying aircraft position and other objects (like waypoints) on a fixed or moving map image. The Map View window links positional data to a real-time moving map display. Latitude, longitude, heading, and altitude are used to display positions on the map. This example uses data streams from the Data Generator (created in Lesson 9) to demonstrate how Map View displays incoming data. The initial Map View background is a view of SeaTac airport. CoPilot users may supply their own unique background images or utilize the topographical and satellite images through the provided Microsoft TerraServer® web service.

Introduced in This Lesson

Map View, Map View Properties, TerraServer Properties

Objective

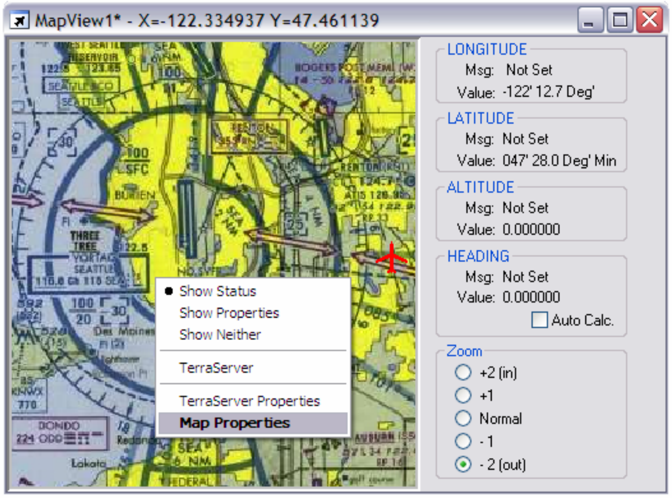
Create a moving map display of a holding pattern (over Sea-Tac airport) by configuring a Map View window, linking positional data fields, and running the simulation. Then, change settings and maps while the simulation is running.

Open a Map View Window

- Click the **New** button to open the New Hardware or 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.

Note: Be sure the Data Generator is present and configured to drive three of the data fields (see Lesson 9). If extra views are open from previous lessons, you can hide them to the Project Explorer with the Minimize button.



MapView1* - X=-122.334937 Y=47.461139

LONGITUDE
Msg: Not Set
Value: -122' 12.7 Deg'

LATITUDE
Msg: Not Set
Value: 047' 28.0 Deg' Min

ALTITUDE
Msg: Not Set
Value: 0.000000

HEADING
Msg: Not Set
Value: 0.000000
□ Auto Calc.

Zoom
○ +2 (in)
○ +1
○ Normal
○ -1
● -2 (out)

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

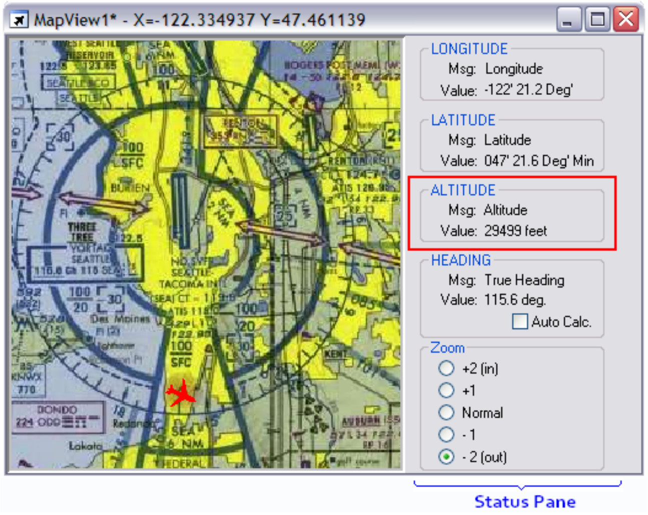
Link Data and View Moving Map

- Drag and drop the BC message **Altitude** field on to the Altitude sink in the status pane

Notice how the plane object reacts to each new link.

- Drag and drop the **True Heading**, **Latitude**, and **Longitude** fields from RT01 in the Hardware Explorer 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.



MapView1* - X=-122.334937 Y=47.461139

LONGITUDE
Msg: Longitude
Value: -122' 21.2 Deg'

LATITUDE
Msg: Latitude
Value: 047' 21.6 Deg' Min

ALTITUDE
Msg: Altitude
Value: 29499 feet

HEADING
Msg: True Heading
Value: 115.6 deg.
□ Auto Calc.

Zoom
○ +2 (in)
○ +1
○ Normal
○ -1
● -2 (out)

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. 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, and choose a style of **2 - Shaded**
- ▶ Click **OK** to close the Map Properties dialog

There are many other options listed under other tabs.

Display TerraServer Maps

Map View can link to Microsoft's TerraServer® web service to 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.

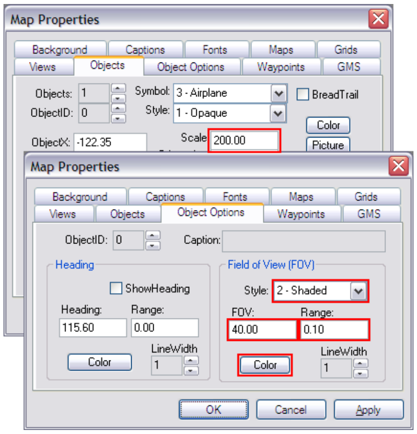
- ▶ Right click in the map pane and choose **TerraServer Properties** from the context menu
- ▶ Click the **Enable** checkbox and choose **Topographical** from the Theme listbox (see figure) and click **OK** to the close TerraServer Properties dialog

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



Map Properties

Objects: 1
ObjectID: 0
Symbol: 3 - Airplane
Style: 1 - Opaque
□ BreadTrail
ObjectX: -122.35
Scale 200.00

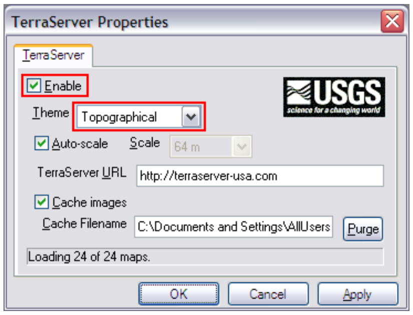
Map Properties

ObjectID: 0
Caption:

Heading
□ ShowHeading
Heading: 115.60
Range: 0.00
LineWidth 1

Field of View (FOV)
Style: 2 - Shaded
FOV: 40.00
Range: 0.10
LineWidth 1

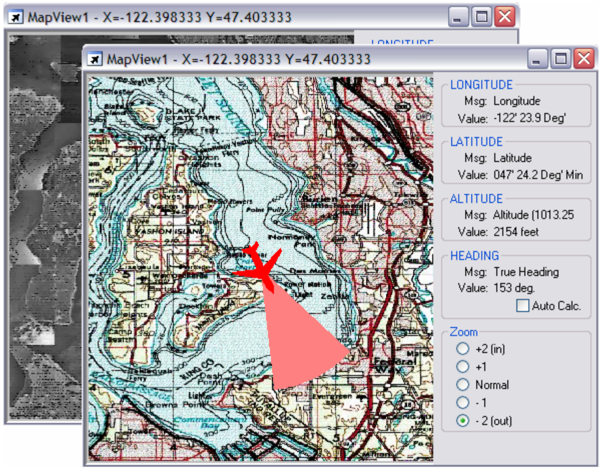
Objects and Object Options tabs with settings described in text



TerraServer Properties

☑ Enable
Theme Topographical
☑ Auto-scale
Scale 64 m
TerraServer URL http://terraserver-usa.com
☑ Cache images
Cache Filename C:\Documents and Settings\AllUsers
Loading 24 of 24 maps.

TerraServer Properties windows with specified settings



MapView1 - X=-122.398333 Y=47.403333

MapView1 - X=-122.398333 Y=47.403333

LONGITUDE
Msg: Longitude
Value: -122' 23.9 Deg'

LATITUDE
Msg: Latitude
Value: 047' 24.2 Deg' Min

ALTITUDE
Msg: Altitude (1013.25
Value: 2154 feet

HEADING
Msg: True Heading
Value: 153 deg.
□ Auto Calc.

Zoom
○ +2 (in)
○ +1
○ Normal
○ - 1
● - 2 (out)

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 the Map View
- how to use TerraServer maps

In the next lesson you will learn how to customize CoPilot with scripting using ATE.

Lesson 16: 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. Additional features of ATE not described in this Getting Started Guide include a Script Debugger, Output Pane, Command Prompt and Output Pane.

Introduced in This Lesson

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

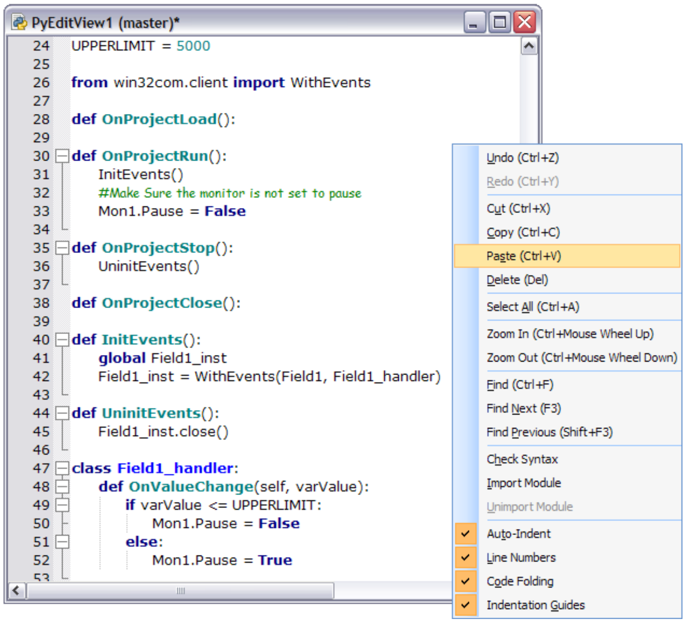
Objective

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**
- Minimize CoPilot and use MS Windows Explorer to locate the example script **MonitorControl.py** file (copied to "…\My Documents\My CoPilot Projects\Samples \Scripts" during first run). Right click the file and select **Edit with IDLE**
- **Copy** the entire file (e.g., **CTRL+A**, **CTRL+C**)
- Then in CoPilot, overwrite the contents of the Python Editor View by **pasting** in the copied script (e.g., **CTRL+A**, **CTRL+V**)

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



```
UPPERLIMIT = 5000

from win32com.client import WithEvents

def OnProjectLoad():

def OnProjectRun():
    InitEvents()
    #Make Sure the monitor is not set to pause
    Mon1.Pause = False

def OnProjectStop():
    UninitEvents()

def OnProjectClose():

def InitEvents():
    global Field1_inst
    Field1_inst = WithEvents(Field1, Field1_handler)

def UninitEvents():
    Field1_inst.close()

class Field1_handler:
    def OnValueChange(self, varValue):
        if varValue <= UPPERLIMIT:
            Mon1.Pause = False
        else:
            Mon1.Pause = True
```

The Python Editor with Monitor Control script and context menu

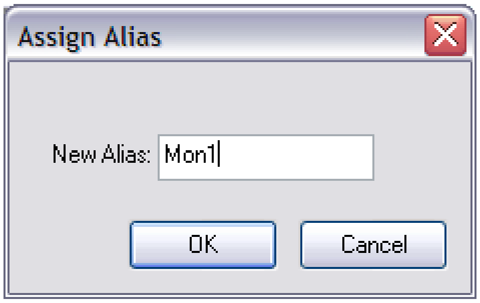
Translate Objectives into Code

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 14, 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**



New Alias: Mon1

The Sequential Monitor object being assigned an alias

- ▶ Drag the **Sequential Monitor** object into the Object Browser and rename the alias **Mon1**

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 (teal icon) is used in the script (see figure). (Methods are represented by a magenta icon and events by yellow icons.)

- ▶ Drag Altitude field from SA02 (not BC message) 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 yellow lightning bolt indicating 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 the **Altitude** field from the Set Altitude BC message (not SA02) into the Data Generator
- ▶ As done with the Data Generator in Lesson 9, set the type to **Sine**, the Min value to **0**, Max value to **20,000**, Period to **20** seconds, and click **OK**

The Bus Monitor has been disabled since Lesson 5.

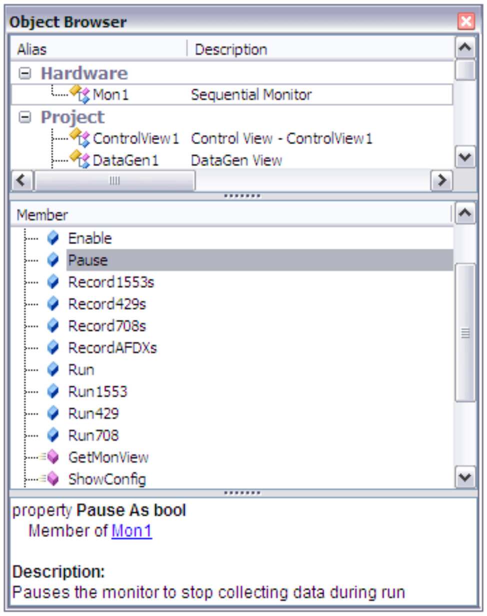
- ▶ Right click the **Bus Monitor** BM and select **Enable** to reenable monitoring (match figure at right)

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, and resumes recording when Altitude falls below 5,000 feet.

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



Object Browser

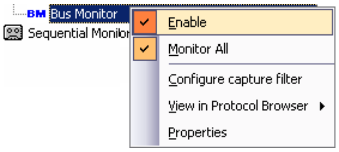
Alias	Description
Hardware	
Mon1	Sequential Monitor
Project	
ControlView1	Control View - ControlView1
DataGen1	DataGen View

Member
- Enable
- Pause
- Record1553s
- Record429s
- Record708s
- RecordAFDXs
- Run
- Run1553
- Run429
- Run708
- GetMonView
- ShowConfig

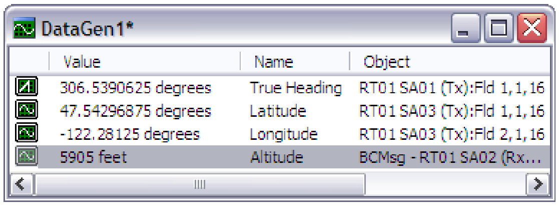
property **Pause As bool**
Member of Mon1

Description:
Pauses the monitor to stop collecting data during run

The Pause property of the Mon1 object



Correctly configured monitors



DataGen1*

Value	Name	Object
306.5390625 degrees	True Heading	RT01 SA01 (Tx):Fld 1,1,16
47.54296875 degrees	Latitude	RT01 SA03 (Tx):Fld 1,1,16
-122.28125 degrees	Longitude	RT01 SA03 (Tx):Fld 2,1,16
5905 feet	Altitude	BCMsg - RT01 SA02 (Rx...

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

Sequential Monitor Paused

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 1553! To learn more about the topics introduced in this guide, please refer to the *CoPilot User's Manual*.

Lesson 17: Legacy CoPilot Scripting

This lesson included for legacy purposes and has been replaced by lesson 16.

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 submenu, then click **Overwrite** (see figure)
- ► In the Select Import File dialog, click the **Browse** button, navigate to the Scripts folder, select the **MonitorControl.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.

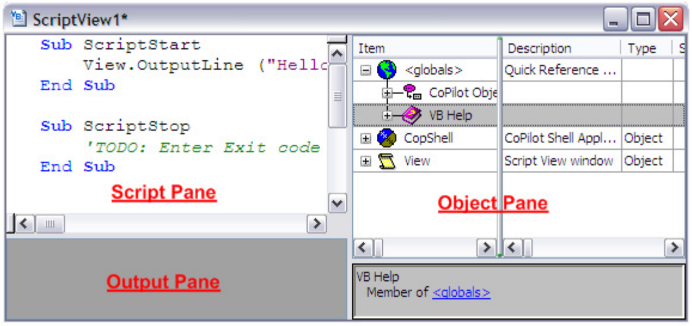
Translate Objectives into Code

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 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 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 (DemoMon1)



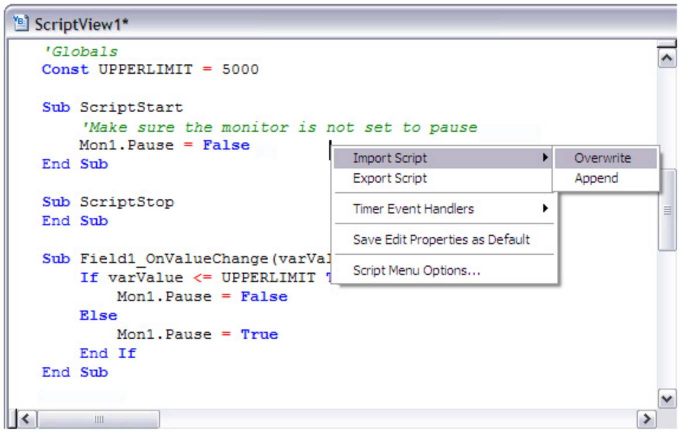
```
    Sub ScriptStart
        View.OutputLine ("Hellc
    End Sub

    Sub ScriptStop
        'TODO: Enter Exit code
    End Sub
```

Item	Description	Type
<globals>	Quick Reference …	
CoPilot Obje		
VB Help		
CopShell	CoPilot Shell Appl…	Object
View	Script View window	Object

VB Help
Member of <globals>

An example Script View window with each pane labeled



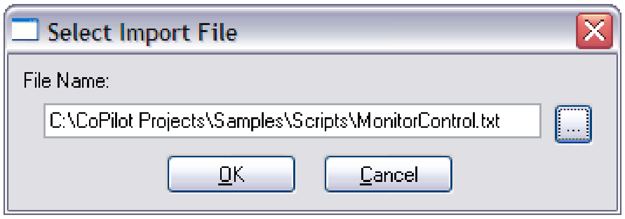
```
'Globals
Const UPPERLIMIT = 5000

Sub ScriptStart
    'Make sure the monitor is not set to pause
    Mon1.Pause = False
End Sub

Sub ScriptStop
End Sub

Sub Field1_OnValueChange(varVal
    If varValue <= UPPERLIMIT T
        Mon1.Pause = False
    Else
        Mon1.Pause = True
    End If
End Sub
```

The Script pane with Monitor Control script and context menu



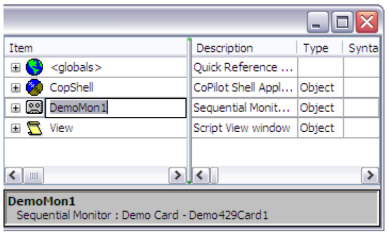
File Name: C:\CoPilot Projects\Samples\Scripts\MonitorControl.txt

The import dialog with the default path to MonitorControl.txt and rename it Mon1 (as identified in the script)

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 (teal icon) is used in the script (see figure). (Methods are represented by a magenta icon ◆ and events by blue ⚡ icons.)

- ▶ Drag Altitude field from SA02 (not BC message) into the Object Pane and rename it **Field1**

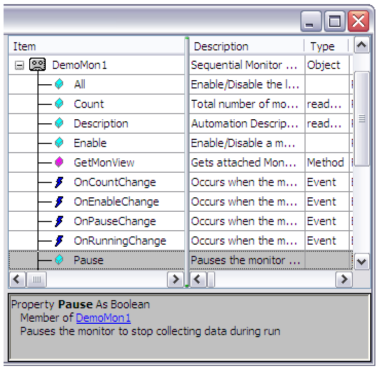
If you browse to the ⚡ OnValueChange method for Field1, you will notice that the icon is the blue lightning bolt indicating that the method is called in response to an event.



Item	Description	Type	Synta
⊞ <globals>	Quick Reference ...		
⊞ CopShell	CoPilot Shell Appl...	Object	
⊞ DemoMon1	Sequential Monit...	Object	
⊞ View	Script View window	Object	

DemoMon1
Sequential Monitor : Demo Card - Demo429Card1

The Sequential Monitor object being renamed



Item	Description	Type
⊟ DemoMon1	Sequential Monitor ...	Object
All	Enable/Disable the l...	
Count	Total number of mo...	read...
Description	Automation Descrip...	read...
Enable	Enable/Disable a m...	
GetMonView	Gets attached Mon...	Method
OnCountChange	Occurs when the m...	Event
OnEnableChange	Occurs when the m...	Event
OnPauseChange	Occurs when the m...	Event
OnRunningChange	Occurs when the m...	Event
Pause	Pauses the monitor ...	

Property **Pause** As Boolean
Member of DemoMon1
Pauses the monitor to stop collecting data during run

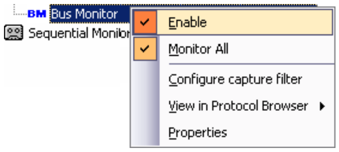
The Pause property of the Mon1 object

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 the Set Altitude BC message (not SA02) into the Data Generator
- ▶ Using what you learned about the Data Generator in Lesson 9, set the type to **Sine**, the Min value to **0**, Max value to **20,000**, Period to **20** seconds, and click **OK**

The Bus Monitor has been disabled since Lesson 5.

- ▶ Right click the **Bus Monitor** BM and select **Enable** to reenable monitoring (match figure at right)



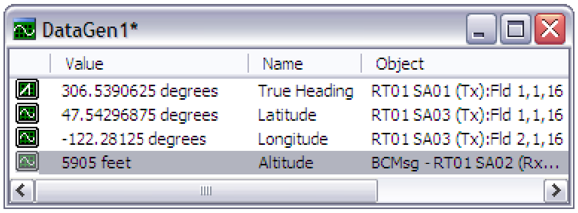
Correctly configured monitors

Execute the Script

- ▶ Click the **Run** button to start the simulation and trigger the 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, and resumes recording when Altitude falls below 5,000 feet.

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



DataGen1*

Value	Name	Object
306.5390625 degrees	True Heading	RT01 SA01 (Tx):Fld 1,1,16
47.54296875 degrees	Latitude	RT01 SA03 (Tx):Fld 1,1,16
-122.28125 degrees	Longitude	RT01 SA03 (Tx):Fld 2,1,16
5905 feet	Altitude	BCMsg - RT01 SA02 (Rx...

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

Sequential Monitor Paused

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 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 16.

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 1553

Just the Beginning

In these brief lessons, you learned how easy it is to simulate and test MIL-STD-1553 terminals using CoPilot with MIL-STD-1553, 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 1553 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 the CoPilot menu, toolbars, and windows using the software-generated 1553 Demo card. If you already have a OmniBus, BUSBox 1553, CM1553-3, PC1553-3, LP1553-3, LC1553-3 or another board, Ballard can install a temporary license 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*